Computers, Materials & Continua
DOI: 10.32604/cmc.2022.030895
Article

Tech Science Press

Compared Insights on Machine-Learning Anomaly Detection for Process Control Feature

Ming Wan[1], Quanliang Li[1], Jiangyuan Yao[2,*], Yan Song[3], Yang Liu[4] and Yuxin Wan[5]

[1]School of Information, Liaoning University, Shenyang, 110036, China
[2]School of Computer Science and Technology, Hainan University, Haikou, 570228, China
[3]School of Physics, Liaoning University, Shenyang, 110036, China
[4]Shenyang Institute of Automation Chinese Academy of Sciences, Shenyang, 110016, China
[5]Department of Electrical Engineering and Information Technology, Karlsruhe Institute of Technology, Karlsruhe, 76131, Germany
*Corresponding Author: Jiangyuan Yao. Email: yaojy@hainanu.edu.cn

Received: 05 April 2022; Accepted: 09 May 2022

Abstract: Anomaly detection is becoming increasingly significant in industrial cyber security, and different machine-learning algorithms have been generally acknowledged as various effective intrusion detection engines to successfully identify cyber attacks. However, different machine-learning algorithms may exhibit their own detection effects even if they analyze the same feature samples. As a sequence, after developing one feature generation approach, the most effective and applicable detection engines should be desperately selected by comparing distinct properties of each machine-learning algorithm. Based on process control features generated by directed function transition diagrams, this paper introduces five different machine-learning algorithms as alternative detection engines to discuss their matching abilities. Furthermore, this paper not only describes some qualitative properties to compare their advantages and disadvantages, but also gives an in-depth and meticulous research on their detection accuracies and consuming time. In the verified experiments, two attack models and four different attack intensities are defined to facilitate all quantitative comparisons, and the impacts of detection accuracy caused by the feature parameter are also comparatively analyzed. All experimental results can clearly explain that SVM (Support Vector Machine) and WNN (Wavelet Neural Network) are suggested as two applicable detection engines under differing cases.

Keywords: Anomaly detection; machine-learning algorithm; process control feature; qualitative and quantitative comparisons

1 Introduction

Machine-learning algorithms, which have been considered as one applied cross-subjects research field, are currently attracting more and more attentions in both ICTs (Information Communication

Technologies) and ACTs (Automatic Control Technologies) [1,2]. On the one hand, machine-learning algorithms can acquire some new and inscrutable knowledge by using computers to simulate and perform the learning actions of real humans; on the other hand, they can also improve their own performance to acquire more detailed and precise results by analyzing the existing knowledge and experiences. Benefiting from the rapid development of computing technologies, machine-learning algorithms have greatly influenced on many emerging research fields, typically including: Big Data [3], Edge Computing [4,5], Automatic Control [6], Intelligent Optimization [7,8], etc. Especially, machine-learning algorithms play an increasingly important role in today's cyber security technologies to resolve various issues caused by malicious attacks or intrusions [9,10].

In general, anomaly detection, aiming to carry out accurate detections and real-time responses, belongs to one of the most popular applications of machine-learning algorithms. Essentially, various machine-learning algorithms are always designed and recognized as critical intrusion detection engines to identify anomalous system actions [11]. More precisely, by using some basic or training data to construct a normal mathematical model which can practically describe some behavior-specific system characteristics, machine-learning algorithms can skillfully analyze new real-time data to realize scientific predictions or judgments. In regular IT (Information Technology) systems, machine-learning anomaly detections have been extensively recognized and applied by both academia and industry. From traditional machine-learning algorithms to deep learning algorithms, the researchers have produced many ground-breaking achievements in anomaly detection researches [9,12]. In recent years, following with the networked development of ICSs (Industrial Control Systems), security incidents in various industrial environments have frequently occurred [13,14]. Different from the ones in regular IT systems [15,16], these incidents always cause immense losses of human lives and properties. Therefore, as one important branch of intrusion detection, machine-learning anomaly detections in ICSs have been developed to explore the possibility and feasibility by both academia and industry. To sum up, the main reasons include two aspects: firstly, the periodicity and finiteness of industrial actions can benefit the application of machine-learning algorithms [17]; secondly, machine-learning anomaly detections can identify some unknown attack behaviors without compromising the availability of industrial control systems [18]. In terms of different analysis methods, the existing industrial anomaly detections have employed multiple machine-learning algorithms, typically including: Support Vector Machine [19], Neural Network [20], Clustering Algorithm [21], Decision Tree [22], Hidden Markov Model [23], etc. Furthermore, all of these anomaly detections have one thing in common: by using machine-learning algorithms, they intelligently recognize the characteristic model and interaction regularity of industrial production activities, and provide the technical support for the design of powerful detection engines.

In practice, machine-learning anomaly detections in ICSs consist of two major components: feature generation approach and machine-learning detection engine. Furthermore, one fine feature generation approach can extract usable features which can maximize the characteristics of industrial original data, and one effective machine-learning detection engine can successfully identify various abnormal behaviors by analyzing the generated features. In other words, the optimal matching between feature generation approach and machine-learning detection engine should be implemented to strengthen its detection performance as much as possible [24]. However, various machine-learning algorithms may exhibit different detection effects even if they match with the same feature generation approach [25], and the main causes include the following two aspects: from the perspective of algorithm design, each machine-learning algorithm has its own complexity and uniqueness, which may directly affect the detection efficiency; from the perspective of data characteristics, each machine-learning algorithm is highly sensitive to different feature distributions and statistic characteristics, which not only have a powerful influence on the training process of detection engine, but also determine its final detection ability. Therefore, after developing one feature generation approach, some effective and applicable detection engines should be desperately selected by comparing distinct properties of each machine-learning algorithm.

