Örgütsel Davranış Araştırmaları Dergisi
Journal Of Organizational Behavior Research
Cilt / Vol.: 7, Sayı / Is.: S, Yıl/Year: 2022, Kod/ID: 22S0-849
2528-9705

Improved Artificial Neural Networks based on Particle Swarm Optimization for Intrusion Detection in Computer Networks

Seyed Hedayat Mohammadi

Graduated in Computer Science from Yasouj Branch Azad University

*Corresponding Author
E-mail: Hedayat0148@gmail.com

ABSTRACT

Over the past few years, the increasing use of computer networks and binary information transmission has highlighted the importance of information security and intrusion detection in computer systems and networks. Simultaneously with the introduction of new and diverse attacks, various methods and systems in intrusion detection systems are proposed. An intrusion detection system is essentially a set of tools, methods, and documentation to detect and report unauthorized network activity. Static techniques, clustering, and learning can be introduced as the most common intrusion detection methods. One of the well-known models in intrusion detection is modeling based on neural networks. This model has shown outstanding performance in various fields, from forecasting to diagnosis and classification. One of the challenges of neural networks is the regulating parameters by an expert. This study aims to improve neural network performance by providing a hybrid model for automatic regulating of network parameters. The NSL-KDD dataset is considered a modified version of KDD-CUP99 to test and evaluate the proposed model. The results proved that the proposed model has a better learning capability than the conventional neural networks.

Keywords: *Neural networks, Parameter regulating, Particle swarm algorithm, Security, Intrusion detection*

INTRODUCTION

Today, computer networks monitor and manage critical infrastructures such as banking, transportation, commerce, and telecommunications. As a result, securing these systems against planned attacks is crucial. Most of these attacks exploit software errors and security vulnerabilities in the target system. It is impossible to eliminate software errors, so all software has security gaps called "software vulnerabilities."

Nowadays, different neural network systems are used to detect and discover attacks on computer networks. These systems provide training, testing, regulating, and applying neural networks in an attack detection system.

Study [1] investigated the performance of ART neural networks (ART - 1 and ART - 2) in intrusion detection and compared the results with the results of self-organizing neural network (SOM)

In [2], rough set theory and neural networks select important features and classify network traffic into normal and attack classes, respectively. Model training and testing were performed on the KDD-CUP99 database. The results show that training and testing time is significantly reduced, attack detection rates are increased, and false alarm rates are reduced.

In [3], an absorption criterion is calculated by the nearest neighbor and the nearest competitor (the closest pattern belonging to another class) for each pattern. The pattern p ϵ T is generally absorbed when | p-x | - | p-w | > δ where w ϵ S, x is the nearest neighbor p and w is the nearest competitor p.

In [4], an intrusion detection mechanism based on an intelligent security architecture using recurrent neural networks (RNN). The performance of the proposed security solution on a wireless sensor network system has been evaluated. It can successfully detect the presence of any suspicious sensor nodes and unusual activity at the base station. The performance overhead of the proposed solution is also calculated.

Study [5] focuses on improving the intrusion system in wireless LAN using support vector machines (SVM). Test data used were extracted from a computer lab. SVM performs intrusion detection based on known attack patterns. The simulation results show that the proposed detection system can detect anomalies and trigger the alarm. In addition, the proposed method has better detection efficiency, lower false alarm rate, better coverage and more effective detection.

In [6], two computational neural network models, namely the general regression neural network (GRNN) model and the multilayer perceptron neural network (MPNN) model, have been used for the host intrusion detection system. The proposed models are tested using log files generated by a computer. The results show that the host intrusion system model (HISM) significantly increases the detection accuracy while having the lowest false notification rate.

Reference [7] has introduced IB2 and IB3 methods as incremental methods. IB2 selects patterns that are not properly classified by 1-NN (such as CNN). IB3 is also an extended version of IB2 using a classification record to identify patterns that need to be maintained.

Study [8] introduced five k-NN-based methods called DROP5, .., and DROPI. The core of these methods is the concept of dependency. Another method that is based on dependency is the iterative case filtering (ICF) algorithm [9]. This algorithm is based on the sets of availability and coverage p, which are called neighboring sets and dependencies, respectively.

According to the above, this study aims to provide improved artificial neural networks based on particle swarm optimization to detect intrusions in computer networks.

