A New Intrusion Detection System Based on Gated Recurrent Unit (GRU) and Genetic Algorithm

Mahdi Manavi[1(✉)] and Yunpeng Zhang[2]

[1] Mirdamad Institute of Higher Education of Gorgan, 0098171 Gorgan, Iran
Mahdi.manavi24@gmail.com
[2] Department of Information and Logistics Technology, University of Houston, Houston 77004, USA

Abstract. Distributed systems are extensive nowadays. The challenge of preventing network penetration by malware and hackers in these systems has been extensively considered and many types of research have been carried out in this field. Due to the high volume of input and output data in distributed systems, definitive and static algorithms that are used for small environments are not appropriate. For this problem, one of the new techniques is the deep learning method, which allows one to find optimal answers. In this paper, deep learning is used to investigate the behavior patterns of requests that enter the distributed network and then attacks are detected based on these patterns, which send an alarm to administrators (Anomaly Detection). In the next step, the genetic algorithm is used with the rule-based database to examine misuse detection. In this paper, considering the results obtained, it can be seen that the proposed algorithm provides high accuracy in detecting attacks with a low false alarm rate.

Keywords: Intrusion detection · Recurrent neural network · Gated recurrent unit · Genetic algorithm · KDD · Hybrid detection method

1 Introduction

An intrusion detection system (IDS) is a system that monitors the network traffic for suspicious activity. In 1987, attack detection systems became a topic of discussion and many researchers have continued to focus on this issue. IDS are security tools that receive and monitor network traffic or scan logs of the system and inform about suspicious activity with alarms they send to the network administrator [1]. Misuse detection consists of rules in the database used for known attacks. These rules are either designed using knowledge-based systems that contain known types of attacks or based on machine training patterns based on the behavior of users on suspicious activities [2]. Machine training techniques

G. Wang et al. (Eds.): SpaCCS 2019, LNCS 11611, pp. 368–383, 2019.
https://doi.org/10.1007/978-3-030-24907-6_28

based on classifier algorithms with the supervisor may not respond properly to new attacks that are patterned much differently than that for which they were trained and they are not suitable for so-called zero day attacks. Anomaly detection is based on the expected behavior of the system, where any behavior that goes out of the model is considered abnormal, and it's suitable for zero day attacks. Hybrid detection is a combination of the above methods that can make the permeability more accurate. Usually, the combination of misuse detection and Anomaly detection methods is used to increase the effect of the permeability detection engines. However, the combination of these two methods is not suitable for detecting intrusion at the level of the hypervisor's IDS implementation methods [1]. The implementation of IDSs is also divided into two categories:

Distributed: In this model, intrusion detection systems are used to exchange data in the detection of an intrusion and to declare attacks in different parts of the network. This model of IDS systems can be effective in detecting attacks by attackers at a specific time.

Non-distributed: This model of intrusion detection systems is located in a network area, such as SNORT [3]. In Fig. 1 the architecture of GRU is shown. GRU is a variant of LSTM which was introduced by Cho [4]. LSTM, a variation of a recurrent network, was proposed as one of the machine learning techniques to solve a lot of sequential data problems. LSTM architecture consists of 4 main components:

1-Input gate 2-forget gate 3-output gate 4-memory cell.

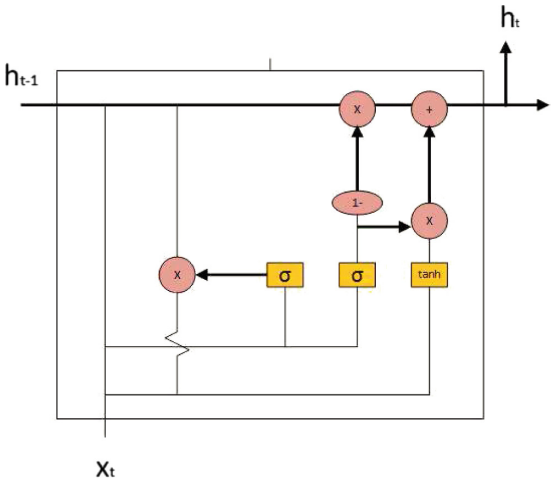

Fig. 1. Architecture of GRU

The cell makes decisions about what to store, read and write via gates that open or close and each memory cell corresponds to a time-step. Those gates pass the information based on set weights. Some of those weights like input and

hidden states are adjusted during the learning process. GRU internal structure is simpler and therefore is faster to train since fewer computations are needed to make an update to its hidden state. GRU has two gates: Reset gate and update gate. The reset gate determines how to combine the new input with the previous memory cell and the update gate defines how much of the previous memory cell to keep around.

