A Novel Kernel SVM Algorithm with Game Theory for Network Intrusion Detection

Yufei Liu and Dechang Pi*
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
Nanjing, 211106, Jiangsu - P.R. China
[e-mail: liuyufei@nuaa.edu.cn, dc.pi@nuaa.edu.cn]
*Corresponding author: Dechang Pi

Received November 27, 2016; revised March 24, 2017; accepted April 28, 2017; published August 31, 2017

Abstract

Network Intrusion Detection (NID), an important topic in the field of information security, can be viewed as a pattern recognition problem. The existing pattern recognition methods can achieve a good performance when the number of training samples is large enough. However, modern network attacks are diverse and constantly updated, and the training samples have much smaller size. Furthermore, to improve the learning ability of SVM, the research of kernel functions mainly focus on the selection, construction and improvement of kernel functions. Nonetheless, in practice, there are no theories to solve the problem of the construction of kernel functions perfectly. In this paper, we effectively integrate the advantages of the radial basis function kernel and the polynomial kernel on the notion of the game theory and propose a novel kernel SVM algorithm with game theory for NID, called GTNID-SVM. The basic idea is to exploit the game theory in NID to get a SVM classifier with better learning ability and generalization performance. To the best of our knowledge, GTNID-SVM is the first algorithm that studies ensemble kernel function with game theory in NID. We conduct empirical studies on the DARPA dataset, and the results demonstrate that the proposed approach is feasible and more effective.

Keywords: Network intrusion detection, SVM, Kernel method, Game theory, Nash equilibrium

This work is supported by the National Natural Science Foundation of China under Grant U1433116, and the Funding of Jiangsu Innovation Program for Graduate Education under Grant KYLX15_0324.

https://doi.org/10.3837/tiis.2017.08.016 **ISSN : 1976-7277**

1. Introduction

Intrusion detection system, a tool of active defense, is an important research topic in the field of information security [1-3]. In the year of 2000, Terran *et al*. [4] put forward that the machine learning techniques can be used in a network intrusion detection system (NIDS), which can divide the network information data into two parts: the normal data and the abnormal data. Therefore, the network intrusion detection problem can be viewed as a pattern recognition problem for which there are many mature machine learning based solution methods [5, 6]. Having enough training samples is the research premise of the traditional pattern recognition methods, thus the existing methods can achieve a good prediction performance in theory when the number of training samples is large enough. However, modern network attacks are diverse and constantly updated, for example, Jo Minho *et al.* [7] identify a new selfish attach type (named Type 3) and proposed the cooperative detection algorithm of the flagship reference in IDS in wireless communications. Comparing with the testing samples in a practical application of NID, the training samples have much smaller size. Therefore, for the problem of network intrusion detection, the traditional machine learning methods often encounter the problems of over-fitting, under-fitting or local minima.

The learning ability is greatly improved by introducing the kernel functions into the Support Vector Machine (SVM) [8], which is a very popular and effective machine learning method, where the kernel function is a key factor to the effective use of the high-dimensional feature space and will directly affect the generalization performance of SVM. Because kernel function is an extremely important part of the SVM, it determines nonlinear processing ability of the SVM, and it is the key to the mature development of the SVM theory. Literature [9] shows that kernel method can improve the computing power of SVM by projecting the original dataset to a high-dimension feature space, where the kernel function implicitly determines the projecting function and the feature space. Meanwhile, the parameters of a kernel function determine the complexity of the obtained sample subspaces, and each subspace corresponds to a hyper-plane that has the best generalization performance. At the same time, the corresponding penalty parameter C is optimal in SVM. Kernel method has gradually permeated many fields of machine learning, such as the regression estimation [10] and the pattern classification [11]. At present, the researches of kernel function mainly focus on the selection, construction and improvement of kernel functions. The effective solution of these problems plays an important role in the future development of SVM theory, but these problems are still open questions. Literature [12] proposed the multi-kernel learning method, but the construction of multi-kernel is still a difficult point in applications. In literature [13], Sam Hare *et al*. put forward a new approach of constructing multi-kernel, using different kernel functions to construct the combined kernel function. In other words, according to the convex combination of kernel functions, a new kernel function satisfying the Mercer condition is constructed. Nevertheless, it has no basis for the selection and combination of kernel functions. Thus, the construction of the kernel combination is always a difficult point. To the best of our knowledge, there are no theories to solve the problems of the selection and construction of kernel function perfectly, such as the problem of how to deal with the relationship among different kernels, and the problem of the parameters' rationality in the multi-kernel learning method. The structure of the kernel function has always been the research focus in machine learning [14].

Different kernel functions have different learning ability and generalization performance. Among all the kernel functions, we observe that the radical basis function kernel has a stronger ability of interpolation and is good at reflecting local properties of the samples, and its disadvantage is that it cannot extract the global features from all training samples. For the polynomial kernel function, its interpolation ability is relatively weaker and is not good at reflecting local properties. However, it can well extract the global features of training samples. Based on this observation, our motivation is to integrate effectively the advantages of the radial basis function kernel and the polynomial kernel based on the game theory for NID to get a better learning ability and generalization performance of SVM classification.

To achieve the goal, we propose a novel kernel SVM algorithm with game theory for NID, called GTNID-SVM, where a new ensemble kernel function is constructed for SVM based on the Game Theory [15] to obtain the optimal classification hyper-plane. Experimental results show that, comparing with the benchmark model, the proposed approach achieves higher testing precision and lower false positive rate. Therefore, the proposed approach improves the adaptability and extensibility of the NID method based on SVM.

The rest of the paper is organized as follows. In Section 2, we review the related work. In Section 3, we discuss the ensemble kernel function constructed on the notion of the game theory, and then a novel SVM classification framework using the ensemble kernel for network intrusion detection is proposed. In Section 4, we empirically compare the GTNID-SVM with a benchmark RBF-Kernel SVM on the standard dataset DARPA that is from the Lincoln Laboratory of MIT by handling many real defense problems. Section 5 concludes this paper.

