A Nature-inspired Multiple Kernel Extreme Learning Machine Model for Intrusion Detection

Yanping Shen[1, 2*], Kangfeng Zheng[1], Chunhua Wu[1] and Yixian Yang[1]
[1] School of Cyberspace Security, Beijing University of Posts and Telecommunications
Beijing 100876, China
[2] School of Information Engineering, Institute of Disaster Prevention
Beijing 101601, China
[e-mail: shenyanping@cidp.edu.cn]
*Corresponding author: Yanping Shen

Received August 6, 2019; revised June 25, 2019; accepted November 13, 2019; published February 29, 2020

Abstract

The application of machine learning (ML) in intrusion detection has attracted much attention with the rapid growth of information security threat. As an efficient multi-label classifier, kernel extreme learning machine (KELM) has been gradually used in intrusion detection system. However, the performance of KELM heavily relies on the kernel selection. In this paper, a novel multiple kernel extreme learning machine (MKELM) model combining the ReliefF with nature-inspired methods is proposed for intrusion detection. The MKELM is designed to estimate whether the attack is carried out and the ReliefF is used as a preprocessor of MKELM to select appropriate features. In addition, the nature-inspired methods whose fitness functions are defined based on the kernel alignment are employed to build the optimal composite kernel in the MKELM. The KDD99, NSL and Kyoto datasets are used to evaluate the performance of the model. The experimental results indicate that the optimal composite kernel function can be determined by using any heuristic optimization method, including PSO, GA, GWO, BA and DE. Since the filter-based feature selection method is combined with the multiple kernel learning approach independent of the classifier, the proposed model can have a good performance while saving a lot of training time.

Keywords: Meta-heuristics, Kernel extreme learning machine, Multiple kernel learning, Intrusion detection

1. Introduction

An endless stream of security vulnerabilities, automatically propagated network viruses, malicious programs that can be downloaded from anywhere on the network, especially the emergence of distributed and collaborative attacks, pose a great threat to network security. The traditional static network defense technology can not meet the needs of modern network security. In this case, intrusion detection system (IDS) becomes an irreplaceable component of the protection system. At present, the technologies used in intrusion detection systems include misuse detection and anomaly detection. Misuse detection can only identify known malicious behavior while anomaly detection system has the concept of normal activity. Since anomaly detection has the ability to discover unknown and new types of attacks, it has become research hotspot.

Machine learning, including artificial neural networks (ANNs) [1], fuzzy sets [2], support vector machines (SVMs) [3-4], and decision trees (DTs) [5] and so on, often forms the basis for the anomaly detection. However, anomaly detection still suffers from high false positive rates and low detection rates. The kernel method has been proven to be an effective method in many application scenarios and has been extensively studied in data mining and machine learning [6-7]. A single defined kernel function is usually adopted in the kernel methods, for example, the Gaussian kernel is a commonly used kernel function. However, since the samples may contain heterogeneous information or could be given in terms of different types of representations, the use of a single predefined kernel function is usually not enough. Therefore, there has been a lot of research on the methods of kernel combination, namely multiple kernel learning (MKL) that can be divided into five categories for determining the kernel combination: fixed rules, optimization methods, Bayesian approaches, boosting and heuristic approaches [7]. The combination of kernel functions includes linear combination, nonlinear combination and data-dependent combination [7].

In recent years, a new efficient machine learning algorithm known as extreme learning machine (ELM) [8] has drawn wide attention. To improve the classification accuracy and generalization performance, the kernel function has been applied to ELM and the kernel ELM (KELM) was proposed by Huang in 2012 [9]. Compared with the ELM and SVM, the KELM has a stable and better performance in classification accuracy and generalization, and can handle multi-class classification problems directly. This paper studies the multiple kernel learning method of kernel extreme learning machine for intrusion detection. In the kernel method, the original samples are mapped from the data space to the feature space through the kernel mapping and the corresponding operations are performed in the feature space. The ideal kernel mapping should make the similarity of the samples with the same label as large as possible, and the similarity of the samples with different labels as small as possible. An ideal kernel (*IK*) matrix can be defined first, and the multiple kernel model can be solved by evaluating the similarity between the actual kernel matrix and the ideal kernel matrix [7]. In recent years, many effective universal kernel evaluation methods have been proposed, including kernel alignment, kernel polarization, kernel class separability and so on [10]. The kernel alignment that can leverage the training data independently of the classifier is the most commonly used assessment method. It has been widely used because of its simplicity, efficiency and theoretical assurance [11].

The heuristic algorithms, including particle swarm optimization (PSO), genetic algorithm (GA), grey wolf optimization (GWO), bat algorithm (BA), differential evolution (DE) and so

on, are inspired by simulating or revealing some natural phenomena. They have good ability of black box optimization and do not require any prior knowledge [12]. The heuristic algorithms have no requirement for the derivable of the objective function and play a pivotal role in many applications. Since there is no perfect theoretical basis for constructing or selecting the kernel function, it is a good choice to use the heuristic algorithms to determine the kernel functions [13]. Based on the feedback, the heuristic algorithms change the input of the system iteratively and randomly until the end of the iteration. The process of changing the variables based on the output is defined by the mechanism of the algorithm. For example, the PSO saves the best solution so far by constantly changing the speed and location of the particles in a certain way.

It is important to note that the feature selection have a direct impact on the detection speed and accuracy. The original network data includes a large number of redundant or useless features that can cause the curse of dimensionality. Feature selection refers to eliminating the useless or redundant features on the basis of preserving the original information to improve detection performance. The feature selection methods can be divided into three types: filter, wrapper and embedding [14]. The filter method is independent of the subsequent learning algorithm. It usually uses the statistical performance of training data to evaluate the features, and this method is highly efficient. The wrapper one uses the training results of the subsequent learning algorithm to determine the feature subset, and it has a large computational complexity. The embedding method integrates the feature selection and the training of the subsequent algorithm as a whole [14].

In this paper, a new hybrid model which combines the ReliefF, kernel ELM (KELM) [9] and nature-inspired methods is proposed for intrusion detection. The ReliefF [15] is adopted to select features and KELM is used as the primary detection engine. Since the selection of the kernel function is critical to the performance of the KELM, a linear combination of multiple Gaussian kernels is used for the KELM. It is worth noting that the parameters of the Gaussian kernel that have great impact on the performance of KELM need to be determined first [9]. In other words, to build the MKELM is the process of combining the Gaussian kernels and determining the kernel parameters. In this paper, the nature-inspired methods are employed for the MKELM to optimize the kernel weights and the kernel parameters. A fitness function based on the kernel alignment that is independent of the detection engine is defined for the nature-inspired method. The numerical results reveal that the optimal composite kernel can be determined by using any heuristic optimization method, including PSO, GA, GWO, BA and DE. Since both the filter-based feature selection method and the multiple kernel learning approach independent of the classifier are used, the proposed model can have a comparable performance while saving a lot of training time.

The paper is organized as follows: Section 2 gives the related work. Section 3 introduces the background knowledge of the kernel ELM, ReliefF and the nature-inspired algorithms. The proposed multiple kernel learning model for intrusion detection is described in Section 4. Section 5 outlines the experimental results. See Section 6 for conclusions and possible extensions.

2. Related Work

In recent years, a few scholars have applied ELMs and its variants to intrusion detection systems (IDSs). Chi et al. [16] applied the basic ELM and KELM to the intrusion detection. The experimental results showed that the basic ELM is superior to SVM in training and testing speed, and the KELM achieves higher detection accuracy than SVM in multi-classification. Singh et al. [17] used the online sequential extreme learning machines (OS-ELMs) in intrusion

detection system. The Beta profiling technique was employed to reduce the size of the training dataset, and the feature selection was performed using an ensemble of Filtered, Correlation and Consistency techniques. Xiang et al. [18] applied ELMs to intrusion detection in a big data environment. Huang et al. [19] designed a parallel ensemble of online sequential extreme learning machine algorithm based on MapReduce for large-scale learning. Al-Yaseen et al. [20] proposed a hybrid model which combines the support vector machine and extreme learning machine. In this model, *K*-means was employed to generate a high-quality training dataset. Shen et al. [21] developed an ensemble pruning method using bat algorithm (BA) to prune the ensemble system. The ELM was chosen as the base classifier in the ensemble method.

