J Electr Eng Technol.2016; 11(5): 1383-1394
http://dx.doi.org/10.5370/JEET.2016.11.5.1383

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

# A Granular Classifier By Means of Context-based Similarity Clustering

**Wei Huang*, Jinsong Wang† and Jiping Liao****

**Abstract** – In this study, we propose a granular classifier (GC) with the aid of a context-based similarity clustering (CSC) method and applied it for network intrusion detection. The proposed CSC supporting the design of information granules is exploited here to determine the so-called contexts. Unlike the conventional similar clustering method, here the CSC built clusters by taking into consideration of both input data and output data. The design of granular classifier is realized based on the if-then rules, which consists two parts: namely premise part and conclusion part. The premise part is developed by using the CSC, while the conclusion part is realized with the aid of supported vector machines. In contrast to typical rule-based classifier, the underlying principle exploited here is to consider a robust classification with the adequate use of output data. In particular, rule-based classifiers or supported vector machines can be regarded as a special case of the proposed granular classifier. Numeric studies show the superiority of the proposed approach.

**Keywords**: Network intrusion detection, Context-based similarity clustering (CSC), Granular classifier

## 1. Introduction

THE recent years have seen a tremendous wealth of classifiers, which have been used in many applications [1-3] such as network intrusion detection, industrial engineering, medical science, and so on. In the design of classifiers, various strategies lead to a diversity set of classifiers.

There have been a suite of rule-based classifiers addressing the classification problems. Pioneering work such as hybrid decision tree [4-6] construct different classification methods. With the use of rule, these classifiers achieve good capabilities to deal with granulation information. Though promising results are obtained for some real-world problems, these methods perform poorly when dealing with high-dimensional problems. To alleviate this problem, lots of associate rules have been used for classification for high-dimensional transactional data and have shown good results on outer membrane localization prediction [7-8]. In spite of advantages, this approach is not free from eventual drawbacks. One limitation is that this method depends on a carefully chosen minimum support, and the performance is not good in comparison with support vector machine (SVM).

The support vector machine is a famous classifier that finds so-called decision hyperplane to obtain the different classes of data. Especially, one can extend SVM to obtain nonlinear decision hyperplanes by exploiting kernelization techniques [9-12]. Pioneering studies by Xue et al. [13], Sebald et al. [14], Alam et al. [15], and Morsier et al. [16] led to different improved SVM models. By now, the research has focused on the efficient of linear or nonlinear classification. Nevertheless, in most cases all these advanced SVM models do not come with capabilities to cope with granular information.

In the field of information security, network intrusion detection is one of key issues. As this issue can also be described as a classification problem, many classical classification approaches such as adabooast algorithms, support vector machines and so on have been applied to address network intrusion detection problem. Although some typical classification methods (e.g. SVM) have lots of advantages, they are still some disadvantages due to its drawbacks of classification strategies. It still necessary to realize the network intrusion detection with the aid of novel classification approaches.

In this paper, we propose design a Granular Classifier (GC) based on a Conditional Similarity-based Clustering (CSC) and SVM, and applied it for solving the network intrusion detection problem. The proposed GC is designed based on the if-then rules that is composed of two parts, namely premise part and conclusion part. The proposed CSC is exploited here to realize the design of information granulation for the premise part, while the conclusion part is realized with the aid of SVM. Alone with the novel architecture of granular classifier, we take the advantage of both rule-based classifiers and SVM. The advantages of the proposed granular classifier are summarized as follows: 1) GC is relative more robust approach when dealing with high-dimensional problems in compared with conventional rule-based classifiers. With the use of SVM, the GC can deal with high dimensional classification problems easily in compared with typical rule-based classifiers. 2) GC

† Corresponding Author: School of Computer and Communication Engineering, Tianjin University of Technology, China. (jswang70@126.com)
* School of Computer and Communication Engineering, Tianjin University of Technology, China. (huangwabc@163.com)
** Tianjin Key Laboratory of Intelligent and Novel Software Technology, Tianjin University of Technology, China.
Received: November 23, 2015; Accepted: January 8, 2016

comes with capabilities to deal with granular information when compared with SVM. By means of CSC, the GC can deal with granular information effectively in comparison with typical SVM.

The rest of this paper is arranged as follows. Section II introduces the underlying idea of granular classifier. Section III provides the architecture of granular classifier. Section IV gives the design procedure of granular classifiers. Section V reports on a comprehensive set of experiments. Finally, concluding statements are made in Section VI.

## 2. Granular Classifier: an idea

This section we first focus on the underlying idea of granular classifiers and context-based similar clustering that is used to realize the information granulation.

### 2.1 Rule-based classifier via information granulation

To show the underlying idea of the granular classifiers, let us recall the classical SVM. SVM is a statistically robust learning method that used for classification and regression analysis, and it can efficiently perform both linear classification and a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces [17]. An illustration example is shown in Fig. 1. In this figure, the classification boundary is described as a decision hyperplane.

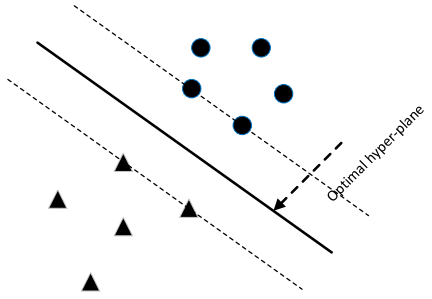



**Fig. 1.** An illustration of linear classifier in a two-dimension space

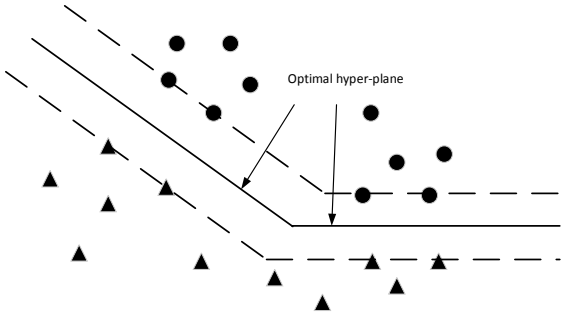



**Fig. 2.** An illustration of rule-based linear classifier in a two-dimension space

With this regard, the two different sets of data can be divided easily. In this case, all of data are classified by using the optimal decision hyperplane. However, sometimes it is very difficult to determine the one optimal decision hyperplane when dealing with the data set with more complex structure as shown in Fig. 2.

