A Method Based on Multiple Population Genetic Algorithm to Select Hyper-Parameters of Industrial Intrusion Detection Classifier

Xuejun LIU[1]*, Hao WANG[1], Xiaoni ZHANG[1], Haiying LUAN[2], Yun SHA[1], Yong YAN[1]

[1] College of Information Engineering, Beijing Institute of Petroleum and Engineering,
19 Qingyuan North Road, Daxing District, Beijing, 102617, China
lxj@bipt.edu.cn (**Corresponding author*), whbeats@163.com, 2019540019@bipt.edu.cn,
shayun@bipt.edu.cn, yanyong@bipt.edu.cn

[2] Fluid Power and Automotive Equipment Center, Beijing Research Institute of Automation
for Machinery Industry CO., LTD, Beijing, 100120, China
lhying1129@aliyun.com

Abstract: The security of industrial control systems is increasingly prominent, and the performance of intrusion detection classifiers depends more on hyper-parameters. This paper proposes an improved multiple population genetic algorithm (IMPGA) used to intelligently search hyper-parameters of classifiers, and the simulated annealing algorithm (SAA) is used to control the evolution rate among various populations. In addition, the hash fitness value is used to reduce resource consumption and the directional evolution operator is introduced to optimize the population. This method can effectively avoid the algorithm falling into local optimal solution and save the optimal solution in the process of evolution. Thus, the optimal or approximate optimal combinations of hyper-parameters of classifiers are obtained and the accuracy of the classifiers is finally improved. In this paper, experiments are conducted on the following datasets: the natural gas pipeline experimental dataset of Mississippi State University from 2014 (a gas dataset), the intrusion detection systems dataset of Canadian Institute for Cybersecurity from 2017 (CICIDS2017 dataset) and an oil depot dataset. The experimental results of those three datasets show that the area under curve (AUC) of the back propagation neural network (BPNN) is more than 98%, of the extreme gradient boosting (XGBoost) is more than 99%, and of the support vector machines (SVM) is more than 98%. This selection method can effectively detect the intrusion attacks.

Keywords: Industrial control network, Intrusion detection, Genetic algorithm, Hyper-parameters optimization.

1. Introduction

Any intrusion attack in the industrial control system will cause serious losses, so an intrusion detection classifier with high accuracy and robustness is needed (Li et al., 2020). Rules-based intrusion detection classifiers have poor mutability, and the construction of the classifier requires an in-depth understanding of the application scenarios. In addition, the selection of the hyper-parameters of classifier directly affects the final classification result.

In recent years, intrusion detection classifiers in the field of industrial control have developed rapidly, including deep neural network exemplified by BPNN (Han et al., 2021), machine learning algorithms exemplified by XGBoost (Wang & Cao, 2020) and SVM. In the research of classifiers, many efforts have been made to improve their performance. For example, Wang et al. (2021) used an adaptive mechanism to optimize XGBoost for classification, Albashish et al. (2021) used SVM to select the optimal solution, Kashef (2021) used a reinforcement algorithm to improve the classification effect of SVM. However, various classifiers need appropriate hyper-parameters to achieve better accuracy. In the method of finding the optimal parameter combination of SVM, the following efforts should be mentioned: Priya et al. (2020) proposed particle swarm optimization (PSO), but PSO is prone to converging prematurely; Zhou et al. (2020) proposed PSO to construct the SVM classifier, but they did not demonstrate other classifiers; the artificial bee colony algorithm proposed by Eesa et al. (2015) is only used in intrusion detection feature selection.

Holland (1973) introduced a genetic algorithm (GA) based on the principles of natural evolution, natural selection and gene recombination. This algorithm selects high-quality individuals in the process similar to natural evolution (Mangano, 1995). The building block hypothesis and the schema theorem were used to evaluate the effectiveness of GAs (Goldberg, 1989). In genetic operation, even if there is a pattern with the same order, there will be different properties, which provide a mathematical basis for explaining the mechanism of genetic algorithms (Jin, 2016). Based on the above-mentioned advantages, genetic algorithms are widely used in optimization issues.

In order to improve the detection effect of intrusion detection classifier in industrial control system, IMPGA is proposed in this paper. It can

https://doi.org/10.24846/v30i3y202104

avoid the population falling into local optimal solution in the process of evolution by special evolutionary rules, and accelerate the convergence rate to find the optimal hyper-parameters for industrial control classifiers.

The structure of this paper is as follows. The first section represents an overview of the full text. The second section introduces the related work and the third section describes the algorithm and the implementation process. In the fourth section, the algorithm is verified by experiments and the fifth section summarizes the paper.

2. Related Works

2.1 The Algorithms and Optimization Strategies Used in This Method

In the study of genetic algorithms to search hyper-parameters, Zhang et al. (2021) proposed a method to optimize XGBoost through GA, but did not verify the effect of other classifiers. The dual population genetic algorithm proposed by Li (2018) and the improved dual population genetic algorithm proposed by He et al. (2020) can enhance the local search performance of the algorithm to a certain extent. It is a preliminary improvement of GA, but there are some defects in the global convergence. Chen et al. (2020) used the probability integral method to improve the multiple population genetic algorithm, but they did not optimize it according to the characteristics of industrial control data. Ma et al. (2020) proposed a multiple population genetic algorithm combined with artificial selection, but its complex evolutionary strategy lacks a certain evolutionary control method. Ni et al. (2021) proposed an interactive multiple population genetic algorithm, which promotes the evolution to a certain extent, but its selection strategy is relatively single. Due to the particularity of industrial control data, it is difficult for the above genetic algorithms to achieve high performance.

