applied sciences

MDPI

Article

An ICS Traffic Classification Based on Industrial Control Protocol Keyword Feature Extraction Algorithm

Changhong Yu *, Ze Zhang and Ming Gao

School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
* Correspondence: yuch@mail.zjgsu.edu.cn

Abstract: Industrial control protocol feature extraction is an important way to improve the accuracy and speed of industrial control protocol traffic classification. This paper firstly proposes a keyword feature extraction method for industrial control protocol, and then designs and implements an industrial control system (ICS) traffic classification based on this method. The proposed method utilizes the characteristics of the relatively fixed format of the industrial control protocol and the periodicity of the protocol traffic in ICS. The keyword features of the industrial control protocol can be accurately extracted after data preprocessing, data segmentation, redundant data filtering, and feature byte mining. A feature dataset is then formed. The designed ICS traffic classifier adopts decision tree and is trained with the feature dataset. Experiments are carried out on the open-source dataset. The results show that the proposed method achieves 99.99% classification accuracy, and the classification precision and classification recall rate reach 99.98% and 99.93%, respectively. The training time and predicting time of classifier are 0.34 s and 0.264 s, respectively, which meets the requirements of high precision and low latency of industrial control system.

Keywords: industrial control system; periodicity; feature extraction; decision tree; protocol traffic classification

Citation: Yu, C.; Zhang, Z.; Gao, M. An ICS Traffic Classification Based on Industrial Control Protocol Keyword Feature Extraction Algorithm. *Appl. Sci.* **2022**, *12*, 11193. https://doi.org/10.3390/app122111193

Academic Editor: Robert Ojstersek

Received: 4 October 2022
Accepted: 2 November 2022
Published: 4 November 2022

1. Introduction

The development of the Internet of Things (IoT) and cloud computing promoted the upgrading of traditional ICSs to intelligence and automation [1], making the application of ICSs more extensive. Nowadays, ICSs are widely used in different critical infrastructures such as intelligent traffic management, smart grids, and smart manufacturing. As a subset of ICSs, supervisory control and data acquisition (SCADA) systems undertake the task of data monitoring, collection, and transmission, in which a large number of industrial control protocols with unknown structures are transmitted. The integration of the Internet and ICSs makes the network environment complicated. ICS protocol traffic classification is an important part of ICS network management, which can be used to prevent security threats and analyze network components.

At present, there are four main internet application protocol traffic classification methods [2]: port-based, load-based, machine-learning-based, and deep-learning-based traffic classification methods. The port-based method classifies traffic according to the specified port number, which is simple and fast. However, the advent of port pooling and dynamic port techniques greatly reduced the accuracy of this approach [3]. The load-based classification method is also called deep packet inspection (DPI). DPI classifies the traffic by extracting the signatures from the protocol packets and matching them in the existing signature database [4]. However, the signature database must be generated in advance and the computational cost is high. It has low accuracy in classifying encrypted traffic. Traffic classification methods based on machine learning (ML) and deep learning (DL) became the hotspots of traffic classification research in recent years [5]. ML-based methods realize traffic classification through feature extraction and training ML classifiers [6]. Dong [7] used network flow-level features to identify the type of traffic. However, expert experience

is required for feature extraction, and features are hard to extract in private protocols, which limits the generalizability of the methods based on ML. DL-based methods avoid manual design and can automatically extract features [8], which can be used to classify traffic easily. Shapira et al. [9] used DL technology to extract features and classify traffic. However, DL-based methods require a large amount of data for learning and have a high time cost, which is not suitable for ICSs that need low latency.

The commonly used network traffic classification methods are mainly used to classify multimedia traffic or internet application traffic with complex characteristics, and have the disadvantages of high algorithm complexity and a high time cost. Different from the internet protocol, the industrial control protocol has a concise structure and relatively fixed functions [10]. In the actual production process, the industrial control network traffic has periodicity and stability for meeting specific industrial processes [11]. Therefore, the method for internet traffic classification cannot be directly applied in ICSs because it cannot meet the high real-time requirement for ICSs. We designed a keyword feature extraction method for private industrial control protocol according to the characteristics of industrial control protocol and periodicity of data flow in ICSs. This algorithm can automatically extract the keyword features through data segmentation, filtering, and association rule mining from protocol payload without manual assistance. Then, we used the dataset after feature extraction to train a more concise decision tree, which can help classify the ICS protocol traffic quickly and accurately.

