J Inf Process Syst, Vol.18, No.1, pp.146~158, February 2022
https://doi.org/10.3745/JIPS.03.0174

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Hybrid PSO-BPSO Based Kernel Extreme Learning Machine Model for Intrusion Detection

Yanping Shen[1,2,*], Kangfeng Zheng[3], and Chunhua Wu[3]

Abstract

With the success of the digital economy and the rapid development of its technology, network security has received increasing attention. Intrusion detection technology has always been a focus and hotspot of research. A hybrid model that combines particle swarm optimization (PSO) and kernel extreme learning machine (KELM) is presented in this work. Continuous-valued PSO and binary PSO (BPSO) are adopted together to determine the parameter combination and the feature subset. A fitness function based on the detection rate and the number of selected features is proposed. The results show that the method can simultaneously determine the parameter values and select features. Furthermore, competitive or better accuracy can be obtained using approximately one quarter of the raw input features. Experiments proved that our method is slightly better than the genetic algorithm-based KELM model.

Keywords

Feature Selection, Intrusion Detection, Kernel Extreme Learning Machine, Parameter Optimization, Particle Swarm Optimization

1. Introduction

Information applications, which have become foundational services of the current society, have been suffering various threats, such as worms, Trojan, denial of service attacks, phishing and botnets. As the second line of security defense after a firewall, intrusion detection becomes an irreplaceable component to ensure system security.

To detect invasion threats, various machine learning models [1,2] have been used in intrusion detection. However, they suffer from long training times, parameter tuning issues and poor generalization [3-5]. On the other hand, the kernel trick has been widely employed in machine learning, as well as the extreme learning machine (ELM) [6]. In recent years, although the ELM and its variants have been applied to intrusion detection [7-14], little work has been published on the optimization of ELM and its variants.

The grid search method [7] is often used to optimize model parameters, which have a great impact on the performance of the model. However, optimal results depend heavily on the step it chooses. The Levenberg–Marquardt (LM) method [13,14], based on the principle of gradient descent, easily obtains the local optimal solution. Particle swarm optimization (PSO) [15], which has few parameters to tune, is

Manuscript received February 5, 2021; first revision July 19, 2021; accepted August 23, 2021.
* **Corresponding Author:** Yanping Shen (shenyanping@cidp.edu.cn)
[1] School of Information Engineering, Institute of Disaster Prevention, Sanhe, China (shenyanping@cidp.edu.cn)
[2] Key Laboratory of Building Collapse Mechanism and Disaster Prevention, China Earthquake Administration, Sanhe, China
[3] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China (kfzheng@bupt.edu.cn, wuchunhua@bupt.edu.cn)

an effective heuristic method. Feature selection, which is divided into two types, wrapper and filter, can also directly affect the accuracy and generalization performance of classifiers. The filter method has nothing to do with the subsequent learning algorithms. The wrapper method takes the data processing tasks (such as higher classification accuracy) as a guide [16,17]. The wrapper method usually generates a better result than the filter method. The grid search method and LM method cannot be used for feature selection, but PSO can be used both to optimize parameters and to select features. Therefore, PSO is the preferred algorithm for model optimization.

This work proposes the combination of kernel extreme learning machine (KELM) [18] and PSO for intrusion detection. To improve the classification accuracy and generalization performance of ELM, a kernel function is applied in KELM. Due to the effectiveness of the Gaussian function and adjustment of fewer parameters, this paper uses the Gaussian function as the kernel function [19]. Two parameters, γ and C, both have great influence on the KELM [7,18], where γ is a kernel parameter impacting the classification result and C is a positive number making the performance stable. Meanwhile, feature selection is also a key problem building an efficient classification model. Although the raw feature set offers a detailed description, it may contain redundant information that directly affects the performance of the classifier [20]. The proper selection of effective features makes it possible to prominently accelerate the training and testing processes and maintain performance. In other words, the effectiveness of the classifier is influenced by two factors: parameter and feature selection [21]. The model parameters and the feature subset must be determined simultaneously because the feature subset and kernel parameter will affect each other [22]. In our model, the hybrid PSO is used to determine the parameters and the feature subset. Simulations conducted on three public datasets have shown that the parameters and the feature subset can be determined simultaneously. The proposed model obtains almost the same or better accuracy using approximately one quarter of the raw input features.

