jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2019, VOL. 65, NO. 4, PP. 657-663
Manuscript received March 15, 2019; revised September, 2019. DOI: 10.24425/ijet.2019.129825

Smart Substation Network Fault Classification Based on a Hybrid Optimization Algorithm

Xin Xia, Xiaofeng Liu, and Jichao Lou

***Abstract*—Accurate network fault diagnosis in smart substations is key to strengthening grid security. To solve fault classification problems and enhance classification accuracy, we propose a hybrid optimization algorithm consisting of three parts: anti-noise processing (ANP), an improved separation interval method (ISIM), and a genetic algorithm-particle swarm optimization (GA-PSO) method. ANP cleans out the outliers and noise in the dataset. ISIM uses a support vector machine (SVM) architecture to optimize SVM kernel parameters. Finally, we propose the GA-PSO algorithm, which combines the advantages of both genetic and particle swarm optimization algorithms to optimize the penalty parameter. The experimental results show that our proposed hybrid optimization algorithm enhances the classification accuracy of smart substation network faults and shows stronger performance compared with existing methods.**

***Keywords*—Smart substation, Network fault classification, improved separation interval method (ISIM), Support vector machine (SVM), Anti-noise processing (ANP)**

I. Introduction

With the increasing coverage of smart substations within the power grid, protecting them from intrusion and failure is critical to power grid safety [1–4]. Among protective measures, the first concern is network security, with fault diagnosis in the smart substation network being an important part of its overall security [5–8]. When a network fault occurs in a smart substation, the data acquisition equipment in the power system uploads a large amount of collected data to the dispatching end at the fastest rate possible for analysis. However, many of these uploaded fault messages have intricately related connections that make it difficult to detect the type of fault. Therefore, it is necessary to use the most suitable fault classification algorithm to classify the collected data accurately for the best fault diagnosis.

Many researchers have explored network security problems in smart substations and have proposed several classification algorithms for diagnosing smart substation network faults more effectively. Some approaches combine neural networks with fault diagnosis to make full use of situational awareness and autonomous learning to classify network faults in smart substations accurately and efficiently [9–12], but the performance is limited when additional noise data are present. Solutions using Bayesian theory find the connection between the cause of a fault and its manifestation, applying machine learning to the manifestation to determine the corresponding cause and obtaining good results [13–18]. However, these algorithms require relatively higher independence of the feature attributes of samples, that is, having very low relevance. In practice, this requirement is often not met, and as the degree of attribute relevance improves, the accuracy of the classification declines greatly. Others have combined Bayesian algorithms and neural networks, but these solutions have their limitations and deficiencies [19, 20].

In this paper, we propose some algorithms to improve the performance of smart substation fault classification. The contributions we make in this paper are as follows.

1. Anti-noise processing. Because outliers and noise appear in much of the sample data, we propose a new method for removing the noise samples from datasets to improve classification.
2. A new method of optimizing kernel parameters. We present our improved separation interval method (ISIM) method to improve classification by taking sample data into account.
3. A new strategy for optimizing penalty parameters. We find that, among heuristic algorithms, the genetic algorithm (GA) [21, 22] and particle swarm optimization (PSO) [23, 24, 25] have their respective advantages and disadvantages in classifying smart substation faults. Therefore, we propose our GA-PSO algorithm using the ISIM method to combine the advantages of both methods for enhancing classification accuracy.

We organize the remainder of our paper as follows. Section 2 describes related work. Section 3 presents our method for improving classification accuracy. Section 4 provides experimental evaluations of the proposed algorithm and compares our algorithm with other methods. Finally, section 5 presents our conclusions.

II. Related Work

In this paper, a support vector machine (SVM) is used to investigate smart substation network fault classification [26, 27, 28]. Since the introduction of SVM in recent years, many new algorithms have been developed using it, and they improve performance in areas such as convergence rate and generalization ability. However, these new algorithms also have their shortcomings. For example, to cope with the considerable noise in the collected data, Lin et al. proposed the fuzzy SVM (FSVM) [29]. The algorithm combines fuzzy math with an SVM to separate noise and outliers from valid samples. In practical applications, researchers have made some

This work was supported by the scientific research project of Zhejiang Provincial Department of Education (project number: Y201738745), the National Natural Science Foundation of China (project number: 61772386).

First Author is with Department of Information Technology, Wenzhou Vocational & Technical College, also with College of Computer Information Technology, Wuhan Institute of Shipbuilding Technology. (e-mail: xinxinfavor@gmail.com).

Second Author is with Department of Information Technology, Wenzhou Vocational & Technical College. (e-mail: timshaw9791@gmail.com).