Research Methodology

Neural network is one of the most common methods of modeling complex and large problems involving hundreds of variables. Neural networks can be used for classification problems (problem output is a class) or regression problems (problem output is a numeric value). The proposed algorithm is briefly introduced below:

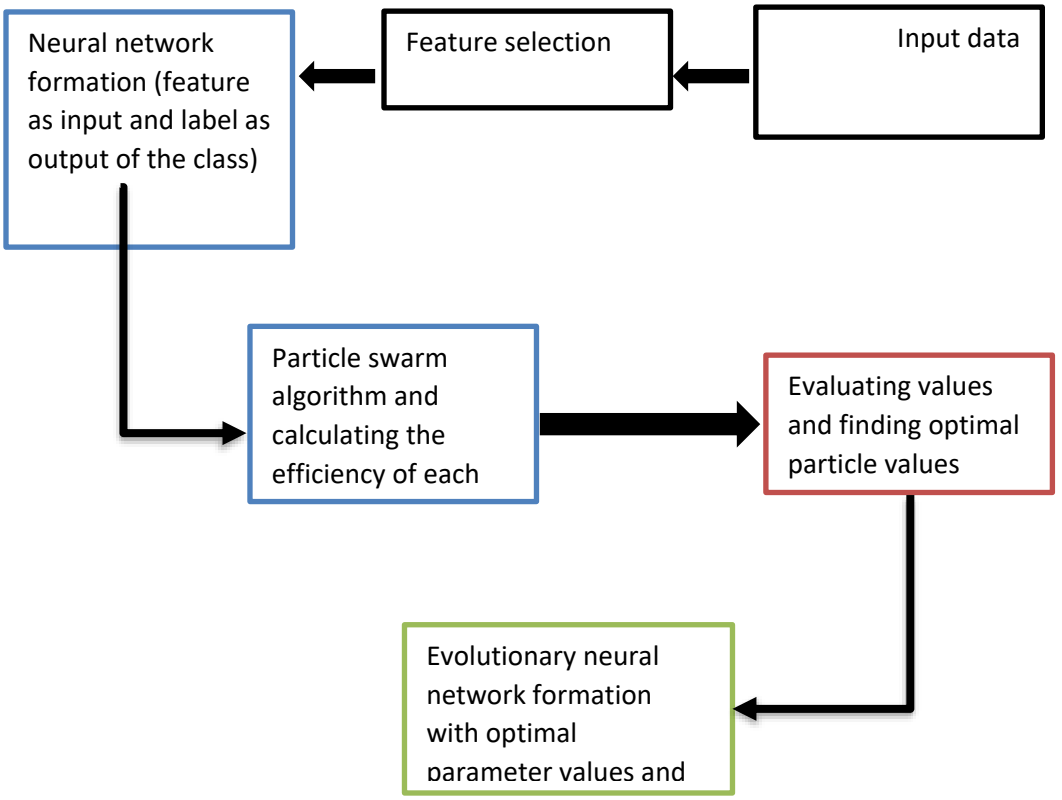

Figure 1: Proposed algorithm

The proposed method is evaluated on the NSL-KDD dataset [60]. This dataset contains selected records from KDD-CUP99 [61]. Challenges in the original dataset, like duplicate records in this dataset, have been addressed. Since 1999, the KDD-CUP99 dataset has been the most common dataset for evaluating anomaly detection and intrusion detection methods. This dataset was created by Stoolfo et al. [62] in the DARPA'98 Intrusion Detection Systems Assessment Program [63].

DARPA'98 contains approximately 5 million records of various connections, each containing 100 bytes. The KDD training set contains approximately 4900000 vectors of different connections, each containing 41 features. Each vector is classified into two classes: normal or attack. Each attack belongs to one of the following categories:

1) Denial of Service (DoS) attack: An attacker overwhelms a processing resource or memory so that it cannot execute authorized requests.

2) User to Root (U2R) attack: An attacker starts by accessing an authorized user's account (via password eavesdropping, dictionary attack, or social engineering) and accessing the system root by finding vulnerabilities in the system.

3) Remote to Local (R2L) attack: The attacker can send packets on the network but does not have an account on the network machines. So attacker accesses the system by examining the vulnerabilities of the system as a user.

4) Surveillance attacks: Such an attack attempts to obtain information about a computer network to threaten network security.

