Network intrusion detection based on Chaotic Multi-Verse Optimizer

Guozhu Liu[1],Bolun Zhang[2], Xinglu Ma[3], Jingjing Wang[4];
{lgz_0228@163.com[1], josephzbl@163.com[2] , qdmxl@aliyun.com[3]}
School of Information Science and Technology ,Qingdao University of Science &
Technology,Qingdao 266061,China[1,2,3]

Abstract. Network data with large amount and high dimensional features usually leads to a not high accuracy of detection when the support vector machine (SVM) is used in the intrusion detection. Principal component analysis (PCA) combined with CMVO-SVM is adopted to improve the accuracy of intrusion detection. Among them, PCA is used for dimensionality reduction and feature extraction on intrusion data and the parameter selection of SVM is optimized by Chaotic Multi-Verse optimizer (CMVO). Performance is tested on KDDCUP99 standard test data set. By comparing the results,the effectiveness of the method is showed through the increased accuracy of intrusion detection,decreased false alarm rate and false negative rate.

Key words: Multi-Verse Optimizer; chaotic maps; intrusion detection; SVM

1 Introduction

Intrusion Detection Systems(IDS), which have become an important complement to most organizations' security infrastructures[1], can protect Internet applications and computer networks from evolving cyber attacks. As an active defense technology, IDS helps to discover, identify and distinguish unauthorized use, duplication, alteration and destruction of information systems by analyzing the information collected by the network, so as to judge the behaviors that endanger the system security and ensure the information safety. At present, the main algorithms used in intrusion detection are neural network algorithm, immune genetic algorithm, support vector machines and so on. Existing intrusion detection algorithms still have some shortcomings, for example, neural network algorithms will lead to the largest false alarm rate[2]; Immune genetic algorithm is more complex, not easy to program, and the matching speed in the large amount of data is slower.

Support Vector Machine (SVM) is a machine learning method based on the theory of VC dimension and minimization of structural risk in statistics. It solves the problem of small sample, nonlinear and high-dimensional pattern recognition, and to a large extent, overcomes the "dimensionality disaster" and "over-study" and other issues[3]. Because of its superior classification performance, Support Vector Machine is widely used in the construction of intrusion detection system, but the penalty factor "C" and the value of kernel function parameters "g" affect the performance of the model, therefore, how to choose better parameters is the key to improve the testing accuracy. Particle swarm optimization(PSO) and genetic algorithm(GA)[4] are the main methods to optimize SVM parameters based on swarm intelligence algorithms. However, they all have some shortcomings in application. The convergence

MOBIMEDIA 2018, June 21-22, Qingdao, People's Republic of China

DOI 10.4108/eai.21-6-2018.2276603

speed of GA is slow and it is easy to fall into partially optimal; Particle Swarm Optimization converges fast, is easy to implement, but also easily fall into the local optimum.

Multi-Verse Optimizer[5](MVO) is a new meta-heuristic algorithm that builds mathematical models using three concepts: white holes, black holes and wormholes. Compared with PSO and GA, it requires fewer parameters to be adjusted, and it has strong robustness, better global optimization ability and is easy to implement. Nowadays, it is used in the reactive power optimization scheduling[6], SVC system regulation[7], annual peak load forecasting application[8]. It's used by Hossam Faris et al.[9] to choose the best features and optimize SVM parameters. Cong Hu et al.[10] use Levy flight principle to improve the MVO algorithm to jump out of the local optimum.

Firstly, the MVO algorithm is introduced. The CMVO algorithm is proposed by combining the advantages of MVO's good convergence accuracy with the ergodicity and randomness of Chaos theory. Afterwards, the intrusion detection strategy based on PCA and CMVO-SVM is put forward. Data dimensionality reduction and feature extraction are carried out by PCA, and then CMVO algorithm is used to select SVM parameters. Finally, KDDCUP99 network intrusion dataset is used as the experimental data in this paper. The experimental results show that this strategy can effectively improve the detection accuracy, reduce the false positive rate and false negative rate.

2 Introduction to Multi-Verse Optimizer algorithm and Chaotic Model

2.1 Multi-Verse Optimizer algorithm

Multi-Verse Optimization is mainly inspired by the three major concepts of white hole, black hole and wormhole in the theory of multiverse in physics. The multiverse has the common influence of white holes, black holes and wormholes, and finally reaches a state of balance. In the optimization problem, the universe represents a feasible solution to the problem. The position of the universe represents the component of the solution, and the inflation rate of the universe represents the fitness value.