2. Related Work

The proposed algorithm, GTNID-SVM, is an ensemble kernel based SVM algorithm for NID. To the best of our knowledge, GTNID-SVM is the first method that studies ensemble kernel function with game theory in network intrusion detection. Existing intrusion detection techniques in machine learning are SVM [16], neural networks [17], C4.5 [18], etc. Although SVM based NID improved the performance in term of detection accuracy, there are considerable rooms for improvement. The proposed approach can adaptively exploit ensemble kernel with the game theory to improve the performance of SVM and reduce false alarm in NID.

2.1 The network intrusion detection with SVM

In NID, we consider a network connection as a sample. For each network connection, assuming that it has n features. The network connection can be represented as $\mathbf{x}=\{x_1, x_2, \cdots, x_n\}$, $\mathbf{x} \subseteq \mathbf{R}^n$, where $x_i (i=1,2,\cdots,n)$ represents the i^{th} feature. Given a training set $\{(\mathbf{x}_i, y_i)\}$, $1 \le i \le l$, $\mathbf{x}_i \in \mathbf{X} \subseteq \mathbf{R}^n$, $y_i \in Y=\{-1,+1\}$, $\mathbf{X}$ is the set of network connections, and represents the input space. Y represents the output field. When the network connection is normal, y =1; when it is abnormal, y =-1; l is the number of samples. Based on the definition, the network intrusion detection is to design an optimal classifier $f(\mathbf{x}): \mathbf{X} \to Y$, which can find an optimal classification hyper-plane. It is not only to be able to properly separate the normal and abnormal connections, but also make the margin maximum.

Literature [19] have suggested host-based anomaly detection method using robust SVM. Literature [18] compared C4.5 with SVM to show the better performance of SVM in NID.

Mukkamala *et al*. [17] applied SVM to network intrusion detection system and compared its performance with neural network based network intrusion detection system; the results show that SVM had better performance than neural network in terms of accuracy and processing speed. Literature [20-22] proposed three network intrusion detection methods based on SVM respectively. Literature [22] proposed a fuzzy SVM algorithm by introducing a fuzzy member function into the multiclass SVM classifier [23], and applied it to NID to achieve good results. On the basis of one-class SVM, literature [21] proposed a SVM method suitable for network traffic anomaly detection. The method generated an SVM classifier based on a statistical analysis of the existing large-scale network data, and adopted the generated classifier to decide the attack type. Literature [20] proposed a general SVM based intrusion detection model, which consists of the auditing data preprocessor, the support vector machine classifier and the decision system. The proposed approach can improves the adaptability and extensibility of the NID method based on SVM.

2.2 Two typical kernels in SVM

Given a training set of sample label pairs $(\mathbf{x}_i, y_i)$, $i = 1, 2, \cdots, l$, where $\mathbf{x}_i \in \mathbf{R}^n$ and $y_i \in \{+1, -1\}$, SVM requires the solution to the following optimization problem.

$$\min\left\{\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i\right\}$$

$$s.t.\ \ y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \xi_i \geq 0 \tag{1}$$

The function $\phi(\mathbf{x})$ can project the training vectors into a high-dimensional (maybe infinite) space. SVM finds a linearly separable hyper-plane with the biggest margin in this high-dimensional space. The penalty parameter of an error term is $C \succ 0$. C is used to regulate the proportion of the credible range and empirical risk in the feature space. Furthermore, the function $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ is the kernel.

In practice, two kernels are usually used, i.e., the radial basis function kernel (a typical representative of local kernels) and the polynomial kernel (a typical representative of global kernels) [24], whose formulae are shown in Eq.(2) and Eq.(3), respectively.

$$K_{rbf}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma^2}\right),\ \sigma \succ 0 \tag{2}$$

$$K_{ploy}(\mathbf{x}_i, \mathbf{x}_j) = \left(\left(\mathbf{x}_i{}^T\mathbf{x}_j\right) + c\right)^d, (d \in N, c \geq 0) \tag{3}$$

Fig. 1 shows that K_{rbf} only has an impact on the test point in a small area nearly. Because the radius of K_{rbf} is constantly changing, the influence of kernel function is changing correspondingly, where σ (radius) is 0.1, 0.2 or 0.4 respectively. The test point is 0.5 according to a prior knowledge and experimental trial in engineering. We observe that the greater the value of σ, the smaller the valid scope of K_{rbf}, besides, its generalization ability decreases with the increase of the parameters.

When we use the polynomial kernel, the samples that are distant from each other in a dataset mainly influence the value of kernel function. From Eq.(3), we observe that no matter how much the actual distance between two samples, each point in the training set has an impact on the kernel function value of the test point. To the best of our knowledge, the parameters of polynomial kernel are obtained by a prior knowledge, user expertise or experimental trial in engineering. Therefore, we select the polynomial degree d =2, 3 or 4, and offset coefficient c =0 according to the usual practice. **Fig. 2** shows that the polynomial kernel has a non-zero value in all points. In other words, the test point 0.5 not only has an effect on the nearby points of small scale but also on the points that are far away from the test point, when the value of parameter d is 1, 2 or 4, respectively.