Fig. 2 illustrates an example of rule-based linear classifier in a two-dimension space. In some sense, the conventional SVM can be regarded as only one rule-based classifier. That is, a general rule-based classifier model comes as an extended SVM. The underlying idea is to divide the SVM classification with a number of rules, which may capture "rough, major structure"; while a SVM is considered as local model, which may capture "subtle, accurate structure". With this regard, we design a rule-based SVM, namely granular classifiers. In the design of granular classifier, the input space is partitioned with the aid of information granulation, while a SVM in one rule is viewed as a local model representing the input-output relationship in a sub-space of the corresponding antecedent. More specifically, a rule-based classifier can be represented in the form of a serial "if-then" rules

$$\mathbf{R}^i : IF \ \mathbf{x} \text{ is } in\ cluster\ A_i\ THEN\ y_i = \mathrm{g_i}(\mathbf{x}) \qquad (1)$$

where $\mathbf{R}^i$ is the $i$th rule, $A_i$ is the $i$th cluster, $i$=1, ···, $n$, $n$ is the number of rules (the number of clusters), $\mathrm{g}_i(\mathbf{x})$ is the consequent output of the $i$th rule, i.e., a local model representing input-output relationship of the $i$th sub-space (local area). Here the way to describe pattern classifiers is realized by using a set of discriminant functions $\mathrm{g}_i(\mathbf{x})$.

It is evident that there is still one important open problem, which is determining the number of rules in the design of granular classifier. In this study the number of rules is determined by information granulation realized by using context-based similar-based clustering.

### 2.2 Clustering method considering output space

Similar clustering [18] is a robust clustering algorithm that is developed based on a total similarity objective function related to the sapproximate density shape estimate. In the similar clustering, the clusters are determined based on the input data without consideration of any output. Unlike the conventional similar clustering method, here we may build the clusters by taking into consideration of both input data and output data. Fig. 3 depicts the two cases of context-based similar clustering method, respectively (denotes case I and case II). In Fig. 3(a) three evident clusters are formed by the patterns according to the features (input variables $x_1$ and $x_2$), while in Fig. 3(b) four clusters are constructed based on both of the features and the output variable ($x_1$, $x_2$, and $Y$). In case II, the patterns are partitioned into two parts (denoted here as ***context*** 1 and ***context*** 2) not only based on their vicinity in the feature space but also considering the output used in this

***Group the data in input space X***

The case II of context-based similar clustering can be described as follows

***Group the data in input space X considering that the output y is in context A***

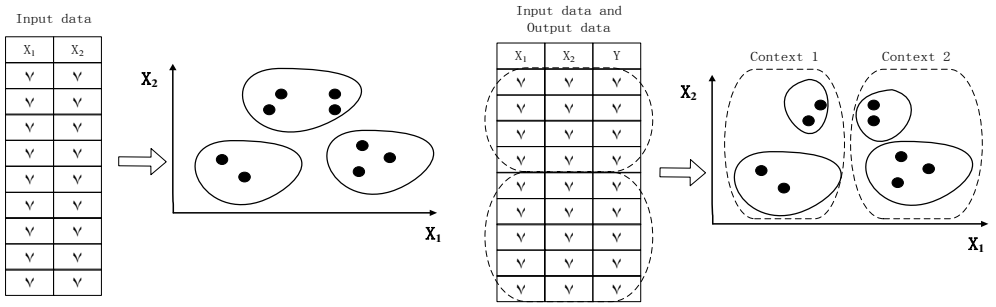



**Fig. 3.** Example of context-based similar clustering (a) case I and (b) case II

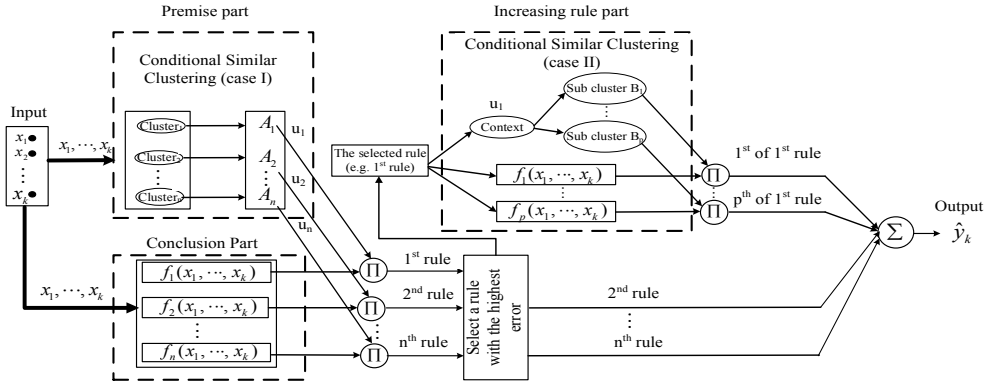



**Fig. 4.** Architecture of granular classifier

data. It noted that the case I of the context-based similar clustering is essentially equal to the conventional similar clustering.

In some sense, the case I of context-based similar clustering can be expressed as follows

## 3. Architecture of Granular Classifier

In (1), we proposed a novel architecture of granular classifier. There are mainly five parts (including input, premise part, conclusion part, increasing rule part, and output) in a general architecture of granular classifier shown in Fig.4. Premise, increasing rule and conclusion parts are corresponding to the form of the "if-then" rules. In this study, conditional similar clustering is used to partition input space to form the premise part as well as the increasing part, while the SVM is utilized to estimate the discriminant functions. It has to be stressed that the GC becomes the "conventional" SVM when there is only one rule in the GC. In other words, the GC emerges as extended classifier of SVM. More specifically, the granular classifier in Fig. 4 can also be represented in the form of a serial "if-then" rules

$$
\begin{aligned}
&\mathbf{R}^1 : IF\ \ \mathbf{x} \text{ is } in\ cluster\ A_1\ with\ B_1\ THEN\ y_i = \mathrm{g}_1(\mathbf{x});\\
&\cdots\cdots\\
&\mathbf{R}^p : IF\ \ \mathbf{x} \text{ is } in\ cluster\ A_1\ with\ \ B_p\ THEN\ y_p = \mathrm{g}_p(\mathbf{x});\\
&\mathbf{R}^{p+1} : IF\ \ \mathbf{x} \text{ is } in\ cluster\ A_2\ THEN\ y_{p+1} = \mathrm{g}_{\mathrm{p+1}}(\mathbf{x});\\
&\cdots\cdots\\
&\mathbf{R}^{p+n-1} : IF\ \ \mathbf{x} \text{ is } in\ cluster\ A_2\ THEN\ y_{p+n-1} = \mathrm{g}_{\mathrm{p+n-1}}(\mathbf{x}).
\end{aligned}
\tag{2}
$$

As shown in Fig. 4, the design meachism of GC is as follows:

Step 1. Generate rules based on the original data.

Step 2. Construct a new data set from the original data based on the rule with worst error.

Step 3. Generate several new rules based on the new data sets.

Step 4. Output the final set of rules.

