Hindawi
Journal of Sensors
Volume 2018, Article ID 5948146, 8 pages
https://doi.org/10.1155/2018/5948146

Research Article

Intrusion Detection System Based on Evolving Rules for Wireless Sensor Networks

Nannan Lu,[1] **Yanjing Sun**,[1] **Hui Liu**,[2] **and Song Li**[1]

[1]*School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China*
[2]*Institute of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing, China*

Correspondence should be addressed to Nannan Lu; lnn_921@126.com

Received 22 December 2017; Accepted 22 March 2018; Published 30 April 2018

Academic Editor: Mucheol Kim

Human care services, as one of the classical Internet of things applications, enable various kinds of things to connect with each other through wireless sensor networks (WSNs). Owing to the lack of physical defense devices, data exchanged through WSNs such as personal information is exposed to malicious attacks. Therefore, intrusion detection is urgently needed to actively defend against such attacks. Intrusion detection as a data mining procedure cannot control the size of rule sets and distinguish the similarity between normal and intrusion network behaviors. Therefore, in this paper, an evolving mechanism is introduced to extract the rules for intrusion detection. To extract diversified rules as well as control the quantity of rulesets, the extracted rules are examined according to the distance between the rules in the rule set of the same class and the rules in the rule set of different classes. Thereby, it alleviates the problem that the quantity of rules expands unexpectedly with the evolving genetic network programming. The simulations are conducted on a benchmark intrusion dataset, and the results show that the proposed method provides an effective solution to evolve the class association rules and improves the intrusion detection performance.

1. Introduction

The Internet of things (IoT) enables a large number of physical things or objects to connect, communicate, and exchange data with each other. IoT techniques span from health care to tactical military, in which human care is a type of classical application. The objects of human care services could include various kinds of medical equipment, even body parts. Wireless sensor networks (WSNs) are crucial for connecting, communicating, and exchanging data among such a large number of things. Although WSNs have the advantages of low installation cost, unattended network operation, and flexible deployment, their deficiency in physical defense devices renders both network and information vulnerable for malicious attacks [1]. To protect human care services from the internal or external attacks, prevention and detection are two main components involved in WSN security. However, as a passive network security mechanism, prevention is aimed at preventing any attack before it occurs and is therefore not sufficient. Thus, an active technique is urgently required to perceive malicious intrusions. Naturally, the focus shifts on the intrusion detection that can detect the actions attempting to compromise the confidentiality, integrity, or availability of one resource.

In general, the intrusion detection has two main techniques: misuse detection and anomaly detection. Misuse detection essentially identifies the previously known attacks from the normal network behaviors, while anomaly detection establishes the normal profiles to detect the new attacks. The combination of these two intrusion detection techniques is the hybrid intrusion detection. All the three techniques have been widely used in IoT. For example, Faisal et al. implemented anomaly detection to detect the external and internal attacks on smart meters [2]. Wang et al. and Pan et al. utilized the hybrid intrusion detection framework to protect the heterogeneous WSN, which was applied to power systems [3, 4]. Whereas, the specific methods of intrusion detection must be reviewed from the classical applications in the wired networks. Early studies on intrusion detection were conducted

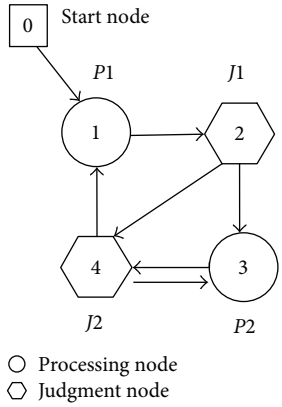

Figure 1: Phenotype of GNP.

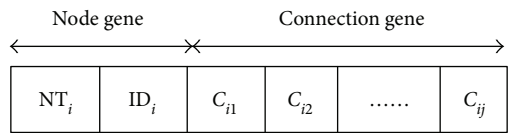

Figure 2: Genotype expression of GNP.

by Denning and Anderson [5, 6]. They aimed to build the monitoring systems for computer security, so that utilized statistics and rules to recognize attacks or viruses from the audited data. Since then, machine learning, data mining, statistic modeling, and pattern matching have been used to construct intrusion detection systems [7]. SVM was used to classify and select the audited data for the intrusion detection [8, 9]. The k-nearest neighbors (KNN) method has also been used to identify the intrusions through measuring distance [10]. Neural networks were also used to realize the intrusion detection systems, such as multilayer perceptron (MLP) [11]. Moreover, MLP was considered the basic unit to form the ensemble classifiers [12] such as AdaBoost. In [13], decision stumps were utilized as the weak classifiers to form a strong classifier.