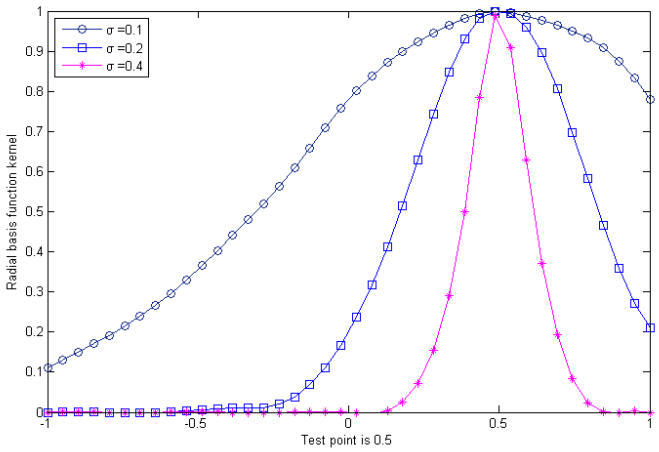

Fig. 1. Characteristic of the radial basis function kernel

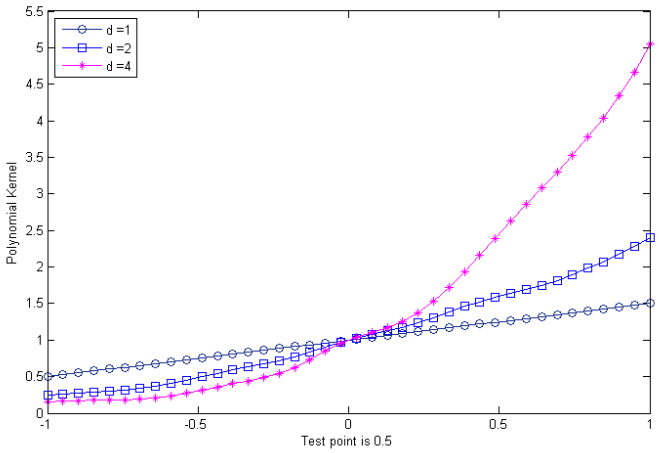

Fig. 2. Characteristic of the polynomial kernel

3. PROPOSED METHOD

In this section, we first design the two-player (two kernels) game theory model. Then, we validate the game theory model to obtain the game kernel function. Next, we prove that the game kernel function satisfies the Mercer condition. Finally, we illustrate the proposed system framework.

3.1 The two-player game model

The game theory mainly studies the problem of making strategic decision. The theory consists of three components: a set of players, a strategy set for each player and a utility function for each player measuring the degree of "satisfaction" of a player [15]. We can use a tuple $G=\{n,S^*,U_n\}$ to represent a game, where $n=\{1,2,\cdots,N\}$ is a set of players, S^* is a set of strategies available to these users, and U_n is the utility function for user n.

Because the two typical kernels classify samples using different hyper-planes, aiming to get better classification results, we try to integrate the two kernels' advantages. To achieve this goal, a game theory model of two players is established. In our model, we regard the two kernel functions as two players. For a training sample, each player has two kinds of possibility, namely "change" or "not change" from K_{rbf} to K_{ploy}. In our model, the strategy set for both players is $\{change, nochange\}$. We describe K_{rbf} 's two options (change inner product, or not change inner product) as the rows of a 2×2 table, and the two columns of the table is for the other player. The set of strategies S^* is $\{change, nochange\}$. **Fig. 3** shows the particular form of our model.

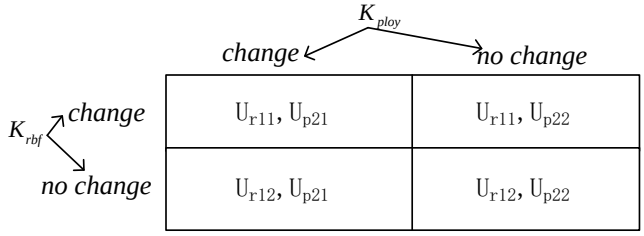

Fig. 3. The two-player game model

Where in U_{xij}, $x=r$ or p represents *rbf* or *ploy*; i =1 or 2 represents the i^{th} player; j =1 or 2 represents the j^{th} strategy.

3.2 Game Theory Model Validation

In game theory, each player just care about his/her own rewards. Hence, we need to give some assumptions to validate our model.

Assumption 1: The principle of game theory requires that players only care about personal rewards. And the personal rewards summarize everything a player cares. It means that players recognize from others for self-satisfaction to maximize his/her personal satisfaction degree. For example, a player who is self-centred may care about both his/her own benefit and the other player's benefit.

Assumption 2: We assume that players are rational during the game. It means that each player expects to maximize his/her own benefit and indeed succeeds in selecting the optimal strategy.

Assumption 3: We assume that players are aware of their list of possible strategies. It means all players know everything about the structure of the game. Furthermore, it is reasonable to assume that each player knows the other' strategies and payoffs corresponding to choices of strategies.

According to the above assumptions, bilateral users will select their optimal strategies when we define the payoffs function. $S_A^i\left(S_A^i \in S_A^*\right)$ is a strategy chose from player *A* and $S_B^i\left(S_B^i \in S_B^*\right)$ is a strategy chose from player *B*, then there is a registration in the payoffs matrix corresponding to the pair of chose strategies $\left(S_A^i, S_B^i\right)$. In the following, we use symbol $U_A\left(S_A^i, S_B^i\right)$ to denote the payoff to player *A* under this pair of strategies and use symbol $U_B\left(S_A^i, S_B^i\right)$ to denote the payoff to player *B* under this pair of strategies. We show the payoffs matrix of our two-player game theory model in **Fig. 4**, where the payoffs function ΔI and ΔV are shown in Eq. (4).

$$\Delta I = 1-\sigma,\ \Delta V = 1-\frac{1}{d} \tag{4}$$

σ is the radius value of K_{rbf} in Eq. (2), and d is the polynomial degree of K_{ploy} in Eq. (3). Because the result of the game is finally that training samples do not be changed, the benefit of each player is 1 at the initial time. When K_{rbf} player chooses different strategies, the payoff functions are $\Delta I+1-1$, $\Delta I+1$, 1-1, 1+1 respectively as shown in **Fig. 4**. When K_{ploy} player chooses different strategies, the payoff functions of K_{ploy} are $\Delta V+1-1$, 1-1, $\Delta V+1$, 1+1 for each strategy pair, as shown in **Fig. 4**.