There have been a variety of multiple kernel learning theories and methods. To obtain the kernel parameters, the composite kernel is usually combined with the SVMs. Then, the objective function is transformed into different optimization problems and solved by different optimization methods. Rakotomamonjy et al. [22] proposed the SimpleMKL model in which a linear combination of multiple kernels was used and the MKL problem was addressed through a weighted 2-norm regularization formulation with an additional constraint on the weights. The objective function was transformed into a convex and smooth function, and they used the gradient descent algorithm to determine the weight coefficient of the kernel function. Wang et al. [23] proposed a multiple-mapping kernel framework based on the SVM for hyperspectral image classification. Unlike using the linear combination, a nonlinear combination method of the multiple kernel learning was realized though repeated nonlinear mappings. Gu et al. [24] introduced a nonlinear MKL (NMKL) based on the SVM to learn a composite kernel. The optimal weight for each kernel matrix was obtained by a projection-based gradient descent algorithm. Hao et al. [6] applied the idea of boosting techniques to the multiple kernel learning, and a novel framework of multiple kernel boosting (MKBoost) was proposed.

Due to the advantages of ELMs and its variants, ELMs-based multiple kernel learning studies have emerged. Ma et al. [25] described a multi-scale Gaussian kernel extreme learning machine based on adaptive artificial bee colony (SABC). A linear multi-scale Gaussian function was used as the kernel function of KELM. They employed SABC to optimize kernel weights and kernel parameters in which the classification accuracy was defined as the evaluation criteria. Obviously, it is a multiple kernel learning method dependent of the classifier. Fossaceca et al. [26] proposed a novel multiple adaptive reduced kernel extreme learning machine (MARK-ELM) framework. The multiple classification reduced KELM that computed the kernel matrix over the randomly chosen subset of the data was chosen as its core classification algorithm. It was emphasized that the mentioned above multiple kernel boosting was combined with the reduced KELM.

The above MKL methods are all classifier-dependent algorithms, their computational complexity is relatively high. There are some MKL algorithms that are classifier-independent. It is similar to the relationship between filter-based and wrapper-based feature selection methods. Liu et al. [27] selected a data-dependent MKL approach and applied the sparse, non-sparse and radius-incorporated theory to the kernel ELM. This optimization can be considered as a different mathematical model for obtaining the kernel parameters and their combination. Wang et al. [28] noted that there are two types of multiple kernel learning methods including one-stage and two-stage methods. Actually, they correspond to the multiple learning methods dependent and independent of the classifier, respectively. A new kernel alignment that combined the global and local information of the basic kernels in the two-stage methods was developed. An alternative algorithm with proved convergence was proposed to determine the multiple kernel coefficients. Niazmardi et al. [13] proposed another

classifier-independent multiple kernel learning framework for multiple feature classification, and the PSO was adopted to determine the composite kernel. Inspired by Niazmardi's idea, this article uses the ReliefF to select features and studies the multiple kernel learning method of kernel extreme learning machine, and Niazmardi's method is still based on SVM.

3. Background

3.1 Kernel Extreme Learning Machine (KELM)

The extreme learning machine (ELM) is a kind of single hidden layer feedforward neural network (SLFN). The algorithm is simple in structure and has the same global approximation ability as well as faster learning speed compared with the traditional neural network.

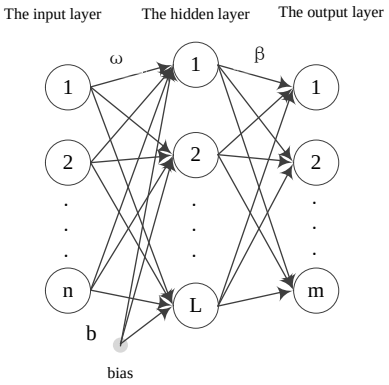


Fig. 1. Model of extreme learning machine

The model of extreme learning machine, including an input layer, a hidden layer and an output layer, is shown in **Fig. 1**. The model contains n input nodes, L hidden layer nodes and m output nodes. Suppose there are Q input instances $\{(x_i, t_i)\}$, where $i=\{1,\ldots,Q\}$, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in R^m$. x_i denotes the features of i-th sample and t_i represents the label of i-th sample. The actual output of the network is

$$f(x) = \sum_{i=1}^{L} g(\omega_i \cdot x_i + b_i)\beta_i \tag{1}$$

where $g(x)$ is the activation function, ω_i and b_i represent the input weight and the hidden layer bias of the i-th hidden neuron, β_i is the output weight connecting the i-th hidden neuron and the outputs. Note that the input weight ω_i and the hidden layer bias b_i are randomly generated. The above formula can be abbreviated as:

$$\mathbf{T} = \mathbf{H}\beta \tag{2}$$

In the case of $L \ll Q$, the output weight β can be determined by calculating the least squares error solution of the linear system [8]:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the hidden layer output matrix $\mathbf{H}$ [8]. Here singular value decomposition method [29] is used to calculate $\mathbf{H}^{\dagger}$:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}} \tag{4}$$

To make the performance of ELM more stable, parameter $\mathbf{I}/C$ was introduced in diagonal matrix [9]. The improved Moore-Penrose generalized inverse matrix can be expressed as:

$$\mathbf{H}^{\dagger} = (\mathbf{I}/C + \mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}} \tag{5}$$

where C is a positive constant. So the output weight can be written as:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} = (\mathbf{I}/C + \mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T} \tag{6}$$

The kernel learning method was introduced into the ELM to enhance the stability and generalization capability [9]. The kernel matrix Ω_{ELM}, constructed to replace $\mathbf{H}\mathbf{H}^{\mathrm{T}}$, can be defined as:

$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^{\mathrm{T}} : \Omega_{ELM_{i,j}} = h(x_i)h(x_j) = K(x_i, x_j) \tag{7}$$

where $h(x)$ represents the hidden layer mapping. The Gaussian kernel function is used, so $K(u,v) = \exp(-(\|u-v\|^2/\gamma))$, where γ is a kernel parameter. Then, the output of KELM can be expressed as follows, and the pseudocodes for the kernel ELM is shown in Algorithm 1.

$$\begin{aligned} f(x) &= h(x)\mathbf{H}^{\mathrm{T}}(\mathbf{I}/C + \mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{T} \\ &= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_Q) \end{bmatrix}^{\mathrm{T}} (\mathbf{I}/C + \Omega_{ELM})^{-1}\mathbf{T} \end{aligned} \tag{8}$$

Algorithm 1 The kernel ELM

Input: the parameter *C* and the kernel function, the training dataset and testing dataset with labels
Output: the confusion matrix.
Training:
1. Get the kernel matrix based on the training dataset;
2. Calculate the output weight;
3. Require the predict label of the training dataset;
Testing:
4. Calculate the new kernel matrix based on the testing dataset;
5. Get the predict label of the testing dataset;

3.2 ReliefF method

The Relief is a supervised and filter-based feature selection method that is suitable for the binary class problem. Kononenko [15] extended Relief and proposed the ReliefF that can deal with the multi-classification problem.