Most importantly, the probability distribution of the test set is not the same as the probability distribution of the training set. The test set has certain types of attacks not present in the training

set. This process brings the evaluations closer to reality. In other words, the training set includes 24 types of attacks, while the test set includes 14 distinct attacks that do not exist in the training set. Table (1) shows these classifications.

Table 1: Classification of attacks in the NSL-KDD dataset

Sub-Class	Class
-	NORMAL
ipsweep, nmap, portsweep, satan	PRB
back, land, neptune, pod, smurf, teardrop	DOS
buffer_overflow, loadmodule, multihop, perl, rootkit	U2R
ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster	R2L

The approximate distribution of different data classes in the training and test data is shown in Table (2). Due to the rounding of percentages, the total number is not equal to 100%.

Table 2: Approximate distribution of training and test data in the NSL-KDD dataset

Test	Training	Class
19 %	48 %	NORMAL
1 %	20 %	PRB
73 %	26 %	DOS
0.07 %	0.02 %	U2R
5 %	5 %	R2L

One of the data normalization process tasks is to convert text feature values to numeric values. Table (1) reflects the characteristics of the NSL-KDD dataset. Some of the 41 feature values in NSL-KDD are text values. The support vector machine uses only numeric data for training and testing, so text feature values must be converted to numeric values. Features with text values include (B) Protoclo_type, (C) Service, and (D) Flag, which are represented by numbers 2, 3, and 4 in Table (2), respectively.

For example, for Protoclo_type feature values, tcp is 1, udp is 2, and icmp is 3. In data normalization, after converting these three features from text format to numeric format, the main challenge is to convert data to binary form or normalized continuous form. In the proposed intrusion detection system, the normal continuous form is used. To normalize the features, a statistical analysis was performed on each feature based on the data in the NSL-KDD and the maximum, and minimum values for the values of each feature were determined. Then, based on Equation (1), normalization was realized in the range [1, 0].

$$Nf = \frac{f-MinF}{MaxF-minF} \qquad (1)$$

where F is the desired feature, f represents the value of the feature, maxF is equal to the maximum value of the feature F, minF is equal to the minimum value of the feature F.

Table 3: NSL-KDD data set features

Label	Name of feature	Label	Name of feature
A	Duration	**V**	Is_guest_login

B	Protocol_type	**W**	Count
C	Service	**X**	Sev_count
D	Flag	**Y**	Serror_rate
E	Src_byte	**Z**	Sev_serror_rate
F	Dst_byte	**AA**	Rerror_rate
G	Land	**AB**	Srv_rerror_rate
H	Wrong_fragment	**AC**	Same_srv_rate
I	Urgent	**AD**	Diff_srv_rate
J	Hot	**AE**	Srv_diff_host_rate
K	Num_failed_login	**AF**	Dst_host_count
L	Logged_in	**AG**	Dst_host_srv_count
M	Num_comprised	**AH**	Dst_host_same_srv_rate
N	Root_shell	**AI**	Dst_host_diff_srv_rate
O	Su_attempted	**AJ**	Dst_host_same_src_port_rate
P	Num_root	**AK**	Dst_host_srv_diff_host_rate
Q	Num_file_creations	**AL**	Dst_host_server_rate
R	Num_shells	**AM**	Dst_host_srv_serror_rate
S	Num_access_files	**AN**	Dst_host_rerror_rate
T	Num_cutbounds_cmds	**AO**	Dst_host_srv_rerror_rate
U	Is_host_login		

Precision and Recall are two well-known parameters used in evaluating data mining and machine learning algorithms. Precision is defined as follows:

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

And Recall is defined as follows:

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

where TP represents data correctly assigned to the positive class, FP represents data incorrectly assigned to the positive class, and FN represents data incorrectly assigned to the negative class. Twenty thousand records from the KDDTrain + set were randomly selected to include 24 known attacks to train the system. Seven thousand records from the KDDTest + set were selected to test the system, including 14 unknown new attacks and 24 known attacks. After selecting important features, the size of the input data is reduced by about 80%.

Results

The advantage of using the feature selection step in the proposed model is the reduction of training time and testing of the support vector machine, reducing the computational costs and necessary computer resources such as memory and CPU time. In this regard, training and testing time without feature selection and feature selection have been calculated and shown in Table (4). The unit of measurement of the time is milliseconds.