Experiments were carried out on the SWAT and EPIC joint dataset. The results show that the industrial control protocol feature extraction method proposed in this paper can successfully extract the protocol keyword features, and the decision tree classifier trained with the feature dataset has a great performance in classification accuracy, precision, recall rate, training time, and classification time, which proves the superiority of our method.

The remainder of this paper is organized as follows. Section 2 describes the related work about network protocol traffic classification. Section 3 describes the proposed feature extraction method of industrial control protocol. Section 4 introduces the characteristics of decision tree briefly, and training process of decision tree classifier for ICS protocol traffic. Section 5 describes the experiments and results. Section 6 compares and discusses the classification effects of different classifiers based on feature datasets, and the latest traffic classification methods. Finally, the conclusion is in Section 7.

2. Related Work

The current research on protocol traffic classification is mainly based on machine learning and deep learning. Table 1 lists several of the latest works on traffic classification based on machine learning and deep learning in recent years and their results.

2.1. Traffic Classification Methods Based on ML

ML-based methods mainly achieve traffic classification by manually selecting features and then training a ML classifier to associate the feature set with known traffic classes [12].

Cao et al. [13] proposed an improved network traffic classification model based on support vector machine to classify network traffic, which achieved a classification accuracy of 97.2% and the training time was 0.31 s. Jiang and Chen [14] proposed an ICS traffic classification method, which achieved 100% accuracy by extracting data features, data imbalance processing, and ensemble learning method. Aouedi et al. [15] designed two feature selection methods to select traffic features, then used the feature data to train a variety of tree-based classification models to classify network traffic. Mokhtari et al. [16] proposed a method for ICS traffic detection based on measurement data. The method used a variety of ML algorithms to monitor the data flow from alternator and selected relevant features from a large amount of data manually. The random forest had the best performance; the accuracy was about 99% and the training time was 2.21 s, while the predicting time was 0.0505 s. Lan et al. [17] applied an ML algorithm to traffic classification,

manually selected flow features and industrial control protocol keyword features, and achieved 99% classification accuracy with a classification time of 0.187 s.

ML-based methods need expert experience when extracting features, which limits the generality of the methods.

2.2. Traffic Classification Methods Based on DL

DL can learn from large amounts of data and extract features directly, which removes the reliance on expert knowledge. However, DL methods usually have larger time costs than ML methods.

In order to solve the problem of low efficiency brought by the DL method, Ling et al. [18] proposed an ICS traffic monitoring method based on a two-way simple recursive unit. The two-way structure in the neural network was optimized and the training effect was improved through using skip connections, and achieved a classification accuracy of more than 92%. However, the training time required at least nearly 100 s and increased with the increase in the number of neural network layers, even though the efficiency of the neural network algorithm was optimized. In order to efficiently manage industrial IoT systems, Lin et al. [19] proposed a traffic classification method based on spatiotemporal features, using DL technology to automatically extract spatiotemporal features in data packets and implement traffic classification, which achieved 95% classification accuracy. Ren et al. [20] proposed a tree-structured recurrent neural network for network traffic classification, which achieved 98.98% classification accuracy with a training time of 54.20 s and a classification time of 0.185 milliseconds. Mendonça et al. [21] proposed a lightweight intelligent intrusion detection system for the industrial Internet of Things using DL algorithms, which achieved 99% classification accuracy through a new predictive model based on sparse evolution training for network monitoring. The training time and predicting time were 61.31 s and 2.36 s, respectively. Zhai et al. [22] used an AM-1DCNN + LSTM DL model to extract features of the ICS traffic, and identified protocols, which achieved accuracy of 93%, but spent 122 min training the model.

The above DL-based methods required high training and predicting time, which is unacceptable for an ICS that requires high real-time performance.

Table 1. Comparison of different traffic classification methods.

