International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

An Intruder Detection System based on Feature Selection using Random Forest Algorithm

G. Madhukar, G. Nantha Kumar

ABSTRACT---In every part of the world, there is tremendous growth in digital literacy in the present era. People are trying to access internet-based applications with the use of digital machines. As a result, the internet has become a primary requirement for everyone, and most business transactions often take place conveniently across the network. On the other hand, intruders involved in making intrusions and doing activities such as capturing passwords, compromise on the route, collecting details of credit cards, etc. Many malicious activities are taking place over the network due to this intruding activity on the internet. Applications such as host-based Intrusion Detection System (IDS) and network-based IDS have previously been used to control network intruders. Mostly when they come with Encrypted packets, spoofed network ids, these techniques were not able to control intruders promisingly. It is essential to examine these types of attacks periodically to identify patterns of recent attacks. In this paper, the authors have proposed a model based on deep learning by using the NSL – KDD dataset to solve these problems. For later train, the model with data with a random forest classifier algorithm, the principal component analysis applied for feature selection. The model is designed to detect patterns of intruders effectively using the knowledge gained from training data. To detect malicious patterns over the network, the model shows a sufficient accuracy of around 90 percent.

Keywords: Feature selection, Intrusion detection, Random forest, Principle component analysis, NSL-KDD dataset

I. INTRODUCTION

Intruder sometimes referred to as a hacker. An Intruder is a person who attempts to gain unauthorized access to a system to harm the system or disturb the system's data. Any unauthorized access or performing malicious activities of the network is known as Intrusion. Data transfer across the network channel is increasing rapidly day by day. There is also a massive increase in intrusions in parallel to user growth in the network. In this case, if sufficient methods for detecting intruders were not used, then various malicious activities such as the denial of service attacks, standard gateway interface scripts, protocol-specific attacks, traffic flooding, Trojans, worms, etc. may regularly disturb the users. Several researchers suggest different solutions to avoid such a network atmosphere. Such approaches are hardware-related, which is often known as intruder detection systems. The use of a firewall is one of the easiest ways to block unauthorized users. However, in many situations, the firewall may fail under different strategies used by hackers. In addition to firewalls other softwares/hardware such as SolarWinds Security Event Manager, Snort, Bro, Suricata, IBM QRadar, Security Onion, Open WIPS-NG, Sagan, Splunk are some of the intruder detection systems available on the market. HIDS is an example program that manages important files and application data. Whereas NIDS is used for inbound network traffic analysis on the network. Detection of data points, objects, observations or events that do not fit the anticipated pattern of a group. These incidents are uncommon but may pose the most important and significant threat, such as cyber intrusions or fraud. Such phenomena of detection are generally classified by signature and intrusion detection based on anomalies.

a) Signature-based Intrusion Detection: Such a detection system stores attack patterns in the database. Whenever there is a connection, patterns will be checked in the database. The system decides whether or not the potential contact is an intruder based on the patterns availabe in the database. This type of model's accuracy is very high. The model works, however, if a hacker does minimum professional attacks. When one type of attack pattern doesn't work, the hacker may adjust the technique that is not available in the database. This model often needs to modify the server frequently.

b) Anomaly-based Intrusion detection: Although detection based on signatures compares behaviour with rules, detection based on anomalies compares behaviour with profiles. Such models work beyond the strategies based on signatures. Such technologies can also identify previously unknown trends by producing the input link activity profile and based on which patterns leading to attacks are defined. By learning from the data, these methods use machine learning techniques to build the models. The following figure1 illustrates both methods' actual execution cycle.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
G. Madhukar*, Research Scholar, Sri SatyaSai University of Technology & Medical Sciences, Sehore (M.P.), India.
G. Nantha Kumar, Associate Professor, Sri SatyaSai University of Technology & Medical Sciences, Sehore (M.P.), India.