Third Author is with School of Computer Science, Wuhan University. (e-mail: jichaolou@whu.edu.cn).

corresponding improvements to the algorithm [30, 31], but many problems remain. For example, if there is a considerable amount of abnormal data or abnormal data with a certain distribution, FSVM loses information when separating the abnormal data. FSVM also requires expensive calculations in the kernel function, a large amount of memory, and significant training time. The Lagrangian SVM cannot handle large sample data in nonlinear problems [32, 33]. The granular SVM performs well on a uniformly distributed dataset, but actual data tend to be unevenly distributed, limiting the performance of the algorithm [34, 35]. Our approach also models the network fault information and optimizes the related algorithms to improve classification accuracy, but it differs from the ones described in this section.

III. METHOD

A. Preliminaries

1) Data normalization

In actual environments, as the amount of network fault data collected and the data size increase, outliers must be detected and considered. We apply a combination technique to the classification problem to make the influence of previous processes observable by later processes. The final result is a weighted combination analysis algorithm. However, for some independent combinations, we use different algorithm values in different parts of the dataset. We combine these different results to detect outliers.

Data normalization is an indispensable part of training an SVM. In the sampled data, the difference in the range of numerical values is very large. For features with such a large range, we can ignore the fractional (decimal) portion of the values as the large range already affects the classifier to a much greater extent than features with a smaller range of values.

We map a given attribute to the range [0, 1] using the normalization formula

$$x_m = \frac{x - x_{min}}{xmin_{max} \times (high - low) + low} \tag{1}$$

where x is the value before the feature value is processed, $x_{\min}$ is the minimum of all of the original features, and $x_{\max}$ is the maximum of all of the original features. *high* and *low* are the maximum and minimum values of the mapping interval, respectively.

2) Anti-noise processing

The widely used SVM classification technique performs the task satisfactorily when no noise is present but performs less well with noise in the dataset, producing different results. Moreover, a given sample may differ significantly from normal data and have a greater similarity to abnormal data. Noise has characteristics indicating that they are equivalent to discrete points. Therefore, admitting a noise sample into the final calculation can make a significant difference between the computed result and the actual value, leading to serious errors in the classifier. Achieving accurate classification requires us to preprocess the existing training samples to remove the noise samples from the initial training sample set. This noise filtering greatly improves classification accuracy.

To make the classifier more robust and less sensitive to noise performance, we propose a scheme to enhance these characteristics greatly. Prior to using the dataset to train the SVM, we remove the outliers using high-dimensional spatial denoising to complete the denoising process.

We introduce some definitions to assist in describing our model. We let *o*, *p*, and *q* denote samples in the sample set *S* and $d(p,q)$ denote the distance between samples *p* and *q*.

- ***Definition* 1.** The k-*dist(p)* is the value of $d(p,o)$ meeting the following requirements: at least k $o' \in S$ samples satisfy $d(p,o') \le d(p,o)$ and at most *(k-1)* $o' \in S$ samples satisfy $d(p,o') < d(p,o)$.
- ***Definition* 2.** $N_k(p)$ is a set of samples that meet the following requirements: the distance between sample *p* and the sample belongs to dataset *S* is less than *k-dist(p)* and $N_k(p) = \{q \in S \setminus \{p\}, d(p,q) \le k - dist(p)\}$.
- ***Definition* 3.** The local density of sample *p* is the reciprocal of the mean value of its k-*dist(p)*, $den_k(p) = 1/avg(k - dist(q) \mid q \in N_k(p))$. The outlier coefficient of sample *p* is $LOF_k(p) = avg\{den_k(q) \mid q \in N_k(p)\} / den_k(p)$ and reflects the discretization between the nearest *k* points.

To separate noise from the samples, we calculate $LOF_k(p)$ for each sample *p*. If $LOF_k(p)$ is higher than a certain threshold value, $LOF_k(p)$ is an outlier (i.e., *p* is a sample that produces noise in the classification), and we should remove it from the dataset. In this way, we obtain better classification precision by training the SVM with the noise-filtered dataset.

B. SVM model optimization

1) Kernel function selection

Given the nonlinearity of the sample data, we need to introduce a kernel function to map the original nonlinear samples to the high-dimensional feature space, so that the samples are linearly separable in the new space. Then, we can use the classification theory of linear samples to solve such problems. Different kernel functions apply to different sample data. Different kernel functions and parameters produce different effects even for the same sample data. Therefore, we should select appropriate kernel parameters to solve the applicable calculations. Commonly used kernel functions include linear kernel functions, polynomial kernel functions, Gauss radial basis kernel functions, and sigmoid kernel functions. We choose the Gauss radial basis kernel function because it has a single parameter σ and can handle the relationship between attribute and category well. It is also superior to several other kernel functions in performance [36, 37].