In modern industry, the process control plays a critical role in realizing the automatic control for the whole production activities by designing some continuous or periodic control functions, which are always triggered by sending the corresponding control commands to key field devices [17,26]. Moreover, each control function can determine one operational action of field devices, and the whole technological process to produce certain product can cover the periodic alteration of operational actions. From another point of view, all sequential control commands extracted from industrial communication data can also has regular and periodic trends. When one sophisticated adversary launches one devastating attack whose main goal is to destroy the process control, these trends appear to be broken due to some redundant or irregular control commands [27,28]. Correspondingly, the change in these trends can offer a breakthrough to design and implement anomaly detection. On this view, this paper proposes one feature generation approach based on directed function transition diagrams, and the generated process control features can adequately describe the dynamic law and changing trend of operational actions. Moreover, this approach defines function patterns of different lengths to reflect varying degrees of operational continuity, and uses each directed function transition diagram to depict all directed transition paths of operational actions over a period of time. In order to further analyze the generated process control features, this paper introduces five different machine-learning algorithms (Support Vector Machine [19], BP Neural Network [29], Decision Tree [22], Naïve Bayes [30] and Wavelet Neural Network [31]) as alternative detection engines, and discusses their matching abilities with the proposed feature generation approach. More narrowly, this paper gives a brief description on the qualitative properties of each machine-learning algorithm, and compares their advantages and disadvantages point by point. Additionally, this paper considers the detection accuracy and consuming time as two quantifiable indicators, and uses the captured communication data from a simulated Modbus/TCP control system to evaluate different performances of five detection engines. In order to facilitate all quantitative comparisons, this paper also defines two attack models and four different attack intensities in the verified experiments, and our final purpose is to select some appropriate machine-learning algorithms as the most effective and applicable detection engines under differing cases.

The major contributions and innovations are listed as follows:

1) by analyzing the main characteristics of operational actions, this paper first presents the feature generation approach based on directed function transition diagrams, and designs the function pattern of different lengths to enhance the correlation and continuity of sequential operational actions.
2) in order to locate the optimal detection engines under differing cases, this paper introduces five different machine-learning algorithms to discuss their matching abilities, and the qualitative and quantitative comparisons are respectively performed for their superiority analysis.
3) so as to enrich the diversity of experiments, this paper not only defines two practical attack models (Targeted Continuing Attack and Blind Haphazard Attack), but also supposes four different attack intensities to evaluate the detection accuracy and consuming time of each detection engine.

2 Process Control Feature Generation Based on Directed Function Transition Diagram

In this feature generation approach, all control commands are extracted to consist of multiple function queues in chronological order. According to the basic idea of state transition diagram [32], each function queue is further used to construct one directed function transition diagram, whose states are set as different functions. Additionally, the directed transition path from one function to another can be described by the function pattern, whose length can be explicitly specified to reflect the degree of correlation and continuity between different functions. Based on the main idea of feature calculation in [33], all feature values in each function queue can be finally obtained by introducing the conditional probability to calculate the transition possibility of function pattern.

2.1 Directed Function Transition Diagram Construction

From industrial communication data over a long period of time, all control commands are successively extracted to form one whole control command sequence, which may cover many periodic processes of operational actions. Furthermore, this sequence is divided to multiple function queues $Q_i = d_i^1 d_i^2 d_i^3 \cdots d_i^n$ ($i \in [1, m]$), and each function queue is composed of n sequential control commands. Here, d_i^j represents the j_{th} ($j \leq n$) control command in the function queue Q_i, and m is the number of function queues. Additionally, all m function queues form the function queue set $Q = \{Q_1, Q_2, Q_3, \cdots, Q_m\}$, which also represents the initial data sample space.

For each function queue, one directed function transition diagram is constructed, and the main construction steps can be described as follows: firstly, the first control command d_i^1 in the function queue Q_i is selected as the source function state, which is also regarded as the initial state in the state transition diagram; secondly, if the next control command d_i^2 in the function queue Q_i is different from the last one d_i^1, one new function state needs to be created, and one directed transition path $ph^{1\to 2}$ from the last function state to this one is set up; thirdly, if the next control command d_i^2 in the function queue Q_i is the same with the last one d_i^1, no new function state needs to be created, and one directed transition path $ph^{1\to 1}$ in the current function state is set up; finally, for the following control command d_i^j ($j \in [3, n]$), the previous two steps are repeatedly executed until traversing all control commands in the function queue Q_i.

Based on the directed function transition diagram constructed by Q_i, the function pattern $F_i^L = (d_i^1, d_i^2, d_i^3, \cdots, d_i^L)$ can be further defined, and it is composed of L control commands. Moreover, each function pattern corresponds to one arbitrary combination of L control commands, and the larger pattern length L can reflect the stronger correlation and continuity of operational actions. Through this definition, each function queue Q_i can be indicated by a limited number of function patterns. In theory, when the number of different control command species is k, the number of function patterns is F_i^L, which may be a large value due to the fast exponential growth. However, the actual number of function patterns is far less than the theoretical one, and the causes mainly include two aspects: on the one hand, the limited behaviors and states in process control determine a relatively narrow range of operational actions, which also reduce the number of control command species; on the other hand, the periodic execution of the same control commands can also restrict the number of function patterns. Fig. 1 gives an example of directed function transition diagram and its function patterns.