Genetic algorithm is one of the evolved computational subsets that have a direct relationship with the topic of artificial intelligence. It can be called a subset of artificial intelligence. Genetic algorithm is a search method that mimics the rules of natural biological evolution. Genetic algorithm applies the law of survival of the fittest to a series of problem solutions to obtain better answers. In each generation, better approximations of the final answers are obtained with the help of a selection process proportional to the value of the answers and reproduction and using operators that mimick natural genetics. This process adapts the new generations with the problem conditions.

In this paper, we propose a solution to detect attacks that go into the network with higher accuracy. Using each method like misuse detection and anomaly detection individually will enable the detection of the intrusions at a lower level. One of the challenges in misuse detection is the lack of recognition of zero-day attacks. Further, in anomaly detection, it can be noted that due to the low number of training data, a number of attacks are not given, so we use the hybrid algorithm for intrusion detection to minimize the challenges as far as possible. In the first section, we use the recurrent neural network in which we implement an anomaly method to detect new attacks by examining their behavior. Then, we use a genetic algorithm in which we detect the attacks using the knowledge base. We used the KDD cup 99 to implement the proposed solution and provide a comparison of our current work with other techniques as well.

The rest of the paper is organized as follows. Related works are introduced in Sect. 2. In Sect. 3, we present the proposed algorithm. The evaluation of the model is analyzed in Sect. 4. Finally, we draw conclusions in Sect. 5.

2 Related Work

Subba et al. [5] presented a simple Artificial Neural Network (ANN) based IDS model featuring feed forward and back propagation algorithms used by the presented Intrusion Detection System (IDS) with different optimization techniques. Xu et al. [6] presented a deep neural network in which the gated recurrent unit was used learn better network and multilayer perceptron network types examined on two KDD99 and NSL-KDD data types. Nguyen et al. [7] presented a convolutional neural network in which DoS attacks were detected and compared with other machine learning methods, such as KNN, Support Vector Machine (SVM) and nare bayes. The CNN introduced in this work was designed with two convolutional layers. Fu et al. [8] used the recurrent neural network and used LSTM-RNN to reach a high detection rate. Their experiment consisted of data preprocessing, feature abstraction, training and detection. LSTM were used

during the training stage to classify whether the traffic was an attack or normal traffic. Balamurugan et al. [9] used a Markov model to predict cyber threat patterns based on the current known features and they used K-means clustering, a neural network, for an intrusion detection system. Kim et al. [10] proposed a deep neural network using 100 hidden units, combined with the ADAM optimizer and the ReLu activation function. Their approach was implemented on a GPU using TensorFlow and evaluated using the KDD data set. Maleh et al. [11] proposed a hybrid, lightweight intrusion detection system. Their intrusion detection model takes advantage of cluster-based architecture to reduce energy consumption. The model uses SVM for anomaly detection and a set of signature rules to detect malicious behaviors and provide global lightweight IDS. Kim et al. [12] constructed an IDS model with Long Short Term Memory (LSTM) architecture to a Recurrent Neural Network (RNN) and trained the IDS model on the KDD Cup 1999 dataset. Ishitaki et al. [13] presented the application of Deep Recurrent Neural Networks (DRNNs) for prediction of user behavior in Tor networks. They constructed a Tor server and a deep web browser (Tor client). Dong et al. [14] discussed different methods which were used to classify network traffic, and used different methods on the open data set. They further experimented with these methods to find out the best way of intrusion detection. Roy et al. [15] used deep neural network as a classifier for the different types of intrusion attacks and did a comparative study with SVM.

In most of the papers presented in this section, a method for detecting intrusions in the network was considered, which makes it possible to detect a shorter range of attacks. Our proposed approach is to study and implement a new model of intrusion detection that combines anomaly detection and misuse detection to cover the suffering of various types of attacks. In some related work, the hybrid methods for intrusion detection have been used, but the proposed approach has been better than others in terms of the accuracy parameter.

3 Proposed Algorithm

The proposed solution is a two-part process that combines the methods of misuse detection and anomaly detection to address the dangers of intruder behavior between vms. Considering the issues examined in the case of IDS and the high volume of input data to the network and the high number of vms, the use of machine training methods can be appropriate, but one of the most important problems in the neural network is the false positive problem. To decrease the rate of false positives, we use the recurrent neural network with GRU and genetic algorithm simultaneously. We use the recurrent neural network to divide the inputs into two categories: the first group, which performs as a normal task, and the second one, which includes the works that are considered an attack. We send normal tasks to the genetic algorithm so that the tasks that have the highest risk are found.

