Hindawi
Mobile Information Systems
Volume 2021, Article ID 6839179, 12 pages
https://doi.org/10.1155/2021/6839179

Research Article

Extracting Function-Driven Tracing Characteristics for Optimized SVM Classification

Ming Wan,[1] **Xinlu Xu**,[1] **Yan Song**,[2] **Quanliang Li**,[1] **and Jiawei Li**[1]

[1]*School of Information, Liaoning University, Shenyang 110036, China*
[2]*School of Physics, Liaoning University, Shenyang 110036, China*

Correspondence should be addressed to Yan Song; song.yan@lnu.edu.cn

Received 10 November 2021; Revised 8 December 2021; Accepted 11 December 2021; Published 26 December 2021

Academic Editor: Jianhui Lv

Due to its openness and simplicity, Modbus TCP has wide applications to facilitate the actual management and control in industrial wireless fields. However, its potential security vulnerabilities can also create lots of complicated information security challenges, which are increasingly threatening the availability of industrial real-time traffic delivery. Although anomaly detection has been recognized as a workable security measure to identify attacks, the critical step to successfully extract data characteristics is an extremely difficult task. In this paper, we focus on the continuous control mode in industrial processes and propose a control tracing feature algorithm to extract the function-driven tracing characteristics from Modbus TCP data traffic. Furthermore, this algorithm can flexibly integrate the time factor with critical functional operations and adequately describe the dynamic control change of technological processes. To closely cooperate with this algorithm, one optimized SVM (support vector machine) classifier is introduced as the practicable decision engine. By designing one applicable attack mode, we develop an in-depth and meticulous analysis on the decision accuracy, and all experimental results clearly explain that the extracted features can strongly reflect the changing pattern of continuous functional operations, and the proposed algorithm can effectively cooperate with the optimized SVM classifier to distinguish abnormal Modbus TCP data traffic.

1. Introduction

Modbus TCP, which is regarded as one representative industrial communication protocol, has been widely applied in various critical infrastructures, including power generation, steel rolling, oil refinery, gas purification, and so on. In particular, with the rapid development of IIoT (industrial internet of things), many industrial wireless fields have collaborated Modbus TCP with other wireless communication protocols to perform various control, management, and monitoring activities of mobile devices or endpoints [1–3]. As a typical case, one wireless gateway can successfully apply Modbus TCP to realize industrial real-time traffic delivery for mobile sensors or actuators. Generally, Modbus TCP is used to accomplish information interactions between different field devices and facilitates the actual management and control in industrial processes [4, 5]. Furthermore, Modbus TCP defines the master/slave communication style, which exploits the request/response way to achieve the data exchange between master and slave stations [6]. For example, when one master station sends one read or write request message, the slave station builds a response message to perform different functional operations. Additionally, one slave station cannot actively send any request to one master station, and all slave stations are forbidden to communicate with each other. According to its protocol specification, Modbus TCP belongs to one real-time communication protocol implemented on the application layer, and its essence is to embed the Modbus frame into the data field of TCP protocol. To be more specific, function code is an important protocol field in the basic Modbus frame structure, which not only can represent the main function of the request message but also can clearly point out the specific type of functional operation to control field devices [7, 8]. In other words, all key functional operations in the whole industrial process can be determined by a range of different function codes, and this design can improve the efficiency of industrial production by simplifying process control and

management. However, it can also be exploited by malicious adversaries to launch targeted attacks due to the potential security vulnerabilities of Modbus TCP. In traditional applications, ICSs (industrial control systems) are physically isolated, and all nonessential external accesses can be disabled. As a consequence, the original design of Modbus TCP misses the privacy and security concerns and only depends on the basic TCP security mechanism. With the tight integration between IT (information technology) and OT (operational technology), the potential security vulnerabilities of Modbus TCP are gradually exposed due to multiple external threats. Currently, many scientists have carried out tremendous researches on Modbus TCP security issues [4, 9–11], mainly including unauthenticated access, function code abuse, unencrypted data delivery, data tampering and replaying, and so on. For example, when one adversary gains access to industrial control systems based on Modbus TCP, it not only can intercept all critical device states and production data but also can insert some abnormal function codes to destroy the regular technological processes. That is to say, any adversary can falsify one normal Modbus TCP request with one malicious function code to change the basic control logic and perform disruptive operations on industrial field devices [12, 13]. To some extent, the potential security vulnerabilities of Modbus TCP have become a breakthrough to attack key control components, whose failures may cause serious production accidents.

In most security measures, anomaly detection has been widely acknowledged and researched by both academia and industry because it can effectively identify known and unknown attacks without impacting ICS's availability [14]. Especially, anomaly detection based on machine learning algorithms can enhance the popular application of artificial intelligence technologies in the domain of information security and achieve relatively good effect in security practices [15–18]. However, before applying anomaly detection, an essential prerequisite is to design an excellent feature generation and extraction algorithm that must inherit the main characteristics from the original data [15, 19]. Actually, the working quality of these algorithms can directly affect the performance of anomaly detection, and one excellent algorithm can enhance the detection efficiency to some extent. Different from the high-dimensional and complex data in traditional IT systems, industrial data traffic presents relatively simple characteristics and trends due to the periodic operating process and stable system structure [20]. In most industrial production activities, process control is an important component, which can indirectly reflect some technological process by using a series of consecutive functional operations. Correspondingly, they are the most direct connection with the changing laws of function codes in Modbus TCP data traffic [21]. From this point of view, this paper proposes a control tracing feature algorithm, which extracts function-driven tracing characteristics by analyzing the continuous control mode from Modbus TCP data traffic. Moreover, this algorithm not only takes into account the time factor caused by the time intervals between every two consecutive functional operations but also associates with the critical characteristics of sequential control predefined by the technological process. That is, this algorithm can flexibly integrate the time factor with critical functional operations and adequately describe the dynamic control change of technological process. In order to closely cooperate with this algorithm, one optimized SVM (support vector machine) classifier is introduced as the practicable decision engine, which is applied in combination with three different intelligent optimization algorithms, including PSO (particle swarm optimization), GA (genetic algorithm), and GRID. At last, we design one applicable attack mode to evaluate SVM's decision accuracy, and our main purpose includes the following two aspects: on the one hand, based on the function-driven tracing characteristics expressed from consecutive function codes, we prove that the extracted features can strongly reflect the changing pattern of continuous functional operations; on the other hand, compared with different intelligent optimization algorithms, we prove that the optimized SVM classifier can effectively cooperate with the proposed feature algorithm to distinguish abnormal Modbus TCP data traffic.

The main contributions of this paper are summarized as follows:

Firstly, according to the continuous control mode in the industrial process, we propose a novel control tracing feature algorithm, and this algorithm can effectively extract function-driven tracing characteristics, mainly including the probability characteristic and time parameter of operational event sequences.