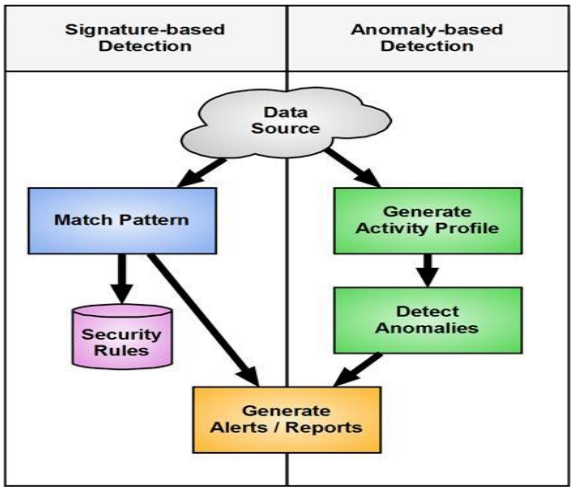

Figure1: Execution process of signature and anomaly detection techniques

Conventional IDS has many disadvantages, such as noise and other errors, which can reduce the accuracy of the algorithm. When new attacks are detected there may an issue in detecting them. Because a signature-based IDS cannot detect unknown attacks, it requires constant updates to keep new signatures up to date. In comparison, most IDS systems do not address protocol-based attacks and are limited to managing network anomalies. In detecting the threats, machine-based learning methods were applied to solve these problems. Supervised learning, unsupervised learning, and semi-supervised learning are the models available in artificial intelligence. The reliability of each model depends mostly on the selected dataset and features used in model construction. Since the dataset may have many features and all the features do not contribute to the development of the system, eliminating the regular feature from the dataset is necessary.

The rest of the paper prepared according to the following: Section 2 describes some research related to the detection of intrusion. Section 3 outlines the approach proposed. The experimental findings and visual analysis have discoursed in Section 4. Section 5 sets out the interpretation and future research.

II. LITERATURE REVIEW

In the study, several intrusion detection methods have described. Intrusion detection has been receiving much attention among researchers in recent times as it is commonly used to maintain protection within a network. Here, we describe a few of the intrusion detection techniques used.

J. Brownlee [1], is discussed about all the types of Machine Learning Algorithms. E. K. Viegas, A. O. Santin and L. S. Oliveira [2] have used Decision tree and Naïve Bayes algorithms for classifying intrusions Probe and DoS attacks on the data source TRAbID. A. Verma and V. Ranga [3] discussed in their paper as Statistical analysis of the dataset CIDDS-001 labels of the flow-based data source. The authors used OpenStack database clustering models and k-means clustering algorithms for network intrusion device evaluation.T. Hamed, R. Dara, and S. C. Kremer [4]authors studied in 2018 about NIDS using recursive feature addition and SVM algorithms on the dataset ISCX-2012. C. R. Wang, S. J. Lee, R. F. Xu, and C. In 2018, H. Lee[5] achieved approximately 99% of binary and multivariable classification using Network Intrusion Detection and Equality Restricted-Optimization-based Intense Data Source Training Systems 10% KDD, KDD DoS, NSL-KDD, UNSW-NB15. G. Fernandes, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proença [6] completed NIDS work using primary element analysis and colony optimization IP flows. The researchers have investigated the identification of anomalies based on network traffic profiling. The authors proposed and compared methods of detection from various algorithm groups. The process of detection is developed utilizing adaptation to patterns, the uses of real and simulated transport to test the methods proposed.