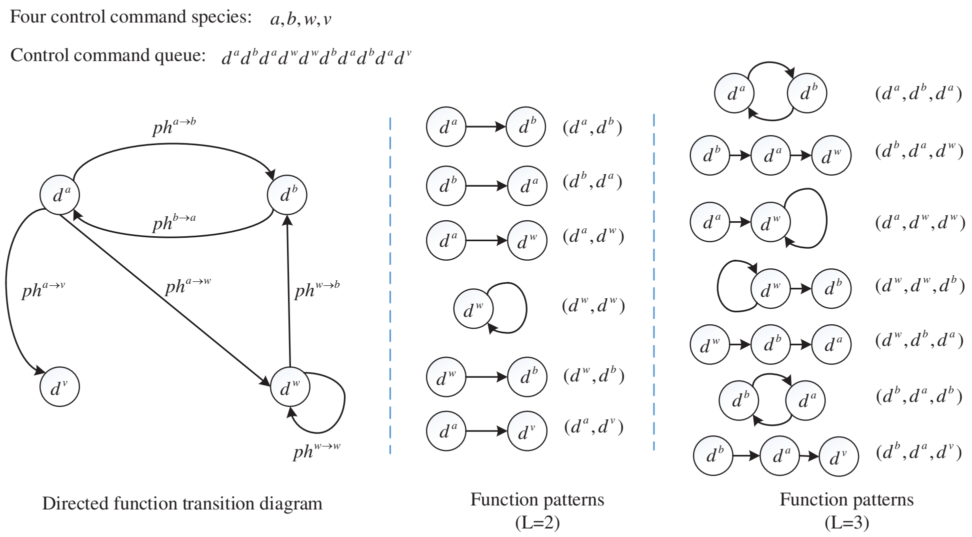


Figure 1: Example of directed function transition diagram and function patterns when L = 2 and L = 3

2.2 Process Control Feature Calculation

In practice, all function patterns can be easily obtained by preprocessing the whole control command sequence, and each function queue can be indicated by this group of function patterns. Therefore, the obtained function patterns are designed as a class of important feature factors to calculate process control feature values. Furthermore, the function pattern F_i^L can be further divided into L sub-patterns $F_i^{\{L,r\}} = (d_i^1, d_i^2, d_i^3, \cdots, d_i^r)$ $(r \in [1, L])$, each of which consists of the first r control commands in the function pattern. For each function queue, the probability of sub-pattern $F_i^{\{L,r\}}$ is defined as $p(F_i^{\{L,r\}})$, and the transition probability from one sub-pattern $F_i^{\{L,r-1\}}$ to another sub-pattern $F_i^{\{L,r\}}$ is calculated by the conditional probability $p(F_i^{\{L,r\}}|F_i^{\{L,r-1\}})$ $(r \in [1, L])$. As a result, the probability of each sub-pattern can be obtained by

$$p(F_i^{\{L,r\}}) = \prod_{l=1}^{r} p(F_i^{\{L,l\}}|F_i^{\{L,l-1\}}) \tag{1}$$

For each function pattern F_i^L, the probabilities of all sub-patterns can form a new probability vector $\overrightarrow{P} = (p(F_i^{\{L,1\}}), p(F_i^{\{L,2\}}), p(F_i^{\{L,3\}}), \cdots, p(F_i^{\{L,L\}}))$, which includes L values. Additionally, the time intervals to implement any two adjacent control commands are not identical, and each sub-pattern requires different time intervals to execute its control commands. Therefore, the time interval can be regarded as another important feature factor, which may play a key role in calculating process control feature values. Similarly, one time interval vector $\overrightarrow{\tau} = (\tau_1, \tau_2, \ldots, \tau_L)$ is constructed, and each time interval τ_r $(r \in [1, L])$ corresponds to the consuming time to implement all control commands of the sub-pattern $F_i^{\{L,r\}}$. According to the above two vectors, the final process control feature value x for each function pattern F_i^L can be calculated by

$$x = \overrightarrow{P} \cdot \overrightarrow{\tau} \tag{2}$$

Here, $(\cdot)$ represents the dot product operation.

To generate the real-time feature samples, each function queue Q_i is constructed by sequentially extracting all control commands from online communication traffics, and the time intervals to implement every two adjacent control commands are recorded. For each function pattern, two vectors are further constructed by analyzing the function queue Q_i: one is the probability vector calculated by Eq. (1) and the other is the time interval vector whose elements are the execution time of all sub-patterns in order. Furthermore, the feature value corresponding to each function pattern can be obtained by Eq. (2), and each function queue Q_i be successfully mapped to the process control feature sample $X_i = (x_i^1, x_i^2, \ldots, x_i^s)$. Here, s is the actual number of function patterns. Fig. 2 depicts the main workflow of process control feature generation algorithm based on directed function transition diagram.

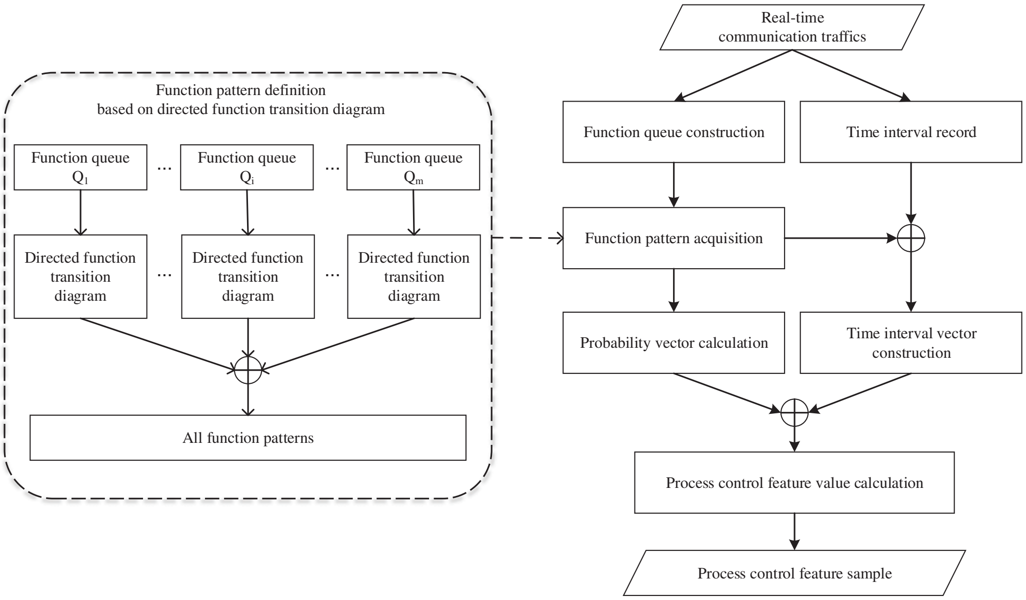

Real-time communication traffics
Function pattern definition based on directed function transition diagram
Function queue Q1
Function queue Qi
Function queue Qm
...
Directed function transition diagram
All function patterns
Function queue construction
Time interval record
Function pattern acquisition
Probability vector calculation
Time interval vector construction
Process control feature value calculation
Process control feature sample