Secondly, in order to closely cooperate with the proposed feature algorithm, we select the SVM classifier as the practicable decision engine and discuss the most appropriate optimization algorithm to strengthen SVM's decision ability.

Thirdly, in the experimental evaluation, we design one attack mode to simulate one possible attack, which fully exploits the security vulnerabilities of unauthenticated access and function code abuse. Also, we give a whole scale analysis on the effects of critical parameter in the proposed feature algorithm.

2. Function-Driven Tracing Characteristics Description and Extraction

In our control tracing feature algorithm, we extract the function-driven tracing characteristics from Modbus TCP data traffic by performing the following steps: firstly, we preprocess Modbus TCP data traffic to extract operational event sequences that consist of a series of consecutive function codes; secondly, based on PST (probabilistic suffix tree) [22, 23], we further perform the control tracing analysis in industrial processes and build a functional association tree; finally, according to the probability of control tracing and time parameter, we obtain the simplified feature samples by the vector calculation.

2.1. Operational Event Sequence Description. From the Modbus TCP data traffic over a period of time T, we extract the corresponding function code of each packet in chronological order and define an operational event sequence as

$O_i = f_1^i, f_2^i, f_3^i, \ldots, f_m^i$ that is composed of m consecutive function codes. Here, we regard a function code as an operational event, and f_j^i represents the j_{th} $(j \le m)$ operational event in the operational event sequence O_i. Additionally, all operational event sequences construct the operational event sequence set $O = \{O_1, O_2, O_3, \ldots, O_n\}$. Because each O_i $(i \in [1, n])$ is composed of m consecutive function codes, its corresponding time interval T_i $(i \in [1, n])$ also includes the time intervals of m consecutive function codes, and $T = \sum_{i=1}^{n} T_i$. Due to the disparate time intervals of any two consecutive function codes, we also have $T_i \neq T_{i+1}$ when $\forall i \in [1, n-1]$.

2.2. Functional Association Tree Construction. In general, PST can describe the probabilistic characteristics of sequence sets and present a k-ary tree whose nodes arrange regularly. Furthermore, each leaf node can denote a tracing path from the root node to this leaf node, and the L-depth tree involves L nodes, which store L symbols. As it describes, the jumping of two adjacent nodes can be expressed by the conditional probability of some prior symbols. Actually, the consecutive functional operations in industrial processes can be considered as a tracing path from the root node to some leaf node, and this tracing path is defined as the functional control tracing in this paper. Here, each leaf node involves a functional control tracing sequence, and the value of each leaf node corresponds to one probability value of its operational event sequence. Similarly, the jumping of two adjacent nodes can depict the changing of two consecutive function codes in functional operations. Figure 1 shows the construction process of the functional association tree. In this tree, the tree depth is L, which indirectly expresses L functional operations. Moreover, we also assume all control tracing sequences include v different function codes $f_a, f_b, f_c, f_d, \ldots, f_v$. The detailed process is described below:

(1) Create the tree root.

(2) According to v different function codes, create v different branches. In each branch, there is a leaf node that includes two variables, such as $\{f_b', p(f_b)\}$ of the first leaf node f_b in the figure. Here, f_b' represents the initial control tracing sequence, and $p(f_b)$ represents the corresponding probability of this sequence, namely, the jumping probability from the root node to this leaf node (indicated by the dotted line in the figure).

(3) Under every first leaf node, create v different branches. In each branch, there is a leaf node that includes two variables, such as $\{f_b' f_a', p(f_b f_a)\}$ of the second leaf node f_a under the first leaf node f_b. Here, $f_b' f_a'$ represents the control tracing sequence of this leaf node, and $p(f_b f_a)$ represents the corresponding probability of this sequence. Additionally, the jumping probability from the first leaf node f_b to this leaf node (indicated by the dotted line in the figure) can be expressed by the conditional probability $p(f_a|f_b)$.

(4) Under every second leaf node, create v different branches. In each branch, there is a leaf node that includes two variables, such as $\{f_b' f_a f_c', p(f_b f_a f_c)\}$ of the third leaf node f_c under the second leaf node f_a. Similarly, $f_b' f_a f_c'$ represents the functional control tracing sequence of this leaf node, and $p(f_b f_a f_c)$ represents the corresponding probability of this sequence. Additionally, the jumping probability from the second leaf node f_a to this leaf node (indicated by the dotted line in the figure) can be expressed by the conditional probability $p(f_c|f_b f_a)$.

(5) In a similar fashion, the L_{th} leaf node is further created. As shown in the figure, suppose the $(L-1)_{\text{th}}$ leaf node is f_d and its next leaf node is f_v. So the corresponding variables are $\{f_b' f_a f_c, \ldots, f_d f_v', p(f_b f_a f_c, \ldots, f_d f_v)\}$. Here, $f_b' f_a f_c, \ldots, f_d f_v'$ represents the control tracing sequence of this leaf node, and $p(f_b f_a f_c, \ldots, f_d f_v)$ represents the corresponding probability of this sequence. Additionally, the jumping probability from the $(L-1)_{\text{th}}$ leaf node f_d to this leaf node (indicated by the dotted line in the figure) can be expressed by the conditional probability $p(f_v|f_b f_a f_c, \ldots, f_d)$.

(6) After all L leaf nodes are created, the whole functional association tree is built successfully.

According to the above construction process, the probability of each control tracing sequence can be calculated by the conditional probability from the root node to the last leaf node.

Suppose the control tracing sequence is $s = f_1, f_2, \ldots, f_L$, then

$$p(s) = p(f_1) \times p(f_2|f_1) \times \cdots \times p(f_L|f_1 f_2, \ldots, f_{L-1}) = \prod_{j=1}^{L} p(f_j|f_1 f_2, \ldots, f_{j-1}), \quad (1)$$

where $p(f_j|f_1 f_2, \ldots, f_{j-1})$ $(j \in [1, L])$ represents the probability of the next function code f_j when the prior $j-1$ function codes are $f_1 f_2, \ldots, f_{j-1}$ in the operational event sequence.