In Fig. 2 you see the flowchart for the proposed approach. First, we normalize the data that is used to train the network. Data normalization is one of the

most important parts of the preprocessing of data so that the recurrent neural network can be properly trained. In the next step, we create the recurrent neural network and train it. After the network is trained, we test the network with new data. Tested data is entered into the network. This training and weighing of the parameters will continue as long as the accuracy of the validation is not improved in 3 epochs. After the network is trained, we test the network with new data. Test data is entered into the recurrent neural network, and After normalization, they are classified. In the genetic algorithm, using the training data, the initial population of chromosomes is constructed. Then the fitness function for each chromosome is calculated and new generations are created. These operations take place as long as the fitness function does not increase in 3 steps. During the generation, mutations in the chromosomes take place because the genetic algorithm can come to a better solution by using mutation. Eventually, a knowledge-base is constructed from rules. The knowledge base contains all known attacks. If a task is entered into the network and in terms of the values of the parameters (duration, src-bytes, wrong-fragment, dst-host-srv-error-rate and hot) is similar to the values in the knowledge-base rules, then it is detected as an attack, and the network administrator will be notified.

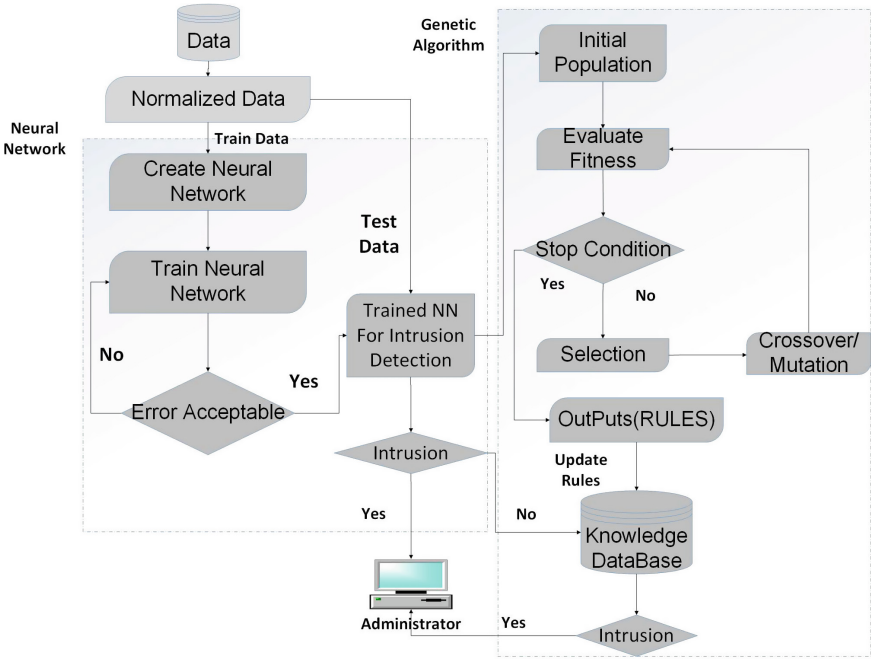

Fig. 2. WorkFlow of proposed IDS

In Fig. 3, we implemented this in the decision-makers (IDS) component. Decision makers (IDS) are deployed as a network-based IDS. The advantages of using

this approach are: its deployment has little impact on the network. Since NIDS are usually passive devices that listen to the network wire without interfering with the normal operation of a network and a large network can be monitored by a few well-placed NIDS. The decision maker (IDS) is a module that the network tasks enter. Then each task is parameterized and, after normalizing the data, the operation of the two stages of intrusion detection is performed on it.

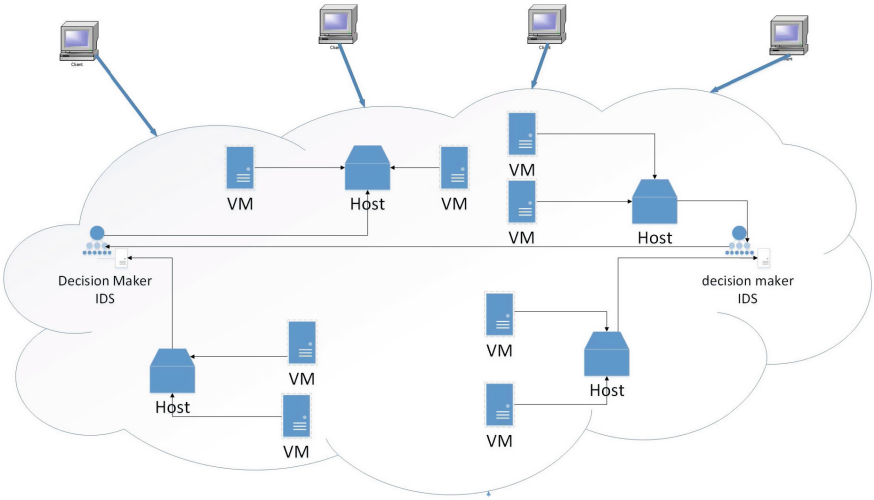

Fig. 3. Intrusion detection systems in network

A typical intrusion detection system consists of three functional components [20]:

Information source: It provides a stream of event records
Analysis engine: It finds signs of intrusions
Decision maker: It applies some rules on the outcomes of the analysis engine and decides what reactions should be done based on the outcomes of the analysis engine [21].

In this paper, we combine three components and call it Decision-maker (IDS). The decision-maker (IDS) is related to hosts for gathering information about vms and jobs that enter the network. First, anomalies are detected using the recurrent neural network. Then, we use the genetic algorithm to implement the misuse detection method. Decision makers inform the administrator to about attacks and update the knowledge database in incremental periods to increase accuracy. Decision makers are also associated with the general state of the network so that their knowledge databases are appropriate and up to date. Decision makers evaluate a task that enters into cloud computing, perform parallel processing and notify the network administrator. The lack of communication between decision makers and virtual machines makes them more secure and all the information required by decision makers (IDS) is collected through hosts.

3.1 Deep Neural Network

In the first step we normalized data and sent it to train the DNN model. The architecture of the GRU model is shown in Fig. 4. We create a four-layer GRU model and use drop out, which is a regularization technique introduced by Google for preventing of overfitting.

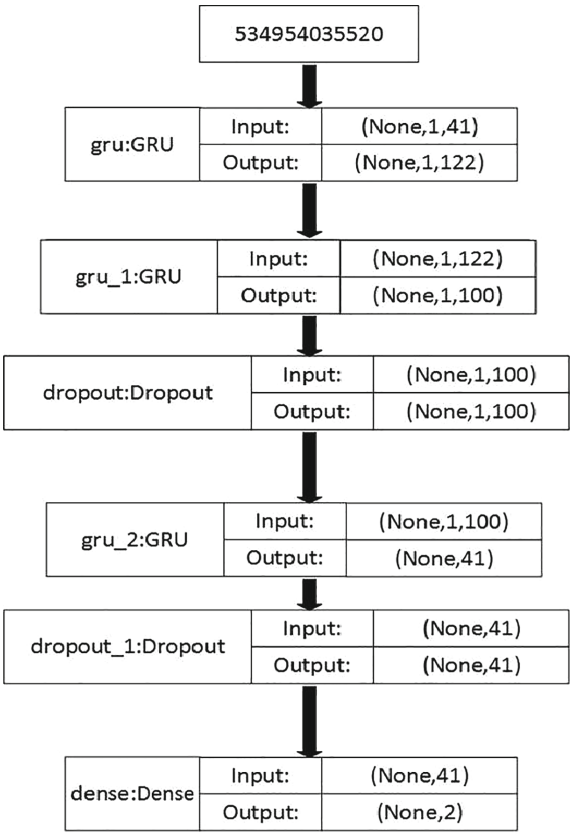

Fig. 4. Architecture of GRU (Proposed approach)

This model of DNN uses categorical-cross entropy for loss function and ADAM as an optimizer. ADAM is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [16]. We use the Softmax activation function in the last layer. Softmax is used to compute probability distribution from an array of real numbers. The softmax function produces an output which is a range of values between 0 and 1. We use parameters for stop training, i.e. the accuracy of validation data and the batch-size is 1000 and the max number of epochs is 100. The compilation time of creating this recurrent neural network is 0.118 s.

3.2 Genetic Algorithm

A genetic algorithm is used to detect attacks using misuse detection, which describes how to configure the genetic algorithm in this section. The number of generations to produce the optimal solution is 20.

Chromosome. In the genetic algorithms: each chromosome represents a possible solution to the problem that each chromosome consists of some constant genes. We use a decimal number to show each chromosome, which is the row of a rule in the training dataset. In Fig. 5 a representation of the chromosome can be observed. We describe five genes for each chromosome in our algorithm.