Figure 2: Process control feature generation workflow based on directed function transition diagram

3 Machine-learning Detection Engines and Qualitative Comparison

3.1 Detection Processes of Five Machine-Learning Detection Engines

In the design of detection engines, five different machine-learning algorithms are introduced to cooperate with the above feature generation approach, and these machine-learning algorithms are SVM (Support Vector Machine), BPNN (BP Neural Network), DT (Decision Tree), NB (Naïve Bayes) and WNN (Wavelet Neural Network), respectively. Furthermore, by using the calculated feature samples, each anomaly detection engine can train a mathematical model or profile to learn the specific characteristics of operational actions, and recognize the statistical deviation for the observed data to identify intrusion activities.

3.1.1 SVM Detection Process

SVM belongs to one generalized linear classifier to realize the binary classification of data samples, and its decision boundary is one solved maximum-margin hyperplane by learning the training samples. The main steps of SVM detection process are outlined below:

Step 1: the observed feature samples are input to the trained SVM classifier, which accepted the penalty factor and the kernel parameter optimized by PSO (Particle Swarm Optimization) [33,34];

Step 2: the optimized SVM classifier introduces the Lagrange function to resolve the convex quadratic programming problem, and uses Gaussian kernel function to simplify the calculation complexity in the high-dimensional feature space.

Step 3: the SVM's decision result can be used to realize the anomaly classification. If the decision result is 1, then the observed data should belong to the normal classification. Differently, if the decision result is-1, then the observed data can be counted among the abnormal classification.

3.1.2 BPNN Detection Process

BPNN is a popular multi-layer feedforward neural network, whose essential characteristics cover two aspects: forward propagation of signals and back propagation of errors. The main steps of BPNN detection process are outlined below:

Step 1: the observed feature samples considered as the input signals in the constructed BPNN network, whose main parameters (including the connection weights, the hidden threshold and the output threshold) have been dynamically improved by the error feedback;

Step 2: according to the connection weights and the hidden threshold, the output results of hidden layer are calculated. Additionally, according to the connection weights and the output threshold, the prediction results of output layer are calculated;

Step 3: the final prediction errors can contribute to identify the outliers. If one prediction error exceeds the basic default threshold, then the observed feature may be normal; conversely, the observed feature should be abnormal.

3.1.3 DT Detection Process

DT is one of the most common algorithms to establish a classification model, which is formed by a hierarchy of branches from the root node to all leaf nodes. In essence, DT anomaly detection can classify different characteristics of feature samples by using a series of rules, and the main steps of DT detection process are outlined below:

Step 1: decision tree growing. Generate an original tree and classification rules based on the trained feature samples;

Step 2: decision tree pruning. Check the original tree, and cut off the redundant branches according the learning mechanisms;

Step 3: real-time anomaly detection. Distinguish abnormal test data through the rule matching.

3.1.4 NB Detection Process

NB generates a particular model, which uses the probability statistics to perform the sample classification. The main steps of NB detection process are outlined below:

Step 1: the observed feature samples are input to the optimal NB model, whose main parameters are first calculated by using the training feature samples;

Step 2: based on Bayes principle, the priori probabilities and posteriori probabilities of all observed feature samples are calculated, respectively;

Step 3: by combining he priori probability and posteriori probability of each observed feature sample, the corresponding normal and abnormal probabilities can be obtained to estimate the risk of abnormalities. If the normal probability of one observed feature sample is larger than its abnormal one, then the observed feature may be normal, and vice versa.

3.1.5 WNN Detection Process

WNN is a layered multi-resolution artificial neural network, which is based on wavelet theory and wavelet transform. Different from BPNN, the wavelet basis function is introduced to activate the hidden units, and the hidden layer wavelons are used to estimate the approximate values of the targets. The main steps of WNN detection process are outlined below:

Step 1: according to the trained feature samples, WNN's network parameters (including the dilation parameter, the translation parameter and the connection weights) should be improved by calculating the predicted errors between the predicted results and the expected outputs;

Step 2: as the input variables, the observed feature samples are analyzed by the optimal WNN network to estimate their predicted results;

Step 3: by comparing with the measured detection threshold, the predicted results are further used to identify intrusion activities. If the predicted result of one observed feature sample complies with the detection threshold, then the observed feature may be normal, and vice versa.

3.2 Differences of Qualitative Properties

Under different circumstances, various machine-learning algorithms present differential detection performances, that is, each machine-learning algorithm can develop its own properties. On the one hand, each machine-learning algorithm has its own distinct solution for the traditional classification problem, whose computational complexity can be reduced in diverse degrees by separately handling several sub-problems; on the other hand, the detection application of machine-learning algorithm is based on the feature engineering, and disparate feature characteristics can have an unequal effect on the detection performance of each machine-learning algorithm. Tab. 1 shows some qualitative properties of 5 machine-learning algorithms, and the advantages and disadvantages can be further compared from this table.