Data mining is a successful solution to actively detect intrusive attacks based on the rules hidden in the network behavior data. Association rule mining is used to discover the correlations among the attribute sets in the data set for intrusion detection. The rules usually form as "$X \rightarrow Y$," which means that the -tuples in the dataset satisfy X is likely to satisfy Y. A RIPPER approach is proposed to generate frequent episodes firstly and then form the rules by associating the frequent episodes [14]. To extract the diversified rules, the fuzzy set theory was widely used to extract compact association rules. Tajbakhsh et al. [15] proposed a fuzzy association rule induction algorithm with two steps. The first step involved finding the significant itemsets with a higher significance factor than the user-specified threshold, and the second step involved generating rules by using the large itemsets induced in the first step. From the other perspective, intrusion detection generally distinguishes normal behavior, known intrusions and unknown intrusions, respectively, which can be taken as a classification procedure. Thus, the classes are considered with association rules

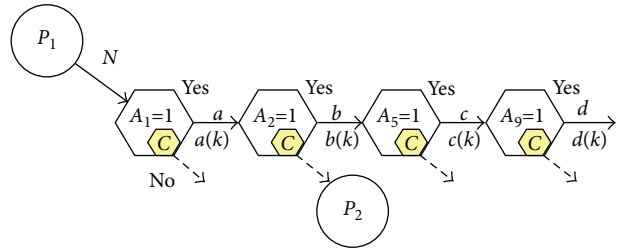

Figure 3: Class association rule mining based on GNP.

to form the class association rules. Different from the association rule, class association rule has the specified class label as its consequent part. Ozyer et al. [16] proposed to use GA boosting to find fuzzy class association rules. They encoded the rules as strings and used GA to evolve them. To extract as many rules as possible for identifying various kinds of intrusions, many algorithms were designed and implemented. Tsang et al. [17] employed a hierarchical GA structure. Each chromosome comprises control and parameter genes, and the parameters of fuzzy member functions were encoded as the parameter genes; the activations of which are managed by the control genes. Thus, the method was also used as a genetic feature selection wrapper to search for an optimal feature subset for dimensionality reduction. Feature selection can be used not only to alleviate the disadvantage of dimensionality and minimize the classification errors but also to improve the interpretability of the rule-based classifiers. Genetic programming (GP) has also been applied to intrusion detection. Some researches utilized GP to extract rules for intrusion detection based on its linear genomes and homologous crossover operators [18, 19]. In conclusion, most of current researches generally pursue the extraction of a large number of rules and overlook the discrimination of the rules [20]. Therefore, it is difficult to identify various types of intrusions with a high detection rate and a low false alarm rate. This could be due to the following two reasons. First, the network behavior data generated rapidly prompts the increase of the rules. Second, the similarity between the normal behavior and the new intrusion behavior limits the discrimination of the rules. Furthermore, this also brings about a considerable amount of redundant, irrelevant, and obvious information into the rule sets. In this case, the important rules are overwhelmed with the useless information. Therefore, the balance between the quality and quantity of rules is crucial for improving the intrusion detection performance.

To improve the quality of the rule sets as well as reserve the diversity of the rules, a new class association rule selection method is proposed based on genetic network programming (GNP) to solve the intrusion detection problem in smart human care services. Specifically, the similarities between rules and between rule sets are checked based on the distances during GNP evolution. The distance between the rules in the rule set of the same class is minimized, and the distance between the rules in the rule sets of the different classes is maximized by adding the newly extracted rules into the rule sets. If the above minimization and maximization criteria are

Table 1: Measurements of class association rules.

Class association rules	Support	Confidence
$(A_1 = 1) \Rightarrow (C = k)$	$a(k)/N$	$a(k)/a$
$(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C = k)$	$b(k)/N$	$b(k)/b$
$(A_1 = 1) \wedge (A_2 = 1) \wedge (A_5 = 1) \Rightarrow (C = k)$	$c(k)/N$	$c(k)/c$
$(A_1 = 1) \wedge (A_2 = 1) \wedge (A_5 = 1) \wedge (A_9 = 1) \Rightarrow (C = k)$	$d(k)/N$	$d(k)/d$

satisfied, the extracted rules are added to the rule sets; otherwise, they are discarded. In this way, redundant information can be avoided during the rule evolution, and the discrimination of the rule sets would be enhanced. In addition, this method also reserves the diversity of the rule sets according to the evolving mechanism. Thus, the GNP evolving method has the ability to discover discriminative class association rules for intrusion detection, which can further improve the intrusion detection performance.