For example, as indicated by the dotted line in Figure 1, if the control tracing sequence of L_{th} leaf node is $s_L = f_b, f_a, f_c, \ldots, f_d, f_v$; the corresponding probability of this sequence can be calculated by

$$p(s_L) = p(f_b) \times p(f_a|f_b) \times p(f_c|f_b f_a) \times \cdots \times p(f_v|f_b f_a f_c, \ldots, f_d). \quad (2)$$

3. Feature Factor Selection and Feature Value Calculation

In the feature factor selection, we cannot only consider a unilateral role of the control tracing sequence's probability. Through our careful observation, the time intervals between

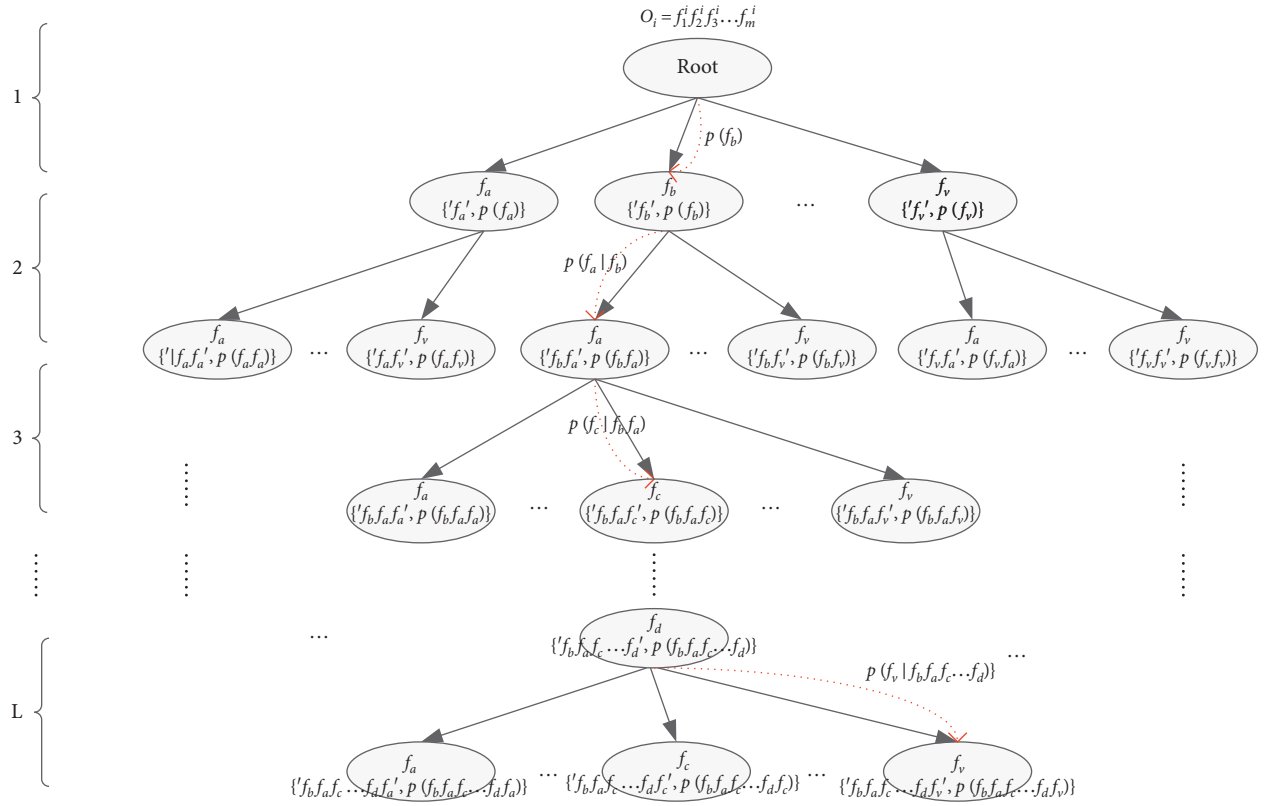

Figure 1: Construction process of functional association tree.

any two consecutive functional operations in each control tracing sequence are different. In other words, the functional operations in industrial processes are closely associated with the time parameter. In order to establish a relationship between the control tracing sequence's probability and the time parameter, we make the following setting and calculation:

According to the tracing path in the functional association tree (supposing the control tracing sequence is $s = f_1, f_2, \ldots, f_L$), we construct the probability vector of functional operation $\overrightarrow{S} = (p(f_1), p(f_1 f_2), \ldots, p(f_1 f_2, \ldots, f_L))$, which contains L different variables. Moreover, in the tracing path of s, it also takes some time intervals to implement the jumping of two adjacent leaf nodes. In this case, we can construct the corresponding the interval vector $\overrightarrow{\tau} = (\tau_1, \tau_2, \ldots, \tau_L)$. Here, $\tau_{j+1} = \sum_1^j t_j$ ($j \in [1, L-1]$), and t_j represents the time interval between two consecutive functional operations in s. From the above analysis, we regard the probability vector $\overrightarrow{S}$ and the interval vector $\overrightarrow{\tau}$ as two feature factors and calculate the feature value by the dot product operation of these two feature factors as follows:

$$x = \overrightarrow{S} \cdot \overrightarrow{\tau}. \tag{3}$$

In summary, if we suppose the operational event sequence O_i involves h different functional operations (namely, h different function codes) and the tree depth is L, we can calculate h^L feature values. That is, there are h^L feature values in each feature sample.

3.1. Dimension Reduction of Feature Sample. In the above-mentioned design, the oversize dimension of the feature sample may become a drawback, and the dimension of the feature sample keeps increasing exponentially with the growth of tree depth. For example, if we suppose one industrial process involves four different functional operations and the tree depth is 6, each feature sample can consist of $4^6 = 4096$ feature values. In practice, the oversize dimension of the feature sample may have a harmful influence on SVM's decision ability because it can cost lots of computing and storage resources. However, due to the limited behaviors and states in each industrial process, the combination number of functional operations may take on an evident downtrend. As a result, the number of different control tracing sequences in one operational event sequence set is also limited and is far less than h^L. More specifically, we can reduce the number of leaf nodes in the functional association tree and the dimension of the feature sample in the following ways: firstly, we rearrange all operational event sequences in chronological order and determine the number r of all control tracing sequences, here $r \ll h^L$; secondly, for each operational event sequence O_i, we build the corresponding functional association tree by using these r control tracing sequences; and finally, we

calculate the feature sample $X_i = (x_1^i, x_2^i, \ldots, x_r^i)$ for each operational event sequence O_i, and the dimension of each feature sample changes to r.

4. Optimized SVM Decision Engine

In the design of the decision engine, we select the classical SVM classifier to match with the above control tracing feature algorithm. Furthermore, by using the above feature samples, this decision engine can train a mathematical model to recognize the statistical deviation for the observed Modbus TCP data traffic. Simultaneously, in order to strengthen its decision ability, we use three different intelligent optimization algorithms to optimize its key parameters respectively, and our ultimate goal is to obtain one optimal decision engine that can effectively cooperate with the proposed feature algorithm.

4.1. SVM Classifier. SVM classifier [18, 24, 25], which is a representative machine learning method based on the statistical learning theory, explores structural risk minimization to improve its adaptive generation ability. Essentially, the SVM classifier belongs to the binary classification, and its core idea is to search for one optimal hyperplane that can make a big effort to separate two categories of samples. For the undivided linear sample space, the SVM classifier can use the nonlinear mapping algorithm to change the original data space to the high-dimensional feature space and further carry out the linear analysis on the nonlinear characteristics of samples in this space.