GA is designed and proposed according to the natural evolution. It converts complex combinatorial optimization issues into evolution issues, that is, through the population selection, crossover and mutation operations, the average fitness value of population is gradually improved. After the pattern theorem proves the validity of the GA, methods of searching for classifier hyper-parameters using GA appear.

In order to avoid trapping itself into a local optimum in the search process of GA, SAA is introduced as the optimization algorithm of GA. SAA is an optimization algorithm based on the Monte-Carlo iterative solution strategy. It is based on the similarity between annealing processes of solids in physics and general combinatorial optimization problems. It was first proposed by Metropolis (2004).

In order to reduce the resource consumption, this method introduces hash fitness value. In the process of GA, there is a certain probability that the individuals will be the same, especially in the late stage of evolution. This means that the search space of genetic algorithms becomes smaller and smaller when the genetic algorithm approaches convergence. Therefore, there is a high probability that repeated solutions would appear. If each individual and its fitness value are recorded in hash tables, the fitness value of repeated solutions can be obtained without computing.

A directional evolution operator combines SAA and elitism strategy. This strategy retains individuals with the highest fitness in each generation for a new population, which is called high-quality population. The individuals in this population generally have high fitness values, so elitism strategy should be used. Specifically, the high fitness individuals in each generation are copied to the next generation, and then participate in the selection, crossover and mutation processes.

2.2 The Classifiers Used in This Method

This paper tested and verified IMPGA by experimenting with three classifiers, BPNN, XGBoost and SVM.

BPNN is a kind of feed-forward neural network (Serban et al., 2020). Its output is propagated forward and the error is propagated backward (Bayar et al., 2019). The neural network is based on the mathematical model of extensive interconnection of a large number of neurons.

XGBoost is called extreme gradient boosting, and it was first developed by Chen Tianqi, which efficiently implemented the gradient boosting decision tree (GBDT) and made many improvements in engineering.

SVM was proposed by Vapnik (1998). It is a machine learning algorithm that relies on the

development of statistical learning theory. It can effectively solve small samples and high dimensional problems, especially for some nonlinear problems.

3. The Proposed Methods

As a meta-heuristic algorithm, GA has been widely used in various scenarios. But its single selection operator makes populations easily fall into the local optimal solution, and the probability of high fitness individuals in the evolution process is slightly lacking. A multiple population genetic algorithm has more abundant selection strategies than the traditional genetic algorithm. But the existing multiple genetic algorithms seldom consider the relationship among different populations, that is, each population evolves independently and lacks communication among different populations. Based on the above issues, IMPGA is proposed.

3.1 Improved Multiple Population GA

Firstly, the genetic algorithms proposed in this paper randomly generate a specified number of individuals. Each individual of the initial population is put into the classifier to get the corresponding fitness value. The individuals are sorted according to the fitness values, from high to low, and divided into three populations according to the set proportion, namely POP1, POP2 and REZ. Then, the three populations are evolved by the combination algorithm which is similar to the natural evolution process, that is, they evolve by different selection operators, crossover operators and mutation operators, and gradually evolve towards the optimal solution. The high crossover rate and high mutation rate of POP1 are controlled by SAA. The rates of POP2 are slightly slower than those of POP1, while the rates of REZ are slower than those of POP2. After the first evolution, the best individual that is in POP1 and POP2 respectively is put into the high-quality population. Then, according to certain rules, REZ provides new genotypes to POP1 and POP2. The above steps are repeated until reaching the specified generation. Finally, the directed evolution operator is used to make the high-quality population evolve again, and SAA is used to control its low evolution rate. The overall process of this method is shown in Figure 1.

IMPGA makes POP1 and POP2 evolve in different directions under the influence of different evolution operators, which can effectively avoid the genetic algorithm falling into a local optimum. Instead of discarding all the eliminated individuals during evolution, they are treated as a reserve population in order to preserve individuals with potential and transport them to POP1 and POP2 so as to provide new genotypes. The new genotype can effectively increase the species diversity of POP1 and POP2, so there is a probability that individuals with high fitness values will appear in the process of evolution, and reduce the probability that the population will fall into the local optimum. Finally, the high-quality population evolves by elite strategy, that is, preserving the top N highest individuals of