The remainder of this paper is organized as follows. Section 2 describes the GNP structure and GNP-based class association rule mining in detail, and Section 3 introduces how to evolve class association rules based on distance. The simulation results are shown in Section 4, and Section 5 concludes this paper.

2. Genetic Network Programming

2.1. Basic Structure of GNP. GA has a string structure, and GP has a tree structure. With the complexity of problems increasing, it is difficult to express the problem using GA, and the GP structure starts bloating. As an extension of GA and GP, GNP has a quite different structure from GA and GP, which is the directed graph structure [21]. Figure 1 shows the phenotype of GNP, and there are three kinds of nodes in each individual. The start node is used to determine the first node to be executed. Judgment nodes work as the decision-making functions and are represented as $J_1, J_2, \ldots, J_m$. Processing nodes represent the functions of actions or processes and are expressed as $P_1, P_2, \ldots, P_n$. Node transition starts from the start node, and then, the next node to be executed is determined by the node transition. In addition, the number of nodes and their functions depend on the specific problem, which are determined by designers. In addition, judgment nodes have conditional branches, whereas processing nodes do not.

Figure 2 illustrates the genotype of the GNP structure.N T_i indicates the node type, the values of which are 0, 1, or 2. 0 is the start node, 1 is the processing node, and 2 is the judgment node. ID_i serves as an identification number. And C_{ij} denotes the node connection between node i and j.

2.2. Class Association Rule Mining Based on GNP. When GNP is used to extract class association rules, the function of the judgment node corresponds to the attribute of each tuple in the dataset, and the processing nodes are used to calculate the measurements of the class association rules. The specific procedure of class association rule mining using GNP is shown in Figure 3. GNP examines the attribute values of tuples in the dataset using judgment nodes and calculates the measurements using processing nodes. The judgment node determines the next node by the judgment result of yes or no, corresponding to the yes side or no side.

The yes side of the judgment node is connected to another judgment node. Judgment nodes can be reused and shared with other class association rules because of GNP's reusability. The no side of the judgment node is connected to another processing node, which represents the end of the current rule and the start of another new rule. The start node is connected to the first processing node. The connections of judgment nodes in Figure 3 are extracted as the candidate class association rules, which are shown below. There are four class association rules that correspond to four connections.

$$\begin{aligned} &(A_1 = 1) \Rightarrow (C = k), \\ &(A_1 = 1) \wedge (A_2 = 1) \Rightarrow (C = k), \\ &(A_1 = 1) \wedge (A_2 = 1) \wedge (A_5 = 1) \Rightarrow (C = k), \\ &(A_1 = 1) \wedge (A_2 = 1) \wedge (A_5 = 1) \wedge (A_9 = 1) \Rightarrow (C = k). \end{aligned} \tag{1}$$

To evaluate the above candidates of class association rules, we can calculate the corresponding support and confidence, which are shown in Table 1.

Let A_i be the item in the data set, and let its value be 1 or 0. Let C be the class label. So, the class association rule can be represented as the following unified form.

$$(A_p = 1) \wedge \cdots \wedge (A_q = 1) \Rightarrow (C = k), \quad k = 0, 1, 2, \ldots, K, \tag{2}$$

where $A_m = 1$ means that attribute A_m equals to 1 and C is the set of suffixes of classes.

3. Evolving Class Association Rules

GNP can extract a great number of class association rules for intrusion detection. However, with an increase in the amount of rules, the detection performance is not always enhanced by the extracted rules. Lots of rules bring redundant and irrelevant information into rule sets. This section describes how to implement the new evolving mechanism on the class association rule mining by GNP.