A. H. Hamamoto, L. F. Carvalho, T. Abrão, L. D. H. Sampaio, and M. L. Proença[7] used the genetic algorithm and fuzzy logic structures to tackle actual network traffic. W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri [8] proposed a system with multi-level devices for hybrid intrusion detection using vector and extreme learning devices to increase the detection of known and unknown attacks. SumaiyaThaseen and C. Aswani Kumar [9] suggested the intrusion detection system with a Chi-square choice function and SVM. A tuning technique is used to improve the kernel parameter of the Radial Base Function. They experimented with findings and reduced false alarm levels. R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, and Y. L. He [10] proposes a new semi-supervised learning strategy based on fuse using unlabeled samples with supervised learning algorithms to enhance the IDS performance of the classifier. The method for generating the fluid vector and non-labeling samples variable quantities (low, medium, high unpredictability), is the Special Hidden Layer Neural Method (SLFN). The hybrid approach proposed by U. Ravale, N. Marathe, and P. Padiya[11] provides a clustering technique such as the k-Means Algorithm and the RBF kernel function of the support vector machine method. The technique proposed aims to decrease every data point's number of attributes. The proposed technique is, therefore, more successful when applied to the KDDCUP'99 data set in terms of detection speed and accuracy. The new approach to the identification, V. Hajisalem and S. Babaie[12] developed through the combination of Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms. By using FCM and Correlation-based Attributes Selection (CFS) techniques for the division of training data and elimination of non-correlated attributes. Moreover, if-then rules are created using the CART technique to distinguish between normal and defect records according to the selected features. The proposed hybrid method also trained through the rules developed. The results of the simulation for NSL-KDD and UNSW-NB15 data sets show that the proposed method is above performance measure and detected at 99%. As search strategy and logistic regression C. Khammassi and S. Krichen [13], we apply a wrapper approach based on a genetic algorithm to choose the subset of network intrusion detection systems.

The experiment is conducted using the KDD99 and UNSW-NB15 data sets. Three different decision tree classifiers calculate the quality of the selected function subsets. M. R. Gauthama Raman, N. Somu, K. Kirthivasan, R. Liscano, and V. S. Shankar Sriram[14] present an efficient and reliable IDS using the parameter setting Hypergraph-based Genetic algorithm and the feature selection accuracy Logistic Regression. The results of this method are accurate, minimum false positive rate, and the optimal number of features. The researchers S. Shitharth and D. Prince Winston proposed for intrusion weighted particle-based optimization of search (IWP-CSO) and neural hierarchical neuron architecture (HNA-NN) [15]. This is aimed at detecting and classifying intrusions on the basis of configuration in a Supervisory Control and Data Acquisition (SCADA) network.

S. M. BamakanHosseini, H. Wang, T. Yingjie, Y. Shi[16] suggested multiple linear programming criteria and optimization of particle swarms to improve the reliability of attack detection. Multiple linear programming criteria (MCLP) is a classification numerical programming approach that has shown the ability to solve problems in actual data mining. H. Wang, J. Gu, and S. Wang [17] proposed an active intrusion detection system using an integrated intrusion charging machine (SVM). The marginal density transformation of the logarithm is mainly applied to provide the original features for achieving new transformed features that greatly enhance SVM system detection.

III. PROPOSED MODEL

This paper proposes the Random Forest model with Feature selection using principal element analysis to resolve the problems in traditional intrusion detection systems. This approach offers a predictive analytical approach. In Figure2, various kinds of attributes available in the NSL-KDD data set, including class, are shown. To train the model, we used the NSL-KDD data set. In Table1, this data set includes four types of attacks. Table2 shows the types of attacks and their attack classes.

The data set consists of 42 features, 41 features grouped into four categories, such as essential features, content features, time-based, and host-based features. The last feature is about all the data of other features

F#	Feature name	F#	Feature name	F#	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

Figure2: Different Features in Data Set

Table1: Features under various categories

Category	Features
Basic Features	Feature 1 to Feature 10 all
Content Features	Feature 11 to Feature22 all
Time-based Features	Feature 23 to Feature 31 all
Host-based Features	Feature 32 to Feature 41all