In particular, by introducing the relaxation factor and the penalty factor C, SVM is always designed to resolve the quadratic programming problem, which can be further simplified by Lagrange function into

$$\begin{cases} \max\limits_{\alpha} \quad \sum\limits_{i=1}^{n} \alpha_i - \dfrac{1}{2} \sum\limits_{j=1}^{n} \sum\limits_{i=1}^{n} \alpha_i \alpha_j y_i y_j k\left(x_i, x_j\right) \\ \text{s.t.} \quad \sum\limits_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C, i = 1, 2, \ldots, n, \end{cases} \tag{4}$$

where $k(x_i, x_j)$ is the kernel function, typically including RBF (radial basis function) kernel, polynomial kernel, linear kernel, and so on. In our decision engine, we select RBF kernel as the final kernel function because it has lower complexity to realize the nonlinear mapping by using fewer kernel parameters. RBF kernel can be calculated by

$$k\left(x, x'\right) = \exp\left(-\frac{\left\|x - x'\right\|^2}{2\sigma^2}\right), \tag{5}$$

where σ is the kernel parameter, which represents the radial action limit of this function. Finally, the decision function can be calculated by

$$\begin{cases} f(x) = \operatorname{sgn}\left(\sum\limits_{i=1}^{n} \alpha_i y_i k\left(x_i, x\right) + b\right), \\ \text{s.t. } b = y_j - \sum\limits_{i=1}^{n} y_i \alpha_i k\left(x_i, x_j\right), \quad j \in \left\{j | 0 < \alpha_j < C\right\}. \end{cases} \tag{6}$$

In practical applications, the performance of SVM classifier can be largely influenced by the penalty factor C and the kernel function's parameters, such as RBF's kernel parameter σ. Furthermore, the penalty factor can affect the margin maximization and sample misclassification, and the kernel parameter can affect the data distribution and the separated hyperplane in the high-dimensional feature space. Different from the experimental selection and settings, some intelligent optimization algorithms have been successfully applied to optimize these parameters and achieve relatively satisfying effects.

4.2. Parameter Optimization Selection. It is universally acknowledged that intelligent optimization algorithms have been widely used in all kinds of information computing fields, and they can further enhance learning efficiencies and working performances by solving single- or many-objective optimization problems [26, 27]. To effectively cooperate with the proposed feature algorithm, the SVM classifier must be optimized for the maximum performance benefit. However, the applicability of different intelligent optimization algorithms may show great variation due to their distinct generalization abilities and fitting degrees. In our decision engine, we select three representative intelligent optimization algorithms to compare different optimization effects, and these three algorithms have been shown to positively affect SVM's decision ability by many researchers. Moreover, these three algorithms are PSO [28, 29], GA [30], and GRID [31], and the training process of the SVM decision engine based on parameter optimization is presented in Figure 2. In each optimization algorithm, the fitness value can be obtained by using the current SVM's decision accuracy, and the optimal parameters can be chosen from the global optimums in all iterations. Actually, for the same training samples, each optimization algorithm can generate a group of optimal parameters, which can be further compared to obtain the optimal SVM decision engine.

5. Experimental Analysis and Discussion

In our experiments, we use the captured Modbus TCP data traffic in [32] to carry out the proposed feature algorithm and introduce the decision accuracy as one evaluation indicator to compare SVM's decision abilities under three optimization algorithms. Practically speaking, our main purpose is to explore the matching degree between the control tracing feature algorithm and SVM classifier and try to find a workable intelligent optimization algorithm to obtain the optimal SVM decision engine. Additionally, we define one applicable attack mode to facilitate the evaluation process, and this attack can continuously send a stream of malicious function codes to disturb normal functional

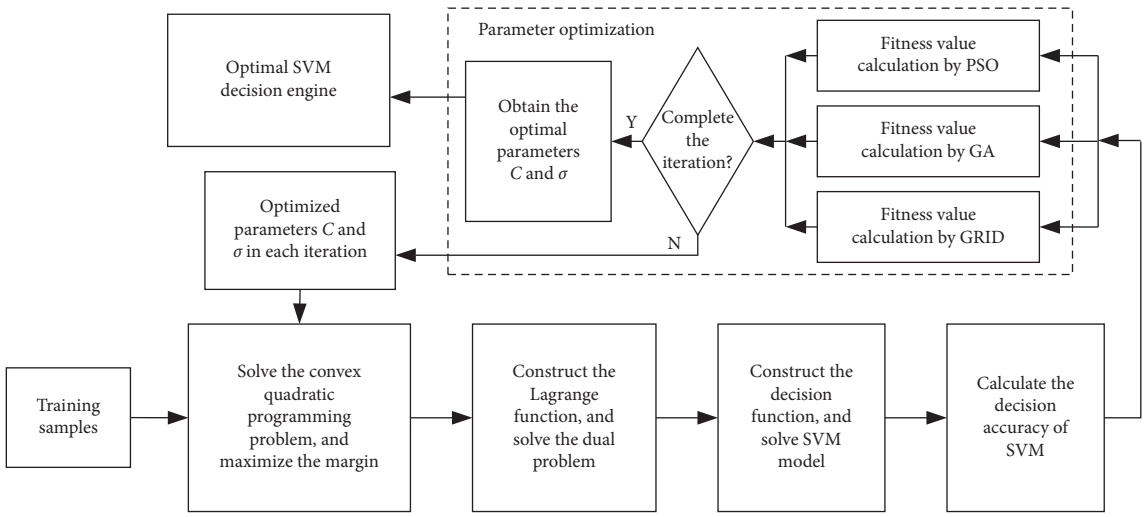

Figure 2: Training process of SVM decision engine based on parameter optimization.

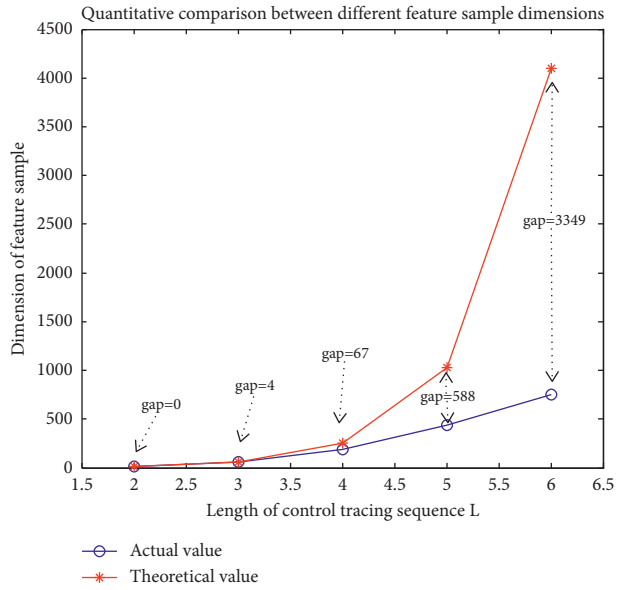

Figure 3: Compared feature sample dimensions under different lengths of control tracing sequence.

operations. In practice, this attack mode possesses a certain level of implementation possibility because it fully exploits the security vulnerabilities of unauthenticated access and function code abuse. Without loss of generality, we also define four different attack powers by changing the number of malicious function codes, whose percentages in each operational event sequence are 2.58%, 3.87%, 5.16%, and 6.45% for light-power attacks, medium-power attacks, heavy-power attacks, and fatal-power attacks, respectively. Furthermore, we discuss the change of SVM's decision abilities under different attack powers.