### 3.1 Design of premise part via context-based similar clustering (Case I)

Let $x_1, x_2, ..., x_N$ be $n$-dimensional patterns defined in

$\Re^n$, the goal of similar clustering approach is to search $v_i$ to maximize the following total similar measure

$$J_s^r(z)=\sum_{i=1}^{c}\sum_{j=1}^{N}\left(S(x_j,v_i)\right)^r \tag{3}$$

where $r$ is a position number that determines the location of the peaks of $J_s(z)$, $S(x_j,v_i)$ represents the similar measure between $x_j$ and the $i$th cluster $v_i$. For convenience, the formulation of $S(x_j,v_i)$ coming from reference [29] can be expressed as follows

$$S_{ij}=S(x_j,v_i)=\exp(-\frac{\|x_j-v_i\|^2}{\beta}) \tag{4}$$

In (4), $\beta$ denotes sample variance that is defined by

$$\beta=\frac{\sum_{j=1}^{N}\|x_j-\bar{x}\|^2}{N}\quad \text{where}\quad \bar{x}=\frac{\sum_{j=1}^{N}x_j}{N}. \tag{5}$$

With the (3) and (4), we have

$$J_s^r(v_i)=\sum_{i=1}^{c}\sum_{j=1}^{N}\exp\left(-\frac{\|x_j-v_i\|^2}{\beta}\right)^r \tag{6}$$

In the design of granular classifier, the premise part is realized by means of context-based similar clustering (case I), while the increasing rule part is realized using context-based similar clustering (case II). To develop the context-based similar clustering method, we define $U, s_i^l$ and $v_i^l$ as shown in (7), (9), (10), respectively.

First, we define $U=[u_{ij}]$ as a $c\times N$ partition matrix, where

$$u_{ik}=\frac{S(x_j,v_i)}{\sum_{k=1}^{c}S(x_j,v_k)}=\frac{1}{\sum_{k=1}^{c}S(x_j,v_k)\Big/S(x_j,v_i)} \tag{7}$$

It follows from (5) that

$$u_{ij}=\left\{u_{ij}\in[0,1]\Big|\sum_{i=1}^{c}u_{ij}=1\ \forall j\right\} \tag{8}$$

Second, we define

$$s_{ij}^l=\begin{cases}\exp(-\dfrac{\|x_j-v_i\|^2}{\beta}), & if\ \ l=0;\\ f(s_{ij}^{l-1}), & if\ \ l\geq 1.\end{cases} \tag{9}$$

At Last, we define

$$v_i^l=\begin{cases}\dfrac{\sum_{j=1}^{N}S_{ij}^r x_j}{\sum_{j=1}^{N}S_{ij}^r}, & if\ \ l=0;\\ f(v_i^{l-1}), & if\ \ l\geq 1.\end{cases} \tag{10}$$

In particular, according to the (4) we have

$$v_i^0=\frac{\sum_{j=1}^{N}S_{ij}^r x_j}{\sum_{j=1}^{N}S_{ij}^r}=\frac{\sum_{j=1}^{N}\exp(-\frac{\|x_j-v_i\|^2}{\beta})^r x_j}{\sum_{j=1}^{N}\exp(-\frac{\|x_j-v_i\|^2}{\beta})^r} \tag{11}$$

The context-based similar clustering (case I) can be summarized as follows

Context-based similar clustering (case I)
Step 1. Estimate the parameter $r$.
  Step 1.1. Set m=1, r=5m, and $\xi_1=0.97$.
  Step 1.2. Calculate the correlation based on the values $J_s^r(z)$ and $J_s^{r+1}(z)$ according to (6).
  Step 1.3. IF the correlation is less than $\xi_1$, go to step 1.5.
  Step 1.4. Set m=m+1, and go to step 1.2.
  Step 1.5. Output r.
Step 2. Set $l=0$.
Step 3. Estimate $s_{ij}^{(l+1)}$ using (4).
Step 4. Estimate $v_i^{(l+1)}$ using (9).
Step 5. $l=l+1$.
Step 6. IF $\max_i\|v_i^{(l+1)}-v_i^{(l)}\|\geq\xi_2$ go to step 3.
Step 7. Output the final states of the data points.
Step 8. Implement Agglomerative Hierarchical Clustering (AHC) [28] with the final states of the data points to identify the c* clusters.
Step 9. Calculate $S_{ij}$ according to the c* clusters.
Step 10. Output the $u_{ij}$ using (7).

### 3.2 Design of increasing rule part via context-based similar clustering (Case II)

The context-based similar clustering (case II) realize the grouping data guided by the contexts expressed in the output space. Unlike the case I, the case II of the context-based similar clustering is to optimize the new reformulated expression

$$\begin{aligned}&\max\ \ J_s^r(t_i)=\sum_{i=1}^{p}\sum_{j=1}^{N}\left(S(x_j,t_i)\right)^r\\ &st:\\ &\quad U\in\mu(f)\end{aligned} \tag{12}$$

The matric $U$ is defined as follows

$$\mu(f)=\left\{\mu_{ik}\in[0,1]\Big|\sum_{i=1}^{p}\mu_{ik}=u_{ij}\quad\forall k\right\} \tag{13}$$

where $u_{ij}$ has been defined in (8).

The maximization of J as completed by the context-based similar clustering is realized by iteratively updating the values of the matrix and centers. The update of the matrix is carried out by using the following expression

$$\mu_{ik}=\frac{u_{ik}}{\sum_{k=1}^{p}S(x_j,t_k)\Big/S(x_j,t_i)} \tag{14}$$

According to (11), the prototypes are expressed as

$$t_i^0=\frac{u_{ik}}{\sum_{j=1}^{N}S_{ij}^r x_j\Big/\sum_{j=1}^{N}S_{ij}^r} \tag{15}$$

where

$$S_{ij}^r=\exp(-\frac{\|x_j-t_i\|^2}{\beta})^r$$

### 3.3 Design of conclusion part via hyper-plane

To find the hyper-plane, here we use the one-again-one method of SVM [8], which constructs $k(k-1)/2$ binary classifiers. The goal of SVM models is to construct $k$ classes, where the $i$th SVM is trained with all of the examples with negative labels. Assume that a given $h$ training data $(x_1,y_1),...,(x_h,y_h)$, where $x_i\in\Re^n, i=1,...,h.$ and $y_i\in\{1,...,k\}$ is the class of $x_i$. A binary classification problem can be formulated as follows

$$\begin{aligned}&\min_{w,b,\xi}\frac{1}{2}\|w\|^2+C\sum_{t=1}^{k}\xi_t\\&st:\\&\quad y_t(w^T\phi(x_i)+b)\ge 1-\xi_t,\quad \xi_t\ge 0, t=1,2,...,k.\end{aligned} \tag{16}$$

where the training data $x_i$ are mapped to a high dimensional space by the function $\phi$, and the parameter $C$ is the penalty parameter controlling the range of $\xi_t$. It is evident that the (16) can be transformed into the following dual problem

$$\begin{aligned}&\min_{\alpha}\frac{1}{2}\alpha^T Q\alpha\\&st:\\&\quad y^T\alpha=0,\quad 0\le\alpha_i\le C, i=1,...,k.\end{aligned} \tag{17}$$

where $Q$ is an $h$ by $h$ positive semi-definite matrix, $Q_{ij}=y_i y_j K(x_i,x_j)$, and $K(x_i,x_j)=\phi(x_i)^T\phi(x_j)$. As to $K(x_i,x_j)$, we use the well-known following Guassian kernel function

$$K(x_i,x_j)=\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right) \tag{18}$$