The general structure of this article is as follows. The second section includes the related works. The background is provided in the third section. Section 4 describes the hybrid PSO-BPSO based KELM model. Section 5 outlines the experimental results and discussion. Conclusions and possible extensions are presented in Section 6.

2. Related Work

The ELM and its modifications have been applied to intrusion detection systems (IDSs). Cheng et al. [7] show that ELM is faster than support vector machines (SVMs), and the accuracy of the KELM is better than that of SVMs in multiclass classification. Because of the large network traffic and low detection rate involved in intrusion detection, Singh et al. [9] presented a method based on the online sequential ELM. The paper uses alpha profiling to reduce the time complexity and filtering to select features. Accounting for the low accuracy of individual classifiers, Fossaceca et al. [10] introduced a framework named MARK-ELM which combines the results of multiple classifiers to obtain a final decision. The KELM was chosen as its key classification algorithm. The author emphasized multiple kernel learning and did not consider the parameter optimization and feature selection problems.

Many techniques can be adopted, such as evolutionary algorithms and swarm intelligence [12], to optimize the classifier models. Kuang et al. [23] applied kernel principal component analysis (KPCA) to determine features and adopted GA to optimize the parameters of the detection engine SVM. Onan et al.

[24] introduced a weighted voting ensemble model in which a differential evolution algorithm is used to determine the weight of each classifier in the ensemble model. Zhang et al. [25] and Huang and Dun [26] applied ant colony optimization (ACO) and PSO to determine the parameters of the SVM. Shen et al. [27] adopted a bat algorithm to optimize an ensemble model. Bao et al. [28] applied a PSO-based memetic method to determine the parameters of an SVM. Ahila et al. [29] used PSO to select a feature subset and the number of hidden nodes in an ELM.

There are many studies to optimize the KELM model. Pan et al. [12] applied quantum PSO to modify the hidden nodes of the kernel extreme learning machine. Jayaprakash and Murugappan [13] employed the LM algorithm to determine the hidden nodes of the kernel extreme learning machine. Jaiganesh and Sumathi [14] presented a kernelized ELM with LM learning, in which the kernel parameters were tuned using LM. The KELM mentioned in [12-14], which only replaced the activation function with the kernel function, is the basic ELM in essence, thus it differs from the optimized KELM described here. Therefore, only the number of hidden layer nodes or the kernel parameter need to be optimized [7]. In addition, the model they proposed is unstable, because they do not apply the positive value *C*. Ma et al. [30] applied a self-adaptive artificial bee colony to optimize the kernel parameters and parameter *C* of the KELM. Unfortunately, the authors neither considered feature selection nor applied the model in the intrusion detection area.

3. Background

3.1 Kernel Extreme Learning Machine (KELM)

The ELM evolved from the single-hidden layer feed-forward neural network (SLFN). The SLFN includes an input layer, a hidden layer and an output layer. The number of nodes in the input, hidden and output layers of the SLFN are *n, L* and *m* respectively. There are *Q* samples $\{(x_i, t_i)\}$, where $i=\{1,...,Q\}$, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^{\mathrm{T}} \in R^n$, and $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^{\mathrm{T}} \in R^m$. x_i and t_i represent the features and the label of the *i*-th instance, respectively. The model can be expressed as

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

where ω_i represents the input weight, b_i represents the threshold of the *i*-th hidden neuron, β_i is the output weight, and $g(x)$ is the activation function. Eq. (1) can be expressed as

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

where ω_i and b_i are produced randomly. In the case of $L \ll Q$, β can be acquired by calculating the least square error solution of the linear system $\mathbf{T}=\mathbf{H}\beta$ [6]:

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

where $\mathbf{H}^{\dagger}$ represents the Moore-Penrose generalized inverse matrix. Here, the singular value decomposition method [31] is adopted to calculate $\mathbf{H}^{\dagger}$.