3.1. Jaccard Distance. Evolving class association rules are aimed at pruning the redundant and irrelevant rules for intrusion detection and at reserving the discriminative rules. In fact, a class association rule is composed of a set of attributes. Thus, the difference between two rules can be regarded as the distance between two sets of attributes, which is computed by the definition of Jaccard distance [22] shown as Definition 1.

```
Input: Target generation of GNP, N;
       Training data base, TrainDB;
Output: Accumulated ruleset, R;
for i = 0; i < N; i++ do
    while ith GNP generation do
        Extract next rule r based on TrainDB
        Distance_N ← CalculateDistance(r, R_N)
        // Calculate distance using the latest normal ruleset
        if Distance_N > CalculateDistance(r, R_N) then
            if the class of rule r equals normal then
                AddOneRule(r, R_N)
            else
                AddOneRule(r, R_I)
            end if
            break
        end if
        Distance_I ← CalculateDistance(r, R_I)
        // Calculate distance using the latest intrusion ruleset.
        if Distance_I > CalculateDistance(r, R_I) then
            if the class of rule r equals normal then
                AddOneRule(r, R_N)
            else
                AddOneRule(r, R_I)
            end if
        end if
    end while
    GNP population comets to (i + 1)th generation
end for
R ← MergeRulePool(R_N, R_I)
return R
```

Algorithm 1: (GNP with rule evolving).

Definition 1. Given two sets A and B, the Jaccard distance is defined as

$$D_J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}, \tag{3}$$

where $|A \cup B|$ states the union of set A and set B, and $|A \cap B|$ indicates the intersection between set A and set B. The Jaccard distance can measure the degree of overlap between the two sets.

3.2. Rule Selection Based on Distance. Pruning the redundant and irrelevant rules is achieved by minimizing the distance of rules in the same class rule set as well as maximizing the distance between the different class rule sets. Therefore, the generated rules are checked according to the similarity of the rules and that of the rule sets. Specifically, when a newly extracted rule is added into the rule set, the distance between the rules in the rule set of the same class is minimized and the distance between the rules in the rule sets of different classes are simultaneously maximized. In this case, the rule is regarded as a distinguishable class association rule.

As the description of a class association rule, it comprises a group of attributes, which can be regarded as the mathematic theory "set." Thus, the distance either between the rules or between the rule sets can be described by the difference of two sets. Based on this principle, the distance between rule r and r' is defined as (4). And the distance between the rule sets can be calculated based on (4). The detailed definition is shown as (5).

$$d\left(r, r'\right) = \frac{|A_r \cup A_{r'}| - |A_r \cap A_{r'}| + \sum_{a \in A_r \cap A_{r'}} d\left(v(r,a), v\left(r', a\right)\right)}{|A_r \cup A_{r'}|}, \tag{4}$$

$$d\left(R, R'\right) = \frac{\sum_{r \in R} \sum_{r' \in R'} d\left(r, r'\right)}{|R|\left|R'\right|}, \tag{5}$$

where R and R' denote the rule set with different classes. r and r' represent the two rules. A_r and $A_{r'}$ are the corresponding attribute sets of rule r and rule r', respectively. a denotes the attribute in the rule. $v(r,a)$ is the value of attribute a of rule r. $d(R,R')$ stands for the distance between rule set R and rule set R'. $d(r,r')$ represents the distance between rule r and rule r', whose range is [0,1]. $d(v(r,a), v(r',a))$ is defined as

$$d\left(v(r,a), v\left(r', a\right)\right) = \begin{cases} 1, & v(r,a) \neq v\left(r', a\right), \\ 0, & v(r,a) = v\left(r', a\right). \end{cases} \tag{6}$$

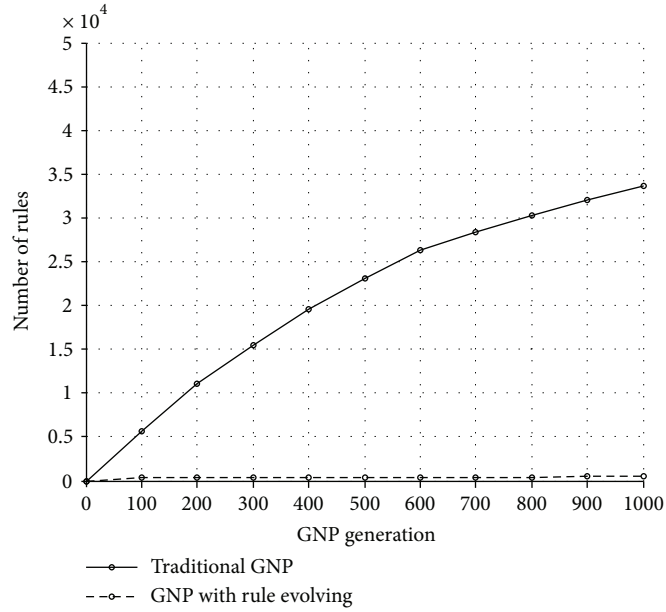

Figure 4: The comparison of rule quantity between the traditional GNP and GNP with rule evolving.