Table2. Different major Classes of Attacks

Class	Type of Attack
DoS (10)	Worm, Back, Processtable, Land, Udpstorm, Neptune, Apache2, Pod, Smurf, Teardrop
Probe (6)	Ipsweep, Satan, Nmap, Saint, Portsweep, Mscan
R2L (16)	Guess_Password, Named, Ftp_write, Imap, Sendmail, Phf, Multihop, Warezmaster, Httptunnel,Warezclient, Spy, Xlock, Xsnoop, Snmpgetattack, Snmpguess,
U2R (7)	Snmpgetattack,Buffer_overflow, Load module, Perl, Rootkit, Xterm, Ps, Sqlattack,

The model proposed for intrusion detection in the network displayed in Figure3. The NSL-KDD data set submitted to the pre-processing mechanism, principal component analysis approach applied to select the number of features from the available key-value features. Once the data obtained for the final feature attributes, then the random forest method used to train the model by considering 70% of final feature attributes. The model tested for the remaining 30% of data attributes to predict the network can be either healthy or attack.

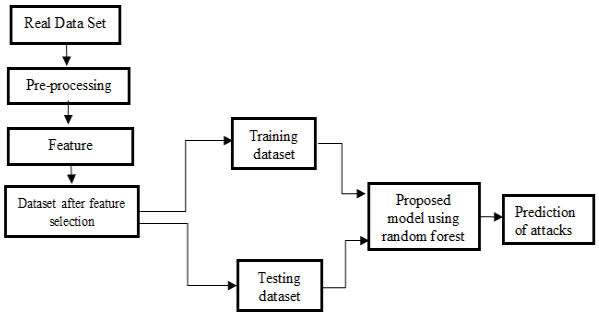

Figure3: Proposed Architecture

The following are the steps in PCA

Algorithm PCA

1. Normalize the data
2. Calculate the covariance matrix using Cov(A, B) = $\frac{1}{n-1}\sum_{i=1}^{n}(Ai - a^{-})(Bi - b^{-})$
3. Use $(\lambda I - A)v = 0$ *to* find the eigenvalues and eigenvectors
4. Choose the components and frame feature vector
5. Form principal components

The following are the steps involved in the random forest algorithm

Algorithm RandomForest(x,K,N,M)

1. Fork = 1 to N

a. Draw a bootstrap sample of size N from the training set

b. Grow a Random Tree Θ(k) on the bootstrap data, by recursing on each un-visited node:

c. While stopping criteria are not reached

i. Select m variables at random from the set of an input variable M

ii. Pick the best variable/split-point among them

iii. Split the node into child nodes on the corresponding variable

2. Return the ensemble of trees{h(x,Θk),k= 1,...}

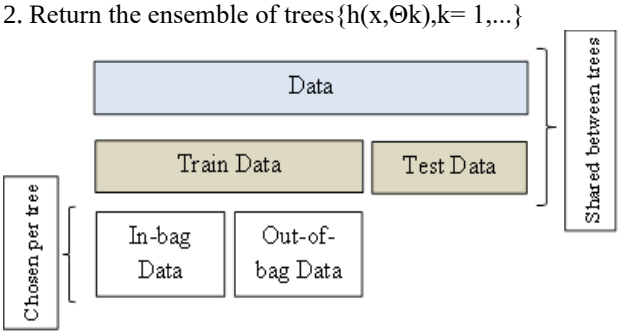

Figure4: The data partition for a Random Forest Classifier

The original algorithm of Random Forest extends the bagging idea by introducing a Random Feature Selection. The Random Forest algorithm builds multiple decision tree classifiers, each built with Random Tree. The Random Tree algorithm modifies the standard construction of a decision tree with randomization. At every node, a subset of features m selected from an original set of features M. After this, the best attribute among m selected as a split point according to an impurity measure. In the original proposal, Breiman used the Gini Index to choose the best variable among randomly selected ones. The trees are built to maximal depth and not pruned. The instance processed for each tree in the ensemble to identify a new example. The majority of recommendations define the resulting class.