Research	Method	Feature Extraction	Training time	Classification Time	Accuracy
Literature [13]	ML-based	Manual	0.31 s	Unknown	97.2%
Literature [14]	ML-based	Manual	Unknown	Unknown	100%
Literature [15]	ML-based	Manual	104.85 s	12.75 s	82.31%
Literature [16]	ML-based	Manual	2.21 s	0.0505 s	99%
Literature [17]	ML-based	Manual	Unknown	0.187 s	99%
Literature [18]	DL-based	Automatic	100 s	Unknown	Unknown
Literature [19]	DL-based	Automatic	Unknown	Unknown	95%
Literature [20]	DL-based	Automatic	54.2 s	0.185 ms	98.98%
Literature [21]	DL-based	Automatic	61.31 s	2.36 s	99%
Literature [22]	DL-based	Automatic	121.9 min	Unknown	94%
Our method	ML-based	Automatic	0.34 s	0.264 s	99.99%

3. Feature Extraction Algorithm for Industrial Control Protocol

The purpose of feature extraction of industrial control protocol is to find out those representative protocol keyword features from a quantity of features, such as protocol identifiers and function codes, which can reduce feature dimension, so as to obtain a set of features with a small number but a large amount of classification information. According to the periodicity of industrial control protocol traffic, we designed a feature extraction method for industrial control protocol based on frequent items, which can effectively extract the keyword features of industrial control protocol.

The process of feature extraction is shown in Figure 1. Data preprocessing strips the protocol payload from the original network data packet, unifies the data length, and converts it into hexadecimal form for subsequent processing. N-Gram model is used for protocol data segmentation, and the value of N is determined by Zipf's law. A large number of length N data items can be obtained after the segmentation. Although it contains frequent and meaningful data items, more are meaningless redundant data units. We use the Jaccard coefficient to filter useless data items and obtain frequent item sets. Finally, the keyword features of the protocol are mined using association rules.

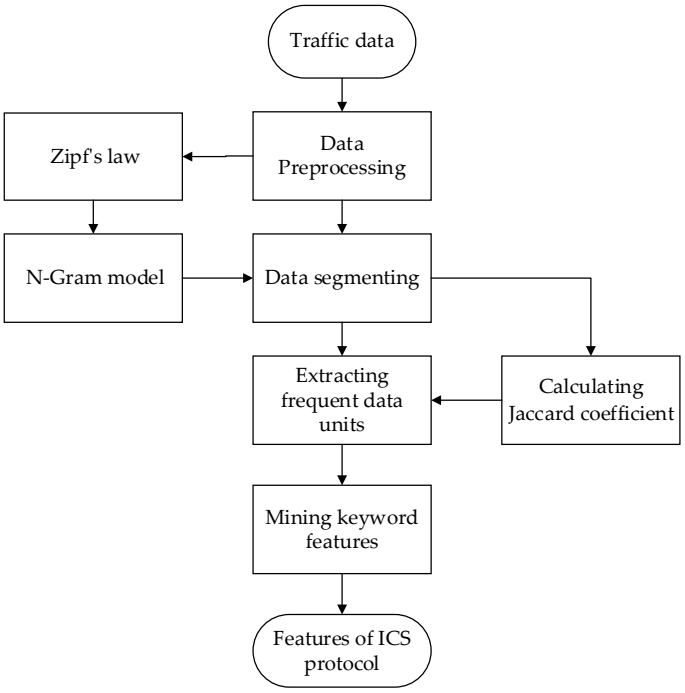


Figure 1. Process of industrial control protocol feature extraction algorithm.

3.1. Data Preprocessing

The industrial control network protocol is usually encapsulated layer by layer from top to bottom based on the transmission control protocol/internet protocol (TCP/IP). We used the protocol payload as our research data. Therefore, it was necessary to strip protocol payload from the original traffic packet. The protocol data can be easily exported into a C array format with the help of Wireshark, then we wrote a program to obtain payload and unify the data length. Finally, the protocol data shown in Table 2 were obtained.

Table 2. Data format after preprocessing.