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

However, in practical applications, the final estimation results may be inaccurate because of multicollinearity. Huang et al. [18] introduced parameter I/*C* in a diagonal matrix, which makes the performance of ELM more stable. The improved generalized inverse matrix and the output weight are represented as

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

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

To further enhance the stability and generalization capability of ELM, KELM was introduced [18]. A matrix Ω_{ELM} can be constructed to replace $\mathbf{H}\mathbf{H}^{\mathrm{T}}$. The kernel matrix can be shown as

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

The Gaussian kernel function is employed in this paper, so $K(u,v) = \exp(-(\| u - v \|^2 / \gamma))$, where γ is a kernel parameter. The output of KELM is

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

3.2 Particle Swarm Optimization (PSO)

PSO can be divided into standard PSO and binary PSO. In standard PSO, the velocity and position are denoted by $V_i=(v_1,v_2,...,v_r)$ and $X_i=(x_1,x_2,...,x_r)$, respectively. There is another important attribute called fitness, which is the metric of measuring a particle. The particles can obtain their own optimal position (*pbest*) by their own experience and obtain the global optimal position (*gbest*) by the experience of all particles. The update rules for velocity and position are as follows:

$$V_i^k = \omega_{pso} \times V_i^{k-1} + c_1 \times rand() \times (pbest_i^k - X_i^k) + c_2 \times rand() \times (gbest_i^k - X_i^k) \tag{9}$$

$$X_i^k = X_i^{k-1} + V_i^{k-1} \tag{10}$$

where *k* indicates the current number of iterations. c_1 and c_2 are positive acceleration coefficients.

The positions in the standard PSO are claimed to be continuous, but in practice, they may be discrete. Therefore, the discrete particle swarm optimization algorithm, namely, binary PSO [32], is developed. In the BPSO, the position of a particle is denoted in binary, which is defined as

$$X_i^k = \begin{cases} 1 & if\ rand() \le s(V_i^k) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where *s*() represents the sigmoid curve.

4. The Hybrid PSO-BPSO based KELM Model

This section describes the hybrid PSO-BPSO based KELM model (hereafter referred to as the hybrid PSO-KELM). The mechanism of the method is shown in Fig. 1.

In the training phase, PSO and BPSO are used to train the KELM on the training dataset to obtain the optimal parameters and features. The training procedure is an iterative process in PSO. After the first stage, the structure of the KELM network is fixed. Then, according to Eq. (8), the final classification result (i.e., the prediction of the labels) is obtained.

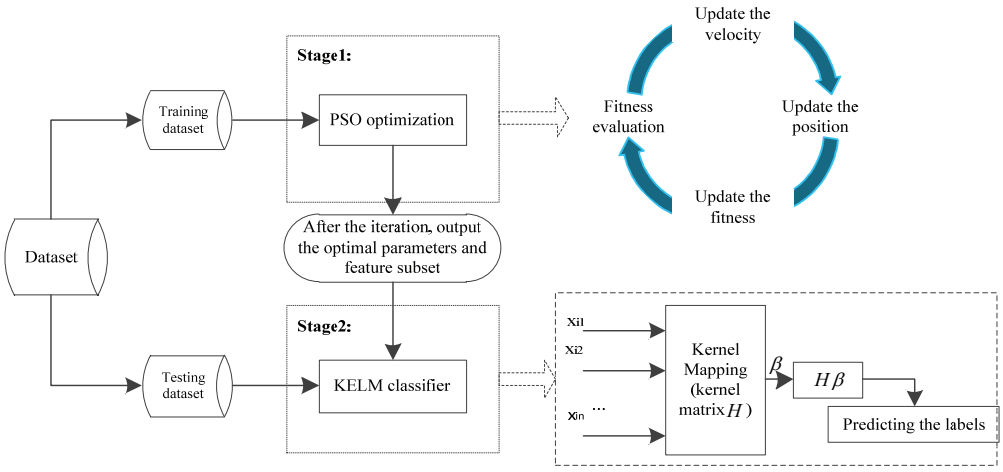


Fig. 1. The mechanism of the hybrid PSO-KELM.

4.1 Particle Representation

The particle consists of the feature mask, parameters C and γ. The feature mask is represented as binary data, where "1" indicates that the corresponding feature is chosen, and "0" indicates that it is not chosen. One can suppose that the dataset has n features, so the feature mask includes n bits. Parameters C and γ are real numbers. Therefore, the particle can be represented by the (n+2)-dimensional vector shown in Table 1.