5.1. Certifiable Dimension Reduction. Based on the basic technological process, we define the length of one operational event sequence as 155, namely $m = 155$ consecutive function codes in each operational event sequence. After the preprocessing, we can obtain a total of 287 original operational event sequences and corresponding intervals. Similarly, we can further get the same number of normal feature samples calculated by the control tracing feature algorithm. However, an extremely important problem is that the large dimension of the feature sample may waste lots of computing resources and affect the decision ability when the SVM decision engine is applied. In theory, each feature sample may consist of h^L feature values at the extreme, and the corresponding dimension may become a very large value. Fortunately, our algorithm can reduce the dimension of the feature sample by determining the number of all the control tracing sequences in actual applications. Figure 3 gives the compared feature sample dimensions under different lengths of control tracing sequence. From this figure, we can see that with the length of the control tracing sequence increasing, the gap between the actual dimension

Table 1: Optimized results under three intelligent optimization algorithms.

	PSO	GA	GRID
C	100,000	19,271.29	24,833.50
σ	0.0478	0.0199	0.0136
Training time (s)	1,283.67	757.99	1,385.14
Training accuracy (%)	99.10	94.93	94.81

and theoretical dimension of the feature sample grows larger and larger. For instance, when the lengths of control tracing sequences are 3, 4, and 5, the corresponding gaps reach 4, 57, and 588, respectively. The above quantitative comparison fully illustrates that our algorithm can produce the desired results on dimension reduction.

5.2. Decision Results and Analysis. As discussed above, we further set the length of the control tracing sequence as 3 (namely, the tree depth $L = 3$), and the corresponding dimension of the feature sample is 60. Based on the obtained feature samples, we use the offline way to train three optimal SVM decision engines, whose main parameters are optimized by PSO, GA, and GRID. Moreover, Table 1 presents the optimized results under these three intelligent optimization algorithms. From this table, we can see that three optimization algorithms can achieve different groups of penalty factor C and kernel parameter σ due to their own optimization mechanisms, and the training results of the SVM decision engine are not identical under different key parameters. More specifically, the SVM decision engine optimized by PSO exhibits the highest training accuracy that can reach 99.10%, and this value is about 4 percentage points higher than the ones optimized by GA and GRID. Additionally, although the training time has failed to meet the expectation of real-time training and optimization, we can ignore its impact by using the way of offline training and online detection.

Based on the defined attack mode, we generate 600 test feature samples in every experiment, mainly including 200 normal feature samples and 400 malicious feature samples. Furthermore, we record the number of feature samples that are correctly classified and calculate its percentage as the decision accuracy. For each level of attack power, we conduct 10 different experiments whose malicious samples are generated by forging and inserting abnormal function codes and present the change of SVM's decision accuracies. Figure 4 compares the decision accuracies of three optimized SVM decision engines under four different attack powers, and Table 2 gives the corresponding average decision accuracies. From Table 2, we can see that three optimization algorithms can fulfill their mandates to achieve three promising SVM decision engines, whose average decision accuracies do not differ significantly under each attack power. As a whole, the SVM decision engine optimized by PSO yields the best decision ability because it has the highest decision accuracy under each attack power. More specifically, its average decision accuracies to detect light- and fatal-power attacks can reach 87.82% and 97.40%, respectively, which are 1 and 0.5 percentage points higher than the lowest one of the SVM decision engine optimized by GA. Additionally, as the attack power increases, the average decision accuracies for all optimization algorithms become larger. In other words, when the number of malicious function codes in each operational event sequence increases, the possibility to successfully detect abnormal functional operations by the optimized SVM decision engine can also be enhanced gradually. For the decision time, the average values of the SVM decision engine optimized by PSO can reach 0.01968 s, which is short enough to meet the real-time demand in industrial production. From the above experimental results and analysis, we can conclude that on the one hand, the proposed feature algorithm can adequately reflect the changing pattern of continuous functional operations, and one slight change of functional operations may bring a great difference of feature values that can be used to improve decision accuracies; on the other hand, SVM classifier can be selected as one applicable decision engine to cooperate with the proposed feature algorithm, and the PSO optimization algorithm can offer the most brilliant contribution to SVM's decision ability by comparing with other representative optimization algorithms.

In practice, the fluctuation of decision accuracy can indirectly reflect the stability of SVM decision ability because the great fluctuation of decision accuracy may cause relatively low decision accuracy in certain circumstances. For each attack power, the larger the divergence between the decision accuracy in each experiment and the average decision accuracy is, the greater the fluctuation of decision accuracy is, the poorer the stability of SVM decision ability becomes. In the related theories of probability statistics, the standard deviation, which describes the dispersion degree between individuals, is most commonly used as the measurement of statistical dispersion. Essentially, the larger standard deviation represents the greater dispersion. In our evaluation, we introduce the standard deviation to further analyze different stabilities for three optimized SVM decision engines, and Figure 5 depicts the corresponding standard deviations of decision accuracies under four different attack powers. As shown in Figures 4 and 5, with the increase of attack power, the standard deviations of decision accuracies for all optimized SVM decision engines gradually decrease. That is to say, the divergence between the decision accuracy in each experiment and the average decision accuracy becomes smaller and smaller, and the stability of each optimized SVM decision ability gets better and better. However, from the experimental results in Table 2, although the SVM decision engine optimized by PSO has the most outstanding decision ability, its standard deviation under light-power attacks is the largest, and the corresponding stability is far from satisfactory. Differently, the other SVM