******document content******
['01', '04', '02', '03', 'F1', '0C', '0C', '02', '00', '00', '02', '19', '4C', '29', '21', '20', '4B', . . .]
['01', '00', '30', 'F1', '0C', '0C', '02', '00', '00', '02', '19', '4C', '29', '21', '3E', 'C4', '01', . . .]
['01', '04', '02', '03', 'F2', '0C', '0C', '02', '00', '00', '02', '19', '4C', '29', '22', '08', '87', . . .]
['01', '00', '30', 'F2', '0C', '0C', '02', '00', '00', '02', '19', '4C', '29', '22', '3E', 'C4', '01', . . .]
. . .

3.2. Protocol Payload Segmentation Based on N-Gram Model

Protocol data are transmitted in binary on the network, which can be recognized as machine language. The N-Gram model [23] is a natural language processing model. This paper applies the N-Gram algorithm to the extraction of features for industrial control protocol. The protocol data are regarded as a corpus, and each protocol packet is regarded as a text for segmentation.

The basic idea of N-Gram is using a sliding window of length N to start from the first character of the text and move backward one character in turn. The N characters contained in the window are an N-Gram data unit. Figure 2 is a schematic diagram of segmentation when N = 2.

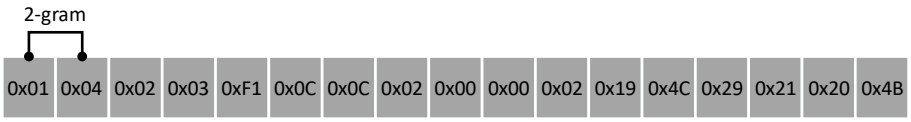


Figure 2. Segmentation of 2-Gram.

In the N-Gram model, the value of N is related to the validity and integrity of the segmentation. The larger the value of N, the better the integrity of the segmented data, but the large range of words after segmentation results in low effectiveness. The smaller the value of N, the more likely the keywords are to be divided, so that the word segmentation segment cannot contain complete lexical information. In this paper, we choose the appropriate value of N through Zipf's law [24]. According to Zipf's law, setting the frequency of a certain data unit in the protocol data to be p and the frequency ranking to be r, then Equation (1) can be obtained.

$$pr = C \tag{1}$$

where C is a constant, take the logarithm to obtain Equation (2)

$$ln(p) + ln(r) = ln(C) \tag{2}$$

Taking $ln(p)$ and $ln(r)$ as the horizontal and vertical coordinates, respectively, we can finally obtain a straight line with a slope of -1, then take N as 1, 2, and 3 to segment the protocol data. If an approximate straight line is obtained, it is considered that Zipf's law is satisfied.

3.3. Frequent Item Extraction Based on Jaccard Coefficient

A large number of N-Gram data units are obtained after dividing the protocol data through the N-Gram model, but few data units are useful for classification. The data units that contribute to protocol classification are meaningful and frequent, so these N-Gram data units need to be further screened in order to obtain frequent data units. The Jaccard coefficient [25] can be used to compare the similarity between sets of N-Gram data units. In this paper, the Jaccard coefficient is used to find out the threshold for screening and filtering out the N-Gram frequent data units that are helpful for classification. The larger the Jaccard coefficient, the higher the similarity between sets of N-Gram data units. At this time, the data units contained in the set are most likely to be frequent and useful.

The process of calculating the Jaccard coefficient is shown in Equation (3). The data units obtained by dividing are randomly divided into two sets, A and B. The similarity between sets A and B can be seen as the ratio of the intersection and union between the two sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{3}$$

Then, calculate the frequency of occurrence of subunits in each set separately and sort them from large to small, recorded as Equations (4) and (5):

$$\{A_1 : f_{A1}, A_2 : f_{A2}, \ldots, A_n : f_{An}\} \tag{4}$$

$$\{B_1 : f_{B1}, B_2 : f_{B2}, \ldots, B_n : f_{Bn}\} \tag{5}$$

Among them, A_i, f_{Ai}, B_i, and f_{Bi} represent the subunits in A and B and their frequency of occurrence, respectively. In order to make Jaccard more suitable for the needs of this paper, it is modified as shown in Equation (6):

$$J(A,B) = \frac{\sum_{i=1}^{n}(f_{Ai} * f_{Bi})}{\sum_{i=1}^{n} {f_{Ai}}^2 + \sum_{i=1}^{n} {f_{Bi}}^2 - \sum_{i=1}^{n}(f_{Ai} * f_{Bi})} \quad (6)$$

Different data units appear at different frequencies in the set. Select different frequencies as thresholds to filter out subunits with low frequencies in the set and then calculate the Jaccard coefficients of the two sets at the corresponding frequencies. We can obtain a list of Jaccard coefficients; this paper selects the frequency value f corresponding to the first maximum value of the Jaccard coefficient as the screening threshold for retaining as many useful data units as possible, and the frequency of a subunit that is greater than f is considered as a meaningful and frequent data unit.