The 1×(n+2) vector represents the position of the particle. However, the basic PSO can only handle a continuous position, while the binary PSO can only update a binary position. To update the position of our hybrid-vector, the basic PSO and binary PSO are combined [31]. If a particle's position of (1–n) dimensions is to be updated, then binary PSO will be applied; if a particle's position of (n+1) – (n+2) dimensions is to be updated, then standard PSO will be applied.

Table 1. Particle structure

Features	C	γ
$F_1 \ldots F_n$	x_1	x_2

4.2 Fitness Function Definition

The fitness function is used to improve the accuracy and reduce the number of features used. It can be defined as

$$fitness = \omega_1 \times acc + \omega_2 \times [1 - \frac{\sum_{i=1}^{n_F} f_i}{n_F}] \tag{12}$$

where acc denotes the accuracy; n_F denotes the size of all the features; f_i is the state of the i-th feature, and f_i=1 if i-th feature used, or f_i=0, otherwise; and w_1 and w_2 represent the weights of the corresponding measures.

5. Experimental Results and Discussion

5.1 Evaluation

The accuracy, detection rate and false positive rate are the most common performance evaluation criteria. Therefore, the performance evaluation of the proposed hybrid model uses these three evaluation parameters as presented in formulas (13), (14) and (15).

The accuracy (Acc) indicates the proportion of samples correctly judged.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

The detection rate (DR) represents the proportion of attack instances correctly detected by the model.

$$DR = \frac{TP}{TP + FN} \tag{14}$$

The false positive rate (FPR) represents the percentage of normal instances judged as attack instances.

$$FPR = \frac{FP}{TN + FP} \tag{15}$$

where TP denotes the number of attack instances correctly judged, FP denotes the number of normal instances misjudged to be attacks, FN indicates the number of attacks classified as normal, and TN denotes the number of correctly judged normal instances.

5.2 Datasets

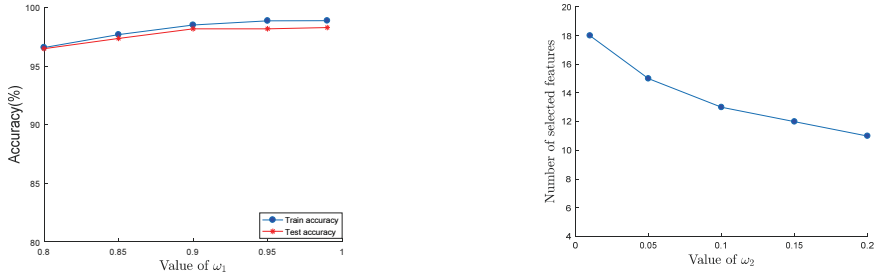
The KDD99 [33], NSL [34], and Kyoto datasets [35] were employed to validate the proposed model. KDD99 is the most widespread dataset for intrusion detection [36]. The NSL evolved from KDD99 and removed a large number of duplicate records. Its data features are the same as those of KDD99. Another more recent labeled dataset named Kyoto is also used. The Kyoto dataset is obtained from diverse types of honeypots consists of 24 features in which 14 statistical features are derived from KDD99 and 10 features are newly added. We selected 17 features, which are shown in Table 2 [37], for the experiment in this paper. Due to space limitations, the features of KDD99 are omitted here.

Table 2. Features used and their representations in the Kyoto dataset

Representation	Feature	Representation	Feature
P_1	duration	P_{10}	dst_host_srv_count
P_2	service	P_{11}	dst_host_same_src_port_rate
P_3	src_bytes	P_{12}	dst_host_serror_rate
P_4	dst_bytes	P_{13}	dst_host_srv_serror_rate
P_5	count	P_{14}	flag
P_6	same_srv_rate	P_{15}	IDS_detection
P_7	serror_rate	P_{16}	Malware_detection
P_8	srv_serror_rate	P_{17}	Ashula_detection
P_9	dst_host_count	-	-