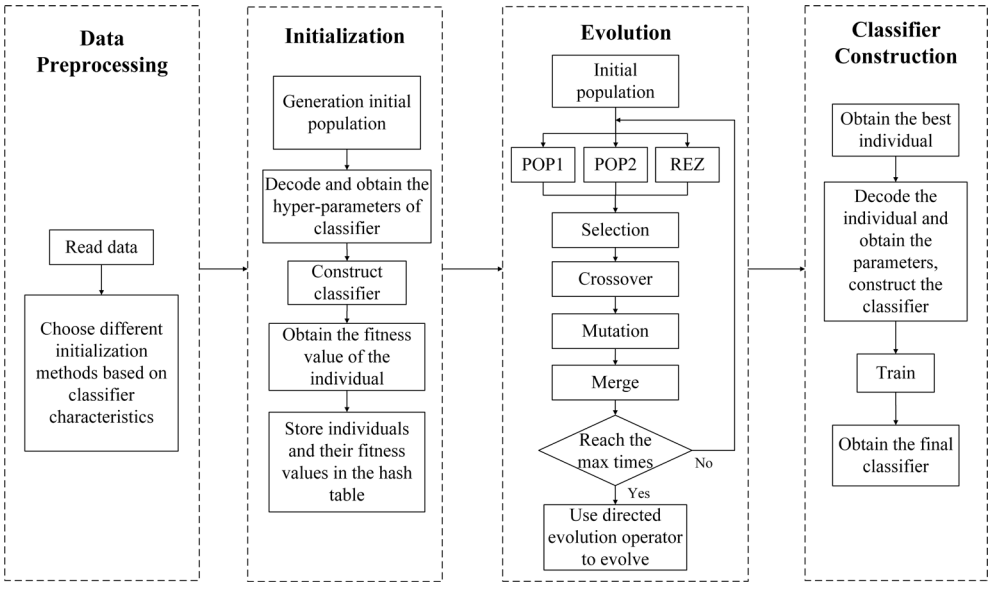

Figure 1. Process of searching hyper-parameters

each generation directly into the next generation to avoid the negative impact of evolution.

3.2 Implementation

The detailed steps needed to implement this method are the following:

(1) Generate the initial population. At the beginning, a specified number of chromosomes are randomly generated, which are binary strings composed of 0 and 1. The encoding methods of different classifiers are different. There are six parameters in BPNN with a total length of 58. Similarly, there are three parameters in XGBoost, and also in SVM, but the total length of their chromosomes is 20 and 22, respectively. The specific encoding method is shown in Figure 2. The pre-processing method is Min-Max or Z-Score and the activation function is Tanh or Sigmoid;

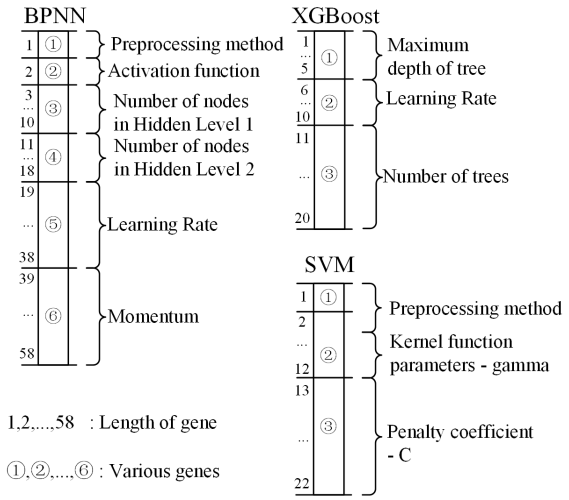

Figure 2. Coding methods for different classifiers

(2) Obtain the fitness values of individuals in the initial population. The individuals in the initial population are decoded and a classifier is constructed to obtain the fitness value of each individual in the initial population. Individuals and their corresponding fitness values are stored in a hash fitness dictionary;

(3) Divide the initial population. First, the initial population is sorted in descending order according to the fitness value, and divided into three populations according to the specified proportion, namely POP1, POP2 and REZ. Then each subpopulation evolves according to different evolutionary strategies, that is, POP1 uses the championship selection strategy, POP2 uses the roulette selection strategy, and REZ also uses the roulette selection strategy;

(4) SAA is used to control the evolution rate of each population. At the beginning of the algorithm, it starts with a high crossover and mutation rates. Through iteration, the crossover rates and mutation rates are gradually reduced due to the cooling factor. In this method, in order to achieve a high-speed search in POP1, given the higher initial temperature and lower cooling rate, POP2 takes a moderate evolutionary rate. REZ is used only to provide new genotypes, so it only needs slow rates. Slow rates not only can improve the fitness of individuals in REZ but they can also save memory resources. Individuals with the highest fitness in POP1 and POP2 respectively are saved in a new dictionary for the next step. The process of steps (3) and (4) is shown in Figure 3;

(5) Merge various populations as the next generation. After the evolution of three subpopulations, some individuals in REZ need to be transferred to POP1 and POP2 respectively. Specifically, in REZ, individuals with top X fitness

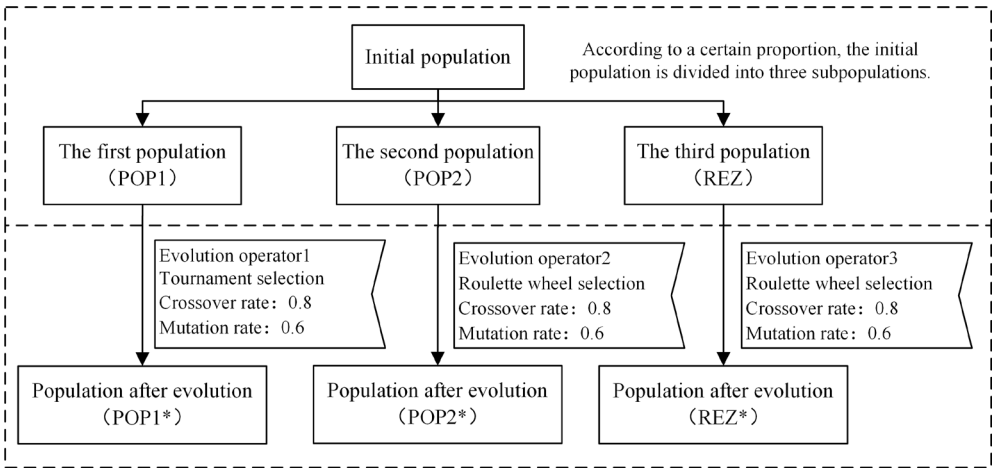

Figure 3. Partitioning and different evolution operators

are selected to replace the X individuals with the last X fitness. Similarly, in the reserved population, the X+1 to 2X individuals with descending order are selected to replace the X individuals with the last X fitness. Through the above steps, the idea of reserving the population to provide new genotypes for POP1 and POP2 is achieved. At the end of this step, all individuals are traversed to see if they exist in the hash table, and, if they do not, they are recorded in the table. The process of step (5) is shown in Figure 4;