2) Kernel parameter optimization

After selecting the kernel function, we must select appropriate kernel parameters. For the Gaussian kernel parameter σ used in our kernel, experimental data show that, if the distance between σ and the sample point is very small, $\sigma \to 0$. Conversely, if the distance between σ and the sample point is large, $\sigma \to \infty$. When σ is very small, the discriminant function obtained by the Gauss kernel function SVM is almost a constant, which leads to overfitting and a reduction in the classification accuracy rate. A large value of σ leads to

reduced classification accuracy as well. Therefore, finding optimal parameter values is necessary for the best classification performance. The traditional separation interval method (SIM) aids in selecting the kernel parameter by taking the smallest distance from the same sample data to the center point of its category. We use two sample sets: $X_1 = \{(x_i, y_i) \mid y_i = 1\}$ and $X_2 = \{(x_i, y_i) \mid y_i = -1\}$. The data quantities are n_1 and n_2, and the respective center points of the sample sets are q_1 and q_2. From this we obtain

$$q_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} x_i \ , \ q_2 = \frac{1}{n_2}\sum_{i=1}^{n_2} x_i \tag{2}$$

We set the kernel function to be $K(x_i, x_j) = <\phi(x_i), \phi(x_j)>$ with kernel parameter σ. After the kernel function maps the selected samples from the lower dimensional to the higher dimensional space, the distance between the center points q_1 and q_2 is

$$Q = \|q_1 - q_2\| = \sqrt{\left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_i) \right\|^2} \tag{3}$$

Then, we express the optimization kernel parameter as

$$\max_{\sigma}\left(Q(\sigma)\right) = \max_{\sigma}\sqrt{\left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_i) \right\|^2} \tag{4}$$

This method only needs to solve for the maximum value in Formula (4) to obtain the value of the kernel parameter, making it easy to implement and fast in theory. However, for a relatively scattered sample set, when solving the maximum distance between the center points, some sample data will be ignored, which is not convincing and does not classify accurately. Given this situation, this paper proposes the ISIM.

In each type of dataset, a distinct feature is always present: the sample data belonging to the same category are always close to each other, and the distribution is relatively aggregated. ISIM first solves their center points q_i according to the sample data of each category and then solves the sum of the distances of different types of sample data to the center points of other categories. As a simple example, consider the use of two categories. For the low-dimensional space, there are two different categories of nonlinear sample sets:

$$\begin{aligned} X_1 &= \{(x_i, y_i) | y_i = 1\}, i = 1, 2, \ldots, n_1 \\ X_2 &= \{(x_i, y_i) | y_i = -1\}, i = 1, 2, \ldots, n_2 \end{aligned} \tag{5}$$

In the above formula, n_1 and n_2 are used to indicate the number of the samples that two categories of data sets contain, respectively, and y_i represents the category of the sample data. If two data belong to the same category, then their y values are equal. Conversely, if they do not belong to the same category, then their y values are not equal.

ISIM calculates the centre points of two different categories of data based on the sample set:

$$Q_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} x_i \ , Q_2 = \frac{1}{n_2}\sum_{i=1}^{n_2} x_i \tag{6}$$

The average distance of the data in the category of X_1 to Q_2 is

$$X_{12} = \frac{1}{n_1}\sum_{i=1}^{n_1}\|x_i - Q_2\| = \frac{1}{n_1}\sum_{i=1}^{n_1}\sqrt{\left\| x_i - \frac{1}{n_2}\sum_{i=1}^{n_2} x_i \right\|^2} \tag{7}$$

Similarly, the calculation of the average distance from the data in the X_2 category to Q_1 is

$$X_{21} = \frac{1}{n_2}\sum_{i=1}^{n_2}\|x_i - Q_1\| = \frac{1}{n_2}\sum_{i=1}^{n_2}\sqrt{\left\| x_i - \frac{1}{n_1}\sum_{i=1}^{n_1} x_i \right\|^2} \tag{8}$$

Then, we use the ISIM selection method to determine the kernel parameter σ:

$$\max(\sigma) = \max\left(X_{12} + X_{21}\right) \tag{9}$$

For the Gaussian kernel function used in this paper, $K(x_i, x_j) = \exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2})$, the above sample set is mapped to a higher dimensional space. The mapping is represented by ϕ. Thus, after the mapping, we can convert Formulae (6)–(8) into the following expressions:

$$Q_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_i) \ , \ Q_2 = \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_i) \tag{10}$$