From (4) and (5), the modified distance considers the actual value of each attribute by adding $d(v(r,a), v(r',a))$ to the traditional Jaccard distance. $d(v(r,a), v(r',a)) = 0$ indicates that the attributes of rule r are completely the same with those of rule r', whereas $d(v(r,a), v(r',a)) = 1$ means that the attributes of rule r are completely different from those of rule r'. Therefore, the larger is the number of the same pairs (attribute, value), the shorter is the distance between r and r'. In this paper, the thresholds of intradistance between the rules in the same rule sets and interdistance between the rules in the different rule sets are all set as 0.98.

3.3. Evolving Class Association Rules. Except for support and confidence, χ^2 is also used to measure the significance of a rule. The class association rule is abbreviated as the form $X \rightarrow Y$, where $X, Y \subseteq I$ and $X \cap Y = \varnothing$, with I being the set of attributes. X and Y are the antecedent and consequent-of the association rule, respectively. Unlike the association rule, the class association rule has a class label as the consequent part. In this way, the support is defined as $\text{support}(X) = x$. x is the fraction of tuples containing X in the database. Confidence is defined by $\text{support}(X \cup Y)/\text{support}(X)$. Therefore, χ^2 of the rule is given by

$$\chi^2 = \frac{N(z - xy)^2}{xy(1-x)(1-y)}, \tag{7}$$

where N is the total number of tuples in the database, z is the value of $\text{support}(X \cup Y)$, and x and y are supports of X and Y, respectively.

Then, the minimum support, minimum confidence, and minimum χ^2 are used to select the candidate rules. After calculating the support, confidence, and χ^2 values of the above candidate class association rules, and if they satisfy the following conditions, $\text{support} \geq \text{support}_{\min}$, $\text{confidence} \geq \text{confidence}_{\min}$, and $\chi^2 \geq \chi^2_{\min}$, the rule is regarded as the important rule and then stored into the ruleset.

Table 2: Confusion matrix of classification results.

	Normal (C)	Misuse (C)	Anomaly (C)	Total
Normal (A)	8202	31	1478	9711
Misuse (A)	0	4776	546	5322
Anomaly (A)	1292	17	6202	7511
Total	9494	4824	8226	22,544

Each individual is evaluated by the fitness function defined by

$$\text{Fitness} = \sum_{r \in R} \left\{ \chi^2(r) + 10(n_{\text{ante}}(r) - 1) + \alpha_{\text{new}}(r) \right\}, \tag{8}$$

where R in $r \in R$ is the set of suffixes of the extracted important rules from the individuals, $n_{\text{ante}}(r)$ is the number of attributes in the antecedent part of rule r, and $\alpha_{\text{new}}(r)$ is the additional constant shown as (9)

$$\alpha_{\text{new}}(r) = \begin{cases} \alpha_{\text{new}}, & \text{when rule } r \text{ is newly extracted,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Therefore, the fitness function of GNP is concerned with importance, complexity, and novelty of rule r.

The pseudocode of evolving class association rules by GNP is summarized in Algorithm 1.

4. Simulations

4.1. Data Set. Owing to the lack of realistic data sets of smart human care services, the benchmark data set NSL-KDD is used to verify the validity of the proposed method