Precision, recall and F-score based on attack class for 41 features and eight features are shown in Tables (5) to (7), respectively. As can be seen from the results, feature selection has led to increased precision, recall, and F-score of the intrusion detection system. The detection rate with

eight features in all five classes is higher than the detection rate with 41 features. In particular, the percentage of attack detection for U2R and R2L classes using feature selection increases significantly. The system also uses important features to have a higher ability to detect new and unknown attacks, i.e., attacks that are not encountered in the training process and are only present in the test data.

Table 4: Training and testing time of the proposed model with and without using feature selection

	41 features	8 features	Time reduction
Training time	238454 ms	71320 ms	70.09%
Testing time	24820 ms	3989 ms	83.94%

Table 5: Precision of detection of the proposed system based on attack class

	Normal	DoS	PRB	U2R	R2L
41 features (Particle Swarm)	82.51 %	98.20 %	99.88 %	55.40 %	65.30 %
8 features (Particle Swarm)	98.60 %	99.16 %	99.97 %	69.46 %	99.91 %

Table 6: Recall of detection of the proposed system based on attack class

	Normal	DoS	PRB	U2R	R2L
41 features (Particle Swarm)	98.12 %	88.23 %	98.51 %	43.19 %	54.51 %
8 features (Particle Swarm)	99.79 %	93.15 %	99.84 %	52.20 %	68.23 %

Table 7: F-score of detection of the proposed system based on attack class

	Normal	DoS	PRB	U2R	R2L
41 features (Particle Swarm)	87.14 %	92.20 %	99.01 %	47.82 %	59.25 %
8 features (Particle Swarm)	98.87 %	93.15 %	99.90 %	52.05 %	79.61 %

In this study, the efficiency of the proposed method is compared with a multilayer perceptron neural (MLP) network based on the back-propagation (BP) algorithm. The BP learning algorithm uses the steepest descent (S.D) algorithm. Regulating the network parameters according to the error signals is calculated based on the feed of each pattern to the network. MATLAB software [73] has been used to implement this method. Comparison of precision, recall and F-score for detection using 41 features and eight features for neural network based on particle swarm algorithm are shown in Figures (2) to (4). The figure shows that the value of the mentioned criteria of 8 features case with the neural network based on the particle swarm algorithm is higher than the others.

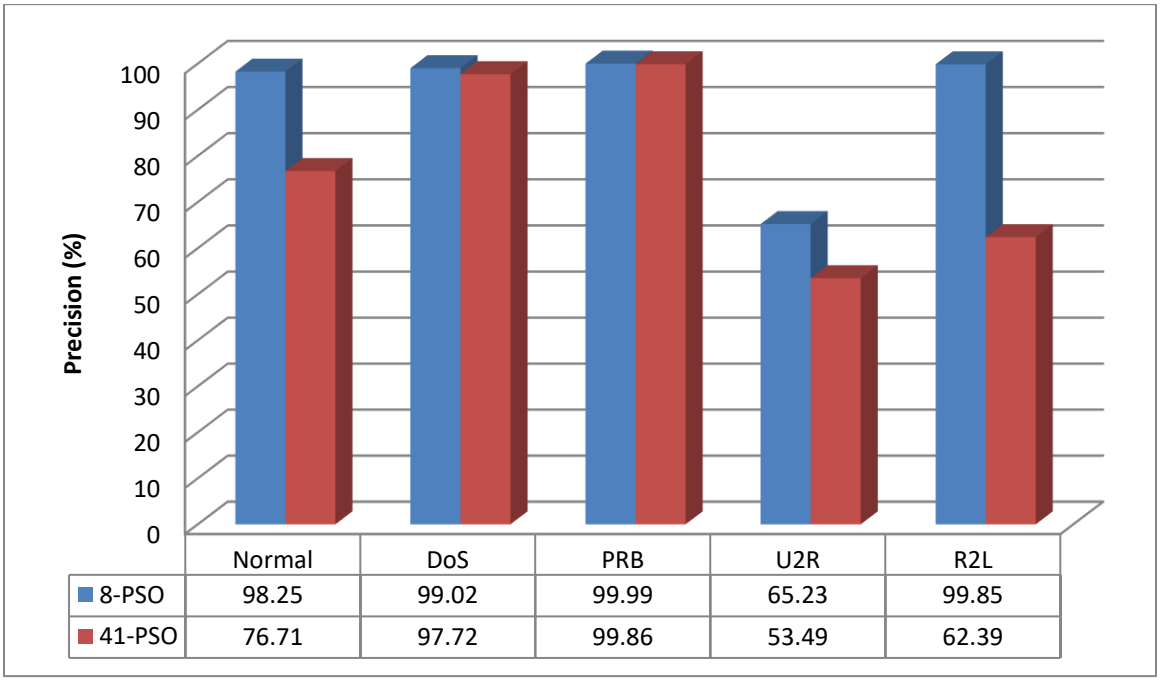

Figure 2: Comparison of precision criteria for detecting attacks based on attack class