(6) Repeat steps (4) and (5) until reaching the maximum evolution times;

(7) In the high-quality population, the directed evolution operator is used to evolve. After reaching the specified generation, the high-quality population with 2N individuals will be obtained. The high-quality population evolves through the elitism strategy to effectively reduce the occurrence of poor genes in a high-precision population, as shown in Figure 5. In addition, elitism strategy allows several optimal solutions to go directly to the next generation without participating in evolution, thus avoiding the destruction of good genotypes. After reaching the set generation times, the best individual in the high-quality population is obtained, which is the optimal sequence of parameters found by the IMPGA. Then, it is decoded to build a classifier.

4. Experiments

4.1 Preparation

The experiments run on Windows 10 system, which holds Ryzen7 4800H CPU, NVIDIA GeForce RTX 2060 GPU, 16.0 GB RAM. Python version is 3.7 and TensorFlow version is 1.14.

The fitness function of GA is AUC, which is a common performance measure in intrusion detection algorithm. AUC is a good trade-off between DR (Detection Rate) metric and FPR (False Positive Rate) metric.

Based on two datasets (Gas Dataset and CICIDS2017 Dataset) and three classifiers (BPNN, XGBoost and SVM), the average AUC of every generation and the optimal AUC of all the individuals in each generation are calculated, then the two diagrams are displayed to prove the effectiveness of the present method.

4.2 Experimentation on Gas Dataset

The dataset is from the Mississippi State University Key Infrastructure Protection Center's 2014 dataset for human intrusion detection and assessment in industrial control systems (Morris & Wei, 2014). These datasets can be used to help researchers evaluate the performance of supervisory control and data acquisition intrusion

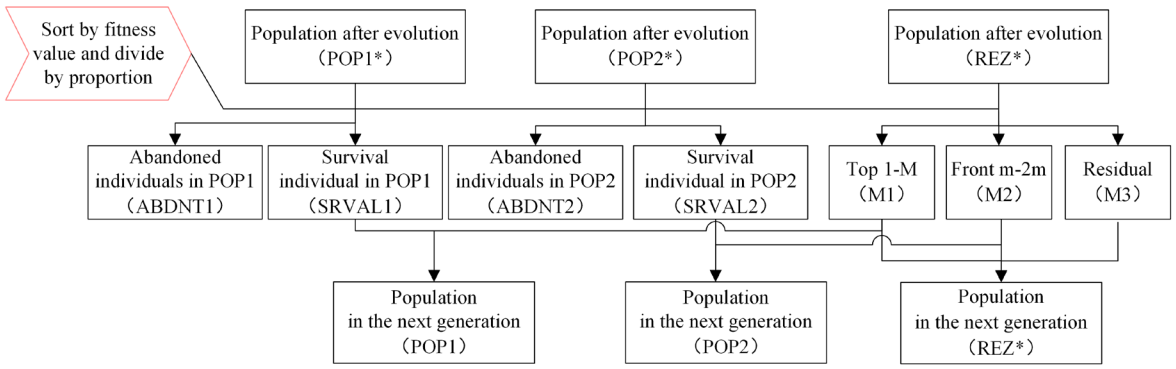

Figure 4. Exchange and merge population

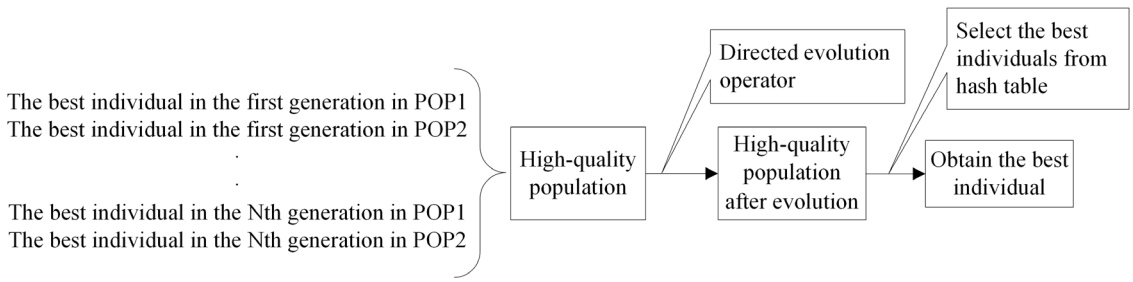

Figure 5. Final evolution with directed evolution operators

detection system (SCADA IDS) by using actual SCADA attack modes and simulated operator operations (Wei, 2013).

Before the experiment, the parameters of GA were set. They are determined by the experiment scale and classifier characteristics. The parameters set in this paper are shown in Table 1. Maximum evolution times and size of initial population are determined by the scale and complexity of the issues; the number of individuals to POP1 and POP2 from REZ is maintained in the same proportion; parameters of the SAA are determined by the function of the divided population.

The maximum AUC and the average AUC curves of the individuals in each generation were plotted, as shown in Figure 6. Group A is the curve of BPNN classifier, group B is the curve of XGBoost classifier, and group C is the curve of SVM classifier. The number 1 is the highest AUC in each generation of POP1. The number 2 is the average individual AUC in each generation of POP2. The number 3 is the highest AUC in each generation of POP2. The number 4 is the average individual AUC in each generation of POP2.