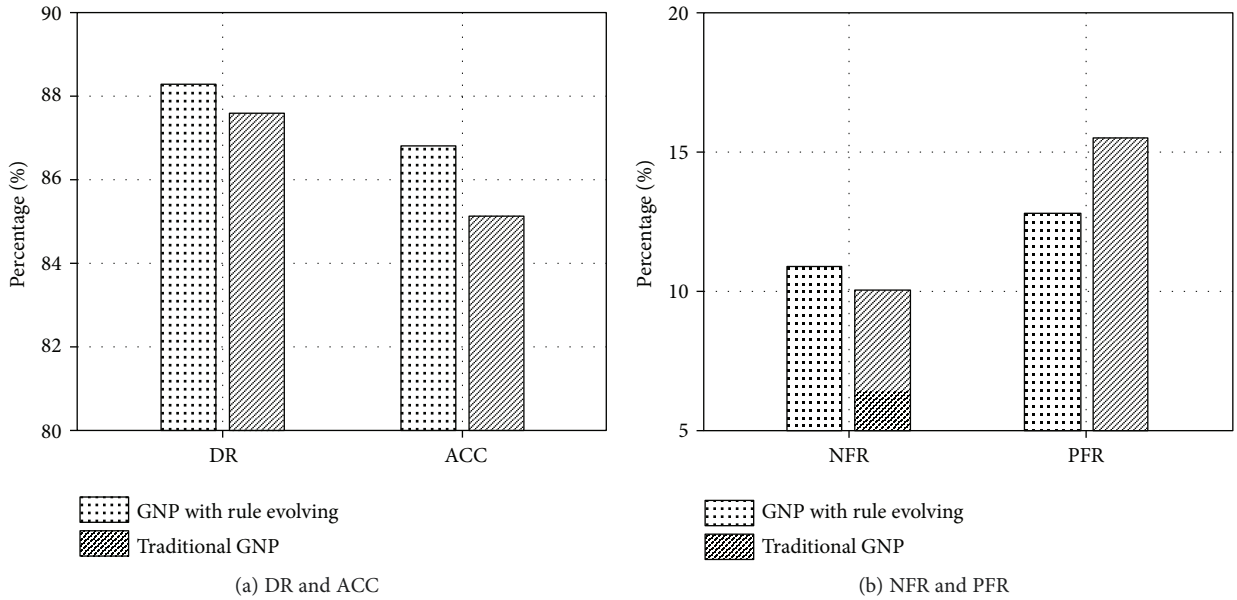

Figure 5: Detection performance comparisons of the traditional GNP and GNP with rule evolving.

Table 3: The accuracy comparisons of the traditional GNP and GNP with rule evolving (%).

Classifier	Traditional GNP (%)	GNP with rule evolving (%)
SVM	72.89	75.98
BP	71.87	74.82
MLP	73.53	74.91
KNN	75.69	77.30
Logic regression	73.56	74.83
Decision tree	79.63	80.22
AdaBoost	75.23	73.50
Naive Bayes	76.31	74.72
Cluster Gaussian	85.07	86.78
Average	**75.97**	**77.00**

[23, 24]. NSL-KDD is a new version of KDD CUP 1999 data set [25]. Both NSL-KDD and KDD CUP 1999 include a wide variety of intrusions simulated in a military network environment, which is difficult for a self-build simulation environment to acquire such diversified categories of intrusions. However, the NSL-KDD data set is different from KDD CUP 1999, which is composed of the most difficult detected data evaluated by the classical classification methods. Thus, NSL-KDD is a challenge data set for evaluating the intrusion detection methods. Moreover, compared to the KDD CUP 1999 data set, the intrusion detection performance on the NSL-KDD data set will not be biased towards the intrusions easily detected.

Each audit data in NSL-KDD consists of 41 attributes including continuous and discrete ones and one class label. Except for normal audit data, there are four types of attacks in this dataset, which are denial of service (DOS), probe, user to root (U2R), and root to local (R2L).

4.2. Parameter Settings. We use $\text{support}_{\min}(N) = 0.1$, $\text{support}_{\min}(I) = 0.075$, $\text{confidence}_{\min} = 0.8$, and $\chi^2_{\min} = 6.64$, where N and I indicate normal and intrusion, respectively. Class association rules are extracted for each class using GNP. The population size of GNP is 120. The number of processing nodes and judgment nodes are 10 and 100, respectively. In addition, the crossover rate is 1/5. The mutation rate for P_{m1} is 1/3 and for P_{m2} is 1/3, in which P_{m1} and P_{m2} mutate the connections of the branches and the contents of the nodes, respectively. The condition of termination is 1000 generations.

4.3. Result Analysis. First, we randomly select 2000 normal data and 2000 intrusion data as the training set. The testing set consists of 9711 normal data and 12,833 intrusion data. Both the training set and the testing set are from NSL-KDD, which avoids redundant records and improves the difficulty level of KDD Cup 1999. GNP-based class association rule mining is conducted on the prepared training set. The proposed method is compared with the traditional GNP-based class association rule mining shown in Figure 4. Different from GNP with rule evolving, the traditional GNP has no action of automatic selection of useful rules, which always extracts a great number of class association rules. After 1000 generations, the traditional GNP increasingly generates rules, while the proposed method has been converged already. We can conclude that GNP with rule evolving has the strong ability to reduce the rule quantity in rule mining. It can be also regarded as an online rule pruning scheme.