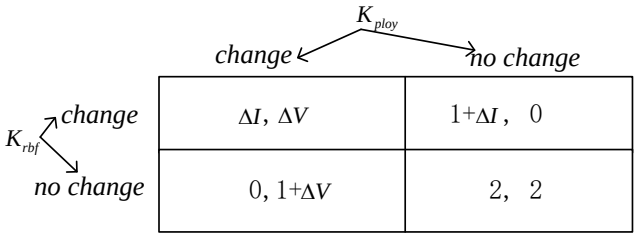

Fig. 4. The payoff matrix of the game theory

The so-called mixed strategy is that players randomly choose different strategies in a certain probability distribution under the condition of the given information [15]. In the game $G=\left\{n, S^*, U_n\right\}$, S^* is the set of strategies available to player i $\left(i=1,2,\cdots n\right)$. Player *i* randomly selects the strategy with certain probability distributions $P_i=\left\{p_{i1}, p_{i2}, \cdots, p_{ik}\right\}$ among the k optional strategies, and this is referred to as a mixed strategy, where $0 \le p_{ij} \le 1\ \left(j \in \left\{1,2,\cdots,k\right\}\right)$ and $p_{i1}+p_{i2}+\cdots p_{ik}=1$. There are two ways for solving the mixed strategy Nash Equilibrium: (i) The maximum payoff; (ii) Payoff equivalent method [25]. Only expectations of the pure strategy equal can we ignore the using of any pure strategy.

In other words, a player of mixed strategy makes the expectations of pure strategy of the other player equal. Therefore, we select the second way to calculate the mixed strategy Nash Equilibrium: first, when two players choose different strategies, the probability of the mixed strategy is shown in Eq. (5). When using K_{rbf}, the probability of the change of the kernel function for samples is p, while the probability is denoted as q when using K_{ploy}.

$$P_{rbf} = \begin{cases} p & change \\ 1-p & nochange \end{cases}, \ P_{ploy} = \begin{cases} q & change \\ 1-q & nochange \end{cases} \tag{5}$$

For K_{rbf} player, the benefit gained by selecting "change" strategy is $V_{rbf}(1,q) = \Delta I * q + (1+\Delta I) * (1-q)$; and the benefit obtained by selecting "no change" strategy is $V_{rbf}(0,q) = 0 + 2 * (1-q)$. Then, we use the principle of solving the mixed strategy Nash Equilibrium to calculate the q value and let $V_{rbf}(1,q) = V(0,q)$, and we can get : $q = 1 - \Delta I$. The situation is symmetric when we consider issues from the K_{ploy} player, and evaluate the payoffs from a probability p by K_{rbf}. It will have $p = 1 - \Delta V$. According to Eq. (4), we can get the final forms of p and q as shown in Eq. (6).

$$p = \frac{1}{d}, \ q = \sigma \tag{6}$$

In **Fig. 5**, the intersection point between the red and blue lines is the Nash Equilibrium, which is a more intuitive description of our game model. Now, we can determine the value of the two players and the ensemble kernel function based on the game theory as shown in Eq. (7).

$$K(x,z) = \frac{1}{d} K_{rbf} + \sigma K_{ploy}, \left(\frac{1}{d} + \sigma = 1 \right) \tag{7}$$

We refer to the expression in Eq. (7) as the game kernel function for NID, and denote it using symbol K_{GT}.

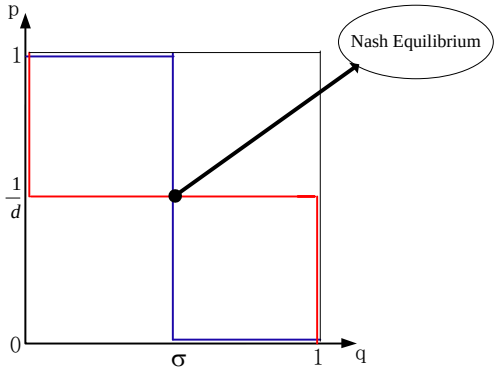

Fig. 5. Mixed Strategy Nash Equilibrium

3.3 System Framework

Kernel function is essentially an inner product, and it is a full symmetric function satisfying the Mercer condition. According to the Mercer condition [12], we assume that K_1 and K_2 are two kernels, and constant α satisfies: $\alpha \succ 0$, then the following functions are kernels:

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=K_1\left(\mathbf{x}_i,\mathbf{x}_j\right)+K_2\left(\mathbf{x}_i,\mathbf{x}_j\right) \tag{8}$$

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\alpha K_1\left(\mathbf{x}_i,\mathbf{x}_j\right) \tag{9}$$

Therefore, our game kernel function (7) satisfies the Mercer condition.

The generalization performance of SVM is dependent on the penalty parameter C and the parameters of the kernel function. Reference [26] reviews the existing parameter setting methods from the training dataset, but all of these methods require the prior knowledge, user expertise or experimental trial to obtain the suitable parameters of SVM. Therefore, we set the polynomial degree d =2, 3 or 4, and offset coefficient $c=0$ according to the practices. Then, according to Eq.(7), we can draw the characteristic of the K_{GT} as shown in **Fig. 6**.

The test point is 0.5. **Fig. 6** shows that the learning ability and generalization performance is satisfactory. The game kernel function not only highlights the local information of the nearby test point but also ensures the global information that is far away from the test point.

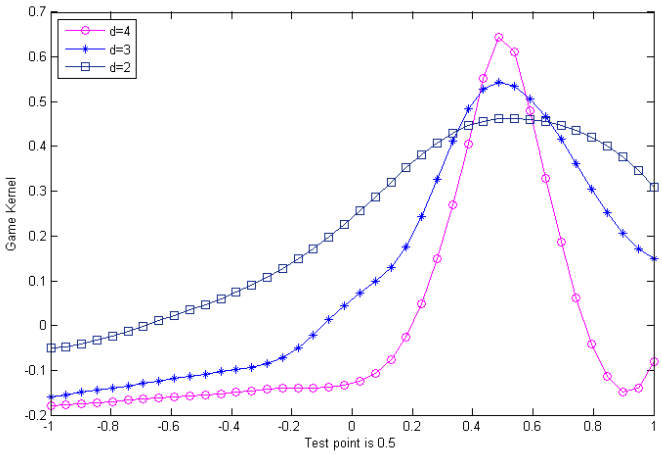

Fig. 6. Characteristic of game kernel function

Following the common network intrusion detection framework [27], we propose a novel GTNID-SVM based system framework of NID, which is shown in **Fig. 7**. It consists of three modules. Firstly, the data acquisition module collects the real-time network data. The data acquisition module is composed of two modules: capturing network data module and extracting connection information module. It can screen out the error data and the invalid data, and it can unify the data for the subsequent processing.