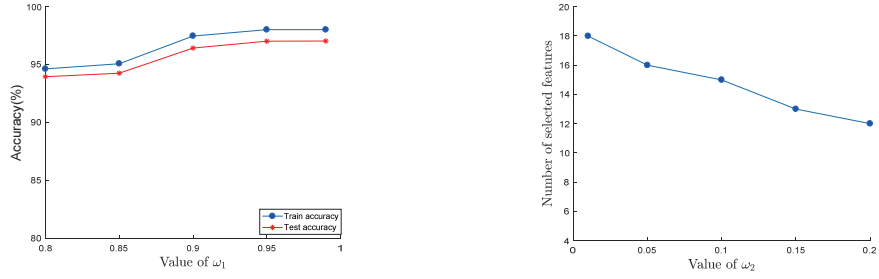
5.3 Experimental Procedure and Results

The value ranges of C and γ are $[2^{-10}, 2^3]$ and $[2^{-3}, 2^{10}]$ respectively. The max value of the velocity is set to approximately 20% of the range of the variables, so the velocity of parameter C is restricted to the range [-1.6, 1.6] and γ to [-204.8, 204.8]. For the discrete particle of binary PSO, the max velocity is restricted between [-4, 4]. The personal and social learning factors (c_1, c_2) are set to be in (2, 2). The inertia weight w_{pso} is set to 0.72. The population quantity and the number of iterations are 20 and 80, respectively. We randomly selected 50 groups of data from the original datasets for the experiment, and recorded the average results as follows.



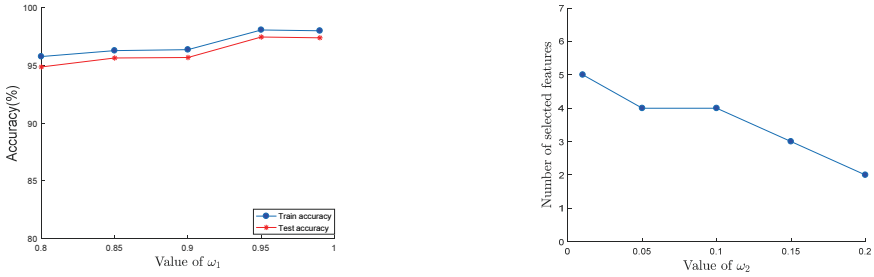
(a) (b)

Fig. 2. Adjusting the values of parameters ω_1 and ω_2 on KDD99: (a) accuracy with ω_1 increased and (b) number of selected features with ω_2 increased.



(a) (b)

Fig. 3. Adjusting the values of parameters ω_1 and ω_2 on NSL: (a) accuracy with ω_1 increased and (b) number of selected features with ω_2 increased.



(a) (b)

Fig. 4. Adjusting the values of parameters ω_1 and ω_2 on Kyoto: (a) accuracy with ω_1 increased and (b) number of selected features with ω_2 increased.

First, as discussed in Section 4.2, suitable values for ω_1 and ω_2 need to be determined. To select proper values for the proposed hybrid PSO-KELM, different values of ω_1 and ω_2 are analyzed on the three datasets as seen in Fig. 2–4. Regardless, the fitness is mainly determined by the accuracy rather than the number of selected features. Therefore, we only selected several sets of weight parameters for the experiment. Using several values adjusted from 0.8 to 0.99 with interval 0.05 and the last interval 0.04 for ω_1, the train accuracy and test accuracy are shown in Figs. 2(a), 3(a), and 4(a), which reveal that the test accuracy remains almost unchanged when ω_1 is greater than 0.95. Furthermore, the number of selected features using different values of ω_2 is presented in Figs. 2(b), 3(b), and 4(b). It is observed that the number of the selected features decreases with the increase of ω_2. This means that the number of chosen features is dependent on the value of ω_2. We chose values ω_1=0.95 and ω_2=0.05 for further testing.