$$\begin{aligned} X_{12} &= \frac{1}{n_1}\sum_{i=1}^{n_1}\|\phi(x_i) - Q_2\| \\ &= \frac{1}{n_1}\sum_{i=1}^{n_1}\sqrt{\| \phi(x_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_i)\|^2} \\ &= \frac{1}{n_1}\sum_{i=1}^{n_1}\sqrt{k(x_i, x_i) + \frac{1}{n_2^2}\sum_{j=1}^{n_2}\sum_{k=1}^{n_2}k(x_j, x_k) - \frac{1}{n_2}\sum_{j=1}^{n_2}k(x_i, x_j)} \\ &= \frac{1}{n_1}\sum_{i=1}^{n_1}\sqrt{1 + \frac{1}{n_2^2}\sum_{j=1}^{n_2}\sum_{k=1}^{n_2}\exp\left(\frac{-\|x_j - x_k\|^2}{2\sigma^2}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2}\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)} \end{aligned} \tag{11}$$

$$X_{21} = \frac{1}{n_2}\sum_{i=1}^{n_2}\|\phi(x_i) - Q_1\| \tag{12}$$

$$\begin{aligned} &= \frac{1}{n_2}\sum_{i=1}^{n_2}\sqrt{\| \phi(x_i) - \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_i)\|^2} \\ &= \frac{1}{n_2}\sum_{i=1}^{n_2}\sqrt{k(x_i, x_i) + \frac{1}{n_1^2}\sum_{j=1}^{n_1}\sum_{k=1}^{n_1}k(x_j, x_k) - \frac{1}{n_1}\sum_{j=1}^{n_1}k(x_i, x_j)} \\ &= \frac{1}{n_2}\sum_{i=1}^{n_2}\sqrt{1 + \frac{1}{n_1^2}\sum_{j=1}^{n_1}\sum_{k=1}^{n_1}\exp\left(\frac{-\|x_j - x_k\|^2}{2\sigma^2}\right) - \frac{1}{n_1}\sum_{j=1}^{n_1}\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)} \end{aligned}$$

After mapping to a higher dimensional space, the full expansion of Formula (9) is

$$\begin{aligned} \max(\sigma) = \max\Bigg(&\frac{1}{n_1}\sum_{i=1}^{n_1}\sqrt{1 + \frac{1}{n_2^2}\sum_{j=1}^{n_2}\sum_{k=1}^{n_2}\exp\left(\frac{-\|x_j - x_k\|^2}{2\sigma^2}\right) - \frac{1}{n_2}\sum_{j=1}^{n_2}\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)} \\ &+ \frac{1}{n_2}\sum_{i=1}^{n_2}\sqrt{1 + \frac{1}{n_1^2}\sum_{j=1}^{n_1}\sum_{k=1}^{n_1}\exp\left(\frac{-\|x_j - x_k\|^2}{2\sigma^2}\right) - \frac{1}{n_1}\sum_{j=1}^{n_1}\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)}\Bigg) \end{aligned} \tag{13}$$

The preceding describes our optimization method for the kernel parameters. The following steps give a specific process.

- *Step 1*: First, obtain the sample dataset and incorporate the sample data in each category into the formulae above to obtain their actual expressions.
- *Step 2*: Select a range of values (f_1, f_2) for the kernel parameter σ and set the condition value for the end of the optimization to $e = 10^{-3}$.
- *Step 3*: Solve for the values $f_3 = (f_1 + f_2)/2$, $\max(f_1)$, and $\max(f_2)$.
- *Step 4*: If $\max(f_1) > \max(f_2)$, set $f_2 = f_3$. If $\max(f_1) < \max(f_2)$, set $f_1 = f_3$.
- *Step 5*: If $|\max(f_1) - \max(f_2)| \le e$, the optimal value is $(f_1 + f_2)/2$, and the entire optimization process ends. Otherwise, repeat *Step 3* and following.

3) Penalty parameter optimization

The penalty parameter *C* is another important factor affecting the performance of SVM algorithms by balancing error and risk. This parameter adjusts the ratio of the confidence range to the empirical risk of the SVM model, improving the SVM's generalization ability. Once the value of *C* is too small, there is a smaller empirical error, and the obtained error becomes greater, increasing the empirical risk value of the SVM, resulting in an "under-learning" condition. If *C* is too large, the accuracy of the model improves at the expense of its generalization ability, and the "over-learning" condition occurs. In addition, reasonable values of *C* leading to better processing of outliers in the sample help keeping the model in a stable state. Therefore, we need to optimize the penalty parameter selection.

We have introduced the GA into the SVM to optimize the parameter set. However, the results show that the convergence rate is slow and the results are not very satisfactory. GAs are general-purpose algorithms that can solve many problems, but the results obtained are not optimal. We also introduce the PSO into the SVM to optimize the parameter set. Our research found that the convergence rate is very fast but with unsatisfactory accuracy. The PSO method slowly loses its diversity as the number of iterations increases. This easily leads to the rapid convergence of the population. However, this produces only a local optimal solution. Given the characteristics of the two algorithms, we propose the GA-PSO algorithm, which introduces genetic operations into the PSO to optimize the penalty parameter C. Although many experts and scholars have spent a considerable amount of research on GA-PSO, our approach is different from theirs [38, 39, 40].