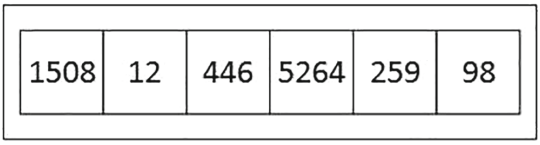

Fig. 5. Chromosome

Initial Population. The first stage of the genetic algorithm is the production of the initial population; to prevent early convergence, the initial population is randomly selected to cover a wide range of data. The fitness of the chromosomes is based on their gene fitness. The initial population size in our algorithm is 10000.

Selection Function. In this operator, from among the chromosomes in a population, some chromosomes are selected for reproduction. The elitism method is used to select the parent chromosomes to produce the children; we select 70% of the best chromosomes in terms of fitness to produce the next generation.

Crossver. As part of the integration process, parts of the chromosomes are replaced randomly. This makes the children have a combination of their parents' characteristics, so they do not exactly resemble their parents. In this solution, the multipoint crossover approach is used. In this method, various parts of the parent chromosomes are selected for the production of children.

Mutation. After completion of the crossover, the mutation operator is performed on the chromosomes. This operator randomly selects a gene from the chromosome and then changes the content of that gene. The mutation operator is used to avoid getting stuck in a local maximum or minimum. The probability of mutation in the solution presented in this solution is 0.1.

Fitness. To solve the problem using the genetic algorithm an appropriate fitness function must first be developed. The most important part of the genetic algorithm is a fitness function. As the function with higher is selected, more upper convergence is obtained, the algorithm runs faster, and the optimal answer is selected. Equation 1 shows the defined fitness function.

$$Fitness = \sum_{i=0}^{SC} \left(\frac{HTE(i) * CRL(i) * PR(i)}{HTR(i)}\right) \tag{1}$$

As seen in Eq. 1, SC is the size of a chromosome, HTE(i) is a count of rule i that is in the validation set of the genetic algorithm and it is known as an attack and HTR(i) is a count of rule i that is in the training set of the genetic algorithm and it is known as an attack. CRL(i) is considered as a conditional variable if the number of rule repetitions of i in the training data is lower than the threshold. This parameter is considered equal to the constant value of 4; otherwise, it is equal to 1 and PR(i) is the difference between the predictions of 2 classes of rule i in the recurrent neural network. If this difference is less than the threshold, then the parameter mentioned is equal to 1.3 and, if greater, it is equal to 1.

Considering the low amount of training data and the low difference between 2 class predictions in the recurrent neural network, the probability of error in the recurrent neural network test is increased. These last two parameters are used to find the optimal solution for the proper convergence of the genetic algorithm.

Early Stop Conditions. An early stop is determined so that the convergence of the algorithm does not go away and does not convert to divergence and it can be used to select more optimal solutions. Three conditions for early stopping of the genetic algorithm are considered:

1. The number of generations of offspring will exceed the threshold.
2. The mean of the fitness function in each generation is lower than the threshold.
3. The mean of the fitness function for the two consecutive generations is reduced.

4 Evaluation

The implementation of this paper has been done using the PYTHON programming language for the genetic algorithm, and it has been used by the TensorFlow library to implement the recurrent neural network. Table 1 shows the hardware system used to run the proposed approach.

In Table 2 the KDD Cup 99 dataset is shown. This dataset includes traffic with both normal and abnormal connections on the network and was used to evaluate the IDS performance objectively in the study. Each record has a field that shows 22 attack patterns or normal traffic and each row contains 41 features.

Table 1. Platform properties.

Properties	Values
Computer	Asus
CPU	Intel(R) Core(TM) i5-3230M CPU @ 2.60GHzCPU @ 2.60 GHz
RAM	6.00 GB (5.45 GB Usable)
Operating System	64-bit, Windows 8
IDE	Jupyter Notebook

Table 2. Data.

Type	Class	Number
ATTACK	DoS	391458
	Probe	391458
	R2L	4107
	U2R	1126
Normal	–	97278

Feature combinations of continuous and symbolic values and a data transformation were needed to convert them. Attacks can be categorized into exactly one of four types, as detailed below:

Denial of Service Attack (DoS): These types of attacks are usually related to busy resources and this high volume of requests sent to resources is related to the attackers to reject legitimate users' requests.

User to Root Attack (U2R): This is a type of security exploitation that allows an attacker to gain access to a normal user through conventional instruments, enabling the root to be accessed through the identification and exploitation of existing vulnerabilities.

Remote to Local Attack (R2L): This is when an attacker attempts access to a system over a network. The attacker can only transmit data packets over the network and the attacker tries to gain access to the machine by exploiting some vulnerability.

Probing Attack (Prob): This is when an attacker attempts to acquire information from a network to avoid the system's security protocols.