A sample x_i is randomly chosen from the training set each time, then the k nearest neighbor samples that have the same label and different label with x_i denoted by H and I are found out. The weight of each feature $w(A)$ is updated:

$$w(A) = w(A) - \sum_{j=1}^{k} diff(A, x_i, H_j)/zk + \sum_{C \neq class(x_i)} \left[\frac{P(C)}{1-P(class(x_i))} \sum_{j=1}^{k} diff(A, x_i, I_j(C))/(zk)\right] \tag{9}$$

where z indicates the sampling times, $I_j(C)$ denotes the j-th nearest neighbor sample with different labels C, $P(C)$ indicates the ratio of the number of samples labeled C to the total samples, $class(x_i)$ represents the label to which x_i belongs. The function $diff(A, I_i, I_j)$ represents the distance between the sample I_i and I_j based on feature A.

$$diff(A, I_i, I_j) = \begin{cases} |I_i(A) - I_j(A)| / (\max(A) - \min(A)), & A \text{ is continuous} \\ 0, & A \text{ is discrete and } I_i(A) = I_j(A) \\ 1, & A \text{ is discrete and } I_i(A) \neq I_j(A) \end{cases} \tag{10}$$

3.3 Nature-inspired methods

3.3.1 Particle Swarm Optimization (PSO)

The PSO [30] is one of the most classical swarm-based optimization algorithms. In the PSO algorithm, each particle represents a potential solution to the optimization problem. Particles constantly adjust their position by individual cognition and social cognition, and gradually approach the optimal solution. The particle *i* adjusts its speed and position according to the following formula:

$$V_i^t = \omega_{pso} \times V_i^{t-1} + c_1 \times rand() \times (pbest_i^t - X_i^t) + c_2 \times rand() \times (gbest_i^t - X_i^t) \quad (11)$$

$$X_i^t = X_i^{t-1} + V_i^{t-1} \quad (12)$$

where V_i^t represents the speed of the *i*-th particle in the *t*-th iteration, X_i^t indicates the position of the *i*-th particle in the *t*-th iteration. $pbest_i^t$ is the best position for the *i*-th particle until iteration *t*, and $gbest_i^t$ is the best position for all particles until iteration *t*. w_{pso} is the inertial weight; *rand()* is a random number evenly distributed over [0,1]; c_1 and c_2 are the acceleration factors.

3.3.2 Genetic Algorithm (GA)

The genetic algorithm that adopts the binary coding is one of the most widely used optimization algorithms [31]. It is evolved in the same strategies as those in nature including the operations of selection, crossover and mutation.

The appropriate individuals are chosen to enter the next evolution according to some certain rule. According to the crossover probability *Px*, some individuals are randomly selected to perform crossover operations at random positions and a new generation of individuals is obtained.

For some individuals in the population, a certain position of the individual is changed according to the variation probability *Pm* to generate a new individual. The mutation strategy can maintain population diversity. This paper will use the Sheffield Genetic algorithm toolbox for testing.

3.3.3 Grey Wolf Optimization (GWO)

Grey wolf optimization [32], a new swarm intelligence algorithm, is proposed based on the tight organization system of the wolves. The wolves are divided into four groups: α, β, δ and ω. The first three groups are in turn the three groups with the best fitness, and these three groups guide other wolves ω to search for the target. During the optimization process, the positions of α, β, δ and ω are constantly updated as follows:

$$D_\alpha = |C_1 \times X_\alpha^t - X^t|,\ D_\beta = |C_2 \times X_\beta^t - X^t|,\ D_\delta = |C_3 \times X_\delta^t - X^t| \quad (13)$$

where $D_\alpha, D_\beta, D_\delta$ indicates the distance between α, β, δ and ω, $X_\alpha^t, X_\beta^t, X_\delta^t$ represents the position of α, β *and* δ in the *t*-th iteration. C_1, C_2 and C_3 represent the random vectors, and X^t represents the current gray wolf position. Equations (14) define the forward step of the wolf ω toward α, β *and* δ, respectively:

$$X_1 = X_\alpha^t - A_1 \times D_\alpha,\ X_2 = X_\beta^t - A_2 \times D_\beta,\ X_3 = X_\delta^t - A_3 \times D_\delta \quad (14)$$

$$X^{t+1} = (X_1 + X_2 + X_3)/3 \quad (15)$$

$$A = 2 \times a \times r - a,\ C = 2 \times r,\ a = 2 - 2(t / t_{\max}) \quad (16)$$

Equations (14) to (15) define the final position of the wolf ω. *A* and *C* are coefficient factor as shown in Equation (16) and *a* represents the converging factor.

3.3.4 Bat Algorithm (BA)

The BA algorithm proposed by Yang [33] in 2010 is a random search algorithm that simulates bats using sonar to detect prey and avoid obstacles. The bionic principle of the BA algorithm is: a bat flies in speed V at position X with a fixed frequency f. It searches for prey with varying f and volume A. Given an D-dimensional space, X_i^t and V_i^t represents the position and velocity of the i-th bat at the t-th moment. X^* and f_i indicates the global best position and current frequency of the i-th bat. The update mechanisms of the position and velocity are

$$V_i^t = V_i^{t-1} + (X_i^t - X^*) \times f_i \tag{17}$$

$$X_i^t = X_i^{t-1} + V_i^t \tag{18}$$

The frequency f, pulse emission rate R and loudness A change as follows:

$$f_i^t = f_{min} + (f_{max} - f_{min}) \times rand() \tag{19}$$

$$R_i^t = R_i^0 \times [1 - exp(-\gamma t)] \tag{20}$$

$$A_i^t = r \times A_i^{t-1} \tag{21}$$

where r and γ are the specified coefficients.

3.3.5 Differential Evolution (DE)

The differential evolution (DE) [34] includes the strategies of variation, intersection and selection. The *RandToBest/2* mutation strategy is used and the variant individual is generated as follows:

$$M_i^t = X_i^t + F \times (X_{best}^t - X_i^t) + F \times (X_{r1}^t - X_{r2}^t) + F \times (X_{r3}^t - X_{r4}^t) \tag{22}$$

where F is the variation factor, r_1, r_2, r_3, and r_4 represents the arbitrary integer between 1 and N, respectively. $M_{i,}^t$ represents the variation of the i-th individual in the t-th iteration.

The mutated individual and the target individual are cross-operated to produce a testing object in the following manner:

$$U_{i,t}^j = \begin{cases} M_{i,t}^j & rand \le CR \text{ or } j=j0 \\ X_{i,t}^j & otherwise \end{cases} \tag{23}$$

where $U_{i,t}^j$ represents the testing individual, $j=1, 2,\ldots, D$, $j0$ is a random integer between 1 and D. The cross probability factor CR is a random number between 0 and 1.

If the fitness of the individual generated in the previous step is better than the fitness of the target individual, the target individual will be replaced by the testing individual.

4. The proposed natured-inspired multiple kernel learning model

The simplest and most common method for constructing the multiple kernel model is the linear combination of multiple basic kernel functions. Assuming there are M basic kernel functions, the linear multiple kernel (*LMK*) can be expressed as:

$$\sum_{i=1}^{M} w_s K_s(x_i, x_j), \quad s = 1, 2, ..., M \tag{24}$$

where w_s denotes the kernel weight between 0 and 1， $K_s(x_i, x_j)$ represents the s-th Gaussian kernel function used.

The kernel alignment (*KA*), a similarity measure between two kernels, is used to determine the actual kernel. The *KA* rule can be calculated as follows [10]:

$$KA(K_p, K_q) = \frac{< K_p, K_q >_F}{\sqrt{< K_p, K_p >_F < K_q, K_q >_F}} \tag{25}$$

where $<K_p, K_q>_F$ represents the Frobenius inner product of the matrix K_p and K_q, KA ranges between −1 and 1. The larger KA indicates the greater similarity between the two matrices.

The ideal kernel matrix IK can represent the training dataset well. There are several ways to define an ideal kernel matrix. For the binary classification problems, the ideal kernel matrix can be defined as yy^T, where y represents the labels of the samples. For the multiple classification problems, the ideal kernel matrix can be defined as [13]:

$$IK_{i,j} = \begin{cases} 1, & t_i = t_j \\ -1, & t_i \neq t_j \end{cases} \quad i, j \in [1,2,\ldots,Q] \tag{26}$$

where the value of $IK_{i,j}$ is 1 when the sample i and the sample j share the same category, -1 otherwise.