The value of parameter $\alpha$ can be estimated by running the Sequential Minimal Optimization algorithm [2], and then the optimal $w$ may be obtained by using the formula

$$w=\sum_{i=1}^{m}y_i\alpha_i\varphi(x_i) \tag{19}$$

The decision function is formulated as follows

$$f(x)=\mathrm{sgn}(w^T\varphi(x)+b)=\mathrm{sgn}(\sum_{i=1}^{h}y_i\alpha_i K(x_i,x)+b) \tag{20}$$

For convenience, the decision function of a binary classification for any $i$th and $j$th is denoted as $f_{ij}(x)$, we define

$$q_{ij}=\begin{cases}1, & if\quad f_{ij}(x)>0;\\0, & if\quad f_{ij}(x)\le 0.\end{cases} \tag{21}$$

$$Q_i=\sum_{j=1,j\ne i}^{n}q_{ij} \tag{22}$$

Then the final output of conclusion part is used as the following discriminant function $g_i(x)$

$$g_i(x)=\arg\max_i\sum_{i=1}^{n}u_i Q_i \tag{23}$$

## 4. Design Procedure of Granular Classifiers

In this section, we elaborate on the algorithm details of the overall developing approach in the architecture of granular classifiers. The design procedure of granular classifiers comprises the following steps.

**[Step 1]** ***Division of dataset***

Training, validation, and testing data set are formed through the division of data set. Assume that a general input–output data set is represented as the following way

$$(x_i,y_i)=(x_{1i},x_{2i},...,x_{ni},y_i),\ i=1,2,...,N$$

where $N$ is the number of data points. And the classification rate ( CR ) is denoted as follows

$$CR=\frac{T}{N}\times 100\% \quad (24)$$

where $T$ is the total number of correct classification cases.

The data set is partitioned into three parts, namely training data, validation data, and testing data. The training and validation data are used to construct the GCs, while the testing data is utilized to evaluate the quality of the classifiers. For convenience, $TR$ represents the classification rate for the training data, $VA$ stands for the classification rate for the validation data, and TE means the classification rate for the testing data. The objective function OF (performance index) includes both the training data and validation data is expressed in the form

$$OF=\frac{TR+VA}{2} \quad (25)$$

**[Step 2]** ***Construct premise part of rules using CSC (Case I)***

To develop the premise part of rules for the design of granulation classification, we run CSC (case I) for getting the three important parameters when determining the premise part of rules. The three parameters are the number of clusters (the number of rules), the clusters, and the partition matrix (membership), respectively.

**[Step 3]** ***Construct conclusion part of rules via finding hyper-plane***

Here we decide upon the essential parameters in the conclusion part of rules for the design of granulation classification. To find the hyper-plane, here we use the one-again-one method of SVM [7-8], which constructs $k(k-1)/2$ binary classifiers. Each model in a rule is constructed by the training data, while TE and VA are obtained based on the validation and testing data of the GCs that is developed by running the training data, respectively.

**[Step 4]** ***Check the termination criterion***

Two conditions are considered here as the termination criterion. One is that the number of current loops is less than a given number. The other is the value of local models with the highest error is less than a given values. It is noted that the size (rules) of granulation classifiers has been experimentally found to form a sound compromise between the high accuracy of the resulting GCs and its complexity as well as generalization abilities.

**[Step 5]** ***Select one rule with the highest error***

According to (25), the error of each rule can be obtained and then one rule with the highest error will be selected for growing new rules in the increasing rule part as shown in Fig. 5.

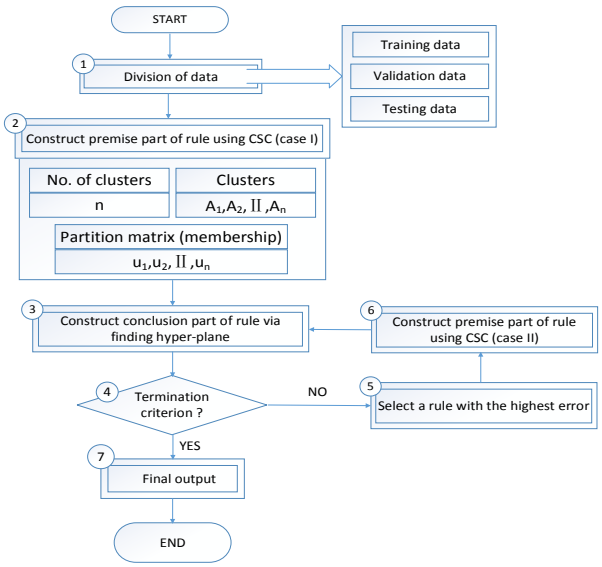



**Fig. 5.** An overall flowchart of design of Granular Classifier

**[Step 6]** ***Construct increasing rule part using CSC (Case II)***

With the use of CSC (case II), the partition matrix (membership) of the selected rule is further sub-partitioned into two or more clusters. Similar to the construct premise part, the increasing rule part is designed for getting the related important parameters for constituting new rules.

**[Step 7]** ***Final output***

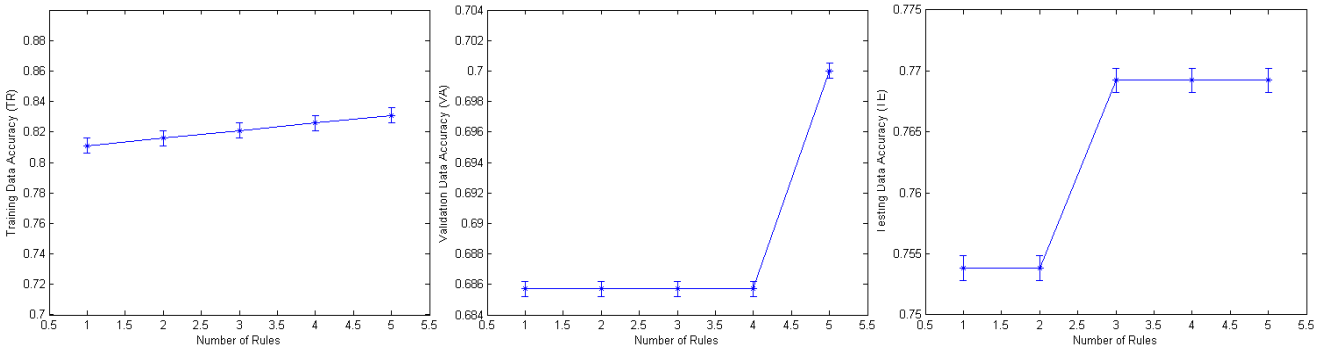
Output the final output.