The chromosome with the highest fitness in the high-quality population is founded and decoded and a classifier is built. The results predicted by the classifier are shown in Table 2. Through AUC, ACC, detection rate, false alarm rate, recall rate, F-score, TNR and FNR, it can be noticed that the classifiers built by this method are at good levels. The AUC of the three classifiers reached 98.41%, 99.12% and 99.03%, respectively.

Table 2. Experimental result on gas dataset

	BPNN	XGBoost	SVM
AUC	98.407%	99.121%	99.031%
ACC	98.430%	99.133%	99.033%
DR	98.247%	99.286%	99.063%
FAR	1.434%	1.044%	1.001%
Recall	98.247%	99.286%	99.063%
F-Score	98.174%	99.190%	99.096%
TNR	98.566%	98.956%	98.999%
FNR	1.318%	0.823%	1.077%

The classifiers built by this method are compared with those built by other methods, as shown in Table 3. It can be seen that the accuracy of classifiers built by this method is at a high level,

Table 1. Initial parameters on gas dataset

	BPNN	XGBoost	SVM
Maximum evolution times	30	30	30
Number of individuals in the initial population	300	100	100
Chromosome length	58	20	22
Individual proportions of various population	3:3:4	3:3:4	3:3:4
Number of individuals provided by REZ	30	10	10
Initial crossover rate	1.0/0.9/0.8	1.0/0.9/0.7	1.0/0.8/0.6
Initial mutation rate	0.8/0.7/0.6	0.7/0.6/0.5	0.7/0.6/0.5
Initial temperature	1.0	1.0	1.0
Cooling rate	0.002	0.002	0.002

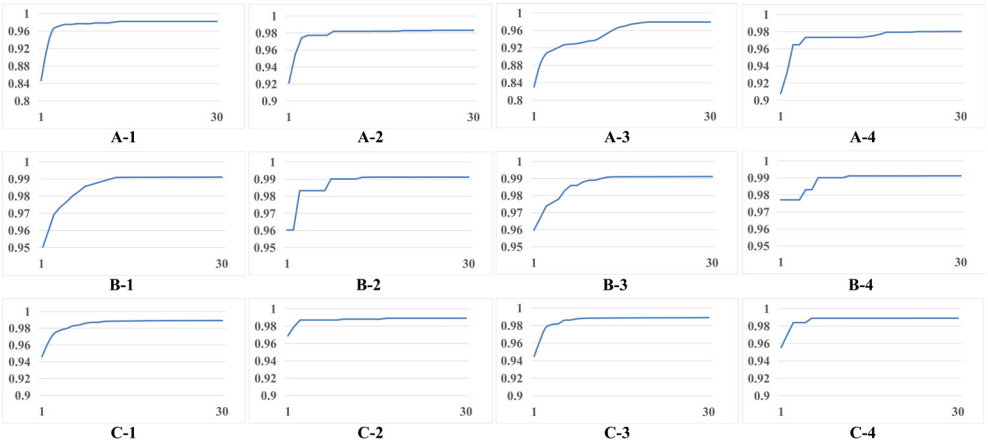

Figure 6. Experimental diagrams on gas

Table 3. Comparison with other classifiers on gas dataset

	Classifiers	AUC	ACC	DR	FNR	F-Score
Gas Dataset	SVM	/	/	90.6%	8.70%	/
	GA-SV	/	/	88.3%	5.98%	/
	Naive Bayes	/	/	89.0%	8.50%	/
	KPSO-SVM	/	/	94.3%	2.74%	/
	MIKPSO-SVM-3I	/	/	95.4%	1.74%	/
	IMPGA-BPNN	**98.407%**	**98.430%**	**98.247%**	**1.434%**	**0.9817**
	IMPGA-XGBoost	**99.121%**	**99.133%**	**99.286%**	**0.823%**	**0.9920**
	IMPGA-SVM	**99.031%**	**99.033%**	**99.063%**	**1.077%**	**0.9910**

and the detection rate is also improved when the AUC is improved.

4.3 Experimentation on CICIDS2017 Dataset

The CICIDS 2017 dataset generated by the Canadian Institute for Network Security in 2017 contains benign and up-to-date common attacks (Sharafaldin et al., 2017). It satisfies 11 indispensable features of an effective IDS dataset: anonymity, attack diversity, full capture, full interaction, complete network configuration, available protocols, full traffic, feature set, metadata, heterogeneity, and tagging (Gharib et al., 2016). It also includes the results of network traffic analysis using CICFlowMeter, using markup streams based on time stamps, source and target IP, source and target ports, protocols and attacks (CSV files) (Roopak et al., 2019). These characteristics enable data to behave as closely as possible to real-world data (Chiba et al., 2019).

Similar to subsection 4.2, the parameters of this method, which is the same as the one from Table 2, need to be set first.

The maximum AUC and the average AUC curves of the individuals in each generation were plotted, as shown in Figure 7. Group D is the curve of BPNN classifier, group E is the curve of XGBoost classifier, and group F is the curve of SVM classifier. The numbers 1, 2, 3, 4 have the same meaning as those from subsection 4.2.

The results are shown in Table 4. The AUCs of the three classifiers are 98.30%, 99.95% and 98.65% respectively, and other indexes such as ACC are also at an excellent level.