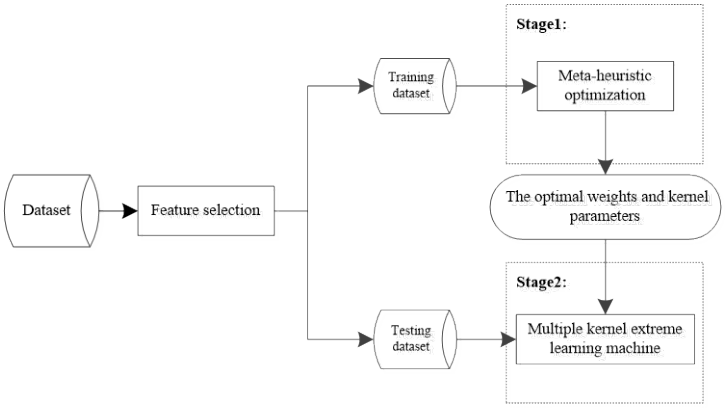


Fig. 2. Architecture of the nature-inspired multiple kernel extreme learning machine

The overall process of the proposed intrusion detection model is shown in **Fig. 2**. In the first stage, the ReliefF is used for feature selection to generate suitable training dataset and testing dataset. In the second stage, the meta-heuristics is used to learn an optimal composite kernel by making the multiple kernel matrix infinitely close to the ideal kernel matrix. The individual in the heuristic algorithm will go through a series of changes based on its position, fitness and other important metrics until the iteration has been completed. In the final stage, a multiple kernel extreme learning machine is set up to classify the testing dataset represented by the optimal feature subset. In the MKELM, the optimal composite kernel can map all the input samples from the input space to the feature space in which the samples can be classified easily. It is worth mentioning that the individual and the fitness function need to be defined first when using the meta-heuristics.

4.1 Individual Representation

An individual is comprised of two parts including the weight and kernel parameter. Suppose there are M kernels used, thus there are a total of $2 \times M$ parameters need to be set. Therefore, the individual is in $2 \times M$-dimensional space. The structure of an individual is shown in **Table 1**.

Table 1. Individual representation

Weight	Kernel parameter
$w_1,\ldots,w_M$	$\gamma_1,\ldots,\gamma_M$

4.2 Fitness Function Definition

The individuals of the meta-heuristics evaluate themselves based on their fitness. Therefore, the definition of the fitness function is critical to the performance of the meta-heuristics. In this paper, a linear combination of multiple Gaussian kernels, shown in equation (24), is used for the KELM. However, how to determine the kernel weights and the kernel parameters?

There is a way to define an ideal kernel matrix as expressed in formula (26). The kernel alignment (*KA*) is adopted as the evaluation measures to assess the quality of the linear multiple kernel (*LMK*). The *LMK* used in this paper can be obtained by evaluating the similarity between the actual kernel function and the ideal kernel function. The larger similarity measure (i.e. *KA*) indicates the greater similarity between the two matrices. In other words, an ideal kernel function will have a higher value of *KA*. Thus, the fitness function *F* can be defined as

$$Maximize\ F = KA(LMK, IK) = \frac{< LMK, IK >_F}{\sqrt{< LMK, LMK >_F < IK, IK >_F}} \tag{27}$$

where *LMK* denotes the linear multiple kernel matrix and *IK* is the ideal kernel matrix. A larger fitness *F* results in a better kernel matrix that is closer to the ideal kernel matrix.

4.3 The Nature-inspired Multiple Kernel Extreme Learning Machine

The nature-inspired methods are all iterative algorithms. Each algorithm is iterated according to its own update mechanism. After the iteration, the algorithms stop at the appropriate fitness value and the corresponding position of the individual denotes the best kernel parameters and weights that can compose the optimal kernel. The pseudocodes for the proposed model is shown in Algorithm 2.

Algorithm 2 The nature-inspired multiple kernel learning method

Input: the important parameters used for the meta-heuristics, parameter *C*, training and testing dataset with labels
Output: the best weights and kernel parameters, training time and the confusion matrix.
1. The ReliefF is used to select appropriated features;
2. Get the ideal kernel matrix based on the training sample labels;
3. **for each** individual **do**
4. Compute the initial fitness of all individuals;
5. **end for**
6. **for each** iteration **do**
7. **for each** individual **do**
8. Update the positions and other important metrics of the individual;
9. Update the fitness of the individual;
10. Find the best individual based on their fitness values;
11. **end for**
12. **end for**
13. Obtain the kernel parameters and weights according to the best individual;
14. Testing:
15. [testing accuracy, confuse matrix] = predict (test label, test scale, the optimal composite kernel);

4.4. Complexity analysis

The main computational cost of the nature-inspired methods will be in the evaluation of the fitness function. Suppose there are $Q/2$ pairs of training samples, the complexity of the fitness function independent of the classifier is $O((Q/2)^2)$ [10]. So the nature-inspired multiple kernel learning method requires $O((Q/2)^2) \times sizepop \times Iter$ to perform the kernel parameters and weights optimization. In contrast, the complexity of kernel ELM is $O(Q^3)$ [35]. That means the complexity of the kernel learning method dependent of the classifier based on the meta-heuristics is $O(Q^3) \times sizepop \times Iter$, where *sizepop* represents the swarm size and *Iter* represents the maximum number of iterations. It can be seen that compared with the multiple kernel learning method relying on the classifier, the multiple kernel learning method indepedent of the classifier can reduce the complexity of the algorithm from $O(Q^3)$ to $O(Q^2)$.

5. Experiments

5.1 Evaluation

In this paper, four indicators of accuracy (*Acc*), detection rate (*DR*), false positive rate (*FPR*) and *F1* measure are used to evaluate the performance of different methods. **Table 2** shows the confusion matrix for the correct and incorrect number of instances detected.

Table 2. Confusion matrix in intrusion detection

	Judged as attack	Judged as normal
Attack	(*TP*)	(*FN*)
Normal	(*FP*)	(*TN*)

The model evaluation parameters are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{28}$$

$$DR = \frac{TP}{TP + FN} \tag{29}$$

$$FPR = \frac{FP}{FP + TN} \tag{30}$$

$$Precision = \frac{TP}{TP + FP} \tag{31}$$

$$Recall = \frac{TP}{TP + FN} \tag{32}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{33}$$

where *TP* represents the number of correctly identified attack samples, *FP* indicates the number of normal samples that are judged to be attacks, *FN* represents the number of attack samples that are judged to be normal, and *TN* indicates the number of normal samples that are correctly identified.

5.2 Experimental Descriptions

Three public datasets are used to evaluate the performance of the proposed method, including the KDD99 [36] dataset, the NSL [37] and the Kyoto dataset [38]. Although the KDD99 has some drawbacks, it is still the most used dataset for evaluating intrusion detection models [39]. All attacks fall into four categories, Probing, Denial of Service (DoS), User to Root (U2R), and Remote to Local (R2L). Each record has 41 features, including the basic characteristics of TCP connections and network traffic statistics. The NSL generated by deleting duplicate records from the KDD99 is also used. It has the same features with the KDD99 and has higher requirements for intrusion detection algorithms. As space is limited, their features are not listed here. The Kyoto dataset, collected from several real honeypots deployed in Kyoto university, has been built on over 2.5 years. The attack consists of known attack and unknown attack. There are 24 features in total and 18 features are selected in this paper shown in **Table 3** [21].

The dataset we use will be split into two equal parts. One half of the dataset represents the training samples and the other half represents the testing samples. The symbolic data is converted to numeric values and all data will be discretized before using the method. The described experiments were carried out in MATLAB R2016b environment, on a 3.30 GHz processor with 16G of memory.

Table 3. Features used in the Kyoto dataset

Feature representation	Feature name	Feature representation	Feature name
$\mathbf{F_1}$	duration	$\mathbf{F_{10}}$	dst_host_srv_count
$\mathbf{F_2}$	service	$\mathbf{F_{11}}$	dst_host_same_src_port_rate
$\mathbf{F_3}$	src_bytes	$\mathbf{F_{12}}$	dst_host_serror_rate
$\mathbf{F_4}$	dst_bytes	$\mathbf{F_{13}}$	dst_host_srv_serror_rate
$\mathbf{F_5}$	count	$\mathbf{F_{14}}$	flag
$\mathbf{F_6}$	same_srv_rate	$\mathbf{F_{15}}$	IDS_detection
$\mathbf{F_7}$	serror_rate	$\mathbf{F_{16}}$	Malware_detection
$\mathbf{F_8}$	srv_serror_rate	$\mathbf{F_{17}}$	Ashula_detection
$\mathbf{F_9}$	dst_host_count	$\mathbf{F_{18}}$	duration

5.3 Experimental Results

The parameter ranges of the optimal composite kernel will refer to that of the single kernel ELM. In the single kernel ELM, there are two parameters to be determined, including C and the kernel parameter γ. Since the kernel function is determined by the method independent of the classifier in this paper, the value of C must be determined first. Previous studies used the Grid-search method to determine the two mentioned above parameters of the single kernel ELM: [8, 0.125]. Therefore, it is reasonable to set C to 8 for the multiple kernel extreme learning machine. And the searching range for γ is set as: $\gamma \in [2^{-10}, 2^{-2}, ..., 2^{3}]$. The value of M that represents the number of basic kernel function used is set to be 4. The k of ReliefF is set to be 10, we will select 15 features for experiment.