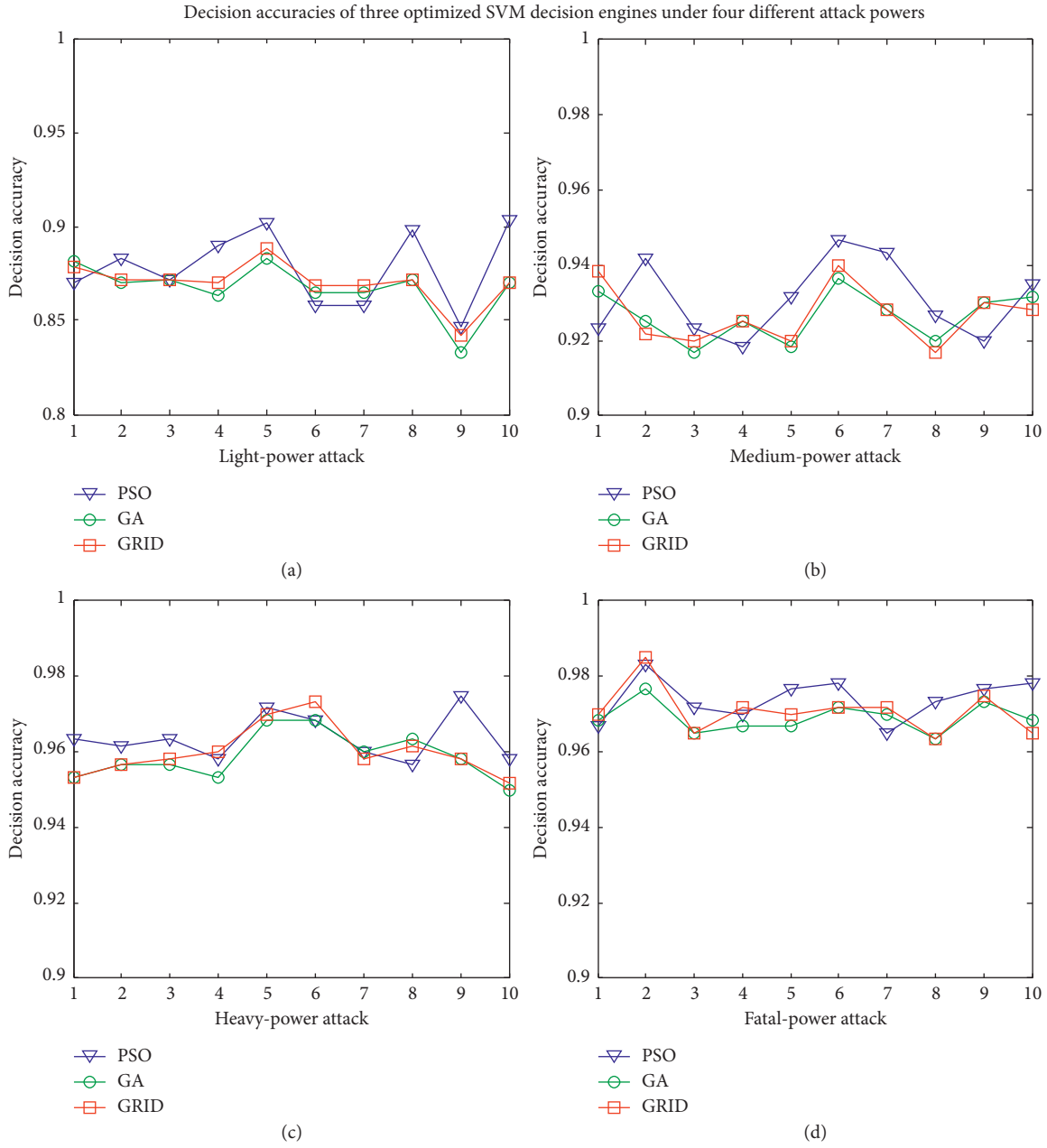

Figure 4: Decision accuracies of three optimized SVM decision engines under four different attack powers: (a) light-power attack, (b) medium-power attack, (c) heavy-power attack, and (d) fatal-power attack.

decision engines have relatively high decision stability, but their decision abilities are slightly inferior. In practical applications, we should select an appropriate intelligent optimization algorithm to optimize the SVM classifier according to the specific scenarios and requirements.

5.3. Influence of Control Tracing Sequence Length. In the proposed control tracing feature algorithm, the length L of the control tracing sequence is a preconfigured variable, which may impact the decision ability of the SVM decision engine. On the one hand, it is actually easy to understand that the decision time will undoubtedly increase with the growth of L because the dimension of the feature sample becomes higher. On the other hand, the decision accuracy can also change in different degrees when L increases or decreases. In our experiments, we select $L = 2$ to illustrate this point because the dimension of the feature sample can get greater than the length of the operational event sequence if $L > 3$. Figure 6 depicts the decision accuracy comparison of three optimized SVM decision engines under different lengths of control tracing sequence, and every average decision accuracy in this figure is also calculated by 10 different experiments. According to the direct presentation of this figure, we can draw the following conclusions: for one thing, when $L = 2$, the SVM decision

Table 2: Average decision accuracies of three optimized SVM decision engines under four different attack powers.

Attack power	Average detection accuracy (%)		
	PSO-SVM	GA-SVM	GRID-SVM
Light-power attack	87.82	86.75	87.00
Medium-power attack	93.10	92.65	92.68
Heavy-power attack	96.37	95.88	96.02
Fatal-power attack	97.40	96.90	97.08

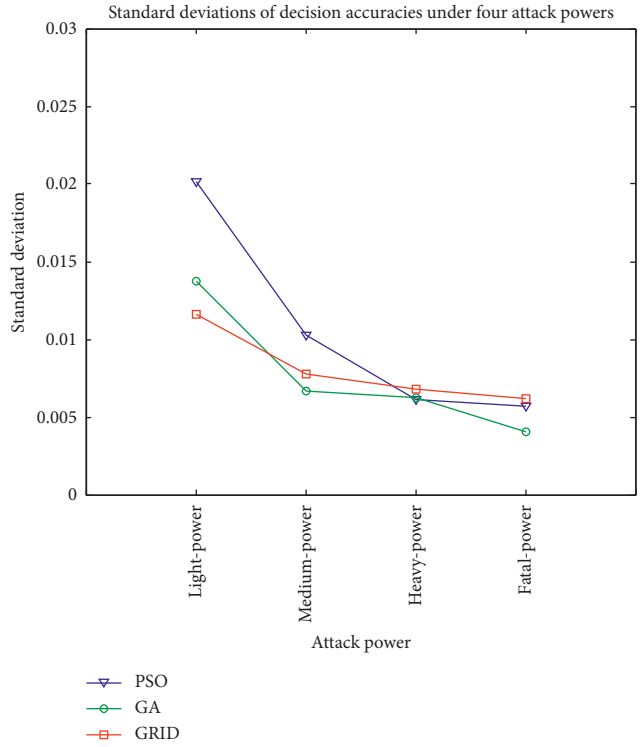

Figure 5: Standard deviations of decision accuracies for three optimized SVM decision engines under four different attack powers.

engine optimized by PSO still has the best decision ability by comparing the average decision accuracies of three optimized SVM decision engines; for another thing, all average decision accuracies of three optimized SVM decision engines are improved to one higher level when the length of control tracing sequence increases under each attack power. To sum up, if we select an SVM classifier as the applicable decision engine, we should design one appropriate L under comprehensive considerations of decision accuracy and computation efficiency.