Secondly, the detection engine module detects the collected data. The module is composed of data preprocessing module, SVM training module, support vector library and SVM testing module. The idea is to select features of the collected data so as to reduce the dimension. Then, the SVM training module can be trained from the preprocessed data and the support vectors are stored in the support vector library. The new samples can be detected by the SVM testing module.

Finally, the detection response module responds to the intrusion and sends an intrusion alert. Meanwhile, the intrusion event is written to the system event log library.

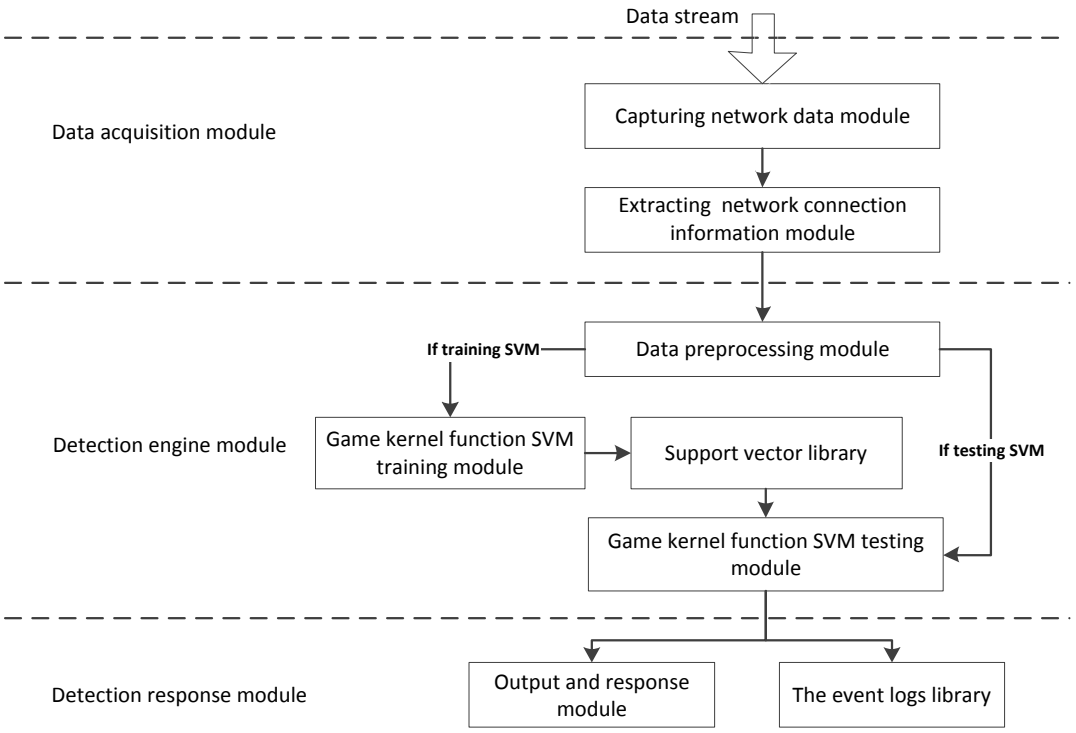

Fig. 7. The system framework of the proposed method

4. Experiments

In this section, we verify the performance of GTNID-SVM on the DARPA dataset by empirically comparing it with the Radial basis function kernel SVM. We conduct all experiments in the MATLAB software on a PC with Intel Core i7 processor, 8GB memory.

4.1 The used dataset

In order to verify the effectiveness of our method, we run experiments on the DARPA network intrusion detection dataset that is from MIT Lincoln laboratory [28]. The dataset is obtained from the simulation of the real network environment, in which the network traffic consists of four types of representative attacks, such as the Denial of Service attack (DoS), the Remote to Local attack (R2L), the User to Root attack (U2R) and the Probing and Scanning attack (Probe). Using the techniques of Data Mining and Knowledge Discovery, the university of California, Irvine (UC Irvine) has extracted 41 features of network connection information from the DARPA dataset for network intrusion detection research, which include 32 continuous-valued features and 9 discrete-valued features as shown in **Table 1** ("C" denotes continuous-valued feature while "D" denotes discrete-valued feature).

There are about 5,000,000 network connections in the DARPA dataset. In order to reduce the training time and ensure the representativeness of the training dataset, we randomly select 1/50 of the total dataset, 97969 network connections, as the training dataset according to the equal interval method. To ensure that the confidence coefficient is 0.95, and the error rate of the best training classification results is less than 0.002 and the total error rate is no more than

$\frac{\hat{\rho}}{1-\beta}$=0.0025, literature [29] has determined that the size of the training dataset for SVM must be larger than 74,894. Obviously, the size of our selected training set satisfies the condition in our experiments.

On the other hand, we select 300,000 to 500,000 data records from DARPA as testing datasets. There are three testing datasets containing about 1,300,000 network connection records, which are named the first testing dataset containing 489844 samples, the second testing dataset containing 489843 samples, and the third testing dataset containing 311029 samples, respectively. **Table 2** shows the characteristics of the training and testing datasets mentioned above.

The training dataset, the first and second testing datasets, contain the same type of attacks. In order to verify the effectiveness of our method on new types of attacks, the third testing dataset contains some new types of attacks besides the common types. **Fig. 8** shows the proportion of new types of attacks in the third testing data.

As shown in **Fig. 8**, the U2R attack type contains the most new attack type, which accounts for a considerable proportion. Therefore, if our algorithm can achieve more accurate classification results on the third testing dataset, we will conclude that it has better generalization performance.