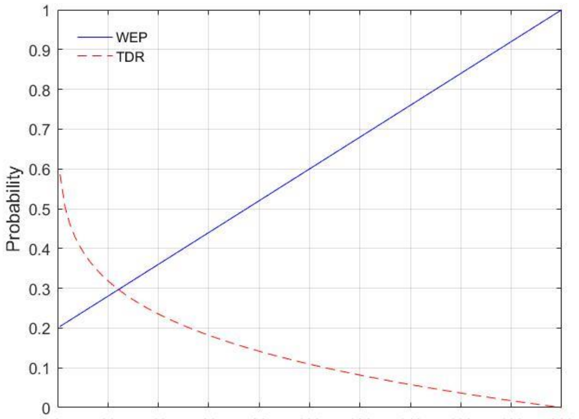

Fig.1 . Wormhole existence probability (WEP) versus travelling distance rate (TDR)

The wormhole existence probability(WEP) and travelling distance rate(TDR) are two important parameters that affect the performance of the algorithm. As shown in Fig. 1, it grows linearly,which emphasize exploitation as the progress of optimization process. TDR is the distance the matter travels through the wormhole around the optimal universe, and contrary to WEP, it is continuously reduced during the iteration to facilitate more accurate local searches.

The MVO algorithm starts with the creation of a random universe, in which the matter in the universe moves through the white holes toward the black holes and can migrate through the wormholes toward the optimal universe until the standard iteration is completed.

The MVO algorithm can be summarized as follows:

Step1:Initialize the universes U , the maximum number of iterations $Max\_iteration$, and variable interval $\left[lb,ub\right]$.

$$\boldsymbol{U}=\begin{bmatrix} x_{1}^{1} & x_{1}^{2} & \cdots & x_{1}^{d} \\ x_{2}^{1} & x_{2}^{2} & \cdots & x_{2}^{d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n}^{1} & x_{n}^{2} & \cdots & x_{n}^{d} \end{bmatrix} \tag{1}$$

In Eq .(2.1) , x is the location of the universe, d is the number of parameters(variables) , n is the number of universes (candidate solutions).

Step2:Arrange the universes according to inflation rates and choose a white hole with the roulette wheel mechanism.

$$x_i^j=\begin{cases} x_k^j & r_1<NI\left(U_i\right) \\ x_i^j & r_1\geq NI\left(U_i\right) \end{cases} \tag{2}$$

In equation (2), x_i^j is the *j*th parameter of the ith universe; U_i shows the ith universe; $NI\left(U_i\right)$ is the normalized inflation rate of the ith universe; r_1 is the random number between $\left[0,1\right]$; x_k^j is the ith parameter of the kth universe selected by the roulette wheel mechanism.

Step3:Update the WEP and TDR according to equation(3) and equation(4), and check the boundary.

$$WEP=min+l\cdot\left(\frac{max\text{ - }min}{L}\right) \tag{3}$$

$$TDR=1-\frac{l^{1/p}}{L^{1/p}} \tag{4}$$

In equation (3), WEP_{min} is the minimum value of WEP (0.2 in this paper) and WEP_{max} is the maximum value of WEP (1 in this paper); l indicates the current iteration number and L shows the maximum iterations , and p (in the paper equals 6) defines the exploitation accuracy over the iterations.

Step4:Calculate the current inflation rate of the universe, if it is better than the preservation of the inflation rate of the universe, then update the preservation of the universe inflation rate, otherwise maintain the current universe.

Step5:Update the location of the universe according to equation (5) and equation (6), and find the best individual in the optimal universe.

When $r_2<WEP$

$$x_i^j=\begin{cases} X_j+TDR\left(\left(ub_j-lb_j\right)\cdot r_4+lb_j\right) & r_3<0.5 \\ X_j-TDR\left(\left(ub_j-lb_j\right)\cdot r_4+lb_j\right) & r_3\geq 0.5 \end{cases} \tag{5}$$

When $r_2>WEP$

$$x_i^j=x_i^j \tag{6}$$

In equation (5) and equation (6), r_2, r_3, and r_4 are random numbers between $[0,1]$; X_j is the jth parameter of the current best universe, and lb_j is the lower bound of jth variable and ub_j is the upper boundary of the jth variable.

Step6:Determine the termination conditions: if the number of iterations reaches the maximum, the corresponding result is output, otherwise, the number of iterations is increased by 1, and execution returns to Step2.