5.4. Effect of Changed Time Parameter. In our control tracing feature algorithm, the time parameter (interval vector $\vec{\tau}$) is selected as an important feature factor to calculate the final feature value, and the change of time parameter can initiate dynamic modifications to the feature sample. Without loss of generality, when one baleful adversary wants to launch the given attack, the time interval between two consecutive functional operations will change as well. In order to illustrate the effect of the changed time parameter, we also perform 10 different experiments to compare the average decision accuracies, and the corresponding attack power belongs to the fatal-power level. In these experiments, we develop two hypothetical scenarios: one is that the given attack can generate changed time parameters, and the other is that the given attack cannot cause the change of time parameter. Additionally, we choose the SVM decision engine optimized by PSO as the experimental object because other optimized SVM decision engines will have a similar presence. Figure 7 shows its decision accuracy comparison under the changed time parameter and the unchanged time parameter. Intuitively, the average decision accuracy under the unchanged time parameter is only 95.22%, which is less than the one under the changed time parameter. According to the compared results, we recognize that the changed time parameters can improve the average decision accuracy of the optimized SVM decision engine to some degree. In other words, the introduced time parameter in our control tracing feature algorithm can provide an effective role in improving the decision ability of the SVM decision engine.

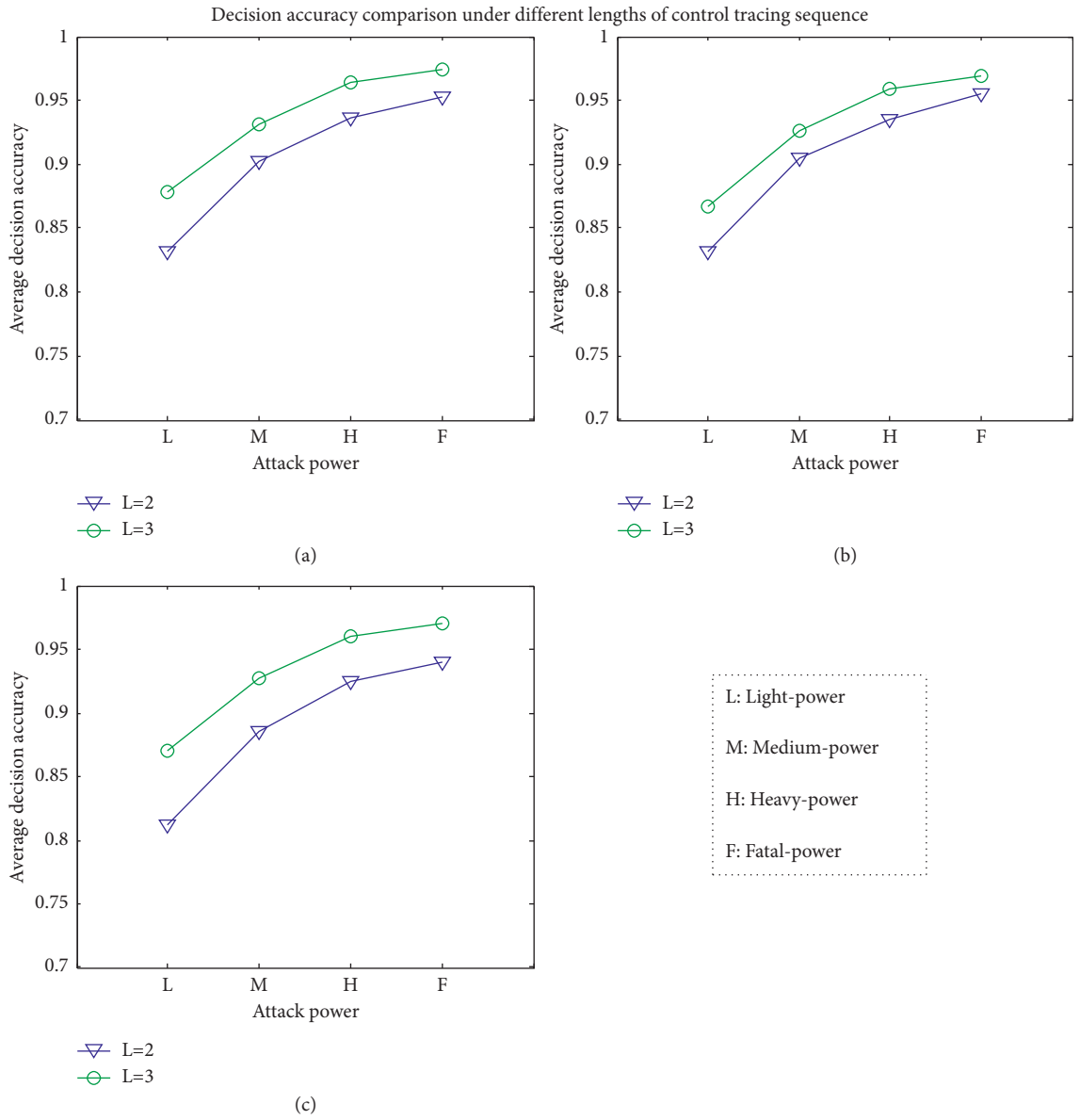

Figure 6: Decision accuracy comparison of three optimized SVM decision engines under different lengths of control tracing sequence: (a) PSO-SVM, (b) GA-SVM, and (c) GRID-SVM.

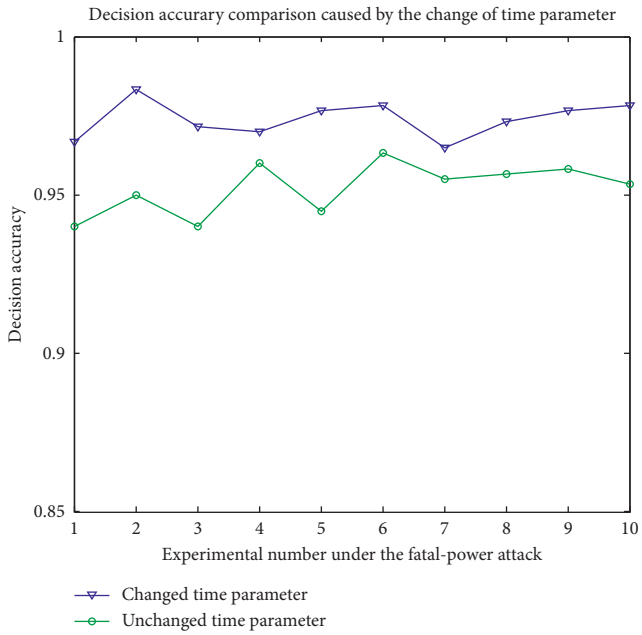

Figure 7: Decision accuracy comparison of SVM decision engine optimized by PSO under the change of time parameter.