The experimental results of the Grid-KELM, continue PSO-KELM, hybrid PSO-KELM and GA-KELM methods on the three datasets are shown in Tables 3–5. For Grid-KELM and continuous PSO-KELM, only parameter optimization can be done, but feature selection cannot be done. For the hybrid PSO-KELM, parameter optimization and feature selection can be carried out simultaneously. GA can also optimize the parameters and select the features simultaneously. Therefore, the writing method in the table is used to show the difference. In Table 3, the average test accuracy of Grid-KELM is 98.2997%. For the continuous PSO-KELM method, the average test accuracy is 98.3165%. For the hybrid PSO-KELM, the average test accuracy is 98.2492%, the average number of selected features is 14, while the testing time is 0.7791 seconds. There is no doubt that the hybrid PSO-KELM method can determine the parameters and the features used at the same time. Compared with the continuous PSO-KELM method, the testing time of the hybrid PSO-KELM decreased by 1.6%. Even when the dimension has been reduced to 14, the reduction of testing time is not too much. This result is understandable because the dimension of the input space has no direct effect on the size of the kernel matrix. However, generally, the computational complexity of the kernel matrix depends on the number of samples and the dimension of the input features.

Table 3. Experimental results of the Grid-KELM, continue PSO-KELM, hybrid PSO-KELM and GA-KELM methods on KDD99

Techniques	Parameters [*C*, γ]	Test time (s)	Test Acc (%)	Feature size
Grid-KELM	[8, 0.125]	0.7922	98.2997	41
Continues PSO-KELM	[7.6925, 0.125]	0.7916	98.3165	41
Hybrid PSO-KELM	[4.2364, 0.125]	0.7791	98.2492	14
GA-KELM without FS	[7.5878, 0.125]	0.7936	98.2828	41
GA-KELM with FS	[7.6059, 0.1302]	0.7780	98.1987	12

Table 4. Experimental results of the Grid-KELM, continue PSO-KELM, hybrid PSO-KELM and GA-KELM methods on NSL

Techniques	Parameters [C, γ]	Test time (s)	Test Acc (%)	Feature size
Grid-KELM	[8, 0.125]	0.7997	95.7071	41
Continues PSO-KELM	[8, 0.125]	0.7943	95.7071	41
Hybrid PSO-KELM	[7.8414, 0.125]	0.7884	96.9697	14
GA-KELM without FS	[7.7509, 0.1485]	0.7938	95.5892	41
GA-KELM with FS	[7.8405, 0.1345]	0.7775	95.1515	13

Table 5. Experimental results of the Grid-KELM, continue PSO-KELM, hybrid PSO-KELM and GA-KELM methods on Kyoto

Techniques	Parameters [C, γ]	Test time (s)	Test Acc (%)	Feature size
Grid-KELM	[8, 0.125]	0.7785	98.4007	17
Continues PSO-KELM	[3.1310, 0.125]	0.7736	97.6936	17
Hybrid PSO-KELM	[8, 0.125]	0.7664	98.2828	4
GA-KELM without FS	[7.9871, 0.2711]	0.7701	97.5084	17
GA-KELM with FS	[7.9878, 0.2533]	0.7636	97.4747	3

In other words, feature selection is still important in reducing the computational complexity of kernel mapping. Table 4 shows that the NSL is more demanding with respect to the method, i.e., the testing accuracy on NSL is not as ideal as that of KDD99. The proposed approach is also compared with GA-KELM. Taken together, although the GA can perform parameter optimization and feature selection, the results are slightly worse than those of the hybrid PSO. Tables 3–5 show that a competitive or better level of accuracy can be achieved with fewer features, which indicates that some features are uncritical to the performance of the classifier.

In the hybrid PSO-KELM, the frequencies of the features used in ten runs are listed in Tables 6 and 7. The feature with frequency equal to or greater than four times the other features' frequencies is considered to be a significant feature. There are 41 features in the KDD99 and NSL datasets, represented by M_1, M_2, ..., M_{41}. The significant features consist of M_2, M_3, M_{16}, M_{23}, M_{32}, M_{33}, M_{35}, M_{36} and M_{40}. The important features of the Kyoto are P_4, P_{10}, and P_{14}. The features chosen above are important in judging invasion. For instance, F_{23} represents the number of connections with the same target host as the current connection in the last 2 seconds. There is a very close relation between F_{23} and the DoS attack. In contrast, the features that are not chosen even once, represented by "*others*" in the last column, are thought to be redundant.