Since we want the maximum classification accuracy, we solve the fitness function in the algorithm, $fit(f(C,\sigma)) = f(C,\sigma)$. The specific algorithm steps are as follows.

- *Step 1*: Set the particle swarm size and then initialize the position and velocity of each particle. Set the variable $i = 1$. Set the initial value of n as the number of evolutional generations.
- *Step 2*: Apply the SVM to each particle, calculating the fitness value fit_i ($i=1,2,\ldots, k$) of each particle in the population and the average fitness value fit_v of the particle swarm by using the classification accuracy of the fivefold cross-validation as the fitness value of the particle.
- *Step 3*: Sort the fitness values of each particle from largest to smallest. Update the current particle and population extreme values based on the current fitness value of the particle and the reserved respective remaining extreme values. According to the fitness order, divide the particle swarm into two parts $\{A\}$ and $\{B\}$. If $fit_i \ge fit_v$, then $P_i \in \{A\}$, where P_i represents the *i*th particle. If $fit_i < fit_v$, then $P_i \in \{B\}$.
- *Step 4*: Perform the crossing and mutation operation for particles in A ($P_i \in \{A\}$) to generate another new particle swarm *C*. Replace swarm *B* with swarm *C*. Reconstruct a new population combined with swarm *A*.
- *Step 5*: Compare the current fitness value of each particle with the best value retained by the particle. If the former is better than the latter, the current position of the particle is set to the best position p_{best} that the particle has experienced; the current fitness value of each particle is compared with the best value reserved by the population, and if the former is better than the latter, then the current position of the particle is set to the best position g_{best} experienced by the population.
- *Step 6*: Increment *i*. When $i=n$ or the optimum fitness value increment is less than a given threshold, break out of the loop. Otherwise, return to *step 3*.
- *Step 7*: Obtain the optimal parameter and use it to calculate classification accuracy.

IV. Experiment

A. Experiment design

We implemented our approach using the open source package LIBSVM and MATLAB. LIBSVM is a simple, easy to use, fast, and efficient SVM pattern recognition and regression software package. The software not only provides compiled executable files for Windows-based systems but also offers source code for easy improvement, modification, and use on other operating systems. The LIB-SVM algorithm in the package implements basic SVM functionality. Our code was written in MATLAB and integrates with LIBSVM. We performed the following steps in our experiment.

- *Step 1*: Preprocess the fault data, deleting records with missing values.
- *Step 2*: Normalize the data as described in Section 3.1.1.
- *Step 3*: Convert the data to the format required by LIBSVM.
- *Step 4*: Perform fault classification using optimized methods.
  - ① Filter noise and optimize the sample set using the anti-noise processing (ANP) method.
  - ② Select the Gaussian radial basis kernel function.
  - ③ Optimize kernel parameters with the ISIM method.
  - ④ Optimize the penalty parameter with the GA-PSO algorithm.
  - ⑤ Calculate the final classification result.

B. Experimental results and analysis

1) Anti-noise processing experiment

According to the sample optimization strategy, we calculated $LOF_k(p)$ for each sample *p* in the dataset *S* and removed noise data according to a corresponding threshold.

Because larger datasets have a greater probability of noise, we selected 800 samples from the dataset collected by the smart substation. We used x samples as the training set and the remainder as the test set. We compared the noise reduction results between our proposed anti-noise processing (ANP)-SVM and the regular LIB-SVM methods. Table I shows the experimental results.

TABLE I
CLASSIFICATION ACCURACY OF ANP-SVM AND LIB-SVM

x value	LIB-SVM classification accuracy (%)	ANP-SVM classification accuracy (%)
100	64.2	66.8
200	74.8	76.9
300	77.8	80.8
400	83	86.1
500	87.9	91.5
600	90.1	93.2

Our ANP-SVM achieved better classification accuracy than LIB-SVM, so we decided to use ANP-SVM in all subsequent experiments.

2) The effect of the parameter σ on performance

We looked for values of the kernel parameter σ in the range [0,100], using ISIM to find the optimal value according to Formula (9). We then conducted an experiment to verify that our improved kernel parameter selection method was better. In the experiment, we set penalty parameter C to 100 (constant) and used both C and σ in the SVM. We used the de-noised sample set from the first experiment and measured the classification performance in terms of accuracy. We selected 800 samples from the dataset collected by the substation, using x samples for training and the remainder for testing as before. Table II shows the results from using the ANP-SVM algorithm for x=600 and compares the performance before and after the improvement of the kernel parameter σ.