3.4. Feature Byte Mining Based on Association Rules

There may be a certain relationship between the frequent data units and their positions in the protocol because the format of the industrial control protocol is relatively fixed. This paper uses association rule [26] to extract the characteristic positions of industrial control protocol. The association rule reflects the correlation and interdependence between things, whose purpose is to extract valuable laws and connections from massive data. The process of association rule is generally divided into two steps: the first is to find out all frequent units according to the support degree and the second is to generate association rules according to the confidence degree. After segmentation through the N-Gram model and filtering by Jaccard coefficient, frequent item sets can be successfully obtained. The location information of the protocol features can be found according to appropriate association rule and confidence threshold.

There are two events, X and Y. The confidence calculation of the association rule $X \Rightarrow Y$ is shown in Equation (7), which means the proportion of the number of events in which the Y event occurs when the X event occurs in the total number of X events.

$$\text{confidence}(X \Rightarrow Y) = \frac{X \cap Y}{X} \quad (7)$$

Presume the event that the frequent item appears in the protocol frame is T, and the event that the frequent item appears in at the position N of the packet is K. Define the association rule $T \Rightarrow K$: the probability that a frequent item occurs at position N of the packet while being present in the data frame. If this probability is greater than the specified confidence threshold, it can be considered that when the frequent item appears in the protocol packet, there is a high probability of appearing at position N, which means the position N is the characteristic position of the protocol. Complex industrial scenarios may lead to many possible values of feature positions, which can result in a reduction in the proportion of each feature value. Therefore, the confidence threshold must be dynamically adjusted according to the complexity of the realistic industrial control scenarios.

4. Industrial Control Network Traffic Classifier Based on Feature Extraction Dataset

4.1. Classifier Selection Based on Discrete Uncorrelated Features

A classifier needs to be trained to realize the industrial control network traffic classification after the feature extraction. We selected the decision tree as the industrial control network traffic classifier because of its superior performance [27]. The decision tree does not need a priori assumptions, the data processing is simple, and it can produce good results for the data source in a short time. The features extracted by the feature extraction method in this paper are discrete protocol keywords with a small quantity, which greatly reduced the possibility of overfitting the decision tree.

4.2. Decision Tree Classifier Training and Testing Based on Feature Dataset

The training process of the decision tree classifier is shown in Figure 3. The decision tree belongs to supervised classification algorithms. It is necessary to know the categories corresponding to different data in advance before training the model, so the data must be labeled first. Then, according to the features extracted by the feature extraction method, redundant data are removed from the original dataset to obtain a new feature dataset. Finally, 80% of the feature dataset is randomly selected as the training set, and 20% is used as the test set to train and test the decision tree classifier on the open-source machine learning platform.