Table 6. Frequency of the selected features on KDD99 and NSL

Feature	M_2	M_3	M_5	M_6	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
Frequency	4	9	1	2	1	2	2	3	3	1	**2**
Feature	M_{16}	M_{17}	M_{18}	M_{20}	M_{22}	M_{23}	M_{25}	M_{26}	M_{29}	M_{30}	M_{31}
Frequency	4	1	2	1	2	6	2	1	3	1	1
Feature	M_{32}	M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{40}		*others*	
Frequency	5	8	1	7	4	3	2	4		0	

Table 7. Frequency of the selected features on the Kyoto

Feature	P_1	P_2	P_4	P_5	P_9	P_{10}	P_{14}	*others*
Frequency	1	3	10	1	1	10	8	0

Finally, performance comparisons of the different methods for KDD99 were carried out. The average simulation results of the measures are shown in Table 8 [7,38]. Feature selection is performed, and the three methods used the same selected features. The proposed model has the best overall performance among the current popular machine learning methods. Compared with other activation functions, ELM with sigmoid (*sig*) activation function has the best performance. Its performance is dependent on the selection of the hidden neuron, which is set to 80 here. Table 8 shows that the hybrid PSO-KELM is better than the ELM in terms of accuracy. This is because KELM takes a stable kernel mapping as an alternative to the random mappings in ELM and has stable network output weights. Therefore, the KELM can avoid random fluctuations in the output of the model caused by random assignments in ELM and enhance the stability and generalization ability of the model. It also shows that SVM has relatively stable performance, but the false positive rate is not as good as our proposed approach.

Table 8. Comparison of accuracy of different models (%)

Model	Acc	DR	Normal	Probe	DoS	U2R	R2L	FPR
ELM (*sig*) [7]	97.81	99.16	97.62	98.92	99.50	10	61.67	2.38
SVM [38]	97.04	99.13	96.01	92.04	99.83	85.71	91.67	3.99
Hybrid PSO-KELM	98.25	99.02	98.44	97.42	99.33	35.00	87.50	1.56

6. Conclusion

A hybrid PSO-BPSO based KELM model is proposed and applied to intrusion detection. The standard PSO and binary PSO are both adopted to optimize the parameter combination and input features. A fitness function is designed for the hybrid PSO. It is proved by experiments that the method can determine the parameters and the appropriate features at the same time. The results also show that the method outperforms the GA-KELM model. The Gaussian kernel is chosen in this work. In future works, other kernels or multiple kernel learning can be studied. In addition, other metaheuristic methods can also be developed to optimize the model.

Acknowledgement

This work is supported by the Fundamental Research Funds for the Central Universities (No. ZY20215151).

References

[1] 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.

[2] S. Elhag, A. Fernandez, 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 Applications*, vol. 42, no. 1, pp. 193-202, 2015.

[3] 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 and Technology*, vol. 8, no. 1, pp. 107-119, 2013.

[4] 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, no. 1, pp. 66-82, 2013.

[5] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127-140, 2014.

[6] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.

[7] C. Cheng, W. P. Tay, and G. B. Huang, "Extreme learning machines for intrusion detection," in *Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, Australia, 2012, pp. 1-8.

[8] Z. Ye and Y. Yu, "Network intrusion classification based on extreme learning machine," in *Proceedings of 2015 IEEE International Conference on Information and Automation*, Lijiang, China, 2015, pp. 1642-1647.

[9] 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, 2015.

[10] 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, 2015.

[11] 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, 2016.

[12] L. J. Pan, W. Jin, and J. Wu, "A novel intrusion detection approach using multi-kernel functions," *Telkomnika*, vol. 12, no. 4, pp. 1088-1095, 2014.

[13] R. Jayaprakash and S. Murugappan, "Intrusion detection based on KELM with Levenberg-Marquardt optimization," in *Proceedings of 2015 International Conference on Communications and Signal Processing (ICCSP)*, Melmaruvathur, India, 2015, pp. 0154-0156.

[14] V. Jaiganesh and P. Sumathi, "Kernelized extreme learning machine with Levenberg-Marquardt learning approach towards intrusion detection," *International Journal of Computer Applications*, vol. 54, no. 14, pp. 38-44, 2012.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of International Conference on Neural Networks (ICNN)*, Perth, Australia, 1995, pp. 1942-1948.