Table 1: Qualitative properties of five machine-learning algorithms

	Advantages	Disadvantages
SVM	• It belongs to one small-sample statistical learning method. • The kernel function (dot product) which realizes the nonlinear mapping in high dimensional feature spaces can reduce the computational complexity. • Support vectors are the trained results, which can help avoid the dimension disaster to some extent. • Its simple algorithm has strengthened abilities of robustness and generalization. • Its excellent learning ability can avoid the overfitting problem.	• For large training samples, it is too difficult to resolve the quadratic programming problem. • It is not applicable to solve the multi-classification problem. • It is very sensitive to the parameter and kernel function selection.
BPNN	• It can realize any complex nonlinear mapping, and fit for the problems whose internal mechanism is complex. • It has some practical abilities of self-adaption and generation. • It can support to solve the multi-classification problem. • Its fault-tolerant ability can guarantee the perfect system operations when the partial damage happens.	• It belongs to one gradient descent method, and the learning speed (or convergence speed) is rather low. • It often gets trapped in the local optimum, and causes the training failure. • The network's approximation and generalization are directly related to the typicality of samples, and the overfitting phenomenon can be easily caused.
DT	• It can be theoretically understood and computationally realized, and possess relatively low computation complexity. • It can deal with the data with many irrelevant features, and is insensitive to the missing median. • It can support to solve the multi-classification problem. • It requires no prior domain knowledge and initial parameter.	• It may cause the overfitting problem. • It ignores the relationships between different attributes. • It is difficult to handle large data sets due to the tree searching characteristics. • It is inappropriate for the high-dimensional data, especially the data containing too many attributes. • Its generalization ability is poor.

(Continued)

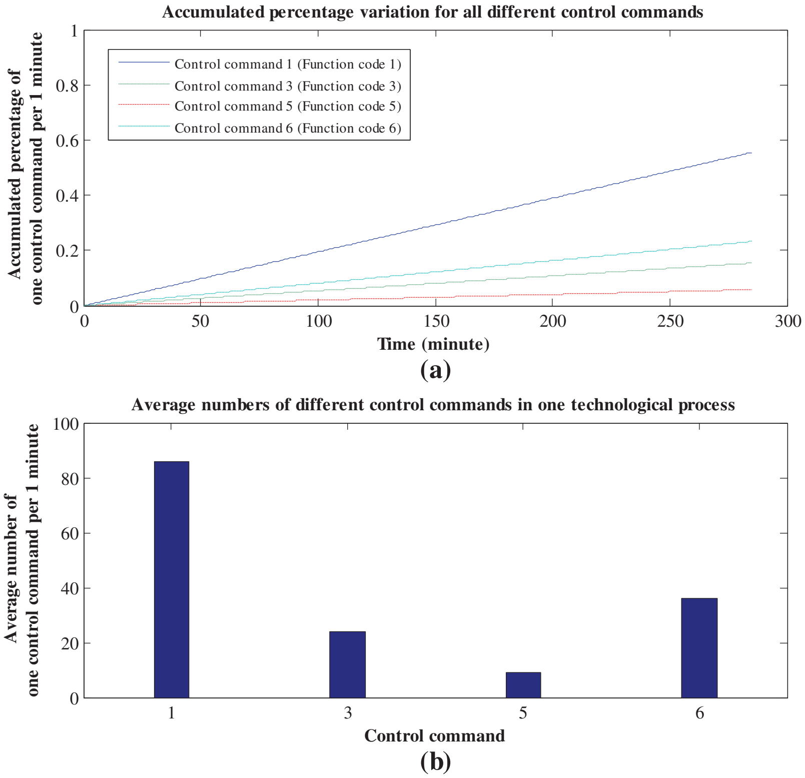
Table 1: Continued

	Advantages	Disadvantages
NB	• It has the solid mathematical foundation and stable classification efficiency. • It can reflect good performance on the small-scale data, and handle the multi-classification task. • Its simple algorithm can work great for the incremental learning, and is insensitive to the data missing.	• Its default assumption cannot stand in many practical applications. • Many cases can affect the classification efficiency, such as numerous attributes and strong correlation between attributes. • It is sensitive to the expression forms of input data.
WNN	• The wavelet theory can avoid the blindness of network structure. • It is practical with good learning ability and self-adaption. • Compared with BPNN, it has simpler network structure, faster convergence speed and higher classification accuracy.	• High-dimensional data may lead to huge network structures, and the corresponding convergence speed decreases greatly. • The number of hidden layer nodes is difficult to determine. • The inappropriate parameters may result in the non-convergence of learning process.

4 Experimental Comparison and Quantitative Analysis

In the experimental analysis, the detection accuracy and consuming time are introduced as two significant aspects to evaluate different performances of five machine-learning detection engines. By performing quantitative comparisons, our final purpose is to locate the optimal detection engines under differing cases. For this purpose, we evaluate all proposed designs by using the simulated Modbus/TCP control system, whose detailed statement can be presented in [18].

Special emphasis is given on the cycle of one complete technological process, which is programmed to 1 min. That is to say, we proportionally balance the operation cycle to achieve considerable Modbus/TCP communication data, which can serve to facilitate all analysis and comparisons. Furthermore, we run this system for about 4 h 46 min, and win a total of 44485 sequential control commands and the corresponding time intervals after the in-depth parsing of Modbus/TCP communication data. Through the preliminary analysis of original data, we plot the changing characteristics of different control commands with the technological process in Fig. 3. From Fig. 3a we can see that we design 4 different control commands in one technological process, which respectively correspond to function codes 1, 3, 5 and 6 in Modbus/TCP protocol, and the average number of each control command per 1 min is not identical. More specifically, the number of control command 1 is considerably larger than the number of control command 5, and it is chiefly because the system need frequently read the states of all valves to monitor the performance of technological process. Moreover, Fig. 3b shows the accumulated percentage variation for all different control commands, and each curve expresses the accumulated percentage of one control command per 1 min. Overall, all curves appear a smooth and upward-sloping trend, and this phenomenon can provide some indirect evidences for the periodic operational actions and relatively changeless running state.



(a)

(b)

Figure 3: Changing characteristics of different control commands with the technological process

4.1 Definition and Assumption

In order to evaluate different detection performances of 5 machine-learning algorithms, we define two practical attack models which can destroy the above technological process by disturbing normal process control. Apart from 4 kinds of control commands in the simulated control system, we do not introduce any additional control command in these attacks. The main reason for this definition is that any additional control command can be easily filtered by industrial firewalls based on defense-in-depth strategies [14,35]. In particular, the definitions of two attack models can be described in Tab. 2.