Table 4. Experimental result on CICIDS2017 dataset

	BPNN	XGBoost	SVM
AUC	98.297%	99.947%	98.645%
ACC	98.295%	99.950%	98.591%
DR	98.310%	99.931%	99.081%
FAR	1.716%	0.036%	1.791%
Recall	98.310%	99.931%	99.081%
F-Score	98.018%	99.942%	98.402%
TNR	98.284%	99.964%	98.209%
FNR	1.275%	0.053%	0.724%

The classifiers constructed by this method are compared with those constructed by other

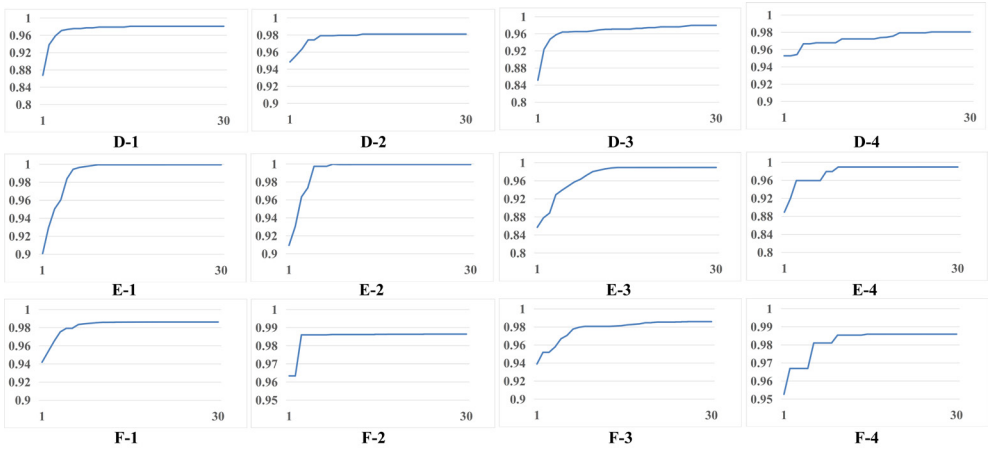

Figure 7. Experimental diagrams on CICIDS2017

Table 5. Comparison with other classifiers on CICIDS2017 dataset

	Classifiers	AUC	ACC	DR	FNR	F-Score
CICIDS2017 Dataset	DNN with SVM&Clustering	/	92.03%	91.35%	/	/
	Clustering&GB	84.25%	/	83.886%	/	0.84
	Multi-Agent System	/	79.8%	72.03%	12.43%	/
	GA and Fuzzy Logic	96.53%	87.97%	76.50%	0.56%	0.8484
	CS-PSO	75.51%	78.14%	59.16%	2.87%	0.73
	IMPGA-BPNN	**98.297%**	**98.295%**	**98.310%**	**1.275%**	**0.9802**
	IMPGA-XGBoost	**99.974%**	**99.950%**	**99.931%**	**0.036%**	**0.9994**
	IMPGA-SVM	**98.645%**	**98.591%**	**99.081%**	**0.724%**	**0.9840**

methods, as shown in Table 5. From the table, it can be seen that the AUC of the classifier constructed by this method is improved by 10-20%, and the detection rate is maintained at more than 98%.

4.4 Experimental Analysis

- Analysis of different experimental results

In the experiment with BPNN, searching appropriate hyper-parameters is difficult, so it is hard to initialize them to a high level, which cause low maximum AUC and average AUC. Through evolution, AUC increases rapidly in the first six generations, then increases at a slower rate, and finally the accuracy is over 98%.

In the experiment with XGBoost, the initial maximum AUC and average AUC are high, and the accuracy can reach more than 99%. This is because XGBoost has few hyper-parameters, and decision trees are more suitable for industrial control datasets. Although the initial fitness is at a high level, we can see that this method can also improve the accuracy of the classifier, which proves the universality of this method.

In the experiment with SVM, both the initial maximum AUC and average AUC are in the upper middle level. Due to the penalty coefficient and gamma that have a significant impact on AUC, SVM finally converges after twelve generations.

- Analysis of the function of the algorithm based on the experimental results

The comparison and analysis show that the performance of the intrusion detection system constructed by this method is better than the performance of other methods. The multiple populations can converge towards the optimal solution more effectively. The combination of the two selection strategies retains the advantages of the two strategies and makes the convergence rate moderate, but there is still a small probability of falling into the local optimal solution. In order to solve this problem, SAA is introduced. In the early stage of evolution, selection, crossover and mutation are carried out at a higher rate, which creates more individuals and improves the probability of producing a better solution. In the later stage, the overall performance is high, so the rates of selection, crossover and mutation need to be slowed down. The directed evolution operator evolves slowly in the high fitness population, so the evolution is promoted under the condition that individuals with high performance are not destroyed.

Experiments on different classifiers and different datasets show that this method is universal. Whether applied in BPNN which has more hyper-parameters and lower initial fitness values, or in XGBoost and in SVM which have fewer hyper-parameters and higher initial fitness values, this method can improve fitness by evolving. It is also shown that considering the features of previous classifiers is significant when a classifier is structured by this method. By analyzing and setting the approximate initial value, the applicability of this method can be effectively improved.

4.5 Safety Prediction of an Oil Depot

After the experiment on public datasets, the research on the self-built oil depot dataset is also carried out. The intrusion detection system of an oil depot is shown in Figure 8.