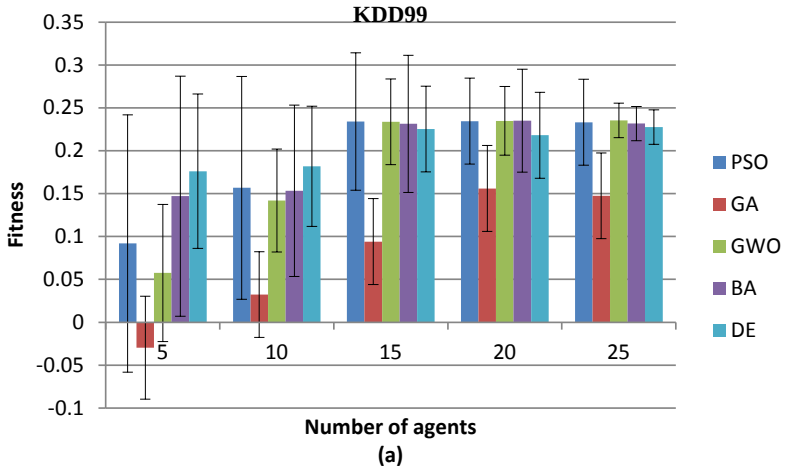
The parameters of the meta-heuristics are chosen empirically, the important parameters used for the meta-heuristics are presented in **Table 4**. As shown in Section 4.4, the algorithm

complexity of the proposed method depends not only on the number of training samples but also on the number of iterations and agents in the heuristic algorithm. To obtain reasonable values for the above two parameters, it is necessary to explore the effect on the fitness value when the "number of iterations × population size" takes different values [40]. Moreover, to evaluate the search and convergence capabilities of different heuristic algorithms, the heuristic algorithms under the following conditions are implemented, i.e. the number of iterations is 10 and 20, and the population size increases from 5 to 25 with steps of 5.

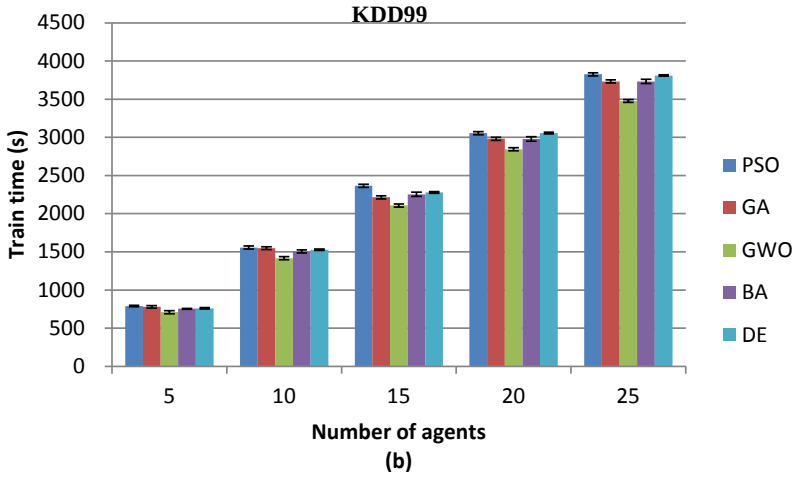
Table 4. Parameters for nature-inspired approaches

Methods	Parameters
Particle swarm optimization	c_1=c_2=2, w_{pso}=0.72
Bat algorithm	r= γ =0.9, R^0=0.5, A=0.25
Grey wolf optimization	The converging factor a decreases linearly from 2 to 0 with the number of iterations.
Genetic algorithm	$GGAP$=0.95, px=0.7, pm=0.01
Differential evolution	F is randomly selected from 0.2 to 0.8, CR=0.2

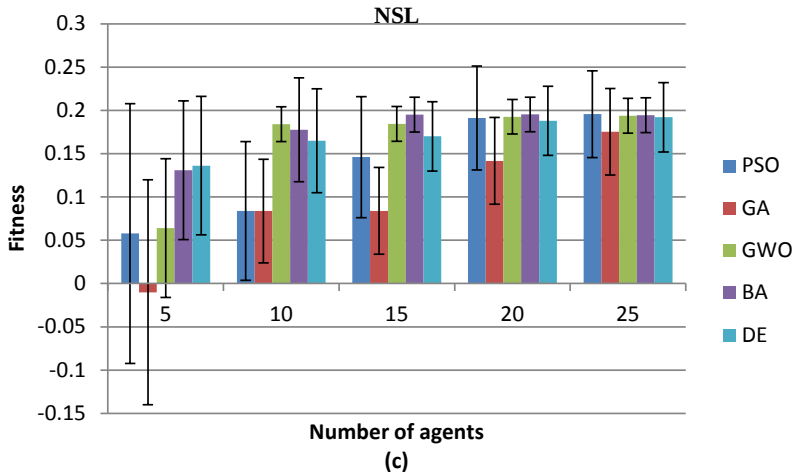
Fig. 3 and **4** give the experimental results of the proposed method when the number of iterations of the heuristic algorithms is 10 and 20 respectively. The heuristic methods, including PSO, GA, GWO, BA and DE, are used to determine the best composite kernel in KELM. Higher iterations and population size may slow down the efficiency of the proposed algorithm. It is worth noting that the experimental results, including the fitness-*KA* and the training time, are the average of each algorithm running five times.



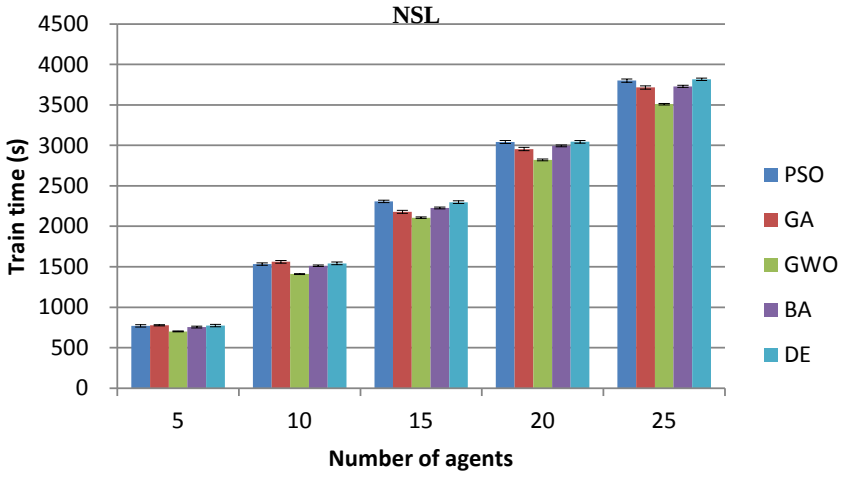
(a)


KDD99
Train time (s)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
5
10
15
20
25
Number of agents
PSO
GA
GWO
BA
DE

(b)


NSL
Fitness
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
5
10
15
20
25
Number of agents
PSO
GA
GWO
BA
DE

(c)


NSL
Train time (s)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
5
10
15
20
25
Number of agents
PSO
GA
GWO
BA
DE

(d)