6. Conclusions

Focusing on the continuous control mode in industrial processes, this paper proposes a control tracing feature algorithm to extract function-driven tracing characteristics from Modbus TCP data traffic. Moreover, this algorithm not only takes into account the time factor caused by the time intervals between every two consecutive functional operations but also associates with the critical characteristics of sequential control predefined by the technological process. In order to effectively cooperate with this algorithm, this paper introduces the classic SVM classifier as one practicable decision engine, and three intelligent optimization algorithms are compared to optimize its main parameters. Finally, the experimental evaluation shows that on the one hand, this paper designs one applicable attack mode to analyze SVM's decision accuracy, and the experimental results prove that the proposed feature algorithm can adequately reflect the changing pattern of continuous functional operations, and the PSO optimization algorithm can offer the most brilliant contribution to SVM's decision ability; on the other hand, this paper gives the quantitative discussion on the impacts of decision accuracy caused by the length of control tracing sequence and the time parameter, which can play a strong part in improving the decision ability of SVM decision engine.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This work was supported by the National Key R & D Plan, Research and Development of Intelligent Factory Industrial Internet System Transmission Performance Optimization Technology and Verification Platform, Ministry of Science and Technology of the People's Republic of China (Grant no. 2018YFB1700103), and the Scientific Research Project of Educational Department of Liaoning Province (Grant no. LJKZ0082).

References

[1] P. R. C. Araujo, R. H. Filho, J. J. C. Rodrigues, J. P. C. M. Oliveira, and S. A. Braga, "Middleware for integration of legacy electrical equipment into smart grid infrastructure using wireless sensor networks," *International Journal of Communication Systems*, vol. 31, no. 1, pp. 1–15, 2018.

[2] S. K. Datta, C. Bonnet, and N. Nikaein, "An IoT gateway centric architecture to provide novel M2M services," in *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 514–519, Seoul, Republic of Korea, March 2014.

[3] S. Dodla, L. Mahendra, K. Jaganmohan, R. K. Senthil Kumar, and B. S. Bindhumadhava, "Wireless real-time meter data acquisition system," in *Proceedings of the 2019 IEEE Region 10 Conference (TENCON)*, pp. 997–1002, Kochi, India, December 2019.

[4] A. Volkova, M. Niedermeier, R. Basmadjian, and H. de Meer, "Security challenges in control network protocols: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 619–639, 2019.

[5] Y. Si, N. Korada, R. Ayyanar, and Q. Lei, "A high performance communication architecture for a smart micro-grid testbed using customized edge intelligent devices (EIDs) with SPI and Modbus TCP/IP communication protocols," *IEEE Open Journal of Power Electronics*, vol. 2, pp. 2–17, 2021.

[6] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, "Performance evaluation and modeling of an industrial application-layer firewall," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2159–2170, 2018.

[7] O. Liu, B. Zheng, W. Sun et al., "A data-driven approach for reverse engineering electric power protocols," *Journal of Signal Processing Systems*, vol. 93, no. 7, pp. 769–777, 2021.

[8] M. Faisal, A. A. Cardenas, and A. Wool, "Modeling modbus TCP for intrusion detection," in *Proceedings of the 2016 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–5, Philadelphia, PA, USA, October 2016.

[9] Q. Li, Y. Liu, S. Meng, H. Zhang, H. Shen, and H. Long, "A dynamic taint tracking optimized fuzz testing method based on multi-modal sensor data fusion," *EURASIP Journal on Wireless Communications and Networking*, vol. 202021 pages, 2020.

[10] C. Alcaraz, G. Bernieri, F. Pascucci, J. Lopez, and R. Setola, "Covert channels-based stealth attacks in industry 4.0," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3980–3988, 2019.

[11] J. Luswata, P. Zavarsky, B. Swar, and D. Zvabva, "Analysis of SCADA security using penetration testing: a case study on modbus TCP protocol," in *Proceedings of the 2018 29th Biennial Symposium on Communications (BSC)*, pp. 1–5, Toronto, Canada, June 2018.

[12] L. Rosa, M. Freitas, S. Mazo, E. Monteiro, T. Cruz, and P. Simoes, "A comprehensive security analysis of a SCADA

protocol: from OSINT to mitigation," *IEEE Access*, vol. 7, pp. 42156–42168, 2019.

[13] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, and P. Sarigiannidis, "A unified deep learning anomaly detection and classification approach for smart grid environments," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1137–1151, 2021.

[14] M. Wan, J. Li, J. Yao, R. Wang, and H. Luo, "State-based control feature extraction for effective anomaly detection in process industries," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1415–1431, 2020.

[15] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.

[16] I. Ahmed, A. Dagnino, and Y. Ding, "Unsupervised anomaly detection based on minimum spanning tree approximated distance measures and its application to hydropower turbines," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 654–667, 2019.

[17] A. Kavousi-Fard, W. Su, and T. Jin, "A machine-learning-based cyber attack detection model for wireless sensor networks in microgrids," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 650–658, 2021.

[18] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1644–1652, 2017.

[19] W. Yan, L. K. Mestha, and M. Abbaszadeh, "Attack detection for securing cyber physical systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8471–8481, 2019.

[20] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 860–880, 2013.

[21] N. Goldenberg and A. Wool, "Accurate modeling of modbus/TCP for intrusion detection in SCADA systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 63–75, 2013.

[22] W. Li, Z. Su, R. Li, K. Zhang, and Q. Xu, "Abnormal crowd traffic detection for crowdsourced indoor positioning in heterogeneous communications networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2494–2505, 2020.

[23] B. Chikhaoui, S. Wang, T. Xiong, and H. Pigot, "Pattern-based causal relationships discovery from event sequences for modeling behavioral user profile in ubiquitous environments," *Information Sciences*, vol. 285, pp. 204–222, 2014.

[24] Q. Ma, C. Sun, B. Cui, and X. Jin, "A novel model for anomaly detection in network traffic based on kernel support vector machine," *Computers & Security*, vol. 104, no. 2, Article ID 102215, 2021.

[25] J. Alvarez Cid-Fuentes, C. Szabo, and K. Falkner, "Adaptive performance anomaly detection in distributed systems using online SVMs," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 928–941, 2020.

[26] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.

[27] L. Ma, N. Li, Y. Guo et al., "Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[28] J. Liu, D. Yang, M. Lian, and M. Li, "Research on intrusion detection based on particle swarm optimization in IoT," *IEEE Access*, vol. 9, pp. 38254–38268, 2021.

[29] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723–6742, 2021.

[30] A. Assiri, "Anomaly classification using genetic algorithm-based random forest model for network attack detection," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 767–778, 2021.

[31] P. Zhang, S. Shu, and M. Zhou, "An online fault detection model and strategies based on SVM-grid in clouds," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 445–456, 2018.

[32] M. Wan, W. Shang, and P. Zeng, "Double behavior characteristics for one-class classification anomaly detection in networked control systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3011–3023, 2017.