The classifier constructed by this method is used as the edge analysis and edge optimization module to predict the attack samples that are in the oil depot dataset. Then the system saves the results to the

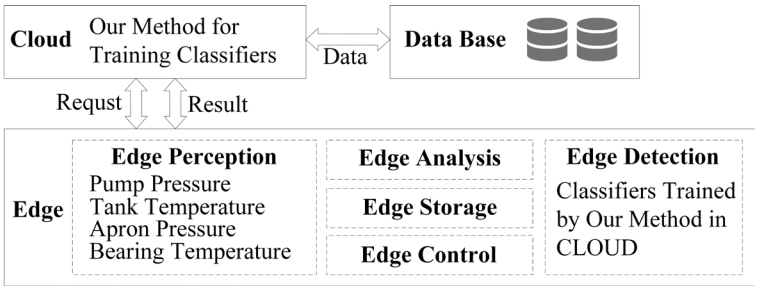

Figure 8. Control system structure of an oil depot

upper computer to realize the "real-time" feature of edge calculation (Mansouri & Babar, 2021). It is very effective and necessary for time-sensitive and precision-sensitive industrial control systems.

Edge is defined as any computing and network resource node between data source and cloud data center (Zhao, 2020). The process of extending computing services from centralized cloud-based paradigms to network edges is called edge computing (Mansouri & Babar, 2021). In theory, edge computing should be analyzed and processed near the data source (Shi et al., 2016). Devices generate and collect large amounts of data that are initially preprocessed and analyzed at the edge of the network. Then, different methods can be used to transmit these data to the centralized cloud to extract depth knowledge (Taneja et al., 2020).

There are 131 attributes of oil depot data, including inlet pressure of oil pump, outlet pressure of oil pump, liquid position in tank, liquid pressure in tank, tank temperature, pressure differential of filter, etc. The attack types are set to seven, as shown in Table 6.

Table 6. Description of seven abnormal types

	Attack Types
1	Tamper PID integral coefficient, resulting in the abnormality of the main oil pipeline.
2	Change the automatic adjustment of PID coefficient to manual adjustment.
3	Frequently switch the manual mode and the automatic mode of the PID.
4	Tamper with the actual value of pipe pressure.
5	When there is no oil, the main pump is forced to start.
6	Tamper frequency of frequency pump.
7	Tamper with the flow of the recovered oil.

The two-months information set was used in this experiment. There were about 59500 samples, including 41400 positive samples and 18100 negative samples. The results are shown in Table 7.

Table 7. Attack prediction of an oil depot

	BPNN	XGBoost	SVM
AUC	90.99%	97.54%	95.80%
ACC	90.64%	97.80%	95.81%
DR	86.36%	95.36%	96.01%
FAR	4.37%	0.26%	4.41%
F-Score	90.85%	97.44%	96.11%
TNR	95.63%	99.74%	95.60%

The experimental results of a certain oil depot show that the hyper-parameters searched by this method can be applied to all kinds of classifiers and obtain good classification results. AUC of BPNN and SVM are 90.99% and 95.80% respectively. The result of XGBoost classifier is the best, and its AUC reaches 97.54%.

5. Conclusion

This paper proposes IMPGA to search the hyper-parameters of intrusion detection classifier. Based on three datasets (gas dataset, CICIDS2017 dataset and self-built dataset), this paper studies and analyses the classification performance of three classifiers (BPNN, XGBoost and SVM). Experiments on public datasets show that this method solves the problem of premature convergence of multiple population genetic algorithms and single genotypes in the process of evolution by exchanging individuals among populations. SAA is used to further search for the global optimal solution, and an accurate industrial control intrusion classifier can be constructed to classify and predict the attack samples. The experimental results of a self-built dataset for industrial control of an oil depot show that the hyper-parameters searched by this method can be adapted to various classifiers, and the classifiers built by this method can obtain good results. It has a good application value and application prospects in the field of industrial control. AUC of self-built datasets is slightly lower than that of

public datasets. The reason may be that there is a large number of features in the self-built dataset. Therefore, feature extraction algorithm could be used to improve the performance of the classifier in the data preprocessing.

Acknowledgements

This research reported in this paper is supported by CNAF KJ2019003, BIPTACF-008, National Key R&D Program 2018YFC0824801.

REFERENCES

Albashish, D., Hammouri, A. I., Braik, M., Atwan, J. & Sahran, S. (2021). Binary biogeography-based optimization based SVM-RFE for feature selection, *Applied Soft Computing, 101*, 107026.

Bayar, H., Terzi, U. K. & Ozgonenel, O. (2019). PCA-ANN Based Algorithm for the Determination of Asymmetrical Network Failures of Network-Connected Induction Generators, *Tehnički Vjesnik-Technical Gazette, 26*(4), 953-959. DOI: 10.17559/TV-20171204220620

Chen, X. D., Yu, X. X., Chi, S. S., Wang, T. & Chen, W. W. (2020). Parameters Inversion of Probability Integral Method Based on Multi-population Genetic Algorithm, *Safety in Coal Mines, 51*(11), 50-54+60.

Chiba, Z., Abghour, N., Moussaid, K., Omri, A. E. & Rida, M. (2019). Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms, *Computers & Security, 86*(3), 291-317. DOI: 10.1016/j.cose.2019.06.013

Eesa, A. S., Orman, Z. & Brifcani, A. M. A. (2015). A new feature selection model based on ID3 and bees algorithm for intrusion detection system, *Turkish Journal of Electrical Engineering & Computer Sciences, 23*(2), 615-622.