IV. RESULTS AND DISCUSSION

The model constructed from 59270 NSL-KDD dataset training records, which we tested with a test data set containing 13130 data records. This model evaluated with various measures like accuracy, precision, recall, F-measure for finding four classes of attacks is calculated with the following formulae.

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

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1-Measure} = \frac{2TP}{2TP+FP+FN}$$

Where TP is TruePositive, TN is TrueNegative, FP is FalsePositive, and FN indicatesFalseNegative

The results calculated for Random Forest classifier on test data of the NSL-KDD feature selection attributes shown in the following Figure5. As per the figure, the random forest classifier shows satisfactory performance with more than 95% accuracy for predicting most of the attack types.

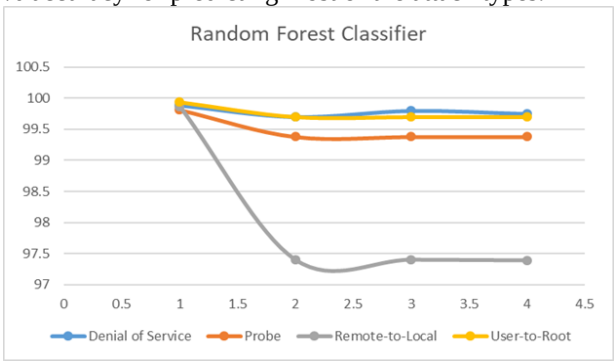

Figure5: Random Forest Classifier measurements

When we interpret the algorithm performance measures under each class, different results are generated and shown in the following figure6 to figure9. These figures show the accuracy obtained by the developed model. As per the results, all four attacks DOS, PROBE, R2L and U2R are predicted with accuracy more than 99%. By observing the results, it is evident that the model can predict the attacks by using machine learning techniques.