2.2 Chaotic model

Chaos is a seemingly random non-linear random phenomenon in nature. It is a stochastic behavior between regular and random, which is shown by a completely deterministic system without external random factors. It is different from muddledness and irregular phenomenon, although seemingly random, but it has a sophisticated internal structure, with randomness, ergodicity, regularity and other characteristics.

Logistic mapping is a typical chaotic model. It’s proved that Logistic mapping is the best chaotic map to improve MVO performance. Its expression is:

$$z_{k+1} = \mu z_k \left(1 - z_k\right) \tag{7}$$

where $z_k \in (0,1)$ is the kth chaos variable, μ is the control parameter. When the value of μ is 4, the system is in complete chaos.

3 Chaotic Multi-Verse Optimization algorithm and intrusion detection strategy

3.1 Chaotic Multi-Verse Optimization algorithm

The strategy of initializing and updating universe positions in MVO algorithm is based on iterative iterations of individual populations. Therefore, the merits and demerits of individual locations directly affect the algorithm's ability to find optimal solutions. The introduction of chaos into the MVO algorithm reduces the likelihood of falling into local extreme when updating the universe. The basic idea of improvement includes the following two points: (1) It uses the ergodicity of chaotic motion to generate a large number of initial values, selects the better universe as the initial universe and improves the quality of the initial solution. (2) When the MVO algorithm appears the phenomenon of "precocious"[11], it can jump out of the local extreme range through the chaotic disturbance.

Observe the overall change in the expansion rate in the MVO algorithm, expressed as a variance, which is defined as:

$$\sigma^2 = \sum_{t=1}^{N} \left(\frac{f_t - f_{avg}}{f} \right)^2 \tag{8}$$

In equation (8) and equation (9), N is the total number of universes in the group, f_{avg} is the average of the universe inflation rate, and f_t is the expansion rate of the tth individual universe. f is the normalization scaling factor used to limit the size of σ^2. The general value is:

$$f=\begin{cases}\max\left\{\left|f_t - f_{avg}\right|\right\}, \max\left\{\left|f_t - f_{avg}\right|\right\} > 1 \\ 1, \max\left\{\left|f_t - f_{avg}\right|\right\} < 1\end{cases} \tag{9}$$

If the difference between two adjacent σ^2 less than the threshold, it indicates that "premature" phenomenon occurs. At this point, we need to solve this problem by chaos disturbance of the optimal location in the universe.

Step1 Parameter settings: CMVO the maximum number of iterations $Max\_iteration$, the number of universes N, SVM penalty factor and the range of kernel functions, and so on.

Step2 Use Chaos to initialize the universe. A vector $z_1=\left(z_{11}, z_{12}, \cdots, z_{1n}\right)$ with n dimensions and each component between $(0,1)$ is generated , and according to equation (8) , we get N components $z_{i+1,j}=\mu z_{i,j}\left(1-z_{i,j}\right)\left(j=1,2,\cdots,n; i=1,2,\cdots,N-1\right)$.

According to the SVM parameter range in Step 1, the universe position is generated according to equation (10).

$$X=lb+z\left(ub-lb\right) \tag{10}$$

After many iterations, the m initial solutions with better performance are selected from the generated $N(N>m)$ initial universes as the final initial universe. At this time, the generated universe has the best quality and distribution.

Step3:Calculate the universe's inflation rate and sort by roulette wheel mechanism to choose a white hole.

Step4:Update the WEP and TDR according to equation (3) and equation (4), and check the boundary.

Step5:Calculate the current inflation rate of the universe, if it is better than the preservation of the inflation rate of the universe, then update the preservation of the universe inflation rate, otherwise maintain the current universe.

Step6:Update the location of the universe according to equation (5) and equation (6), and find the best individual in the optimal universe.

Step7:Calculate the coefficient of variance σ^2 of the universe inflation rate , if the σ^2 difference between two adjacent less than the threshold ν (its value is set to 10^{-5}), we can assume that "premature" phenomenon occurs .Then $m-1$ two-dimensional vectors with each component value between $(0,1)$ are randomly generated, and their respective component carriers are within the chaotic disturbance range $\left[-\beta, \beta\right]$. The disturbance Δx is calculated as : $\Delta x=-\beta+2*\beta*z$, and the new universe location $x`=x+\Delta x$ is calculated. Calculate the inflation rates of the two old and new universes, compare them, choose one with larger expansion rate and set its universe position as the new universe position.