TABLE II
EXPERIMENTAL RESULTS OF KERNEL PARAMETER OPTIMIZATION

Parameters	Values
Number of training samples	600
Number of test samples	200
σ before improvement	65.1
σ after improvement	63.7
Classification accuracy before improvement (%)	93.2
Classification accuracy after improvement (%)	94.8

This experiment shows that optimizing σ alone improved our classification accuracy while holding other parameters constant. For further comparison, we used 10%, 20%, 30%, 40%, 50%, 60%, and 70% of the sample dataset as training samples. For each training sample set, we calculated the classification accuracy before and after optimization, as shown in Fig. 1. The classification accuracy was better when the kernel parameter σ was optimized. Therefore, we confirm that our ISIM method improves the accuracy of smart substation network fault classification. This stage also prepared for the following GA-PSO experiment.

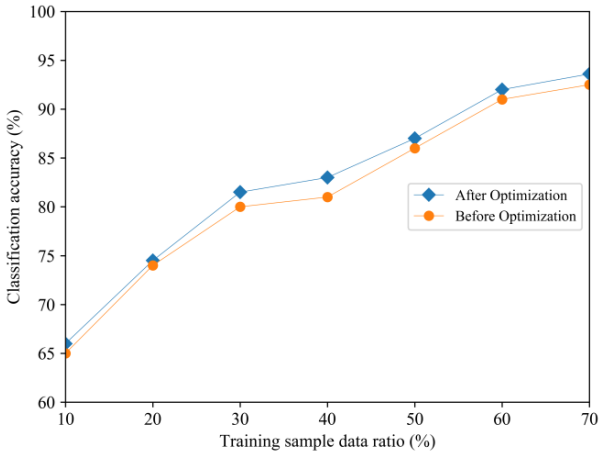

Fig. 1. Classification results for kernel parameter optimization

3) Optimization of penalty parameter C

In this experiment, we searched for the optimal value of the penalty parameter C in the range [0,200]. We set the value of the two learning factors c_1 and c_2 to 1.5 and 1.7, respectively, as these are common values and would have no effect on the comparison results of the experiments in this section. We used a population size of 30 and a generation limit of 100. We used 600 samples as the training dataset and the remaining 200 as the test dataset. The optimized kernel parameter σ together with the specified penalty parameter C (C,σ) obtained from the previous experiment is set to the initial value of g_{best} in GA-PSO algorithm that we proposed. However, GA and PSO need to calculate their respective classification accuracies without the ISIM optimization. After repeated training, our GA-PSO algorithm determined the optimal penalty parameter C and the corresponding σ. Table III shows the results of the classification prediction.

TABLE III
OPTIMIZATION RESULTS OF CLASSIFICATION PREDICTION

Algorithms	**(C,σ)**	optimization time (s)	classification accuracy
GA	(98.6,78.5)	529	95.2%
PSO	(129.7,70.2)	463	93.9%
GA-PSO	(116.2,64.8)	458	97.8%

Table III shows that when the classification performance was optimal, not only the C value was optimized but also the σ value. The experimental data from the GA algorithm show that the algorithm had better performance optimization but with slower convergence speed. The experimental data from the PSO algorithm show that PSO took less time than GA but with slightly reduced classification accuracy. The data from our GA-PSO method show that it had the shortest optimization time and the best resulting classification accuracy. Notice that the optimization time taken by GA-PSO consists of the time for ANP denoising, ISIM kernel parameter optimization, and GA-PSO for penalty parameter optimization. Since the initial value of g_{best} was optimized by the ISIM method, the optimization time of the penalty parameter was greatly reduced. Therefore, the hybrid optimization algorithm we have proposed consists of three parts: ANP, ISIM, and GA-PSO.

To further compare and the performance of this hybrid algorithm with others, we also ran tests using 10%, 20%, 30%, 40%, 50%, 60%, and 70% of the input as the training set to see the effects on classification accuracy. Fig.2 shows the results. The results demonstrate that our hybrid optimization algorithm significantly outperforms existing methods. Therefore, we believe that the hybrid algorithm offers the best fault classification accuracy.