Based on the above attack models, we further generate 600 test feature samples in every experiment, mainly including 200 normal feature samples and 400 malicious feature samples. In practice, the detection performance may change with different attack intensities, and the aggressor may launch an attack with a higher success probability by increasing the attack intensity. Obviously, the number of abnormal control commands in each function queue can indirectly reflect the attack intensity. So, we define 4 distinguishing levels to depict different attack intensities: micro-intensity level (L1), low-intensity level (L2), medium-intensity level (L3) and high-intensity level (L4). More specifically, we suppose that these 4 levels correspond to 4, 6, 8 and 10 abnormal control commands in each function queue, respectively. Additionally, we set the length of function pattern is 3, namely the pattern length $L = 3$, and the corresponding dimension of feature sample is 60.

Table 2: Definitions of two attack models

Attack definition	Description
Targeted Continuing Attack (TCA)	The baleful aggressors can continuously launch a whole string of malicious operational actions, and each attack may start at different target locations in one technological process. In other words, this attack type can cause an unbroken series of abnormal control commands to appear in one function queue.
Blind Haphazard Attack (BHA)	The baleful aggressors can randomly launch each malicious operational action, and the attack target has certain blindness in one technological process. In other words, this attack type can cause all abnormal control commands to randomly spread in one function queue.

4.2 Detection Performance Comparison and Analysis for TCA

As previously described, we choose the detection accuracy and consuming time as two important aspects to compare the differences among 5 machine-learning detection engines. For each level of attack intensity, we conduct 12 different experiments whose malicious samples are generated by forging and inserting abnormal control commands, and calculate the average detection accuracy and consuming time. Tab. 3 shows the experimental results of 5 machine-learning detection engines for TCA. Seen from the high-intensity of view, the average detection accuracy of DT is lowest, and the corresponding average consuming time is longest. There follow two key reasons causing the above results: for one thing, DT ignores the relationships between different attributes; for another, the tree searching characteristics may waste too much time. Comparatively speaking, the average detection accuracy of WNN becomes a relatively ideal value which has exceeded 93% for each level of attack intensity, and it average consuming time is moderate, for example, the maximum consuming time is only 78.32 ms, which is less than the one of the least effective NB. Additionally, it is intuitively plausible that the average detection accuracy of BP can reach 95.72% for the high-intensity level, but its average detection accuracy for micro-intensity level is in an undesirable situation. Differently, SVM yields the best detection performance in all experiments, and obtains the highest detection accuracy and the least consuming time. Because the detection accuracies of SVM precede the ones of BP for all attack intensities, we suggest SVM and WNN as two applicable detection engines to match with the proposed feature generation approach. To be more specific, SVM has better detection performance whose highest detection accuracy can reach 97.51%, and it seems more appropriate for the attacks with medium-intensity and high-intensity levels. Differently, although WNN can give an impression of relatively smooth detection accuracies for all attack intensities and have powerful effect to identify the attacks with the micro-intensity level, its consuming time is not necessarily ideal.

Table 3: Detection performances of 5 machine-learning detection engines for TCA

Attack intensity	Average detection accuracy (%)					Average consuming time (ms)				
	SVM	BP	DT	NB	WNN	SVM	BP	DT	NB	WNN
L1	88.00	86.43	77.47	68.61	93.83	19.91	19.82	510.25	80.09	77.48
L2	93.03	91.25	82.67	77.58	93.92	19.57	19.45	502.83	86.03	75.51
L3	96.44	94.49	85.39	84.99	93.50	19.74	19.32	499.40	86.17	76.62
L4	97.51	95.72	87.04	89.05	93.55	19.49	20.12	553.49	90.26	78.31

4.3 Detection Performance Comparison and Analysis for BHA

Similarly, we take the same way to evaluate the detection performance for blind haphazard attack, and 12 different experiments are also performed for each level of attack intensity. Tab. 4 shows the experimental results of 5 machine-learning detection engines for BHA, and the basic detection performance of each machine-learning detection engine changes like the one for TCA. Differently, the whole average detection accuracies are significantly higher than the ones for TCA, because the probability of each function pattern seems more sensitive to the random distribution of abnormal control commands in each function queue. Obviously, SVM still gives a fantastic detection performance, and its lowest detection accuracy for the micro-intensity level is well above 93%. Compared with other detection engines, SVM possesses the highest detection accuracy and requires a minimum of consuming time to identify the blind haphazard attacks with all attack intensities.

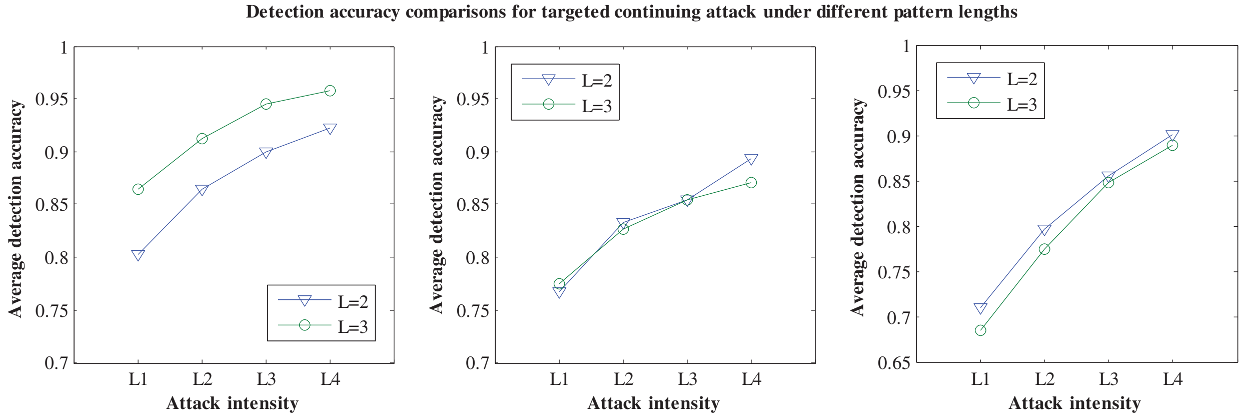
Table 4: Detection performances of 5 machine-learning detection engines for BHA