Step8:Determine the termination conditions: If the number of iterations reaches the maximum, the corresponding result is output, otherwise, the number of iterations is increased by 1, and execution returns to Step2.

3.2 Data reduction based on PCA

Network intrusion data usually has the characteristics of high dimension and large amount of data. The "dimensionality disaster" brought by high dimension data not only increases the time complexity but also affects the accuracy of algorithm classification. PCA can reduce the dimensionality of data by extracting a small number of information primitives that can reflect data attributes from multidimensional features.

Step1:Set the number of original input data samples be m, and each sample has p feature attributes. equation (11) is the input data matrix.

$$x_i = \left(x_{i1}, x_{i2}, \ldots, x_{ip}\right)^T \left(i = 1, 2, \ldots, m\right) \tag{11}$$

equation(12) is the calculated covariance matrix S of sample X , and equation(13) is the mean vector of X .

$$\mathrm{S} = \sum_{i=1}^{m} \left(x_i - U\right)\left(x_i - U\right)^T \tag{12}$$

$$\mathrm{U} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{13}$$

Step2:Use eigenvalue decomposition to find the corresponding eigenvectors $\boldsymbol{E} = \left(\theta_1, \theta_2, \ldots, \theta_p\right)$ of

P eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ of S and sort eigenvalues by size .

Step3: Calculate the ith principal component according to equation (14), and the contribution of the ith contribution rate of the principal component samples according to equation (15).

$$Y_i = e_i^T X \tag{14}$$

$$\lambda_i \left(\sum_{j=1}^{p} \lambda_j\right)^{-1} \tag{15}$$

Step4:Statistics the cumulative contribution rate to determine the number of sample components of the principal component, and select the first n input variables instead of the original value, which can achieve the purpose of data reduction.

3.3 Intrusion detection based on CMVO algorithm

Intrusion detection strategy based on CMVO algorithm reduces the dimension of network data through PCA, uses CMVO to optimize SVM parameters, and establishes a classification model to detect abnormal behavior. The kernel functions of SVM are mainly radial basis function(RBF), polynomial function(Poly) and Sigmoid function(Sigmoid). The radial basis function has good generalization performance and performs well. Therefore, Radial Basis Function is chosen as the kernel function of SVM. And CMVO is applied to the optimization and selection of penalty factor C and kernel function of SVM in intrusion detection. Radial base kernel function expression is shown in equation (16):

$$\mathrm{K}\left(x_i, x_j\right) = \exp\left(-\frac{\left|x_i - x_j\right|^2}{2r^2}\right) \tag{16}$$

Its main components include feature extraction of training data and test data, CMVO algorithm optimizing SVM parameters, SVM model training and detection, and decision response. The framework of the intrusion detection system is shown in Fig. 2.

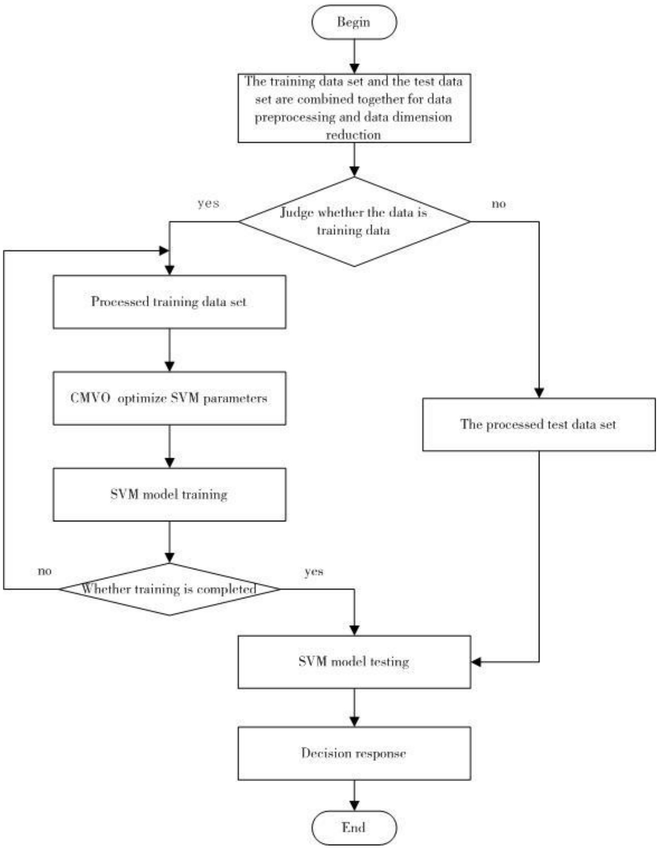
Begin
The training data set and the test data set are combined together for data preprocessing and data dimension reduction
Judge whether the data is training data
yes
no
Processed training data set
CMVO optimize SVM parameters
SVM model training
The processed test data set
no
Whether training is completed
yes
SVM model testing
Decision response
End