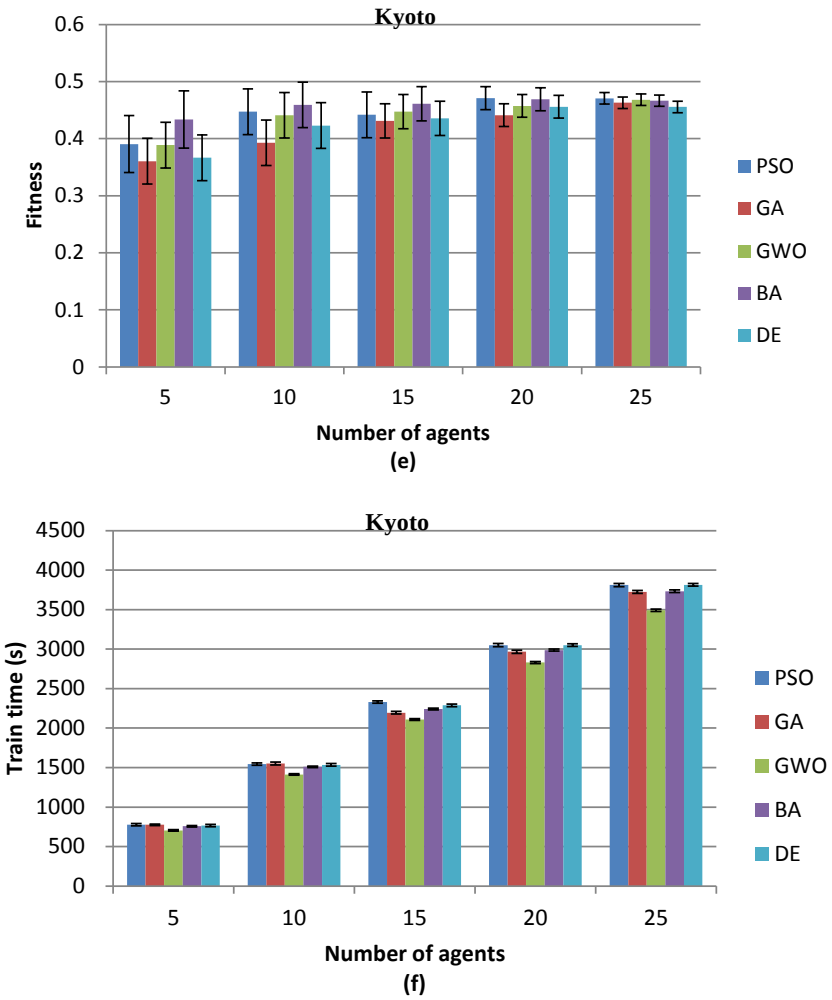
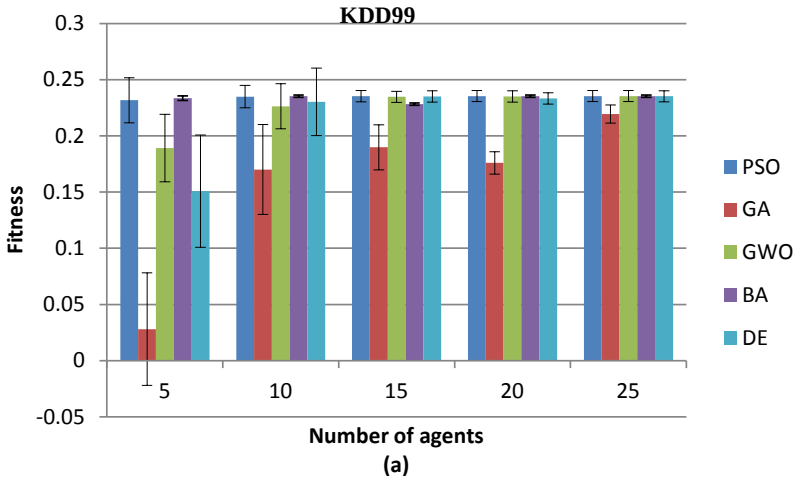
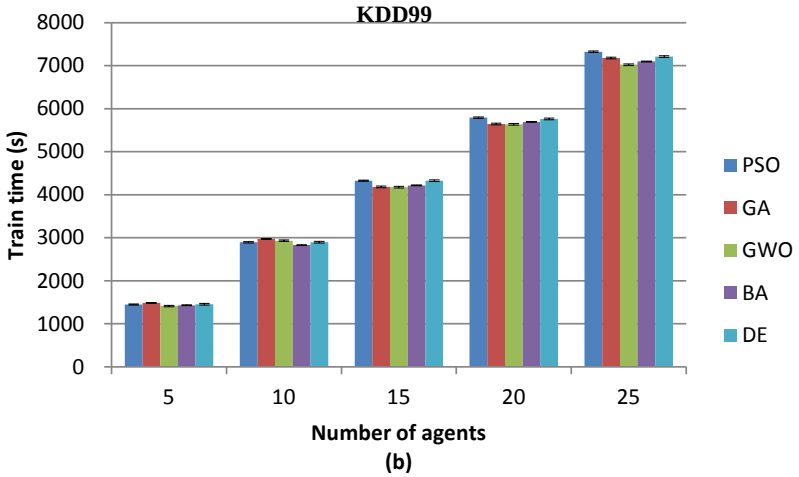
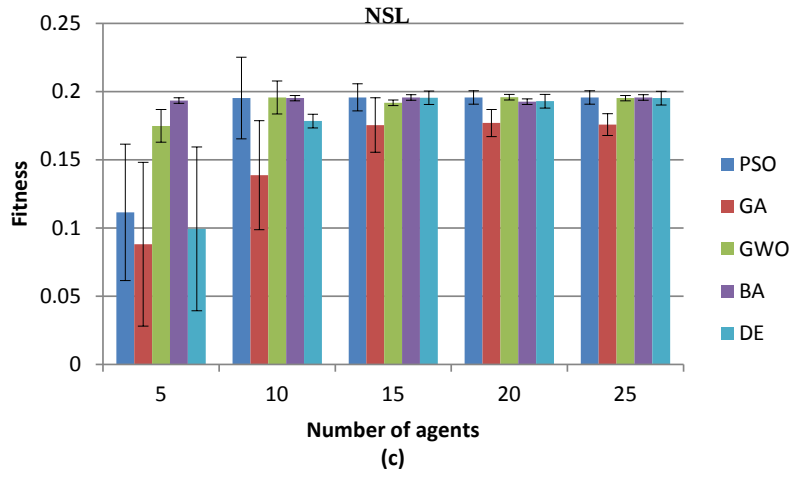


Fig. 3. The fitness and running time of the heuristic algorithms when the iteration is 10

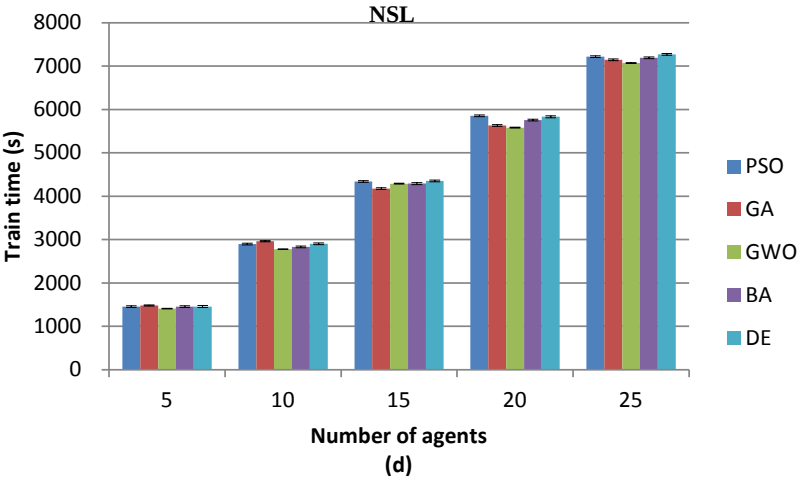



KDD99
Train time (s)
8000
7000
6000
5000
4000
3000
2000
1000
0
5
10
15
20
25
Number of agents
PSO
GA
GWO
BA
DE

(b)


NSL
Fitness
0.25
0.2
0.15
0.1
0.05
0
5
10
15
20
25
Number of agents
PSO
GA
GWO
BA
DE

(c)


NSL
Train time (s)
8000
7000
6000
5000
4000
3000
2000
1000
0
5
10
15
20
25
Number of agents
PSO
GA
GWO
BA
DE

(d)