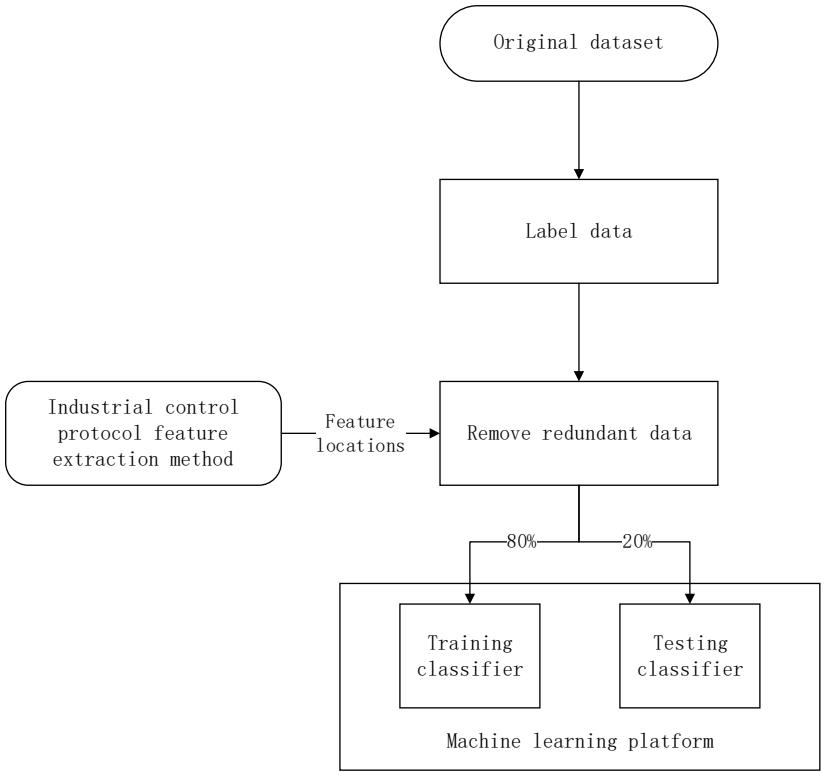


Figure 3. The training and testing process of the decision tree classifier.

5. Experiments and Results

5.1. Experimental Environment

To evaluate the effectiveness of the method in this paper, experiments were carried out on a Windows 10 system with Inter(R)Core(TM) i7-6500U and NVIDIA GeForce 940MX. The experiments were programmed in Python and used the scikit-learn framework of machine learning to build a classification model and all machine learning classification models were trained with the default parameters.

5.2. Dataset Description

We used the SWAT dataset and EPIC dataset provided by Singapore University of Technology and Design as our research data. Figure 4 shows the experimental environment for the acquisition of the two datasets. The SWAT dataset was collected from a real water treatment bench. The water treatment process includes water supply, storage, filtration, and backwashing, etc., and communicates using the ethernet/IP protocol [28]. The EPIC dataset was collected from a small power test bench that simulated a real power system in a small smart grid, communicating based on the IEC 61850-MMS protocol [29].

(**a**) SWAT testbed (**b**) EPIC testbed

Figure 4. SWAT and EPIC testbeds by iTRUST in Singapore.

The EPIC test bench was also used to power the SWAT test bench, resulting in a cascaded ICS using multi-protocol communication, which fits well with the research topic of this paper (classification of ICS protocol traffic based on multi-communication protocols). Therefore, these two datasets are selected as the experimental data of this paper. However, the amount of data in the SWAT and EPIC datasets is huge. In order to facilitate the experimental processing, we randomly select 70,000 protocol data streams from each of the two datasets, and a total of 140,000 samples are used for experiments.

5.3. Classifier Evaluation Metrics

The performance metrics are mainly composed of two parts: (1) the time efficiency of the classifier, and (2) the accuracy of the classifier. The time efficiency of classification refers to the time it takes to train a classifier and the time it takes to predict the category of a sample using the classifier. The correctness of the classifier includes three criteria: accuracy, precision, and recall. Accuracy is used to measure the overall classification effect of the classifier, and precision and recall measure the classification effect of a single protocol category of the classifier. The calculation formula is shown in Formulas (8)–(10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

The meanings of TP, TN, FP, and FN are shown in Figure 5.