Gao, W. (2013). *Cyberthreats, attacks and intrusion detection in supervisory control and data acquisition networks*. Dissertations & Theses - Gradworks.

Gharib, A., Sharafaldin, I., Lashkari, A. H. & Ghorbani, A. A. (2016). An evaluation framework for intrusion detection dataset. In *2016 International Conference on Information Science and Security (ICISS),* (pp. 1-6). IEEE. DOI: 10.1109/ICISSEC.2016.7885840

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addisin-Westly Publishing Company, Inc.

Han, H., Xu, L., Cui, X.Y. & Fan, Y. Q. (2021). Novel chiller fault diagnosis using deep neural network (DNN) with simulated annealing (SA), *International Journal of Refrigeration, 121*, 269-278.

He, G. Q., Li, B. C. & Wang, D. X. (2020). Research on Vehicle Routing Problem Based on Improved Double-population Hybrid Genetic Algorithm, *Supply Chain Management, 1*(07), 108-118.

Holland, J. (1973). Genetic Algorithms and the Optimal Allocation of Trials, *SIAM Journal on Computing, 2*(2), 88-105.

Jin, Y. P. (2016). Research on Mathematical Model of Genetic Algorithm based on Schema Theorem, *Journal of Mudanjiang Normal University (Natural Sciences Edition)*, *04*, 16-17.

Kashef, R. (2021). A boosted SVM classifier trained by incremental learning and decremental unlearning approach, *Expert Systems with Applications, 167*, 114154.

Li, Y. F. (2018). Research on Production Line Balance Problem of L Company Based on Double Population Genetic Algorithm, *Value Engineering, 37*(33), 272-273.

Li, Y. M., Xu, Y. Y., Liu, Z., Hou, H. X., Zheng, Y. S., Xin, Y., Zhao Y. F. & Cui, L. Z. (2020). Robust detection for network intrusion of industrial IoT based on multi-CNN fusion, *Measurement*, *154*(2), 107450.

Ma, H. X., Wang, D. H. & Luo, W. L. (2020). PID Parameter Optimization of Hydraulic System Based on Multi-population Genetic Algorithm, *Packaging Engineering, 41*(23), 204-210.

Mangano, S. (1995). Genetic algorithms solve seemingly intractable problems, *Computer Design, 34*(5), 70-74.

Mansouri, Y. & Babar, M. A. (2021). A review of edge computing: Features and resource virtualization, *Journal of Parallel and Distributed Computing, 150*, 155-183.

Metropolis, N. (2004). Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics, 21*, 1087-1092.

Morris, T. & Gao, W. (2014) Industrial Control System Traffic Data Sets for Intrusion Detection Research. In: Butts, J. & Shenoi, S. (eds.), *Critical Infrastructure Protection VIII. International Conference on Critical Infrastructure Protection ICCIP 2014. IFIP Advances in Information and Communication Technology, 441* (pp. 65-76). Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-662-45355-1_5

Ni, S. P., Qi, H. T. & Li, H. F. (2021). Hybrid Algorithm Based on Multiple Population Genetic and Mind Evolution, *Computer Engineering,* 1-9. DOI: 10.19678/j.issn.1000-3428.0060297

Priya, R. M. S., Maddikunta, P. K. R., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L. & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture, *Computer Communications, 160*, 139-149.

Roopak, M., Tian, G. Y. & Chambers, J. (2019). Deep learning models for cyber security in IoT networks. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, (pp. 0452-0457).

Serban, F. M., Grozav, S., Ceclan, V. & Turcu, A. (2020). Artificial Neural Networks Model for Springback Prediction in the Bending Operations, *Tehnički Vjesnik-Technical Gazette, 27*(3), 868-873. DOI: 10.17559/TV-20141209182117

Sharafaldin, I., Gharib, A., Lashkari, A. H. & Ghorbani, A. A. (2017). Towards a Reliable Intrusion Detection Benchmark Dataset, *Software Networking, 1,* 177-200.

Shi, W.S., Cao, J., Zhang, Q. & Xu, L. (2016). Edge Computing: Vision and Challenges, *IEEE Internet of Things Journal, 3*(5), 637-646.

Taneja, M., Byabazaire, J., Jalodia, N., Davyab, A., Olariuc, C. & Malonea, P. (2020). Machine learning based fog computing assisted data-driven approach for early lameness detection in dairy cattle, *Computers and Electronics in Agriculture, 171, 105286.*

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons Inc.

Wang, N. & Cao, C. W. (2020). A Dynamic Multi-output Prediction Model of the Hydrogen Network in a Real-World Refinery Based on XGBoostModel, *Journal of East China University of Science and Technology, 46*(1), 77-83.

Wang, X., Wang, L., Wang S. Y., Chen, J. F. & Wu, C.G. (2021). An XGBoost-enhanced fast constructive algorithm for food delivery route planning problem, *Computers & Industrial Engineering, 152*, 107029.

Zhang, Y. W., Feng, B., Chen, Y., Liao, W. H. & Guo, C. X. (2021). Fault diagnosis method for oil-immersed transformer based on XGBoost optimized by genetic algorithm, *Electric Power Automation Equipment, 41*(02), 200-206.

Zhao, M. (2020). Survey on Technology and Application of Edge Computing, *Computer Science, 47*(S1), 268-272+282.

Zhou, W. W., Chen, M. Y., Yang, Z. L. & Song, X. B. (2020). Real Estate Risk Measurement and Early Warning Based on PSO-SVM, *Socio-Economic Planning Sciences,77*(5), 101001.