Fig. 2. Intrusion detection system structure

The whole system is divided into three stages:

(1) Data processing stage: The types of network data are more complex, so the data sets are first numerically quantified and normalized to eliminate differences between different attributes. Because of the high dimensionality of data set used in intrusion detection, irrelevant attributes and redundant attributes exist, so PCA is used to reduce dimensionality and extract features of network intrusion data to decrease the computation load.

(2) Model training stage: The processed training set data is continuously iterated through the CMVO algorithm to find better support vector machine parameters and establish a classifier model.

(3) Data inspection stage: The tested dataset after data processing is tested by trained SVM classification model. The decision response module makes the corresponding decision based on the result.

4 Data simulation

4.1 Data Sources

In order to verify the validity and superiority of the intrusion detection strategy proposed in this paper, KDDCUP99 experimental data set is selected for simulation and further analysis and research. The dataset was derived from an intrusion detection project conducted in collaboration with DARPA in 1998 by MIT Lincoln Laboratories to simulate real-world network environments. The simulation attacks in the experiment are divided into four types: PROBE, DOS, U2R, R2L. Each link in the KDD Cup99 dataset was recorded with 41 features.

4.2 Data Preprocess

The experiment is implemented under the platform of Matlab2016a and libsvm toolbox. Due to the complexity of the types of data, existing character types (such as protocol type, tcp, udp, ...) also have numeric types (duration, number of sent packets, etc.) and are therefore treated separately. The data preprocessing process is divided into the following three steps:

①The network connection record character attributes into digital attributes, such as protocol_type tcp, udp, icmp mapped to the numerical values 1, 2, 3 respectively.

② Data are normalized to reduce the impact of large attribute values on small attribute values, while reducing the numerical calculation difficulties. The data normalization function is:

$$f(x)=\begin{cases}\dfrac{x-x_{min}}{x_{max}-x_{min}},x_{max}\neq x_{min}\\ 0,x_{max}=x_{min}\end{cases} \qquad (17)$$

Adjust all data to the range $\left[0,1\right]$.

This paper selects the kddcup.data_10_percent_corrected file from the KDD Cup 99 dataset as the training data of the model, and the corrected file serves as the test data of the model. Since the dataset is too large, we randomly select 2000 training datasets as training datasets, randomly select 4000 datasets from the test dataset as the test dataset. The data types and numbers are shown in Table 1.

Table1. Experimental data types and numbers

type of data	Normal	Probe	Dos	U2R	R2L
The number of training sets	1500	180	270	40	10
The number of test sets	3000	360	540	80	20

4.3 Result analysis

The detection effect of intrusion data is to evaluate the experimental results and consider the detection rate, false alarm rate, and false alarm rate as evaluation indicators. Its definition is as follows:

$$\text{Detection rate}=\frac{\text{Accurately detected samples}}{\text{The total number of samples}}\times 100\% \qquad (18)$$

$$\text{False alarm rate}=\frac{\text{The number of normal samples that were misclassified}}{\text{Normal number of samples}}\times 100\% \qquad (19)$$

$$\text{False negative rate}=\frac{\text{Number of normal intrusion samples incorrectly predicted}}{\text{The total number of intruded samples}}\times 100\% \qquad (20)$$

Randomly extract data from the dataset and use the particle swarm algorithm, GA genetic algorithm, MVO algorithm, and CMVO algorithm to detect the KDDCUP99 dataset. Perform thirty experiments for each set of data, and take the average of the experiment results for 30 times. The test results are shown in Table 3.

In the experiment, the MVO and CMVO parameters are set as follows: the number of universes is 5 and the maximum number of iterations is 200; the minimum probability of wormhole is 0.2 and the maximum probability is 1; the maximum velocity in PSO algorithm is 5; the minimum velocity is: -5, the two parameters of speed update are 1.5 and 1.7 respectively. The genetic algorithm (GA) parameters are set as follows: the crossover probability is 0.8 and the mutation probability is 0.01. The optimal range of parameter C in SVM is $[0.1,100]$, and the range of parameter g is $[0.01,1000]$.