In this section, we use metrics for assessing the classification of the recurrent neural network that defines these metrics as follows:

True Positive (TP). Attack data that is correctly classified as an attack.

False Positive (FP). Normal data that is incorrectly classified as an attack.

True Negative (TN). Normal data that is correctly classified as normal.

False Negative (FN). Attack data that is incorrectly classified as normal.

We will be using the following measures to evaluate the performance of our proposed approach:

in Eq. 2 the accuracy measures the proportion of the total number of correct classifications.

$$Accuracy = (\frac{TP + TN}{TP + TN + FP + FN}) \tag{2}$$

in Eq. 3 the precision measures the number of correct classifications penalized by the number of incorrect classifications.

$$Percision = (\frac{TP}{TP + FP}) \tag{3}$$

in Eq. 4 The recall measures the number of correct classifications penalized by the number of missed entries.

$$Recall = (\frac{TP}{TP + FN}) \tag{4}$$

in Eq. 5 The false alarm measures the proportion of benign events incorrectly classified as malicious.

$$FalseAlarm = (\frac{FP}{FP + TN}) \tag{5}$$

in Eq. 6 The F1-score measures the harmonic mean of precision and recall, which serves as a derived effectiveness measurement.

$$F1\text{-}score = 2 * (\frac{Precision * Recall}{Precision + Recall}) \tag{6}$$

In this step, we show a graph of different parameters during training of DNN.

Figure 6 shows the degree of accuracy in the correct classification of the data. At the initial steps gradient of the graph, it is ascending and the validation accuracy rate and training accuracy rate are more than 99%.

The loss function is an important section in recurrent neural networks, which is used to measure the inconsistency between a predicted value and true value. It is a non-negative value. In Fig. 7 we show a graph of loss during the training. In each step, the level of the loss function is descending and lower than 0.02 in validation inputs.

In Fig. 8 the ROC graph is present. The receiver operating characteristic (ROC) curve it is a tool for visualization that can be utilized to determine whether a classifier is appropriate in terms of cost sensibility. It is the ratio of the false positive (FP) to true positive (TP) rate. This graph shows that fewer false positives than true positives have been created. The micro-average calculated the ROC curve between y-test and y-prediction and the macro-average calculated between all FPR (False Positive Rate) and mean TPR (True Positive Rate). In Table 3 the values of important parameters of two classes in the recurrent neural network are shown.

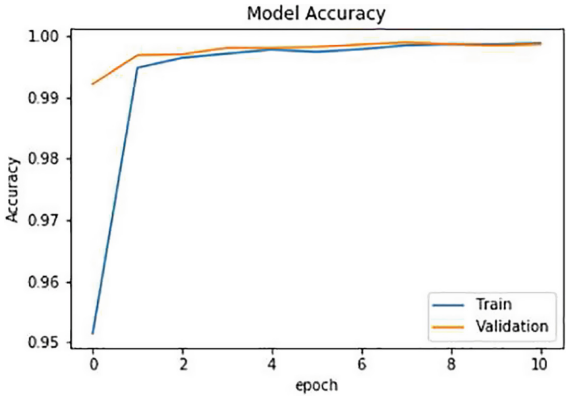

Fig. 6. Model accuracy

In Fig. 9, you can see the mean of the fitness function of the generation of chromosomes in each generation. As stated in the previous section, after sorting out chromosomes, 70% of the best are selected and some of them are used to produce children. In this graph, the improvement in the amount of fitness function is shown in general in each generation. At each stage, due to the reduction in the number of chromosomes and the selection of more relevant chromosomes, the total value of the fitness function is higher.

The condition intended to stop the genetic algorithm is that the algorithm stops if the algorithm does not provide an improvement in the two successive