[16] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, et al., "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1106-1119, 2012.

[17] Z. Zhu, Y. S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 70-76, 2007.

[18] G. B. 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, 2012.

[19] S. W. Lin, K. C. Ying, S. C. Chen, and Z. J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1817-1824, 2008.

[20] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling, "Feature selection using linear classifier weights: interaction with classification models," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, 2004, pp. 234-241.

[21] C. L. Huang and C. J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231-240, 2006.

[22] H. Frohlich, O. Chapelle, and B. Scholkopf, "Feature selection for support vector machines by means of genetic algorithm," in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, Sacramento, CA, 2003, pp. 142-148.

[23] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing*, vol. 18, pp. 178-184, 2014.

[24] A. Onan, S. Korukoglu, and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification," *Expert Systems with Applications*, vol. 62, pp. 1-16, 2016.

[25] X. Zhang, X. Chen, and Z. He, "An ACO-based algorithm for parameter optimization of support vector machines," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6618-6628, 2010.

[26] C. L. Huang and J. F. Dun, "A distributed PSO–SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1381-1391, 2008.

[27] Y. Shen, K. Zheng, C. Wu, M. Zhang, X. Niu, and Y. Yang, "An ensemble method based on selection using bat algorithm for intrusion detection," *The Computer Journal*, vol. 61, no. 4, pp. 526-538, 2018.

[28] Y. Bao, Z. Hu, and T. Xiong, "A PSO and pattern search based memetic algorithm for SVMs parameters optimization," *Neurocomputing*, vol. 117, pp. 98-106, 2013.

[29] R. Ahila, V. Sadasivam, and K. Manimala, "An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances," *Applied Soft Computing*, vol. 32, pp. 23-37, 2015.

[30] 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 optimisation strategy," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1342-1357, 2016.

[31] C. R. Rao and S. K. Mitra, "Further contributions to the theory of generalized inverse of matrices and its applications," *Sankhyā: The Indian Journal of Statistics Series A*, vol. 33, no. 3, pp. 289-300, 1971.

[32] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of 1997 IEEE International Conference on Systems, Man, And Cybernetics: Computational Cybernetics and Simulation*, Orlando, FL, 1997, pp. 4104-4108.

[33] The UCI KDD Archive, "KDD Cup 1999 Data," 1999 [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[34] University of New Brunswick, "NSL-KDD dataset," 2006 [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.

[35] Kyoto University, "Traffic data from Kyoto University's Honeypots," 2016 [Online]. Available: https://www.takakura.com/Kyoto_data/.

[36] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets," in *Proceedings of the 3rd Annual Conference on Privacy, Security and Trust*, St. Andrews, Canada, 2005, pp. 1723-1722.

[37] M. M. Najafabadi, T. M. Khoshgoftaar, and N. Seliya, "Evaluating feature selection methods for network intrusion detection with Kyoto data," *International Journal of Reliability, Quality and Safety Engineering*, vol. 23, no. 1, article no. 1650001, 2016. https://doi.org/10.1142/S0218539316500017

[38] D. S. Kim and J. S. Park, "Network-based intrusion detection with support vector machines," in *Information Networking*. Heidelberg, Germany: Springer, 2003, pp. 747-756.

Yanping Shen **https://orcid.org/0000-0001-7581-9203**

She received a Ph.D. degree in computer application technology from Beijing University of Posts and Telecommunications in 2021. She is an associate professor who is working in the School of Information Engineering, Institute of Disaster Prevention, Sanhe, China. Her research interest centers on network security.

Kangfeng Zheng **https://orcid.org/0000-0002-1160-5596**

He received a Ph.D. degree in computer application technology from Beijing University of Posts and Telecommunications in 2006, where he is currently a professor with the School of Cyberspace Security. His research interests are network security, artificial intelligence security application and cognitive security.

Chunhua Wu **https://orcid.org/0000-0001-6445-8000**

She received a Ph.D. degree in computer application technology from Beijing University of Posts and Telecommunications in 2008, where she is an associate professor with the School of Cyberspace Security. Her research interests are machine learning, network security and malware detection.