Table 1. The 41 extracted features of network connection

No.	Feature Name	Feature Type	Category	No.	Feature Name	Feature Type	Category
1	duration	C	Basic Features	23	count	C	Flow Properties Features with time
2	protocol_type	D		24	serror_rate	C	
3	service	D		25	rerror_rate	C	
4	src_bytes	C		26	same_srv_rate	C	
5	dst_bytes	C		27	diff_srv_rate	C	
6	flag	D		28	srv_count	C	
7	land	D		29	srv_serror_rate	C	
8	wrong_fragment	C		30	srv_rerror_rate	C	
9	urgent	C		31	srv_diff_host_rate	C	
10	hot	C	Connection contend Features	32	dst_host_count	C	Flow Properties Features with hosts
11	num_failed_logins	C		33	dst_host_srv_count	C	
12	logged_in	D		34	dst_host_same_srv_rate	C	
13	num_compromised	C		35	dst_host_diff_srv_rate	C	
14	root_shell	D		36	dst_host_same_srv_port_rate	C	
15	su_attempted	D		37	dst_host_srv_diff_host_rate	C	
16	num_root	C		38	dst_host_serror_rate	C	
17	num_file_creations	C		39	dst_host_srv_serror_rate	C	
18	num_shells	C		40	dst_host_rerror_rate	C	
19	num_access_files	C		41	dst_host_srv_rerror_rate	C	
20	num_outbound_cmds	C					
21	is_hot_login	D					
22	is_ guest_login	D					

Table 2. Characteristics of the training and testing datasets

Datasets	Training datasets	First testing datasets	Second testing datasets	Third testing datasets
Number of overall samples	97969	489844	489843	311029
Number of normal samples	19458	97278	97280	60593
Number of abnormal samples	78511	392566	392563	250436
Number of DoS samples	77667	388338	388335	229853
Number of R2L samples	20	109	110	16189
Number of U2R samples	1	4	6	228
Number of Probe samples	823	4115	4112	4166

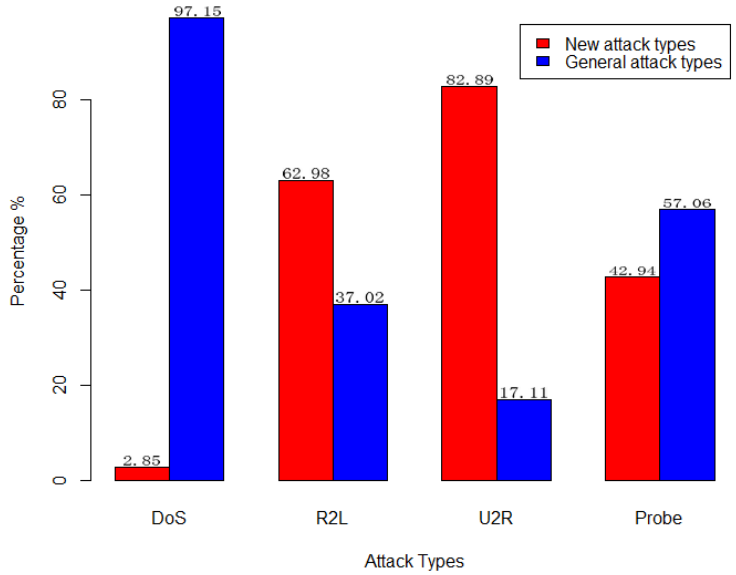

Fig. 8. The bar graph of the proportion of new types of attacks

4.2 Experimental setting

As the features of DARPA dataset are either continuous or discrete, it is a heterogeneous dataset. Researches induced the traditional SVM algorithms under inner product space, which were not suitable for the heterogeneous datasets. Therefore, we cannot directly adopt SVM in our experiments. To normalize the values of the continuous features, we preprocess these features in the heterogeneous dataset using the distance metric function HVDM proposed by D. Randall Wilson *et al*. [30].

We train our model ten times using the selected training dataset, after that, get a set of support vectors and put them into the support vector library. The SVM testing module is the core part of the proposed framework, which uses the support vector library to test the actual network connection records. The testing result, i.e., the output value, has two possible values: “1” or “-1”, where “1” represents the normal state (no intrusion occurs), while “-1” denotes the exceptional state (intrusion occurs). We list the GTNID-SVM algorithm in the following.

Algorithm: GTNID-SVM

1: Input the normalized data for training SVM.

2: Choose the game kernel function (Eq. (7)) as the kernel function of SVM.

3: Use the cross-validation and grid search strategy to choose the most appropriate values for parameter σ and the punish parameter C.

4: Use the obtained parameters to train SVM models on the available training samples, and get a set of support vectors.

5: Test the testing datasets by using the training results of step 4.

The game kernel function corresponds to the inner product of a space. The computational complexity of SVM classification is related to the natural dimension d of the input space, the number of the training samples N, and the number of the support vectors M. Generally, we have the following relation: $\frac{M}{N}$ is far less than 1, and it has been concluded that the computational complexity of SVM is $O\left(M^3 + NM^2 + dNM\right)$.

4.3 Experimental results and analysis

We obtain the parameters by using the cross-validation method. Generally, it is a reasonable and efficient way to generate grid search point according to the exponential growth. After a grid search point is established, it can be along the two different growth directions of parameters through a grid. By the parallel search, we show the partial data in **Table 3**. The highest accuracy of the search result is 92.5%. The values of corresponding parameters is C=256, σ=0.5.

Table 3. Partial data of parameters by cross-validation

Serial number	C	σ^2	Accuracy	Serial number	C	σ^2	Accuracy
1	128	1024	0.915555	9	128	4	0.91555
2	128	512	0.915555	10	128	2	0.921100
3	128	256	0.91555	11	128	1	0.872229
4	128	128	0.915555	12	128	0.5	0.847773
…	…	…	…	…	…	…	…

In our experiments, we use the *testing precision*, the *false positives rate*, and the *false negative rate* to show the performance of the compared methods quantitatively. The definitions of these evaluation measures are as follows.

Testing precision = Number of true positive samples / Number of overall samples.

False positive rate = Number of normal samples mis-classified as abnormal samples / Number of normal samples.

False negative rate = Number of abnormal samples mis-classified as normal sample / Number of abnormal samples.