Furthermore, the detection performance of the proposed method on the NSL-KDD data set is investigated. Here, we use the classifier of cluster Gaussian referred to the literature [26]. The cluster Gaussian classifier utilizes

Table 4: The accuracy comparisons of CBA, CMAR, and GNP with rule evolving (%).

Classifier	Decision tree	SVM	KNN	AdaBoost	Cluster Gaussian
CBA	70.64	74.47	71.75	72.83	86.09
CMAR	72.39	72.28	72.99	72.19	85.48
GNP with rule evolving	80.22	75.98	77.30	73.50	86.78

the information of known normal and intrusion data to find the extract boundary of normal and known intrusions. In terms of the data distribution, it clusters the similar data which are supposed to have the similar network behaviors. And the classifier further uses Gaussian functions to find the cluster boundaries and data distribution to determine the cluster number. Table 2 shows the confusion matrix of the detection results. "A" is the actual class of the data and "C" is labeled by the classifier. From the confusion matrix, detection rate (DR), accuracy, positive false rate (PFR), and negative false rate (NFR) are calculated. DR indicates the rate of the data that are correctly classified into normal or intrusion. ACC (accuracy) is the rate of the data that are accurately classified as normal, misuse intrusion, or anomaly intrusion. PFR represents that the classifier identifies the normal data as misuse intrusion or anomaly intrusion. NFR represents that misuse and anomaly intrusions are identified as normal.

Therefore, according to Table 2, DR, accuracy, PFR, and NFR are calculated as follows.

$$\begin{aligned} \text{DR} &= \frac{8202 + 4776 + 6202 + 17 + 546}{22544} = 87.58\%, \\ \text{Accuracy} &= \frac{8202 + 4776 + 6202}{22544} = 85.08\%, \\ \text{PFR} &= \frac{31 + 1478}{9711} = 15.54\%, \\ \text{NFR} &= \frac{0 + 1292}{5322 + 7511} = 10.07\%. \end{aligned} \tag{10}$$

From the results, misuse intrusion and normal are easy to distinguish by the evolved rules. Though most of anomaly intrusions have been identified, a lot of anomaly intrusions are still difficult to detect. Furthermore, we compare the traditional GNP with the proposed method on DR, accuracy, NFR, and PFR. As shown in Figure 5, GNP with rule evolving obtains higher DR and ACC and lower PFR, but NFR is a little bit higher in GNP with rule evolving than in the traditional GNP. Therefore, anomaly intrusions are still difficult to distinguish. The similarities between anomaly intrusions and normal patterns account for this phenomenon.

In order to further demonstrate the proposed method, classical machine learning algorithms are taken as comparative classifiers, including support vector machine (SVM), back propagation (BP) neural network, multilayer perception (MLP), *k*-nearest neighbor (KNN), logic regression, decision tree, AdaBoost, naive Bayes, and cluster Gaussian.

Table 3 shows the detection accuracy of the traditional GNP and GNP with rule evolving based on different classifiers. By evolving 1000 generations, GNP extracts 33,723 rules and the proposed GNP extracts 436 rules. Then, 9 classifiers are used to evaluate the intrusion detection performance based on the two GNP. Among them, 7 classifiers on GNP with rule evolving acquire higher accuracy than those on the traditional GNP. In addition, the average accuracy of GNP with rule evolving is also better than that of the traditional GNP. The results demonstrate that the proposed method can evolve better rules for intrusion detection. Besides, we evaluate the proposed method by comparing it with the classical classification rule mining methods such as classification based on associations (CBA) [27] and classification based on multiple association rules (CMAR) [28]. Both CBA and CMAR contain the rule pruning procedure. With the default classifiers, CBA and CMAR obtain accuracies of 74.63% and 72.17%, respectively, which are lower than the average accuracy of 77% obtained using GNP with a rule selection mechanism. In addition, we select some of the classical classification methods to evaluate the effectiveness of the proposed method, which are the decision tree, SVM, KNN, AdaBoost, and cluster Gaussian. Table 4 illustrates the accuracy comparisons of CBA, CMAR, and GNP with rule evolving. As shown in Table 4, GNP with rule evolving has higher classification accuracies than the other rule-based methods. Thus, it is necessary to consider the rule evolving technique in the rule mining. And the rule evolving is capable of selecting useful rules and reducing the redundant and irrelevant rules.