Attack intensity	Average detection accuracy (%)					Average consuming time (ms)				
	SVM	BP	DT	NB	WNN	SVM	BP	DT	NB	WNN
L1	93.22	90.99	88.72	80.50	94.10	15.80	19.21	417.06	84.38	75.57
L2	98.21	96.46	94.83	90.30	93.60	16.51	19.49	416.96	83.78	75.40
L3	99.54	97.96	97.32	94.53	93.89	16.23	20.15	446.99	92.65	79.05
L4	99.87	98.44	97.73	95.85	93.76	16.02	19.41	433.00	84.43	75.79

Above all, no matter targeted continuing attacks or blind haphazard attacks, both SVM and WNN exhibit their own unique advantages: for WNN, its detection accuracy is stable at a moderate level, but its real-time ability appears insufficient; for SVM, although it has the characteristic of strong real-time ability, its detection accuracy is not perfect for all attack intensities, especially for the micro-intensity level. In practical applications, if industrial production activities demand for the time sensitivity, SVM is more applicable to the proposed feature generation approach. If we want to improve the capability of detecting the attacks of micro-intensity level, WNN is suggested to match with the proposed feature generation approach. For taking advantages of both SVM and WNN, the combined use of these two detection engines is better than either of the two in effect.

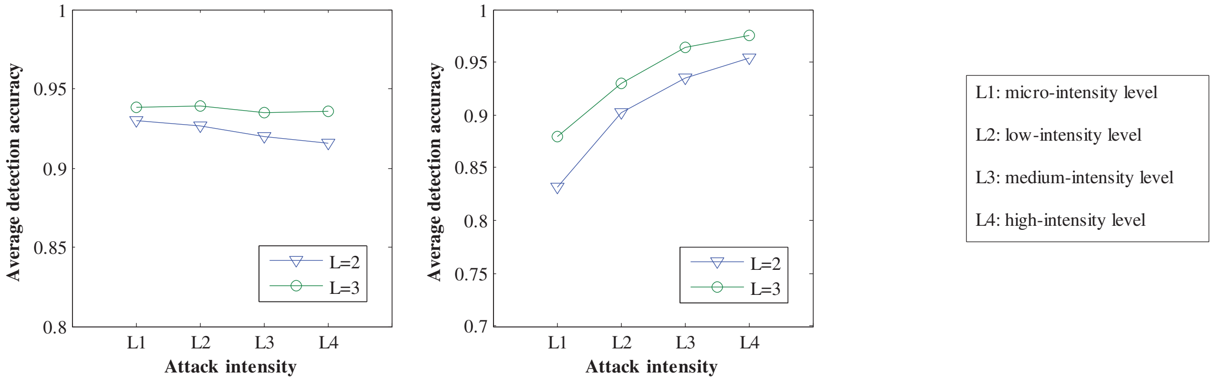
4.4 Discussion on Detection Effects of Different Pattern Lengths

The function pattern plays a significant role in the proposed feature generation approach, and its pattern length L is directly related to the calculation of feature values. In practice, the fine feature values not only reflect the inherent characteristics of original data, but also contribute positively to the effective anomaly detection. In other words, although the pattern length L is a pre-set parameter, different pattern lengths may make a big difference to the detection performance of each machine-learning detection engine. Obviously, due to the higher feature dimension caused by the increasing of pattern length, the corresponding consuming time may become longer. Additionally, when the pattern length changes, the detection accuracies of 5 machine-learning detection engines may have their distinctive changing characteristics. To give a more explicit description, we choose two pattern lengths to compare different detection accuracies of 5 machine-learning detection engines, and these pattern lengths are $L = 2$ and $L = 3$, respectively. Fig. 4 compares different detection accuracies of each machine-learning detection engine for TCA under two pattern lengths, and every detection accuracy is also the average value calculated by 12 experiments. From Sub-figures (a), (d) and (e) we can see that, the average detection accuracies of BP, WNN and SVM are improved to the higher level, when the pattern length increases. Differently, it turns out to be just the opposite in Sub-figures (b) and (c), that is, the average detection accuracies of DT and NB cause a slight decrease. The main reason of this result is that the machine-learning algorithms DT and NB ignore the relationships between different attributes, and the directed function transition diagram can strengthen the correlations with the growth of pattern length. To sum up, if we regard SVM and WNN as the applicable detection engines, one appropriate pattern length should be designed to balance detection efficiency and computation consumption based on the needs of practical application.



(a) BP (b) DT (c) NB

Figure 4: (Continued)



(d) WNN **(e) SVM**

Figure 4: Detection accuracy comparisons for TCA under different pattern lengths

5 Conclusion

Based on the process control features generated by directed function transition diagrams, this paper introduces 5 different machine-learning algorithms as alternative detection engines to compare their distinctive detection performances. On the one hand, this paper gives a brief description on the qualitative properties of each machine-learning algorithm, and compares their advantages and disadvantages point by point. On the other hand, this paper compares the detection performances of 5 different machine-learning algorithms by analyzing all experimental results, and picks out SVM and WNN as two relatively appropriate detection engines. Additionally, two attack models and four different attack intensities, which may reasonably occur in current industrial control networks, are defined to present an in-depth and meticulous analysis on the important properties: detection accuracy and consuming time. Also, this paper gives the quantitative discussion on the impacts of detection accuracy caused by different pattern lengths. Based on the above quantitative and qualitative analysis, we believe that the proposed contents in this paper are fascinating and promising, and can make academic contributions to the related research work.

Acknowledgement: The authors are grateful to the anonymous referees for their insightful comments and suggestions.