## 5. Experimental Studies

In this section, we report the experimental results. First, the proposed classifier is evaluated based on several machine learning data. Then, the proposed classifier is applied on the network intrusion detection data.

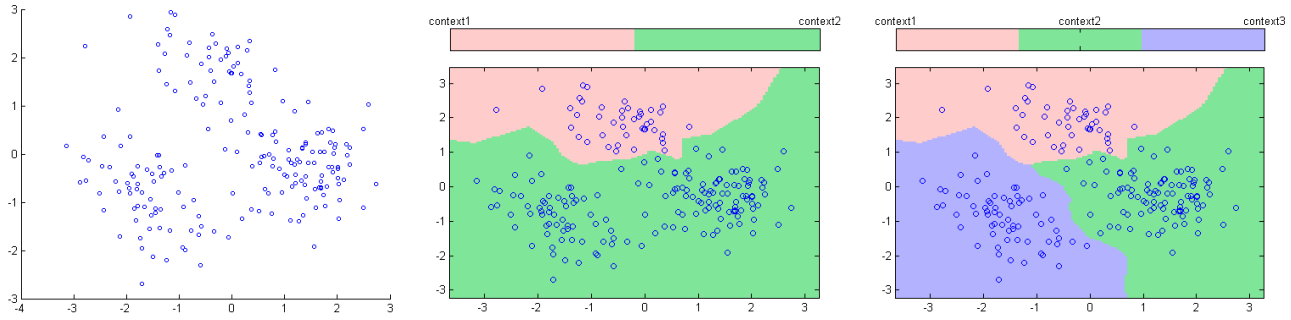
### 5.1 Machine learning data

To illustrate the performance of the proposed granular classifiers, we experimented several well-known machine learning data sets [19-22]. Three data sets of them are first studied, and then the results of some other real-world problems of varying levels of complexity (different sizes of data sets) are summarized in the Appendix. Table 1 shows the description of all data sets used in the experiments. In all experiments, each data set is divided into three parts: 60% of data set is considered for training; 20% of data set is used as validation data; and the remaining 20% data set is utilized for testing. In the premise part of GC, the number of clusters (contexts) of CSC is set as two; while in the consequence part Gaussian kernel is utilized. The symbols in these experiments are summarized as follows: TR represents for the performance index (classification rate)

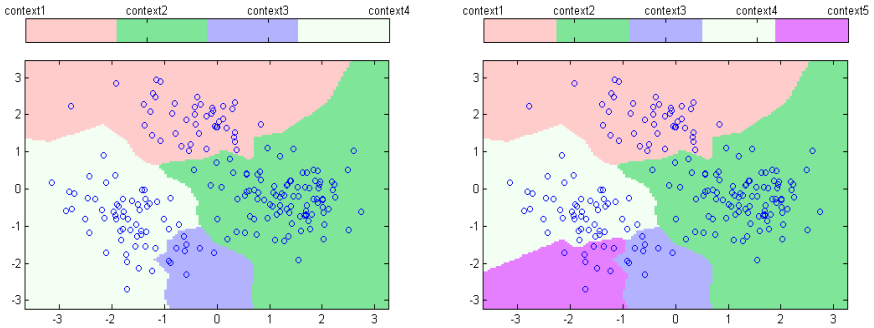



(a) Training data (b) Validation data (c) Testing data

**Fig. 6.** Performance index of GCs for the Ecoli data




(a) One context (b) Two contexts (c)Three contexts




(d) Four contexts (e) Five contexts

**Fig. 7.** Identification process quantified in term of the context partition for the Ecoli data

for the training data; VA stands for the validation data; and the TE denotes the testing data.

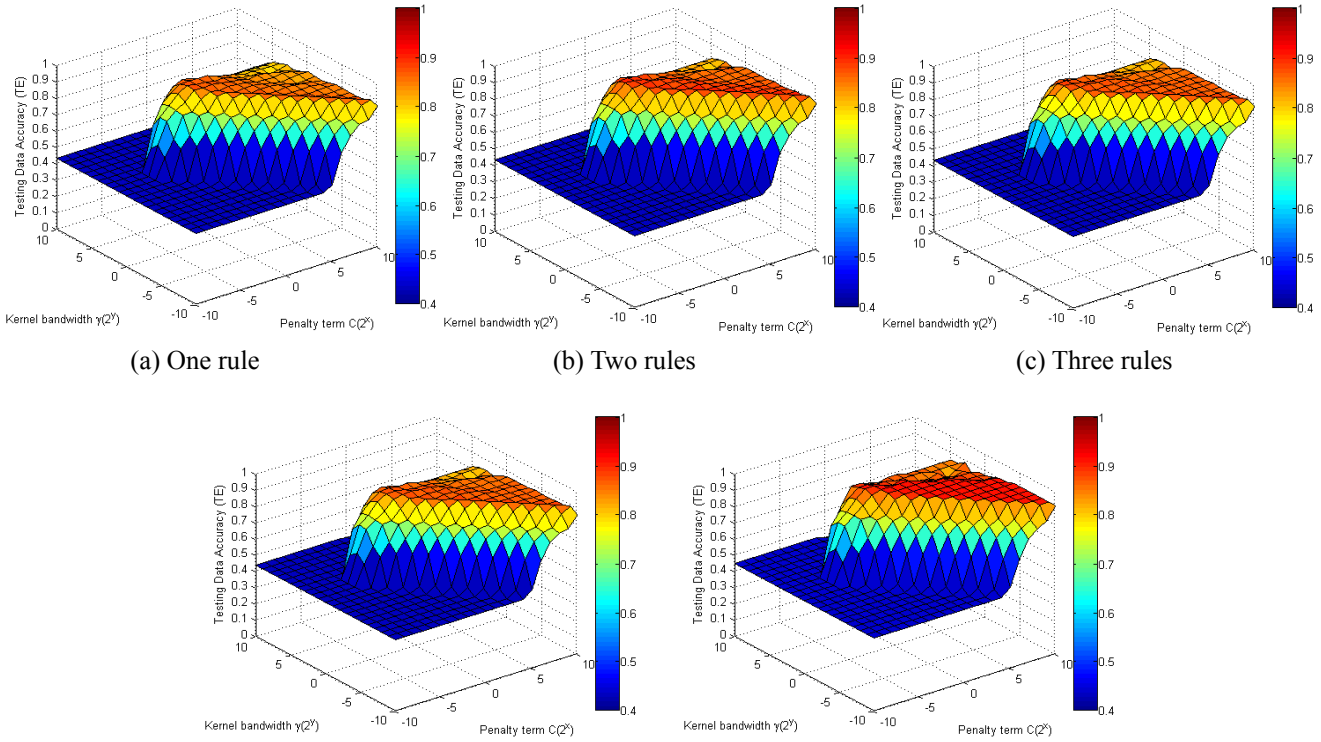
### *A. Ecoli Data*

The GCs are applied to the Ecoli data set, which involves 336 patterns. Each pattern is described by eight input variables.