We use the same experimental steps, but use a SVM with the radial basis function kernel to run the same experiments, and **Table 4** shows the experimental results. From **Table 4**, it can be observed that the GTNID-SVM achieves much higher *testing precision* while lower *false positive rate* and *false negative rate*, which indicates that the GTNID-SVM algorithm can

achieve better performance for the NID problem. The training dataset, the first testing dataset, and the second testing dataset are independent identically distributed. By using the estimation algorithm, under the conditions of the estimation of error rate $\hat{\rho} \approx 0.002$, the confidence value 0.95, and $\beta = 0.2$, the total error rate should be less than the upper bound 0.0025. Obviously, the error rates of the first testing dataset and the second testing dataset are respectively 0.0009 and 0.001043, which are much lower than the upper bound 0.0025. Experiments show that the expectation of the error rate can be guaranteed as long as the size of the training dataset follows the independent identically distribution. Because the third testing dataset contains a large number of new types of attacks, they are not the independent identically distribution with respect to the other testing datasets. Thus, the error rate 0.079944 is higher than the upper bound. However, in general, our method achieves a higher *testing precision* on the third testing dataset, indicating that our algorithm has better generalization performance and stronger practicability.

Table 4. Overall performances of the two methods

Items	GTNID-SVM				RBF kernel SVM			
	Training Dataset	First Testing Dataset	Second Testing Dataset	Third Testing Dataset	Training Dataset	First Testing Dataset	Second Testing Dataset	Third Testing Dataset
Number of true positive samples	97857	489403	489332	286475	97847	489205	489186	285955
Testing precision (%)	**99.8857**	**99.9100**	**99.8957**	**92.1056**	99.8755	99.8696	99.8659	91.9384
Number of normal samples	19458	97278	97280	60593	19458	97278	97280	60593
Number of false positive samples	9	54	68	926	10	57	70	930
False positive rate (%)	**0.0463**	**0.0555**	**0.0699**	**1.5282**	0.0514	0.0586	0.0720	1.5348
Number of abnormal samples	78511	392566	392563	250436	78511	392566	392563	250436
Number of false negative samples	109	579	585	24139	112	582	587	24144
False negative rate (%)	**0.1388**	**0.1475**	**0.1490**	**9.6388**	0.1427	0.1483	0.1495	9.6408

In order to verify further the effectiveness of our method on new types of attacks, we investigate the testing performance of new types of attacks (DoS, Probe) in the third testing dataset. In the third dataset, the number of overall samples of DoS and Probe are 229853 and 4166, respectively; and the number of new attack samples of DoS and Probe are 6555 and 1789, respectively. They are under the condition of the independent identically distribution. **Table 5** shows the performance of the two compared methods on the two new types of attacks.

Table 5. Performance of the two methods on testing new type attacks in the third testing dataset

	GTNID-SVM		RBF kernel SVM	
Attack type	DoS	Probe	DoS	Probe
Testing precision of new attacks (%)	**7.5374**	**86.1291**	6.8192	85.1314
Testing precision of general attacks (%)	**99.5058**	**92.7575**	99.4926	89.9453
The total detection rate (%)	**98.7466**	**89.9844**	96.8497	87.8781

We can observe from **Table 5** that the proposed method has higher detection rates in NID especially for those types of attacks that have been appeared. For the Probe type, its testing rate of new attack is higher. But for the DoS type, the testing rate of new attack is only 7.5374 percent, which is much lower. The reason for this phenomenon is that there is a large gap between the features of new type of attacks and those of the general types of attacks. Therefore, in the case of constantly changing network attacks, it deserves our further study that extracting the features of attack types more accurately.

SVM classifies samples by searching for an optimal classification hyper-plane. The game theory method in NID can help to find the optimal classification hyper-plane. From **Table 4** and **Table 5**, we can see that the proposed GTNID-SVM makes the error-classifying samples nearby the classification hyper-plane correct, and the correct-classifying samples has not changed. Therefore, by introducing the game kernel function constructed on the notion of game theory in SVM, the total testing precision and the testing precision of all kinds of attacks are higher than the state-of-the-art SVM algorithm in NID.

In addition, the literature [17] had compared the two approaches (Neural Networks and SVM) of intrusion detection. RBF-SVM in NID has a slightly higher accuracy of making the correct detection than neural networks in NID. Also, the neural network needs a very long training time, and required more parameters. Therefore, the proposed approach GTNID-SVM is superior to the neural network method in NID.

5. Conclusion

We comprehensively studied the problem of network intrusion detection in machine learning by simultaneously taking SVM, kernel functions and game theory into account. In NID, there are still several important problems that need solved: how to effectively identify network attacks that emerge continuously, and how to obtain the more accurate detection under the conditions that the number of available samples is small and the prior knowledge is hard to obtain. Therefore, existing learning methods often encounter the problems of over-fitting, under-fitting or local minima. Although kernel function can improve the learning ability of SVM, there are no theories to solve the problem of the construction of kernel functions perfectly in practice. In this paper, we selected a typical representative of the local kernel (concretely, the radial basis function) and a typical representative of the global kernel

(concretely, the polynomial function), and then integrates them effectively based on the notion of game theory. In this way, a mixed strategy based game model of two players is established and a game kernel method for NID is deduced, which provides a kernel structure that is flexible for NID. Based on the new kernel, a game kernel SVM algorithm (GTNID-SVM) is proposed to obtain a better optimal classification hyper-plane, and it is a new solution for the problem of network intrusion detection. To verify the effectiveness of the GTNID-SVM, we conducted a series of comparison experiments on the DARPA dataset. Experimental results indicated that the proposed method outperforms the state-of-the-art SVM method in NID.