Funding Statement: This work is supported by the Scientific Research Project of Educational Department of Liaoning Province (Grant No. LJKZ0082), the Program of Hainan Association for Science and Technology Plans to Youth R & D Innovation (Grant No. QCXM201910), the National Natural Science Foundation of China (Grant Nos. 61802092 and 92067110), the Hainan Provincial Natural Science Foundation of China (Grant No. 620RC562) and 2020 Industrial Internet Innovation and Development Project-Industrial Internet Identification Data Interaction Middleware and Resource Pool Service Platform Project, Ministry of Industry and Information Technology of the People's Republic of China.

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

References

[1] I. B. Taha and D. A. Mansour, “Novel power transformer fault diagnosis using optimized machine learning methods,” *Intelligent Automation & Soft Computing*, vol. 28, no. 3, pp. 739–752, 2021.

[2] M. Aazam, S. Zeadally and K. A. Harras, “Deploying fog computing in industrial internet of things and Industry 4.0,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[3] Z. Hu, X. Yu, J. Shi and L. Ye, “Abnormal event correlation and detection based on network big data analysis,” *Computers, Materials & Continua*, vol. 69, no. 1, pp. 695–711, 2021.

[4] M. Kim, S. Hong, M. Kang and J. Seo, “Performance comparison of PoseNet models on an AIoT edge device,” *Intelligent Automation & Soft Computing*, vol. 30, no. 3, pp. 743–753, 2021.

[5] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi *et al.,* “TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial internet of things,” *IEEE Transactions on Mobile Computing*, pp. 1–14, 2021. [Online]. Available: http://dx.doi.org/doi:10.1109/TMC.2021.3064314.

[6] J. Duan, Y. Qu, J. Hu, Z. Wang, S. Jin *et al.,* “Fast and stable learning of dynamical systems based on extreme learning machine,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1175–1185, 2017.

[7] 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. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2020.3041212.

[8] 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.

[9] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2019.

[10] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 1–27, 2020.

[11] K. Satpute, S. Agrawal, J. Agrawal and S. Sharma, “A survey on anomaly detection in network intrusion detection system using particle swarm optimization based machine learning techniques,” in *Proc. the Int. Conf. on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, Odissa, India, pp. 441–452, 2012.

[12] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[13] N. Jhanjhi, M. Humayun and S. N. Almuayqil, “Cyber security and privacy issues in industrial internet of things,” *Computer Systems Science and Engineering*, vol. 37, no. 3, pp. 361–380, 2021.

[14] M. Wan, J. Li, Y. Liu, J. Zhao and J. Wang, “Characteristic insights on industrial cyber security and popular defense mechanisms,” *China Communications*, vol. 18, no. 1, pp. 130–150, 2021.

[15] J. Dong, K. Wang, W. Quan and H. Yin, “InterestFence: Simple but efficient way to counter interest flooding attack,” *Computers & Security*, vol. 88, pp. 1–12, 2020.

[16] Y. Liu, L. Kong and G. Chen, “Data-oriented mobile crowdsensing: A comprehensive survey,” *IEEE Communications Surveys and Tutorial*, vol. 21, no. 3, pp. 2849–2885, 2019.

[17] A. Valdes and S. Cheung, “Communication pattern anomaly detection in process control systems,” in *Proc. 2009 IEEE Conf. on Technologies for Homeland Security*, Waltham, England, pp. 22–29, 2009.

[18] 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.

[19] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong *et al.,* “Fine-grained abnormal driving behaviors detection and identification with smartphones,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2198–2212, 2017.

[20] G. S. Sestito, A. C. Turcato, A. L. Dias, M. S. Rocha, M. M. da Silva *et al.,* "A method for anomalies detection in real-time Ethernet data traffic applied to PROFINET," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2171–2180, 2018.

[21] S. Huang, Y. Guo, N. Yang, S. Zha, D. Liu *et al.,* "A weighted fuzzy C-means clustering method with density peak for anomaly detection in IoT-enabled manufacturing process," *Journal of Intelligent Manufacturing*, vol. 32, pp. 1845–1861, 2021.

[22] J. Hosi, J. Lamps and D. H. Hart, "Evolving decision trees to detect anomalies in recurrent ICS networks," in *Proc. 2015 World Congress on Industrial Control Systems Security (WCICSS)*, London, UK, pp. 50–57, 2015.

[23] C. Zhou, S. Huang, N. Xiong, S. H. Yang, H. Li *et al.,* "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1345–1360, 2015.

[24] K. El-Khatib, "Impact of feature reduction on the efficiency of wireless intrusion detection systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 8, pp. 1143–1149, 2010.

[25] A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, "Machine learning for anomaly detection: A systematic review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021.

[26] 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.

[27] 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.

[28] C. Kim and D. Robinson, "Modbus monitoring for networked control systems of cyber-defensive architecture," in *Proc. 2017 Annual IEEE Int. Systems Conf. (SysCon)*, Montreal, Canada, pp. 1–6, 2017.

[29] M. Tsukada, M. Kondo and H. Matsutani, "A neural network-based on-device learning anomaly detector for edge devices," *IEEE Transactions on Computers*, vol. 69, no. 7, pp. 1027–1044, 2020.

[30] M. Cui, J. Wang and M. Yue, "Machine learning-based anomaly detection for load forecasting under cyberattacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5724–5734, 2019.

[31] S. Kanarachos, S. G. Christopoulos, A. Chroneos and M. E. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets neural networks and Hilbert transform," *Expert Systems with Applications*, vol. 85, pp. 292–304, 2017.

[32] C. Huang and L. Li, "Architectural design and analysis of a steer-by-wire system in view of functional safety concept," *Reliability Engineering & System Safety*, vol. 198, pp. 1–15, 2020.

[33] M. Wan, X. Xu, Y. Song, Q. Li and J. Li, "Extracting function-driven tracing characteristics for optimized SVM classification," *Mobile Information Systems*, vol. 2021, pp. 1–12, 2021.

[34] A. Alhudhaif, A. Saeed, T. Imran, M. Kamran, A. S. Alghamdi *et al.,* "A particle swarm optimization based deep learning model for vehicle classification," *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 223–235, 2022.

[35] 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.