The number of rules impact the performance of GCs are illustrated in Fig. 6. Their increase produces the improving classification accuracy rates for the training set, validation set, and testing set. This tendency also shows that increasing the number of rules enhances the prediction abilities.

The identification process ranged from one context to five contexts for the Ecoli data are described as shown in Fig. 7. Fig. 8 displays the values of performance index TE range from one rule to five rules for the Icoli data. In case of one rule, the value of performance index with the best parameters selection is still less than the 90%; while in case of five rules, the value of performance index TE is more than 90% with the adaptive parameter selection. This tendency illustrates that the value of TE is enhanced with the increasing number of rules.

Table 1 reports the experimental results of comparative classification rate of the proposed GC with some previous classifiers. It is shown in Table 1 that the proposed classifier provides substantially higher classification rate for Ecoli data.




(a) One rule (b) Two rules (c) Three rules

(d) Four rules (e) Five rules

**Fig. 8.** Performance index (TE) range from one rule to five rules for the Ecoli data

**Table 1.** Comparison of classification rate with previous classifiers (Ecoli data)

| Classifier | TR | VA | TE |
|---|---|---|---|
| SVM[19] | - | - | 89.18 |
| LDA[20] | - | - | 88.12 |
| NDA[21] | - | - | 88.29 |
| SVM+LDA[20] | - | - | 89.54 |
| HLDA[22] | - | - | 90.24 |
| Our classifier (rules=5) | 92.21±0.29 | 86.67±0.82 | 90.77± 0.64 |

### *B. Selected Machine Learning Data*

To further evaluate the proposed granular classifier, five well-known data sets which have different number of variables and available data are tested. Table 2 describes the main features of the seven data sets.

A comparative analysis considering some classifiers reported in the literatures are summarized as shown in Table 3. It is evident that the granular classifier outperform the better classification rate in comparison with these existing classifiers.

**Table 2.** Description of seven machine learning data sets

| Datasets | Classes | Variables | Patterns |
|---|---|---|---|
| Brest-Cancer | 2 | 9 | 277 |
| Ionosphere | 6 | 10 | 351 |
| Balance | 3 | 4 | 625 |
| Diabetes | 2 | 8 | 768 |
| German | 2 | 20 | 1000 |
| Flare-solar | 2 | 9 | 1066 |
| Banana | 2 | 2 | 5300 |

**Table 3.** Comparison of classification rate with some selected classifiers

| Classifier | Dataset | TR | VA | TE |
|---|---|---|---|---|
| The best classifier listed in reference | Brest-Cancer [19, 20, 21, 22] | - | - | 73.83 |
| | Ionosphere[23, 24] | - | - | 90.31 |
| | Balance[19, 23, 24] | - | - | 94.72 |
| | Diabetes[19-22] | - | - | 78.41 |
| | German[19-22] | - | - | 76.77 |
| | Flare-solar[19-22] | - | - | 66.59 |
| | Banana[19-22] | - | - | 62.78 |
| The proposed GC | Brest-Cancer | 85.61±7.81 | 74.64±2.15 | 81.29±0.51 |
| | Ionosphere | 99.05±0.96 | 96.67±0.00 | 97.40±0.27 |
| | Balance | 99.82±0.31 | 97.88±1.21 | 97.13±0.62 |
| | Diabetes | 85.61±0.82 | 76.82±0.29 | 78.43±0.34 |
| | German | 83.27±0.11 | 78.66±0.34 | 81.11±0.13 |
| | Flare-solar | 66.08±0.82 | 66.52±0.73 | 71.17±0.35 |
| | Banana | 93.47±0.27 | 89.08±0.10 | 89.77±0.05 |

## 5.2 KDD CUP 99 data

To illustrate the performance of the proposed RCs for solving the network intrusion detection problem, here we test it on the well-known KDD Cup 99 dataset. There are 5,000,000 labeled records and 41 attributes in the original KDD dataset, which consists one type of normal data and 24 different types of attacks that are categorized in four groups of DDOS, Probe, U2R, and R2L [25-27]. It is noted that some researchers recommend using 10% KDD Cup 99 when resolving many issues with the dataset. Table 4 describes the filtered 10% KDD Cup 99 dataset [25-27].

**Table 4.** Distribution for the 10% KDD Cup 99 NSL-KDD dataset tested

| Datasets | Classes | Variables | Patterns |
|---|---|---|---|
| Brest-Cancer | 2 | 9 | 277 |
| Ionosphere | 6 | 10 | 351 |
| Balance | 3 | 4 | 625 |
| Diabetes | 2 | 8 | 768 |
| German | 2 | 20 | 1000 |
| Flare-solar | 2 | 9 | 1066 |

In our experiments, we also used the 10% KDD Cup 99 training data with duplicates removed for the first set of tests with a dataset that consists of 145,585 samples. The data set is split into two parts, 50% of data set is used for training, while the remaining 50% are considered for testing. To compare with other methods, we utilized the following performance measures [25-27]

True Positive (TP). A "True Positive" is a correct detection of an attack in the intrusion detection.

False Positive (FP). A "False Positive" is an indication of an attack on traffic that should have been classified as "normal".

True Negative (TN). A "True Negative" is a correct identification of "Normal Traffic" in the intrusion detection.

False Negative (FN). A "False Negative" is a real attack that was misidentified as "Normal" traffic.

Accuracy. "Accuracy" is the common metric utilized for assessing the overall effectiveness of a classifier. The expression of Accuracy is as follows

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \qquad (8)$$

The number of rules impacts the Accuracy of RCs as shown in Fig. 9. Their increase leads to the increase of accuracy in case of the number of rules is less than four, while it becomes low when the number of rules arrives at five. This tendency illustrates that the relative optimal classifier could be emerged with the adaptive selecting the number of rules. Fig. 10 further depicts the values of Accuracy ranging from one rule to five rules with different parameters.