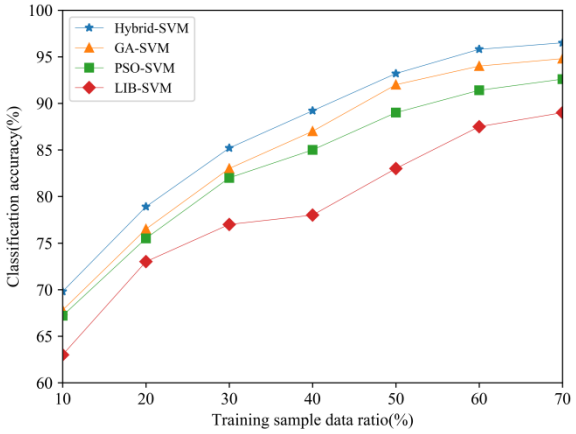

Fig. 2. Comparison of classification accuracy

V. Conclusions

In this paper, we have studied the characteristics and related technologies of network fault diagnosis in smart substations. We have preprocessed and applied ANP to the data to remove noise and increase classification accuracy. From there we investigated kernel parameter optimization with our ISIM classification method to further improve classification. Finally, we used our improved GA-PSO to enhance performance further. The experimental results show that the classification algorithm proposed in this paper is advantageous and suited to smart substation network fault classification.

References

[1] A. Belaout, F. Krim, and A. Mellit, "Multiclass adaptive neuro-fuzzy classifier and feature selection techniques for photovoltaic array fault detection and classification," *Renewable Energy*, vol. 127, pp. 548-558, 2018.

[2] A. Chattopadhyay, A. Ukil, and D. Jap, "Toward Threat of Implementation Attacks on Substation Security: Case Study on Fault Detection and Isolation," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2442-2443, 2018.

[3] W. Chen, M. Nguyen, and W. Chiu, "Optimization of the plastic injection molding process using the Taguchi method, RSM, and hybrid GA-PSO," *International Journal of Advanced Manufacturing Technology*, vol. 83, no. 9, pp. 1873-1886, 2016.

[4] L. Dalton, and E. Dougherty, "Optimal classifiers with minimum expected error within a Bayesian framework," *Pattern Recognition*, vol. 46, no. 5, pp. 1288-1300, 2013.

[5] U. Carmo, D. Sadok, and J. Kelner, "IEC 61850 traffic analysis in electrical automation networks," *IEEE International Conference on Smart Grid Communications*, Miami, USA, pp. 466-471, 2015.

[6] A. Elgargouri, R. Virrankoski, and Elmusrati M., "IEC 61850 Based Smart Grid Security," *IEEE International Conference on Industrial Technology*, Seville, Spain, pp. 2461-2465, 2015.

[7] H. Meizi *et al.*, "Research on Intelligent Diagnosis Strategy of Secondary Device Abnormity in Smart Substation," *2018 2nd International Conference on Power and Energy Engineering*, Xiamen, China, 2018.

[8] J. Hong, C. Liu, and M. Govindarasu, "Integrated Anomaly Detection for Cyber Security of the Substations," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1643-1654, 2014.

[9] C. Hou, F. Nie, and C. Zhang, "Multiple rank multi-linear SVM for matrix data classification," *Pattern Recognition*, vol. 47, no. 1, pp. 454-469, 2014.

[10] K. Qader and M. Adda, "Fault Classification System for Computer Networks Using Fuzzy Probabilistic Neural Network Classifier," Communications in Computer and Information Science, vol. 459, , pp. 217-226, 2014.

[11] O. Khaled, A. Marín, and F. Almenares, "Analysis of Secure TCP/IP Profile in 61850 Based Substation Automation System for Smart Grids," *International Journal of Distributed Sensor Network*, vol. 2016, no. 2, pp. 1-11, 2016.

[12] K. Knuth, M. Habeck, and N. Malakar, "Bayesian evidence and model selection," *Digital Signal Processing*, vol. 47, pp. 50-67, 2015.

[13] O. Kocadağlı, "A novel hybrid learning algorithm for full Bayesian approach of artificial neural networks," *Applied Soft Computing*, vol. 35, pp. 52-65, 2015.

[14] M. Kordestani, M. Samadi, and M. Saif, "A New Fault Diagnosis of Multifunctional Spoiler System Using Integrated Artificial Neural Network and Discrete Wavelet Transform Methods," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 4990-5001, 2018.

[15] C. Lin, and S. Wang, "An intelligent lung cancer diagnosis system using cuckoo search optimization and support vector machine classifier," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 1, pp. 267-293, 2019.

[16] Y. Lin, F. Lv, and S. Zhu, "Large-scale image classification: Fast feature extraction and SVM training," *IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, USA, pp. 1689-1696, 2011.

[17] H. Livani, and C. Evrenosoğlu, "A fault classification method in power systems using DWT and SVM classifier," *IEEE/PES Transmission and Distribution Conference and Exposition*, Orlando, FL, USA, pp. 1-5, 2012.

[18] D. Lu, and W. Qiao, "A GA-SVM hybrid classifier for multiclass fault identification of drivetrain gearboxes," *IEEE Energy Conversion Congress and Exposition*, Pittsburgh, PA, USA, pp. 3894-3900, 2014.