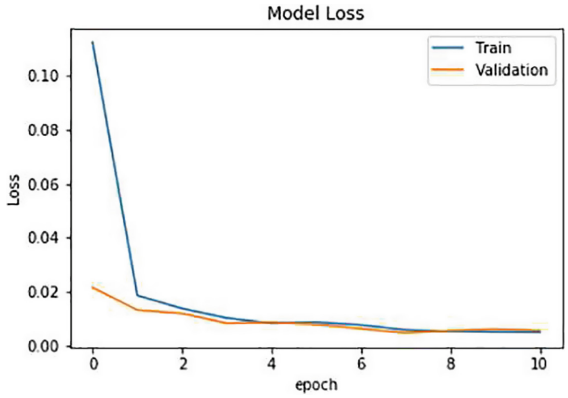

Fig. 7. Model loss

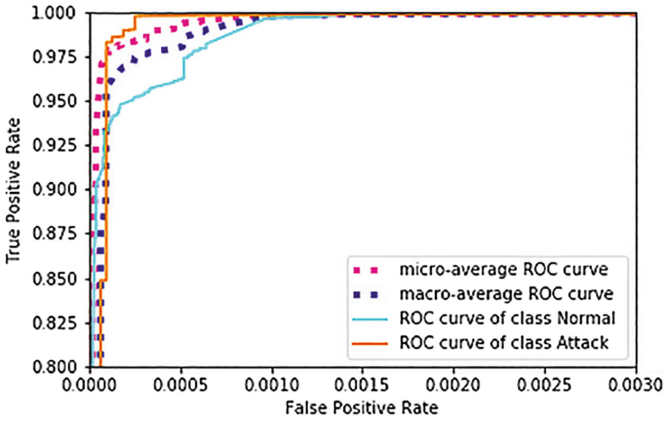

Fig. 8. ROC

Table 3. Results of recurrent neural network.

Type	F1-score	Recall-score	Precision-score
Normal	99.71	99.87	99.54
Attack	99.92	99.88	99.97

generations. This provides a good convergence representation of the genetic algorithm in obtaining the optimal set.

In Fig. 10, the fitness of each chromosome has shown in all generations. This graph shows the total number of chromosomes created in all generations and it shows that after an initial generation with a chromosome number of 10,000, there is an upward trend in the amount of fitness function. The most important challenge in heuristic algorithms is avoiding getting stuck in a local maximum

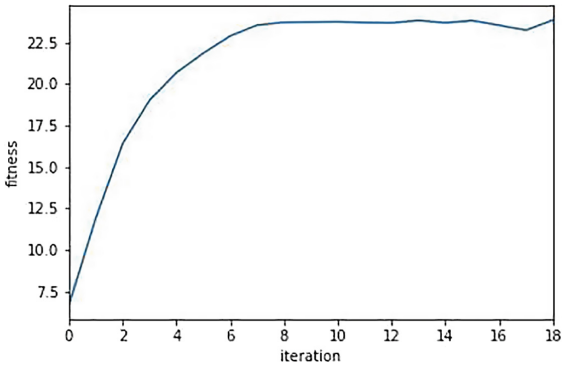

Fig. 9. Average of fitness in each generation

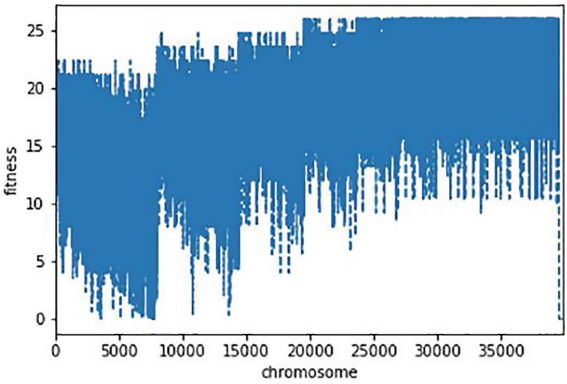

Fig. 10. Fitness function

or a local minimum. In the proposed algorithm we have tried to pass through this local maximum or local minimum with proper configuration, such as with proper use of the mutation operator. You can see the results of the mutation operator in this graph.

In this section, we compared our proposed approach with other papers in this field. In Table 4, the comparison between the proposed solution and the related work that has been carried out in recent years in this field suggests an improvement in the accuracy parameter in detecting attacks on the network.

Our proposed approach is highly efficient at decreasing the false positive rate, but it is challenging in terms of its running speed, which can be eliminated to a large extent by taking into account the strong hardware equipment. We have already seen good results and we are trying to increase the speed and accuracy with new technologies such as GRU in the recurrent neural network and the proper configuration of the genetic algorithm.

Table 4. Results of all competitive methods.

Techniques	Accuracy
CHI-SVM [17]	98
GA+FLN [18]	99.69
PSO+FLN [18]	99.68
LSTM [12]	96.93
GRU+SVM [19]	84.15
Proposed (GRU+GA)	99.91

5 Conclusion

Today, intrusion detection has become one of the most important challenges facing networks and this has led many researchers to focus on this field. In the proposed method, given that the existing networks are wide and the use of traditional methods is not responsive, machine learning methods and heuristic algorithms have been used. In our proposed approach, we have tried to consider the accuracy and speed of the network intrusion detection process by using new technologies such as GRU in the recurrent neural network, as well as the proper configuration of the genetic algorithm that can reduce the false positive rate. In this method, both anomaly detection and misuse detection methods are used simultaneously so that a wide range of attacks can be reported to administrators of the network. We have implemented our proposed model in TensorFlow and performed extensive evaluations on its capabilities. For our evaluations, we have utilized the benchmark KDD Cup '99 dataset and achieved desirable results. Our results have demonstrated that our approach offers high levels of accuracy with reduced training time because we have a suitable configuration in GRU and the genetic algorithm. In the last step, we compared the proposed algorithm with other related work over the last few years, which demonstrated the proposed solution's superiority to other solutions, given that the proposed solution has shown more than 99.91% accuracy.