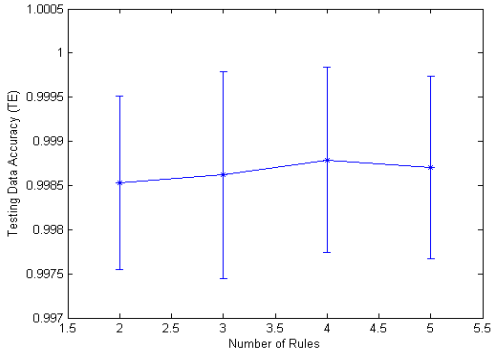
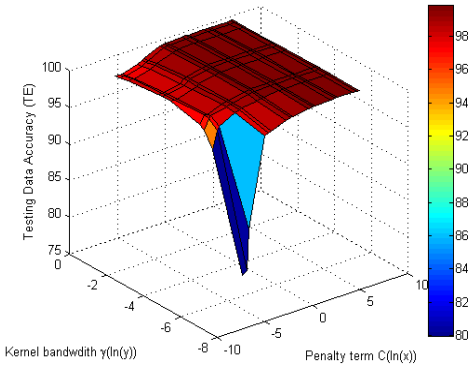
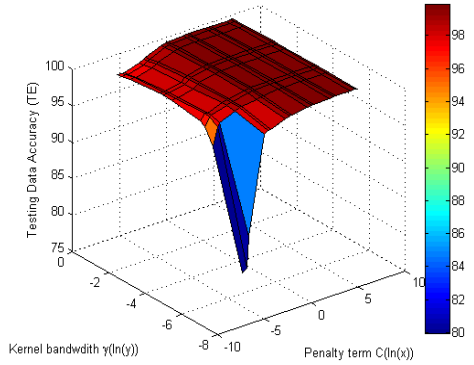


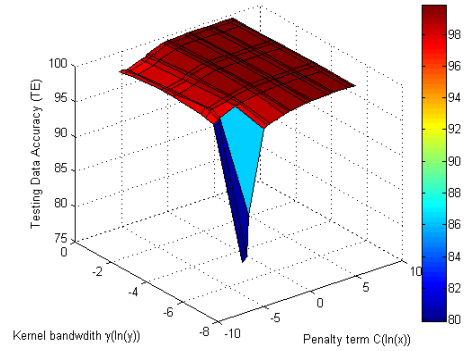

**Fig. 9.** Accuracy of RCs with different rules



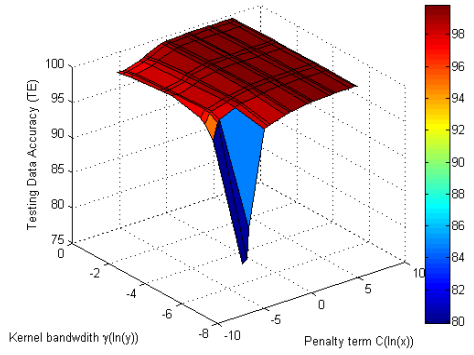

(a)Two rules




(b)Three rules




(c) Four rules




(d) Five rules

**Fig. 10.** Accuracy of RCs with different parameters

**Table 5.** Comparison of performance with some selected approaches

| Classifier | Testing samples | TP/ FP | DDoS | Probe | U2R | R2L | Mean TP/FP ratio |
|---|---|---|---|---|---|---|---|
| ID3 estimated [25] | 311,029 | TP | 99.9 | 99.7 | 49.1 | 93.5 | 983.50 |
| | | FP | 0.03 | 0.55 | 0.15 | 0.98 | |
| Naïve Bayes [26] | 311,029 | TP | 99.69 | 99.11 | 64 | 99.11 | 795.50 |
| | | FP | 0.04 | 0.45 | 0.14 | 8.02 | |
| ID3 [25] | 311,029 | TP | 99.52 | 97.85 | 49.21 | 92.75 | 756.66 |
| | | FP | 0.04 | 0.55 | 0.14 | 10.03 | |
| SVM [27] | 10,000 | TP | 76.7 | 81.2 | 21.4 | 11.2 | 371.32 |
| | | FP | 0.09 | 0.36 | 0.08 | 0.08 | |
| JRip [28] | 15,437 | TP | 97.4 | 83.8 | 12.8 | 0.1 | 322.73 |
| | | FP | 0.3 | 0.1 | 0.1 | 0.4 | |
| BayesNet [29] | 15,437 | TP | 94.6 | 83.8 | 30.3 | 5.2 | 306.82 |
| | | FP | 0.2 | 0.13 | 0.3 | 0.6 | |
| Random forest classifier [29] | 77,287 | TP | 98.91 | 55.12 | 100 | 66.67 | 234.00 |
| | | FP | 3.15 | 0.45 | 0.13 | 5.18 | |
| NBTree [28] | 15,437 | TP | 97.4 | 73.3 | 1.2 | 0.1 | 40.00 |
| | | FP | 1.2 | 1.1 | 0.1 | 0.5 | |
| MLP [28] | 15,437 | TP | 96.9 | 74.3 | 20.1 | 0.3 | 40.00 |
| | | FP | 1.4 | 0.1 | 0.1 | 0.5 | |
| Our approach (RC) | 15,437 | TP | 99.98 | 96.83 | 100 | 97.55 | 3055 |
| | | FP | 0.01 | 0.044 | 0 | 0.88 | |

Table 5 summarizes the comparison of RC per class detection and false positive performance to some conventional approaches. For each line, the results of DDos, Probe, U2R, and R2L based on TP and FP are listed, respectively. Among the cited approaches, there is a wide range of results but very few approaches have consistently good detection performance across all four attack classes except from the proposed RC. From the result of DDos, Probe, U2R, and R2L, our approach not only have better accuracy, but also the mean TP/FP ratio is much higher than the existing methods. It is clear that the proposed RC obtains better performance in comparison with other approaches listed in the literatures.

## 6. Concluding Remarks

Classical rule-based classifiers train the data set without adequate use of output data. With this regard, we first propose a context-based clustering method and then develop a granular classifier to alleviate this limitation. The work contributes to the research on classification can be summarized as the following two aspects: 1) we have proposed a context-based similar clustering method, which is exploited here to realize the premise part of the granular classifier, and 2) we have designed the architecture of granular classifiers. By combining context-based clustering method and support vector machine, we take advantages of the two technologies when dealing the classification problems.

For future study, granular classifiers may be improved by constructing new clustering method or new architectures. Furthermore, the proposed context-based clustering method can be applied to design new granular classifiers by combining other classical classifiers.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant nos. 61301140, 61673295, 61272450, 61562024), supported by The Key Technology Research and Development Program of Tianjin (Grant no. 14ZCZDGX00072), supported the Foundation of Educational Commission of Tianjin City, China (Grant no. 20120703), supported by the Open Foundation of State Key Laboratory of Digital manufacturing Equipment & Technology (Grant no. DMETKF2015012), and supported by the Open Foundation of State Key Laboratory of Virtual Reality and Technology and Systems, Beihang University, China (Grant no. BUAA-VR-14KF-11).