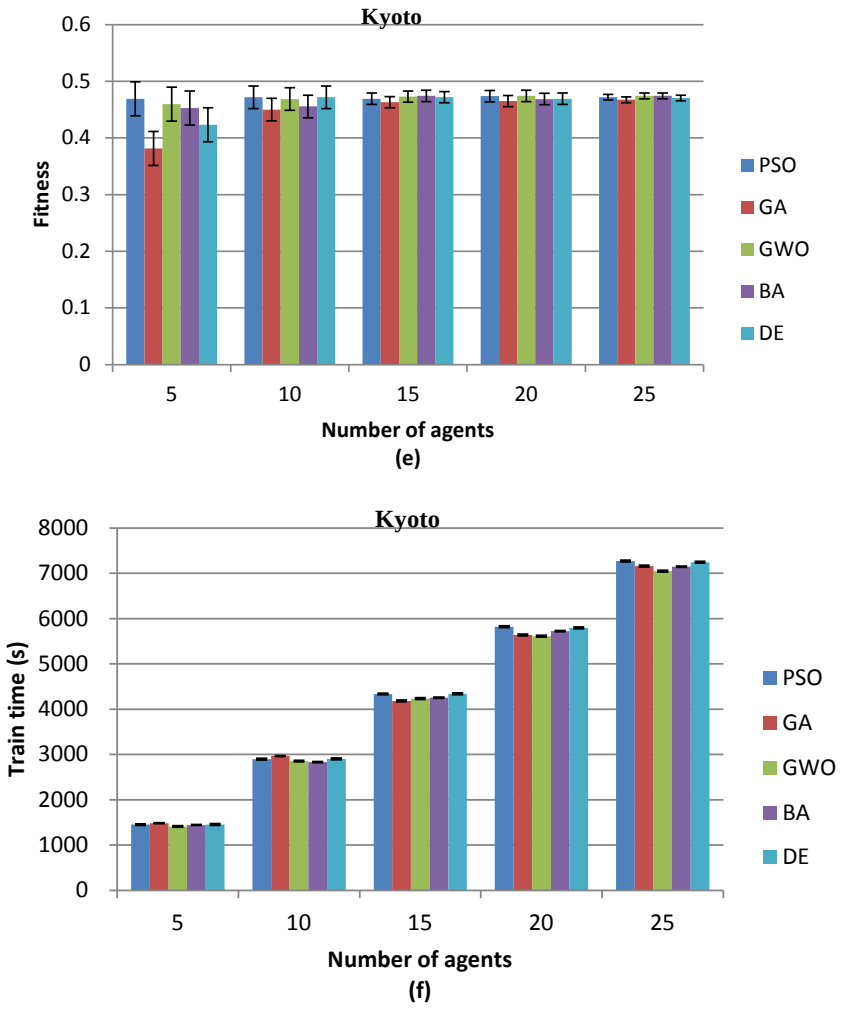


Fig. 4. The fitness and running time of the heuristic algorithms when the iteration is 20

As shown in Section 4.2, a larger fitness value indicates that a better kernel function is obtained. From **Fig. 3** and **4**, as the number of iterations and size of the population increase, the algorithm tends to converge, and the fitness value and training time gradually increase. In the slowest case, with 20 iterations and 25 agents, the fitness value based on the KDD99 dataset is close to 0.25 and the fitness based on the NSL does not exceed 0.2, which proves that the NSL has higher requirements on the intrusion detection model.

The PSO is widely used due to its simplicity of operation, however, it is almost the slowest method (see **Fig. 3**(b), (d) and (f), and **Fig. 4**(b), (d) and (f)). It is worth mentioning that GA does not perform well in terms of the training efficiency and search ability, especially for a small number of agents and iterations. It can be seen that the value of the fitness is negative on the KDD99 and NSL datasets when the agents is 5 and the number of iteration is 10. Although the PSO algorithm is almost the slowest technology in a different number of agents and iterations, it has a significant performance advantage.

It can be seen from the above figures that the GWO, as a relatively new swarm intelligence algorithm who has a performance comparable to that of PSO, is almost the fastest technology on the three public datasets. Like PSO, BA can achieve a competitive fitness value in most

cases and is more efficient than PSO. Both DE and GA belong to the evolutionary algorithms, and the performance of DE is better than that of GA on the three datasets. Anyway, we can draw a conclusion that as long as we choose the appropriate combination of the iteration number and the agents, the optimal kernel function can be determined by using any heuristic optimization method.

The PSO is used for further experiments due to its effectiveness and wide application. **Table 5, 6** and **7** shows the performance of the single kernel independent method, the multiple kernel independent method, i.e. the proposed method and the multiple kernel dependent method on the three datasets, including the testing accuracy (*Acc*), *DR*, *FPR*, *F1* and training time. The feature selection is performed in all methods. The number of iterations and the population size of the PSO applied in the dependent method are the same as that applied in the independent methods. From **Table 5, 6** and **7**, it is observed that the proposed multiple kernel independent method can improve the performance of the single kernel ELM. However, it takes longer time to train. As shown in **Table 5, 6** and **7**, our proposed method has comparable advantage compared with the multiple kernel dependent method. Note that its training time, shortened from 11496s to 7323s on the KDD99 dataset, from 10601s to 7216s on the NSL dataset and from 9519s to 7269s on the Kyoto dataset, is reduced by about 40% compared with the dependent method. It is concluded that the proposed model can have a comparable performance while saving a lot of training time.

Table 5. Comparison of the kernel learning methods independent and dependent of the classifier on the KDD99

Techniques	*Acc* (%)	*DR* (%)	*FPR* (%)	*F1* (%)	Training Time (s)
Single kernel independent method	98.95	98.72	0.76	99.05	2464
Multiple kernel independent method	99.24	99.26	0.78	99.32	7323
Multiple kernel dependent method	99.17	98.99	0.60	99.26	11496

Table 6. Comparison of the kernel learning methods independent and dependent of the classifier on the NSL

Techniques	*Acc* (%)	*DR* (%)	*FPR* (%)	*F1* (%)	Training Time (s)
Single kernel independent method	97.61	97.32	2.04	97.83	2477
Multiple kernel independent method	98.85	98.70	0.96	98.97	7216
Multiple kernel dependent method	98.72	98.44	0.92	98.85	10601

Table 7. Comparison of the kernel learning methods independent and dependent of the classifier on the Kyoto

Techniques	*Acc* (%)	*DR* (%)	*FPR* (%)	*F1* (%)	Training Time (s)
Single kernel independent method	99.36	99.23	0.51	99.36	2208
Multiple kernel independent method	99.60	99.60	0.40	99.60	7269
Multiple kernel dependent method	99.66	99.86	0.55	99.66	9519

Finally, the proposed method is compared with other techniques. **Table 8** shows the experimental results including *DR*, *FPR*, *F1* and detection rate for each category. It can be seen that very few techniques have a good detection rate for each category. Since the U2R category has the smallest sample size, the detection result is the lowest. The SVM has the best

performance in each category, but its false positive rate is the highest. It is observed that the proposed method has comparable advantage compared with other existing methods.

Table 8. Performance comparison of different methods on the KDD99 (%)

Methods	DR	FPR	F1	Normal	Probe	DoS	U2R	R2L
The proposed method	99.26	0.78	99.32	99.22	98.10	98.38	73.08	95.25
ELM (sigmoid) [16]	99.23	0.86	99.28	99.14	98.40	99.82	42.31	95.50
SVM [41]	99.51	1.99	98.97	98.01	99.39	99.85	86.67	94.40
Random forest [42]	99.75	1.09	99.45	98.91	99.75	99.90	77.78	97.20
KNN [43]	99.09	1.16	99.09	98.84	98.35	97.02	57.69	93.75

6. Conclusions

The performance of the kernel function varies greatly in different applications, and there is no perfect theoretical basis for the construction or selection of the kernel function. In recent years, there has been a lot of research on multiple kernel learning methods. A nature-inspired multiple kernel extreme learning machine model for intrusion detection has been proposed in this paper. In the model, MKELM is the core engine because of its fast efficiency and good performance. Additionally, MKELM can perform the multi-category classification directly, without any modification. For the multiple kernel learning, the linear combination of Gaussian kernel functions is used and the nature-inspired methods are employed that aims at optimizing the kernel weights and the kernel parameters. Three public datasets are employed to confirm the performance of the model. The experimental results indicate that any heuristic optimization method, including PSO, GA, GWO, BA and DE, can be adopted to determine the optimal composite kernel function. It has to be noted that the GA is less effective in exploring the search space than other optimization techniques, and the GWO has better result in search performance and efficiency. This model combines the filter-based feature selection method with the multiple kernel learning approach independent of the classifier. That means this method does not require multiple runs of the classifier and it is relatively a low computational demanding strategy. Future work will include the research of other multiple kernel learning methods and how to use the heuristic algorithms to optimize them.