References

[1] S. Rastegari, P. Hingston, and C.-P. Lam, "Evolving statistical rulesets for network intrusion detection," *Applied Soft Computing*, vol. 33, pp. 348-359, August, 2015. Article (CrossRef Link)

[2] M.-H. Chen, P.-C. Chang, and J.-L. Wu, "A population-based incremental learning approach with artificial immune system for network intrusion detection," *Engineering Applications of Artificial Intelligence*, vol. 51, pp.171-181, May, 2016. Article (CrossRef Link)

[3] J. Wei, R. Zhang, J. Liu, X. Niu, and Y. Yang, "Defense Strategy of Network Security based on Dynamic Classification," *Ksii Transactions on Internet & Information Systems*, vol. 9, pp. 5116-5134, December, 2015. Article (CrossRef Link)

[4] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, vol. 36, pp. 11994-12000, December, 2009. Article (CrossRef Link)

[5] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, pp. 1-8, June, 2016. Article (CrossRef Link)

[6] M. M. a. M. V. Valter Vasić, "Lightweight and adaptable solution for security agility," *KSII Transactions on Internet and Information Systems*, vol. 10, pp. 1212-1228, March, 2016. Article (CrossRef Link)

[7] M. Jo, L. Han, D. Kim, and H. P. In, "Selfish attacks and detection in cognitive radio Ad-Hoc networks," *IEEE Network*, vol. 27, pp. 46-50, June, 2013. Article (CrossRef Link)

[8] Z. Qi, Y. Tian, and Y. Shi, "Robust twin support vector machine for pattern classification," *Pattern Recognition*, vol. 46, pp. 305-316, January, 2013. Article (CrossRef Link)

[9] S. Maji, A. C. Berg, and J. Malik, "Efficient Classification for Additive Kernel SVMs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 66-77, January, 2013. Article (CrossRef Link)

[10] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates," *Journal of Machine Learning Research*, vol. 16, pp. 3299-3340, December, 2015. Article (CrossRef Link)

[11] S. F. Jianjun Li, Zhihui Wang, Haojie Li and Chin-Chen Chang, "An Optimized CLBP Descriptor Based on a Scalable Block Size for Texture Classification," *KSII Transactions on Internet and Information Systems*, vol. 11, pp. 288-301, January, 2017. Article (CrossRef Link)

[12] X. Zhang and M. H. Mahoor, "Task-dependent multi-task multiple kernel learning for facial action unit detection," *Pattern Recognition*, vol. 51, pp. 187-196, March, 2016. Article (CrossRef Link)

[13] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, *et al.*, "Struck: Structured Output Tracking with Kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 2096-2109, December, 2015. Article (CrossRef Link)

[14] 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, pp. 573-587, April, 2011. Article (CrossRef Link)

[15] Myerson RB. *Game Theory*. Harvard University Press Books, 2013. Article (CrossRef Link)
[16] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, pp. 42-57, January, 2013. Article (CrossRef Link)
[17] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proc. of Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, pp. 1702-1707, May 12-17, 2002. Article (CrossRef Link)
[18] M. Ektefa, S. Memar, F. Sidi, and L. S. Affendey, "Intrusion detection using data mining techniques," in *Proc. of 2010 International Conference on Information Retrieval & Knowledge Management (CAMP)*, pp. 200-203, March 17-18, 2010. Article (CrossRef Link)
[19] W. Hu, Y. Liao, and V. R. Vemuri, "Robust Support Vector Machines for Anomaly Detection in Computer Security," in *Proc. of International Conference on Machine Learning and Applications - Icmla 2003*, pp. 168-174, June 23-24, 2003. Article (CrossRef Link)
[20] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, *et al.*, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol. 38, pp. 306-313, January, 2011. Article (CrossRef Link)
[21] H. LI, X.-H. GUAN, X. ZAN, and C.-Z. HAN, "Network intrusion detection based on support vector machine," *Journal of Computer Research and Development*, vol. 6, pp. 799-807, June, 2003.Article (CrossRef Link)
[22] K. L. Li, H. K. Huang, S. F. Tian, Z. P. Liu, and Z. Q. Liu, "Fuzzy multi-class support vector machine and application in intrusion detection," *Chinese Journal of Computers*, vol. 28, pp. 274-280, February, 2005. Article (CrossRef Link)
[23] H. Chih-Wei and L. Chih-Jen, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415-425, August, 2002. Article (CrossRef Link)
[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1-27, April, 2011. Article (CrossRef Link)
[25] P. J. Reny, "Nash equilibrium in discontinuous games," *Economic Theory*, vol. 61, pp. 553-569, March, 2016. Article (CrossRef Link)
[26] J. Chorowski, J. Wang, and J. M. Zurada, "Review and performance comparison of SVM- and ELM-based classifiers," *Neurocomputing*, vol. 128, pp. 507-516, March, 2014. Article (CrossRef Link)
[27] J. M. Fossaceca, T. A. Mazzuchi, and S. Sarkani, "MARK-ELM: Application of a novel Multiple Kernel Learning framework for improving the robustness of Network Intrusion Detection," *Expert Systems with Applications*, vol. 42, pp. 4062-4080, May, 2015. Article (CrossRef Link)
[28] R. P. Lippmann and R. K. Cunningham, "Guide to Creating Stealthy Attacks for the 1999 DARPA Off-Line Intrusion Detection Evaluation," *Computer Networks*, vol. 34, pp. 579-595, January, 1999. Article (CrossRef Link)
[29] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank, "Online Adaboost-Based Parameterized Methods for Dynamic Distributed Network Intrusion Detection," *IEEE Transactions on Cybernetics*, vol. 44, pp. 66-82, January, 2014. Article (CrossRef Link)
[30] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *Journal of Artificial Intelligence Research*, vol. 6, pp. 1-34, June, 2000. Article (CrossRef Link)

Yufei Liu was born in Luoyang, China. He received the M.S. degree in computer applications technology from Northwest Minzu University, Lanzhou, China. He is currently pursuing the Ph.D. degree in Nanjing University of Aeronautics and Astronautics (NUAA) of China. His current interests include social network, machine learning, and network security.

Dechang Pi was born in 1971. He was a Ph.D. of Nanjing University of Aeronautics and Astronautics (NUAA) of China and now he is a professor and Ph.D. supervisor in NUAA. His research interests include data mining and big data analysis.