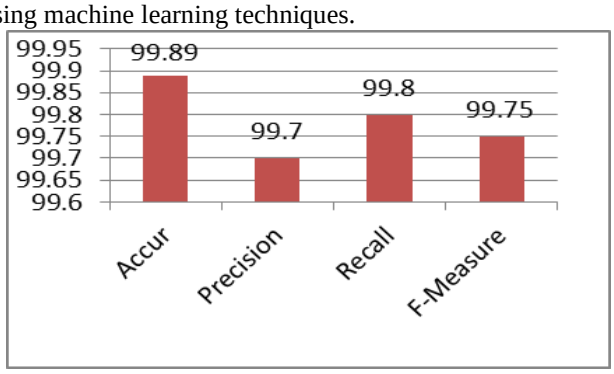

Figure6: Attack Class: DoS

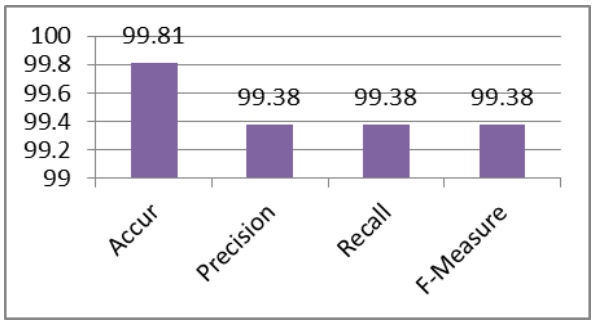

Figure7: Attack Class PROBE

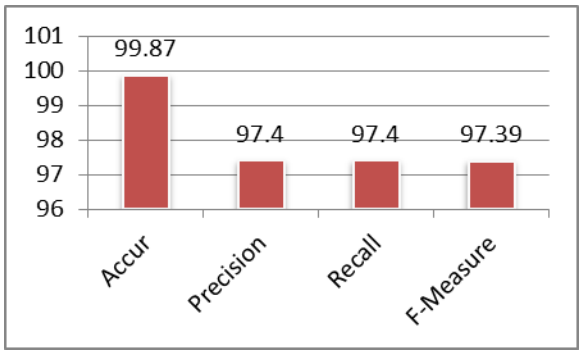

Figure8: Attack Class R2L

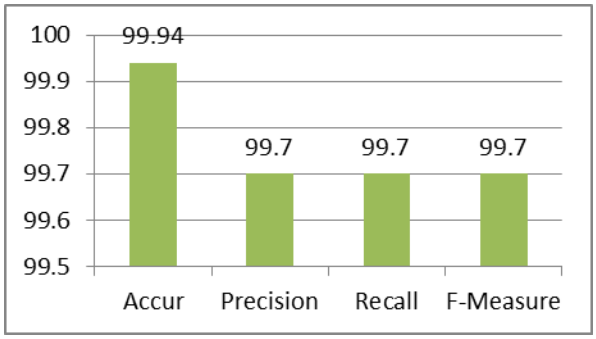

Figure9: Attack Class U2R

V. CONCLUSION

In this paper, we implemented a machine learning technique to train the model with NSL-KDD data set, namely Random forest classifier. This article includes all 41 features with 72400 records of the NSL-KDD data set. For the selection of most contributing attributes, the principal component analysis algorithm. A random forest classifier applied to the attained components for detecting high precision attacks. The experimental results for User-to-Root accuracy show that the accuracy is high about 99.94% in identifying all types of attacks. Whereas precision and recall of U2R shown as 99.7, and F-measure outline 99.7 percent for assessing U2R attack.

REFERENCES

1. J.Brownlee, "A Tour of Machine Learning Algorithms," https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ 2013
2. Soldatos, John. "Part IV: Real world AI applications in." *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies* 160 (2007): 291.
3. Patil, Purushottam R., Yogesh Sharma, and Manali Kshirsagar. "MINDS: Machine Intelligence Based Network Intrusion Detection System."
4. T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," Computer Security, vol. 73, pp. 137–155, 2018.
5. C. R. Wang, R. F. Xu, S. J. Lee, and C. H. Lee, "Network intrusion detection using equality constrained-optimisation-based extreme learning machines," Knowledge-Based Syst., vol. 147, pp. 68–80, 2018.
6. G. Fernandes, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proença, "Network anomaly detection using IP flow with Principal Component Analysis and Ant Colony Optimization," Journal of Networks Computer Applications, vol. 64, pp. 1–11, 2016.
7. A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, "Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic," Expert Systems Applications, vol. 92, pp. 390–402, 2018.
8. W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," Expert Systems Applications, vol. 67, pp. 296–303, 2017.
9. I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using a fusion of chi-square feature selection and multi-class SVM," Journal of King Saudi University - Computer Information Sciences, vol. 29, no. 4, pp. 462–472, 2017.
10. R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, and Y. L. He, "Fuzziness based semi-supervised learning approach for an intrusion detection system," Information Science, vol. 378, pp. 484–497, 2017.
11. U. Ravale, N. Marathe, and P. Padiya, "Hybrid intrusion detection system based on feature choice with K means and the RBF kernel function," Procedia Computer Science, vol. 45, no. C, pp. 428–435, 2015.
12. V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," Computer Networks, vol. 136, pp. 37–50, 2018.
13. C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," Computer Security, vol. 70, pp. 255–277, 2017.
14. M. R. Gauthama Raman, N. Somu, K. Kirthivasan, R. Liscano, and V. S. Shankar Sriram, "An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine," Knowledge-Based Systems, vol. 134, pp. 1–12, 2017.
15. S. Shitharth and D. Prince Winston, "An enhanced optimization-based algorithm for intrusion detection in SCADA network," Computer Security, vol. 70, pp. 16–26, 2017.
16. S. M. Hosseini Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," Neuro Computing, vol. 199, pp. 90–102, 2016.
17. H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," Knowledge-Based Systems, vol. 136, pp. 130–139, 2017.