## References

[1] D. Parikh and T. Chen, "Data fusion and cost minimization for intrusion detection," *IEEE Transacions on Information Forensics and Security*, Vol. 3, No. 3, pp. 381-389, 2008.

[2] V. A. Sotiris, P.W. Tse and M.G. Pecht, "Anomaly detection through a Bayesian support vector machine," *IEEE Transactions on Reliability*, Vol. 59, No. 2, pp. 277-286, 2010.

[3] C. Thomas and N. Balakrishnan, "Improvement in intrusion detection with advances in sensor fusion," *IEEE Transacions on Information Forensics and Security*, Vol. 4, No. 3, pp. 542-551, 2009.

[4] N. Li, W. L. Tsang and Z.H. Zhou, "Efficient optimization of performance measures by classifier adaption," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 5, pp. 2683-2695, 2014.

[5] A.J. Ma, P.C. Yuen and J.H. Lai, "Linear dependency modeling for classifier fusion and feature combination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 5, pp. 1135-1148, 2013.

[6] B. Verma and A. Rahman, "Cluster-Oriented ensemble classifier: impact of multi-cluster characterization on ensemble classifier learning," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 4, pp. 605-618, 2012.

[7] L. Lu, L. Di and Y. Ye, "A decision-tree classifier for extracting transparent plastic-mulched land cover from landsat-5 TM images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 7, No. 11, pp. 1314-1330, 2014.

[8] S.W. Thomas, M. Nagappan, D. Blostein and A.E. Hassan, "The impact of classifier configuration and classifier combination on bug localization," *IEEE Transactions on Software Engineering*, Vol. 7, No. 11, pp. 1314-1330, 2014.
[9] C. Gao, Q. Ge and L. Jian, "Rule extraction from fuzzy-based blast furnace SVM multiclassifier for decision-making," *IEEE Transactions on Fuzzy Systems*, Vol. 22, No. 3, pp. 586-596, 2014.
[10] D. Fisch, E. Kalkowski and B. Sick, "Knowledge fusion for probabilistic generative classifiers with data mining applications," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 3, pp. 652-666, 2014.
[11] Z. Ma, Y. Yang, N. Sebe, K. zheng and A.G. Hauptmann, "Multimedia event detection using a classifier-specific intermediate representation," *IEEE Transactions on Multimedia*, Vol. 15, No. 7, pp. 1628-1637, 2013.
[12] X. Zhai, K. Appiah, S. Ehsan, G. Howells, H. Hu, D. Gu, and K.D. McDonald-Maier, "A method for detecting abnormal program behavior on embedded devices," *IEEE Transacions on Information Forensics and Security*, Vol. 10, No. 8, pp. 1692-1704, 2015.
[13] H. Xue, S. Chen and Q. Yang, "Structural regularized support vector machine: A framework for structural large margin classifier," *IEEE Transactions on Neural Networks*, Vol. 22, No. 4, pp. 573-587, 2011.
[14] D. J. Sebald and J.A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Transactions on Signal Processing*, Vol. 48, No. 11, pp. 3217-3226, 2000.
[15] M.R. Alam and K.M. Muttaqi, A. Bouzerdoum, "An approach for assessing the effectiveness of multiple-feature-based SVM method for islanding detection of distributed generation," *IEEE Transactions on Industry Applications*, Vol. 50, No. 4, pp. 2844-2852, 2014.
[16] F.D. Morsier, D. Tuia, M. Borgeaud, V. Gass and J.P. Thiran, "Semi-Supervised novelty detection using SVM entire solution path," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 51, No. 4, pp. 1939-1950, 2013.
[17] C. Cortes and V. vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, No. 3, pp.273-297, 1995.
[18] M.S. Yang and K.L. Wu, "A similarity-based robust clustering method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 4, pp. 434-448, 2004.
[19] V. Vapnik, "The Nature of Statistical Learning Theory," *Spring-Verlag*, 1995.
[20] T. Xiong and V. Cherkassy, "A combined SVM and LDA approach for classification," *Proceedings of the International Joint Conference on Neural Networks*, pp. 1455-1459, 2005.
[21] S. Mika et al, "Fisher discriminant analysis with kernels," *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, pp. 41-48, 1999.
[22] R. Ksantini and B. Boufama, "Combining partially global and local characteristics for improved classification," *International Journal of Machine Learning*, Vol. 3, pp. 119-131, 2012.
[23] B. Minaei-Bidgoli, H. Parvin, H. Alinejad-Rokny, H. Alizadeh and W.E. Punch, "Effects of resampling method and adaptation on clustering ensemble efficacy," *Artif. Intell. Rev.*, Vol. 41, No. 1, pp. 27-48, 2014.
[24] H. Parvin, M. Mirnabibaboli and H. Alinejad-Rokny, "Proposing a classifier ensemble framework based on classifier selection and decision tree," *Engineering Applications of Artificial Intelligence*, Vol. 37, pp. 34-42, 2015.
[25] V. Jaiganesh, S. Mangayarkarasi, and P. Sumathi, "An efficient algorithm for network intrusion detection system," *International Journal of Computer Applications*, Vol. 90, No. 12, pp. 12-16, 2014.
[26] D. M. Farid, N. Harbi, and M.Z. Rahman, "Combining Naïve Bayes and decision tree for adaptive intrusion detection," *International Journal of Network Security and Its Applications*, Vol. 2, No. 2, pp. 12-25, 2010.
[27] X. Xu, "Adaptive intrusion detection based on machine learning: Feature extraction, classifier construction and sequential pattern prediction," *International Journal of Web Services Practices*, Vol. 2, No.1, pp. 49-58, 2006.
[28] H. A. Nguyen, and D. Choi, *Application of data mining to network intrusion detection: Classifier selection model*. Berlin Heidelberg: Springer. 2014.
[29] M. A. M. Hasan, M. Nasser, B. Pal, and S. Ahmad, "Support vector machine and random forest modeling for intrusion detection system (IDS)," *Journal of Intelligent Learning Systems and Applications,* Vol. 6, pp. 45-52, 2014.

**Wei Huang** received the M.Sc. degree from the School of Information Engineering, East China Institute of Technology, Jiangxi, China, in 2006, and Ph.D. degree at State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China, in 2011. He is currently an Associate Professor in the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China. His research interests include evolutionary computation, opeartions research, fuzzy system, fuzzy-neural networks, and advanced computational intelligence.

**Jinisong Wang** received the B.Sc., M.Sc. degrees at dept. of computer science from Tianjin University of Technology, and Ph.D. degree at Tianjin University, respectively, in China. He is currently a Professor in the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China. His main research interests include network security, computer networks, and advanced computational intelligence.

**Jiping Liao** is an undergraduate student in the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China. His main research interests include network security and computational intelligence.