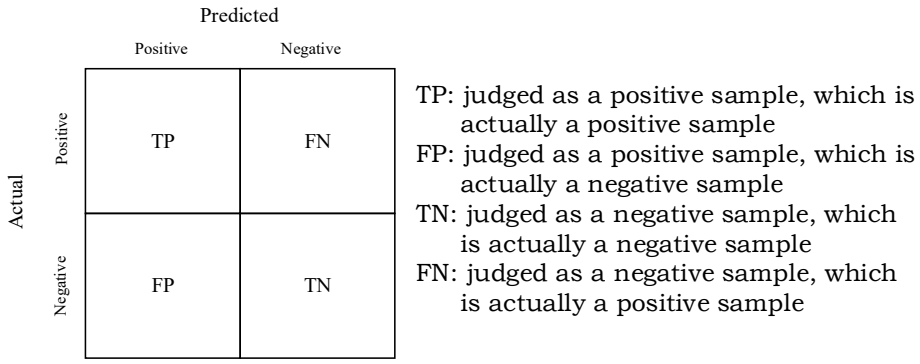
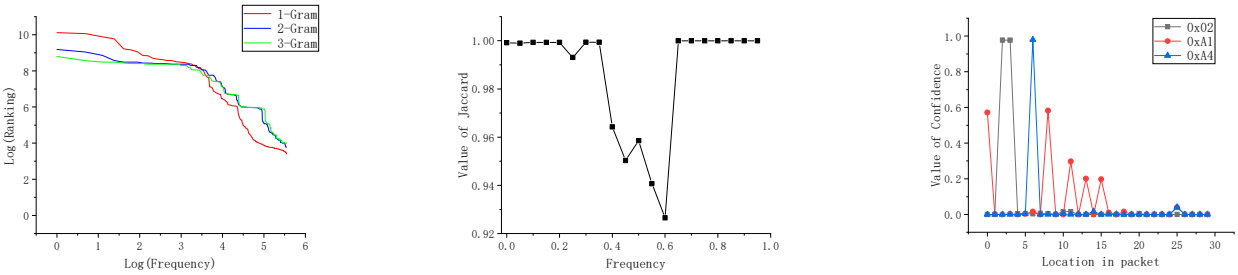


Figure 5. The meaning of TP, FP, TN, FN.

5.4. Results

We take the IEC 61850-MMS protocol as an example to illustrate the accuracy of the keyword feature extraction method for the industrial control protocol in this paper. Since

the keyword features are usually concentrated in the protocol header, we unify the length of the protocol payload as 50 bytes and add "−1" to the packet whose length is less than 50. Experiments are carried out according to the feature extraction method described in this paper and the confidence threshold is set as 0.5. The results of each step when extracting features are shown in Figure 6. Finally, the 0th, 2nd, 3rd, 6th, and 8th bytes are extracted as the keyword bytes of the IEC 61850-MMS protocol.



(**a**) Zipf curve (**b**) Jaccard curve (**c**) Degree of confidence curve

Figure 6. The diagram of each step of the feature extraction method.

The format of the IEC 61850-MMS protocol is shown in Table 3. It can be seen that the 0th, 2nd, 3rd, and 6th bytes are successfully extracted as the keywords of the IEC 61850-MMS protocol by comparing with the protocol format. The 8th byte is also regarded as a keyword, the reason is that this byte represents a service type (tag). The 11th, 13th, and 15th bytes also belong to the tag flag, while their confidence levels are lower than the threshold specified, which means not enough packets use the bytes as a tag, so they are not considered a keyword byte. On the whole, the key feature extraction method of the industrial control protocol proposed in this paper can successfully extract the feature information of the industrial control protocol.

Table 3. IEC 61850-MMS protocol message format.

Protocol Message Type	**Subsequent Data Length**	**Frame Start Flag**
1 Bytes	1 Byte	2 Bytes
invoke id	**Service type**	**subsequent data length**
2 Bytes	1 Byte	1 Byte
	content	
	Accumulated data of tag–length–value	

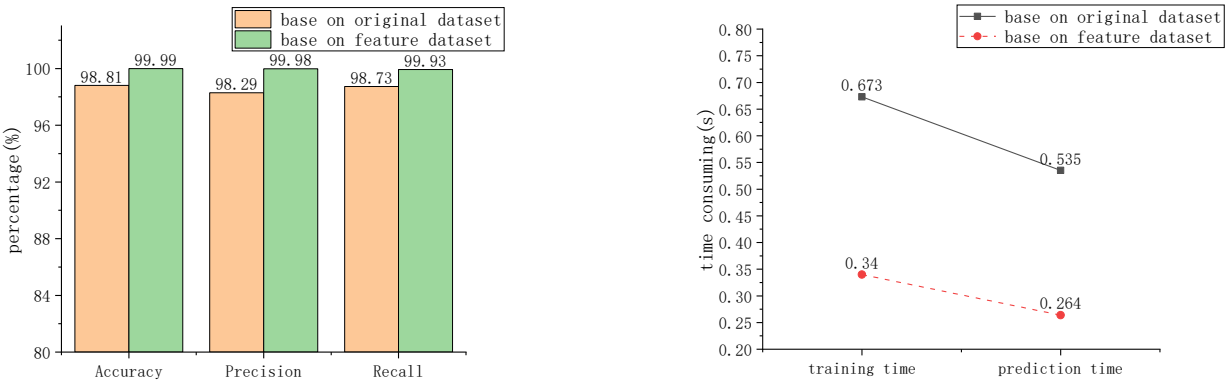
Feature extraction is performed on the IEC 61850-MMS and ethernet/IP protocols in the dataset. The comparison between the feature dataset and the original dataset is shown in Table 4. It can be found that the number of features is reduced from 50 to 13, and data memory footprint is reduced by 74%.