Acknowledgements

This research was supported by the National Key Research and Development Program of China (2017YFB0802803), the National Natural Science Foundation of China (61602052), the Science and Technology Research and Development Project of Langfang (2017011027) and the Fundamental Research Funds for the Central Universities (2017011027).

References

[1] L. M. Ibrahim, D. T. Basheer, and M. S. Mahmod, “A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network,” *Journal of Engineering Science*, vol. 12, no. 3, pp. 11-16, Mar, 2013.

[2] S. Elhag, A. Fernndez, A. Bawakid, S. Alshomrani, and F. Herrera, “On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems,” *Expert Systems with Application*, vol. 42, no. 1, pp. 193-202, Jan, 2015.
Article (CrossRef Link)

[3] W. M. Hu, J. Gao, Y. Wang, O. Wu and M. Stephen, "Online Adaboost-Based parameterized methods for dynamic distributed network intrusion detection," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 66-82, 2014. Article (CrossRef Link)
[4] W. Y. Feng, Q. Zhang, and G. Hu, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127-140, Jul, 2014. Article (CrossRef Link)
[5] S. W. Lin, K. C. Ying, C. Y. Lee, and Z. J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Applied Soft Computing*, vol. 12, no. 10, pp. 3285-3290, 2012. Article (CrossRef Link)
[6] X. Hao, S. C. H Hoi, "MKBoost: A framework of multiple kernel boosting," *IEEE Transactions on Knowledge & Data Engineering*, vol.25, no.7, pp.1574-1586, 2013. Article (CrossRef Link)
[7] M. Gönen, E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol.12, pp. 2211-2268, 2011.
[8] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, Dec, 2006. Article (CrossRef Link)
[9] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513-529, Feb, 2012. Article (CrossRef Link)
[10] T. Wang, D. Zhao, S. Tian, "An overview of kernel alignment and its applications," *Artificial Intelligence Review*, vol.43, no.2, pp.179-192, 2015. Article (CrossRef Link)
[11] S. Zhong, D. Chen, Q. Xu, et al., "Optimizing the Gaussian kernel function with the formulated kernel target alignment criterion for two-class pattern classification," *Pattern Recognition*, vol.46, no.7, pp. 2045-2054, 2013. Article (CrossRef Link)
[12] S. Mirjalili, "SCA: A Sine Cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol.96, pp.120-133, 2016. Article (CrossRef Link)
[13] S. Niazmardi, A. Safari, S. Homayouni, "A novel multiple kernel learning framework for multiple feature classification," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, vol.10, no.8, pp.3734-3743, 2017. Article (CrossRef Link)
[14] G. Chandrashekar, F Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol.40, pp. 16-28, 2014. Article (CrossRef Link)
[15] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. of European Conference on Machine Learning on Machine Learning*, pp. 171-182, 1994. Article (CrossRef Link)
[16] C. Chi, W. P. Tay, and G. B. Huang, "Extreme learning machines for intrusion detection," in *Proc. of WCCI 2012 IEEE World Congress on Computational Intelligence*, pp. 1-8, Oct, 2012. Article (CrossRef Link)
[17] R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8609-8624, Dec, 2015. Article (CrossRef Link)
[18] J. Xiang, M. Westerlund, "Using extreme learning machine for intrusion detection in a big data environment," in *Proc. of The Workshop on Artificial Intelligent & Security Workshop*, pp. 73-82, 2014. Article (CrossRef Link)
[19] S. Huang, B. Wang, J. Qiu, J. Yao, G. Wang, and G. Yu, "Parallel ensemble of online sequential extreme learning machine based on MapReduce," *Neurocomputing*, vol.174, pp. 352-367, Jan, 2016. Article (CrossRef Link)
[20] W. L. Al-Yaseen, Z. A. Othman, M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol.67, pp.296-303, 2017. Article (CrossRef Link)
[21] Y. Shen, K. Zheng, C. Wu, et al., "An ensemble method based on selection using bat algorithm for intrusion detection," *The Computer Journal*, vol.61, no. 4, pp. 526-538, 2018. Article (CrossRef Link)
[22] A. Rakotomamonjy, F. R. Bach, S. Canu, Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol.9, no.11, pp. 2491-2521, 2008.

[23] L. Wang, S. Hao, Q. Wang, et al., "A multiple-mapping kernel for hyperspectral image classification," *IEEE Geoscience & Remote Sensing Letters*, vol. 12, no.5, pp. 978-982, 2015. Article (CrossRef Link)
[24] Y. Gu, T. Liu, X. Jia, et al., "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 54, no. 6, pp. 3235-3247, 2016. Article (CrossRef Link)
[25] C. Ma, J. Ouyang, H. L. Chen, and J. C. Ji, "A novel kernel extreme learning machine algorithm based on self-adaptive artificial bee colony optimization strategy," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1342-1357, 2016. Article (CrossRef Link)
[26] 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, no. 8, pp. 4062-4080, May, 2015. Article (CrossRef Link)
[27] X. Liu, L. Wang, G. B. Huang, et al., "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, pp. 253-264, 2015. Article (CrossRef Link)
[28] Y. Wang, X. Liu, Y. Dou, et al., "Multiple kernel learning with hybrid kernel alignment maximization," *Pattern Recognition*, vol. 70, pp.104-111, 2017. Article (CrossRef Link)
[29] C. R. Rao and S. K. Mitra, *Generalized inverse of matrices and its applications*, Wiley , New York, 1971.
[30] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. of Conference on Neural Network*, pp. 1942-1948, 1995. Article (CrossRef Link)
[31] F. Kuang, W. H. Xu, S. Y. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing*, vol. 18, no. C, pp. 178-184, 2014. Article (CrossRef Link)
[32] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey Wolf Optimizer," *Advances in engineering software*, vol. 69, pp. 46-61, March, 2014. Article (CrossRef Link)
[33] X. Yang, A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464-483, 2012. Article (CrossRef Link)
[34] R. Storn, K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, December, 1997. Article (CrossRef Link)
[35] A. Iosifidis, A. Tefas, I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11-17, 2015. Article (CrossRef Link)
[36] Archibe, U.K. KDD cup 1999 data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. 1999.
[37] Archibe, U.K. NSL data. http://nsl.cs.unb.ca/NSL-KDD. 2006.
[38] Kyoto university. Traffic data from Kyoto university honeypots. http://www.tatakura.com/Kyotodata/. 2009.
[39] O. Atilla and E. Hamit, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *Peer J Preprints*, vol. 4, pp. e1954vl, 2016. Article (CrossRef Link)
[40] K. A. P. Costa, L. A. M. Pereira, R. Y. M. Nakamura, et al., "A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks," *Information Sciences*, vol. 294, no. 10, pp. 95-108, 2015. Article (CrossRef Link)
[41] S. K. Dong, J. S. Park, "Network-based intrusion detection with support vector machines," in *Proc. of International Conference on Information Networking*, pp. 747-756, 2003. Article (CrossRef Link)
[42] J. Zhang, M. Zulkernine, A. Haque, "Random-forest-based network intrusion detection systems," *IEEE Transactions on Systems man & Cybernetics*, vol. 38, no. 5, pp. 649-659, 2008. Article (CrossRef Link)
[43] H. A. Nguyen, D. Choi, "Application of data mining to network intrusion detection: classifier selection model," *Challenges for Next Generation Network Operations and Service Management*, pp. 399-408, 2008. Article (CrossRef Link)

Yanping Shen is a Ph.D. student in the Information Security Center, School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing, China. Her research interest centers on network security.

Kangfeng Zheng is an associate professor who is working in the Information Security Center, School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include networking and system security, network information processing and network coding.

Chunhua Wu is a lecturer who is working in the Information Security Center, School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include network and information security.

Yixian Yang is a professor who is working in the Information Security Center, School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include information security and cryptography.