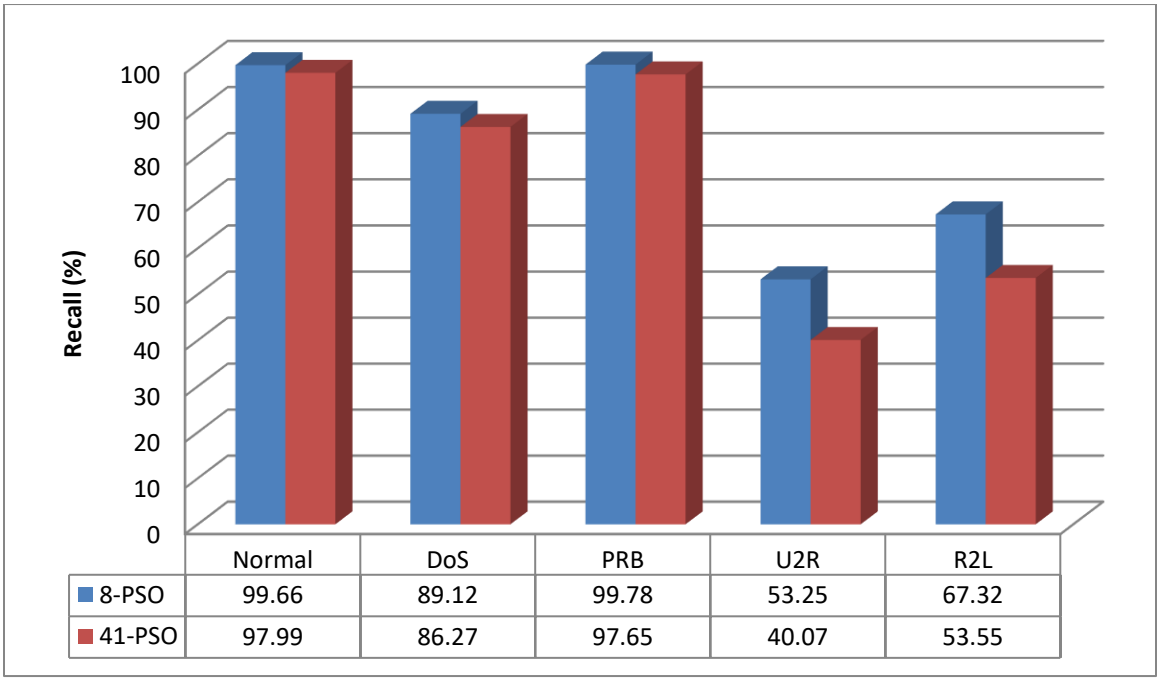

Figure 3: Comparison of recall criteria for detecting attacks based on attack class