Table 2. Comparison of model establishment time and detection rate

	MVO time（s）	MVO detection rate（%）	CMVO time（s）	CMVO detection rate（%）
Use PCA to reduce dimensions	96.72	96.11	118.74	96.28
Without PCA to reduce dimensions	155.57	96.07	181.94	96.19

It can be seen from Table 2 that using PCA to extract features from the data set can reduce the amount of computation and speed up the model establishment. At the same time, the detection rate is improved due to the removal of redundant data.

Table 3. Comparison of experimental results of different algorithms

4 different optimization algorithms	Detection rate（%）	False alarm rate（%）	false negative rate（%）
PSO-SVM	94.35	1.12	19.31
GA-SVM	91.07	0.13	35.33
MVO-SVM	96.11	1.64	10.64
CMVO-SVM	96.28	1.61	10.05

As we can see in Table 3, compared with the PSO and GA algorithms, MVO and CMVO algorithms have better detection rate, lower false alarm rate and false negative rate, so as to establish a better network intrusion detection model. CMVO algorithm compared with MVO, through chaotic initialization and chaotic disturbance, to avoid falling into a local optimum, thus improving the detection rate.

5 Conclusion

This article explores the application of the combination of PCA and CMVO-SVM in network intrusion detection. PCA is used to achieve the purpose of dimensionality reduction as the feature extraction method when facing a large number of high-dimensional features of the network data. In order to improve the detection accuracy of SVM , CMVO algorithm is used to optimize the parameters of SVM.

The CMVO algorithm makes use of the "ergodicity" and "randomness" of chaos to generate a random "universe", and optimizes its iteration by choosing the initial universe position to speed up the convergence rate; Through chaotic disturbance of the universe position, the solution can be decomposed out of the local extreme and the accuracy of calculation can be improved. However, this algorithm also has some deficiencies, spending more time, looking for only the suboptimal solution rather than the optimal solution, etc. These aspects need to be focused on in the future. At the same time, the CMVO algorithm proposed in this paper only tries to improve the algorithm in the direction of avoiding the local optimal solution, and it can be merged with other optimization algorithms to propose a new algorithm, which is also worth studying.

References

[1] Ambusaidi M A, He X, Nanda P, et al. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm[J]. IEEE Transactions on Computers, 2016, 65(10):2986-2998.

[2] Bhati B S, Rai C S. Intrusion detection systems and techniques: A review[J]. International Journal of Critical Computer-Based Systems, 2016, 6(3):173.

[3] VAPNIK V N. Statistical learning theory [M],Berlin:Springer Verlag,1998:123-167.

[4] Aslahi-Shahri B M, Rahmani R, Chizari M, et al. A hybrid method consisting of GA and SVM for intrusion detection system[J]. Neural Computing & Applications, 2016, 27(6):1-8

[5] MIRJALILI S,MIRJALILI S M,HATAMLOU A.Multi-Verse optimizer:a nature inspired algorithm for global optimization[J].Neural Computing and Applications,2016,2(27):495-513

[6] Jangir P, Parmar S A, Trivedi I N, et al. A novel hybrid Particle Swarm Optimizer with multi verse optimizer for global numerical optimization and Optimal Reactive Power Dispatch problem[J]. Engineering Science & Technology An International Journal, 2017, 20(2).

[7] Karthikeyan K, Dhal P K, Karthikeyan K, et al. Multi verse optimization (MVO) technique based voltage stability analysis through continuation power flow in IEEE 57 bus[J]. Energy Procedia, 2017, 117:583-591.

[8] Zhao H, Han X, Guo S. DGM (1, 1) model optimized by MVO (multi-verse optimizer) for annual peak load forecasting[J]. Neural Computing & Applications, 2016:1-15.

[9] Faris H, Hassonah M A, Al-Zoubi A M, et al. A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture[J]. Neural Computing & Applications, 2017:1-15.

[10] Cong H, Zhi L, Tian Z, et al. A Multi-Verse Optimizer with Levy Flights for Numerical Optimization and Its Application in Test Scheduling for Network-on-Chip[J]. Plos One, 2016, 11(12):e0167341.

[11] Ewees A A, Aziz M A E, Hassanien A E. Chaotic multi-verse optimizer-based feature selection[J]. Neural Computing & Applications, 2017(1):1-16.