References

1. Mishraa, P., Varadharajan, V., Pilli, E., Varadharajan, V., Tupakulab, U.: Intrusion detection techniques in cloud environment: a survey. J. Netw. Comput. Appl. **77**, 18–47 (2017)
2. Barbara, D., Jordia, S.: Applications of Data Mining in Computer Security, vol. 6. Springer, New York (2002). https://doi.org/10.1007/978-1-4615-0953-0
3. Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., Payne, B.D.: Evaluating computer intrusion detection systems: a survey of common practices. ACM Comput. Surv. (CSUR) **48**(1), 12 (2015)
4. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP (2014)
5. Subba, B., Biswas, S., Karmakar, S.: A neural network based system for intrusion detection and attack classification. In: Twenty Second National Conference on Communication (NCC). IEEE, India (2016)
6. Xu, C., Shen, J., Du, X., Zhang, F.: An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access **6**, 48697–48707 (2018)
7. Nguyen, S., Nguyen, V., Choi, J., Kim, K.: Design and implementation of intrusion detection system using convolutional neural network for DoS detection. In: Proceedings of the 2nd International Conference on Machine Learning and Soft Computing, ICMLSC 2018. ACM, Vietnam (2018)
8. Fu, Y., Lou, F., Meng, F., Tian, Z., Zhang, H., Jiang, F.: An intelligent network attack detection method based on RNN. In: Third International Conference on Data Science in Cyberspace (DSC). IEEE, China (2018)

9. Balamurugan, V., Saravanan, R.: Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation. Cluster Comput. 1–13 (2017)
10. Kim, J., Shin, N., Jo, S., Kim, S.: Method of intrusion detection using deep neural network. In: International Conference on Big Data and Smart Computing (BigComp). IEEE, South Korea (2017)
11. Maleh, Y., Ezzati, A., Qasmaoui, Y., Mbida, M.: A global hybrid intrusion detection system for wireless sensor networks. Procedia Comput. Sci. **52**, 1047–1052 (2015)
12. Kim, J., Kim, J, Thu, HLT., Kim, H.: Long short term memory recurrent neural network classifier for intrusion detection. In: International Conference on Platform Technology and Service. IEEE, South Korea (2016)
13. Ishitaki, R.T., Obukata, Y., Oda, T., Barolli, L.: Application of deep recurrent neural networks for prediction of user behavior in Tor networks. In: 31st International Conference on Advanced Information Networking and Applications Workshops. IEEE, Taiwan (2017)
14. Dong, B., Wang, X.: Comparison deep learning method to traditional methods using for network intrusion detection. In: 8th IEEE International Conference on Communication Software and Networks. IEEE, China (2016)
15. Roy, S.S., Mallik, A., Gulati, R., Obaidat, M.S., Krishna, P.V.: A deep learning based artificial neural network approach for intrusion detection. In: Giri, D., Mohapatra, R.N., Begehr, H., Obaidat, M.S. (eds.) ICMC 2017. CCIS, vol. 655, pp. 44–53. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-4642-1_5
16. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference for Learning Representations. arXiv, USA (2015)
17. Thaseen, I.S., Kumar, C.A.: Intrusion detection model using fusion of chi-square feature selection and multi class SVM. J. King Saud Univ. Comput. Inf. Sci. **29**(4), 462–472 (2017)
18. Ali, M.H., Al Mohammed, B.A.D., Ismail, A., Zolkipli, M.F.: A new intrusion detection system based on fast learning network and particle swarm optimization. IEEE Access **6**, 20255–20261 (2018)
19. Agarap, A.F.M.: A neural network architecture combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for intrusion detection in network traffic data. In: Proceedings of the 2018 10th International Conference on Machine Learning and Computing, ICMLC 2018. ACM, China (2018)
20. Gurley Bace, R.: Intrusion Detection. Sams Publishing, USA (2000)
21. Yao, J., Zhao, S., V. Saxton, L.: A study on fuzzy intrusion detection. In: Proceedings Volume 5812, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security (2005)