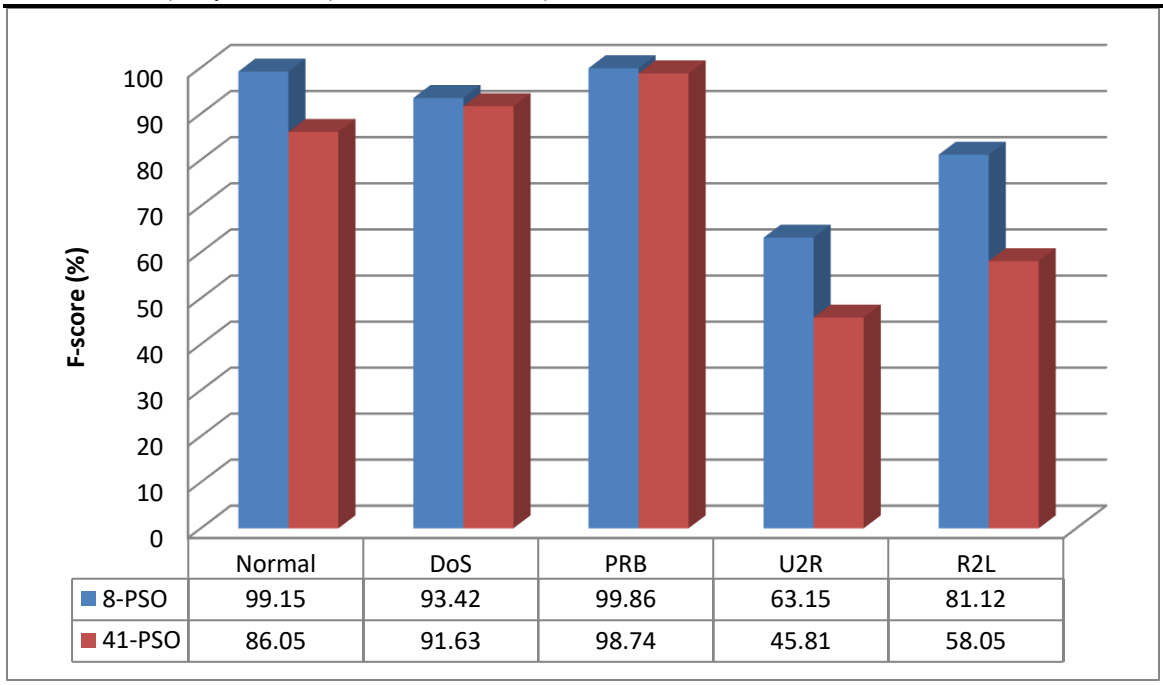

Figure 4: Comparison of F-score criteria for detecting attacks based on attack class

Conclusion

In recent years, the role of intrusion detection systems in ensuring the security of computer networks has become very prominent. Intrusion detection systems are security management systems used to detect malicious activity, unauthorized use, and misuse of computers or networks. An intrusion detection system is a hardware or software tool detecting attacks by monitoring the flow of events. Intrusion detection allows organizations to protect their systems against threats from increased interconnection networks and increase the reliability of their information systems.

Intrusion detection methods are divided into two classes, including detection of abuse and detection of anomalies. In detecting abuse, pre-made patterns are preserved in the form of law. Each pattern contains different types of intrusion. If such a pattern occurs in the system, an alarm will be triggered.

In the anomaly detection method, normal behavioral patterns are identified for the various entities under consideration (user, host, or the entire system). Any deviation from this behavior is considered an anomaly that could be the probability of an attack. One of the advantages of the model presented in this study is the reduction of computational costs and the reduction of computer resources such as memory and CPU time, which are necessary for detecting attacks. Also, feature selection in the proposed model has increased the intrusion detection system's precision, recall, and F-score. Finally, the performance of the proposed model was compared with the results of the neural network and support vector machine. The results showed that the proposed model is more efficient than the mentioned methods.

References

[1] Ramgovind, S, Eloff, MM, Smith, E. The Management of Security in Cloud Computing 2010.

[2] Zhuolin Yang, Jianyong Chen " Virtualization security for cloud computing service 2011.
[3] Chandrasekhar, A.M.; Raghuveer, K., "Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers," in The 2013 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-7, 4-6 Jan. 2013.
[4] Chun-Wei Tsai, "Incremental particle swarm optimisation for intrusion detection," IET Networks, vol. 2, no. 3, pp. 124-130, Sept. 2013.
[5] Abduvaliyev, A.; Pathan, A.-S.K.; Jianying Zhou; Roman, R.; Wai-Choong Wong, "On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp.1223-1237, Third Quarter 2013.
[6] Aljarah, I.; Ludwig, S.A., "MapReduce intrusion detection system based on a particle swarm optimization clustering algorithm," in The 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 955-962, 20-23 June 2013.
[7] Sui Xin, "Research of Intrusion Detection System," in The 2013 Fifth International Conference on Computational and Information Sciences (ICCIS), pp.1460,1462, 21-23 June 2013.
[8] Shen Li; Feng Lin, "An efficient architecture for Network Intrusion Detection based on Ensemble Rough Classifiers," in The 2013 8th International Conference on Computer Science & Education (ICCSE), pp.1411,1415, 26-28 April 2013.
[9] Butun, I.; Morgera, S.D.; Sankar, R., "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol.16, no.1, pp. 266-282, First Quarter 2014.