5. Conclusions

In this paper, an intrusion detection system based on evolving class association rules is proposed as a security solution for smart human care services. In general, it utilizes a class association rule evolving strategy to construct the intrusion detection system. As a data mining solution, GNP with rule evolving can generate diversified class association rules and control the quantity of the rules simultaneously. For intrusion detection, the significance test is performed to ensure the importance of generated rules. In order to generate the more discriminative class association rules, the Jaccard distance is modified to measure the similarity between rules and different rule sets. In this way, the distance of the rules in the rule set with the same class is minimized and the distance between rules in the rule sets with different classes is maximized. The simulations conducted on the NSL-KDD dataset theoretically verify that GNP with rule evolving efficiently controls the quantity of generated rules and improves the detection performance by reducing the redundancy of the rules. In the future, we plan to verify the effectiveness of the

intrusion detection system on the self-build simulation environment.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by Basic Scientific Research Fund for Central Universities under Grant no. 2015QNB20, in part by the Natural Science Foundation of Jiangsu Province under Grant no. BK20150204, in part by China Postdoctoral Science Foundation under Grant no. 2015M581884, and in part by National Natural Science Foundation of China under Grant no. 51734009 and no. 51504255.

References

[1] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266–282, 2014.

[2] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, "Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: a feasibility study," *IEEE Systems Journal*, vol. 9, no. 1, pp. 31–44, 2015.

[3] S. S. Wang, K. Q. Yan, S. C. Wang, and C. W. Liu, "An integrated intrusion detection system for cluster-based wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15234–15243, 2011.

[4] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104–3113, 2015.

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

[6] J. P. Anderson, "Computer security threat monitoring and surveillance," Tech. Rep., James P. Anderson Co., Fort, Washington, PA, USA, 1980.

[7] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: a review," *Applied Soft Computing*, vol. 10, no. 1, pp. 1–35, 2010.

[8] A. A. Aburomman and M. B. I. Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Information Sciences*, vol. 414, pp. 225–246, 2017.

[9] E. Kabir, J. Hu, H. Wang, and G. Zhuo, "A novel statistical technique for intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 303–318, 2018.

[10] C. Guo, Y. Ping, N. Liu, and S. S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[11] J. Cannady, "Artificial neural networks for misuse detection," in *Proceedings of the 1998 National Information Systems Security Conference*, pp. 443–456, Arlington, VA, USA, 1998.

[12] D. Parikh and T. Chen, "Data fusion and cost minimization for intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 381–389, 2008.

[13] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 577–583, 2008.

[14] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on Machine learning*, pp. 115–123, Tahoe City, California, 1995.

[15] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," *Applied Soft Computing*, vol. 9, no. 2, pp. 462–469, 2009.

[16] T. Ozyer, R. Alhajj, and K. Barker, "Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 99–113, 2007.

[17] C. H. Tsang, S. Kwong, and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection," *Pattern Recognition*, vol. 40, no. 9, pp. 2373–2391, 2007.

[18] J. V. Hansen, P. B. Lowry, R. D. Meservy, and D. M. Mcdonald, "Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection," *Decision Support Systems*, vol. 43, no. 4, pp. 1362–1374, 2007.

[19] D. Song, M. I. Heywood, and A. N. Zincir-Heywood, "Training genetic programming on half a million patterns: an example from anomaly detection," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 225–239, 2005.

[20] E. Lazcorreta, F. Botella, and A. Fernández-Caballero, "Towards personalized recommendation by two-step modified apriori data mining algorithm," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1422–1429, 2008.

[21] S. Mabu, K. Hirasawa, and J. Hu, "A graph-based evolutionary algorithm: genetic network programming (GNP) and its extension using reinforcement learning," *Evolutionary Computation*, vol. 15, no. 3, pp. 369–398, 2007.

[22] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, pp. 34-35, 1971.

[23] A. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 53–58, Ottawa, ON, Canada, 2009.

[24] http://nsl.cs.unb.ca/NSL-KDD/.

[25] http://kdd.ics.uci.edu/databases/kddcup99/.

[26] N. Lu, S. Mabu, Y. Li, and K. Hirasawa, "Classification with clustering and Gaussian functions in intrusion detection system," *IEEJ Transactions on Electronics Information and Systems*, vol. 134, no. 12, pp. 1908–1915, 2014.

[27] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in *KDD '99 Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 337–341, San Diego, California, USA, 1999.

[28] W. Li, J. Han, and J. Pei, "Cmar: accurate and efficient classification based on multiple class-association rules," in *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 369–376, San Jose, CA, USA, 2001.