[19] D. Mack, G. Biswas, and X. Koutsoukos, "Learning Bayesian Network Structures to Augment Aircraft Diagnostic Reference Models," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 358-369, 2017.

[20] M. TanveerD., "Newton method for implicit Lagrangian twin support vector m achines," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 6, pp. 1029-1040, 2015.

[21] H. Faris *et al.*, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks," *Information Fusion* , vol. 48, pp. 67-83, 2019.

[22] T. RenatoR *et al*, "NK Hybrid Genetic Algorithm for Clustering," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 748-761, 2018.

[23] H. Md Alamgir *et al*, "Modified PSO algorithm for real-time energy managem ent in grid-connected microgrids", vol. 136, pp. 746-757, 2019.

[24] Y. Qian, H. Zhang, and D. Peng, "Fault diagnosis for generator unit based on RBF neural network optimized by GA-PSO," *International Conference on Natural Computation Natural Computation*, Chongqing, China, pp. 233-236, 2012.

[25] N. Raj, G. Jagadanand, and S. George, "Fault detection and diagnosis in asymmetric multilevel inverter using artificial neural network," *International Journal of Electronics*, vol. 105, no. 4, pp. 559-571, 2018.

[26] A. Hadbah, A. Kalam, and A. Zayegh, "Powerful IEDs, ethernet networks and their effects on IEC 61850-based electric power utilities security," *2017 Australasian Universities Power Engineering Conference*, Melbourne, Australia,, pp. 1-5, 2017.

[27] F. Saraiva, W. Bernardes, and E. Asada, "A framework for classification of non-linear loads in smart grids using Artificial Neural Networks and Multi-Agent Systems," *Neurocomputing*, vol. 170, pp. 328-338, 2015.

[28] J. Steckenrider, and T. Furukawa, "Detection and classification of stochastic features using a multi-Bayesian approach," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Daegu, South Korea, pp. 150-155, 2017.

[29] F. Sun, N. Wang, and J. He, "Lamb Wave Damage Quantification Using GA-Based LS-SVM," *Materials*, vol. 10, no. 6, pp. 1-21, 2017.

[30] K. Topuz, F. Zengul, and A. Dag, "Predicting graft survival among kidney transplant recipients: A Bayesian decision support model," *Decision Support Systems*, vol. 106, pp. 97-109, 2018.

[31] A. Valdes, R. Macwan, and M. Backes, "Anomaly Detection in Electrical Substation Circuits via Unsupervised Machine Learning," *IEEE 17th International Conference on Information Reuse and Integration*, Pittsburgh, PA, USA, pp. 500-505, 2016.

[32] S. Valuvanathorn, S. Nitsuwat, and M. Huang, "Multi-feature face recognition based on PSO-SVM," *10th International Conference on ICT and Knowledge Engineering*, Bangkok, Thailand, pp. 140-145, 2012.

[33] M. Wang, and X. Li, "Power Cable Fault Recognition Using the Improved PSO-SVM Algorithm," *Applied Mechanics and Materials*, vol. 427, pp. 830-833, 2013.

[34] W. Wang, Z. Xu, and W. Lu, "Determination of the spread parameter in the Gaussian kernel for classification and regression," *Neurocomputing*, vol. 55, no. 3, pp. 643-663, 2003.

[35] M. Panda, A. Abraham, and BK. Tripathy," Soft granular computing based classification using hybrid fuzzy-KNN-SVM," *Intelligent Decision Technologies*, vol. 10, no. 2, pp. 115-128, 2016.

[36] Y. Wang, and M. Simaan, "A Suction Detection System for Rotary Blood Pumps Based on the Lagrangian Support Vector Machine Algorithm," *IEEE Journal of Biomedical & Health Informatics*, vol. 17, no. 3, pp. 654-663, 2013.

[37] Z. Xu, M. Dai, and D. Meng, "Fast and Efficient Strategies for Model Selection of Gaussian Support Vector Machine," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 5, pp. 1292-1307, 2009.

[38] H. Yoo, and T. Shon, "Novel Approach for Detecting Network Anomalies for Substation Automation based on IEC 61850," *Multimedia Tools & Applications*, vol. 74, no. 1, pp. 303-318, 2015.

[39] H. Zheng, Y. Zhang, and J. Liu, "A novel model based on wavelet LS-SVM integrated improved PSO algorithm for forecasting of dissolved gas contents in power transformers," *Electric Power Systems Research*, vol. 155, pp. 196-205, 2018.

[40] T. Zhou, S. Popescu, and A. Lawing, "Bayesian and Classical Machine Learning Methods: A Comparison for Tree Species Classification with LiDAR Waveform Signatures," *Remote Sensing*, vol. 10, no. 1, pp. 1-27, 2018.