Table 4. The comparison between feature dataset and original dataset.

Dataset	**Number of Samples**	**Number of Features**	**Data Memory**
Feature dataset	140,000	13	7.28 MB
Original dataset	140,000	50	28 MB

We take 80% of the samples in the original dataset and feature dataset as the training set and 20% as the test set, then train the decision tree classifier and test the performance of the classifier. The accuracy and time cost comparison of the two classifiers are shown in Figure 7, and the precision and recall rate of a single category are shown in Tables 5 and 6. The results show that the feature dataset removes the interference features in the original

data. The decision tree classifier trained with the feature dataset is more concise and efficient; the average precision of the classifier is 99.99%, the average recall rate is 99.98%, the training time is only 0.34 s, and the predicting time is only 0.264 s, outperforming the classifier trained by the original dataset.



(**a**) Accuracy comparison of classifiers

(**b**) Time cost comparison of classifiers

Figure 7. Accuracy and time cost comparison of classifiers based on different datasets.

Table 5. Protocol classification precision and recall based on original dataset.

Protocol Category	Precision	Recall
IEC 61850-MMS	98.26%	98.95%
Ethernet/IP	98.31%	98.52%

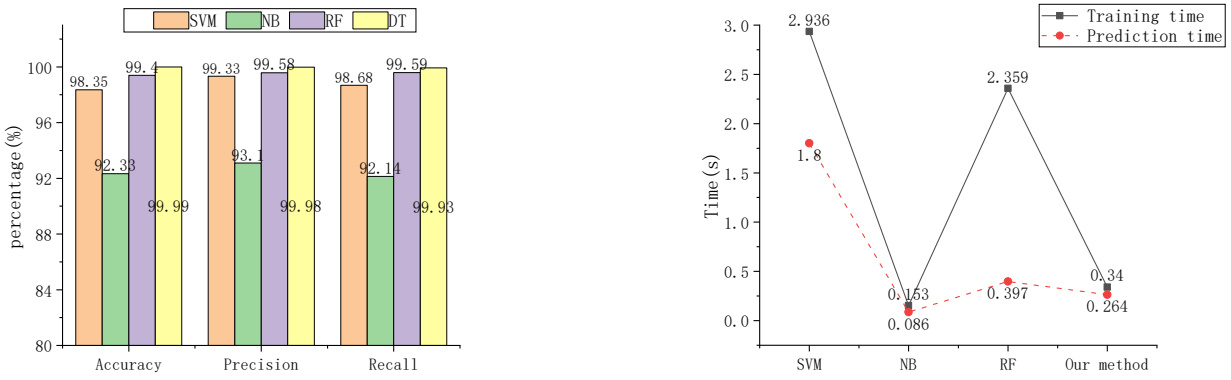
Table 6. Protocol classification precision and recall based on feature dataset.

Protocol Category	Precision	Recall
IEC 61850-MMS	100%	99.87%
Ethernet/IP	99.96%	99.98%

6. Discussion

6.1. Classification Effect of Different ML Algorithms

We also trained other ML classifiers using the feature dataset, including support vector machine (SVM), naive Bayes (NB), and random forest (RF). The results are shown in Figure 8 and the precision and recall rate of a single category are shown in Tables 7–9. SVM regards each set of data as a point in the p-dimensional space (p is the number of features), tries to construct a (p-1)-dimensional hyperplane, and uses this plane to separate points belonging to different categories. The method can achieve high accuracy, but it is computationally expensive and performs poorly in training time and predicting time. NB is based on the Bayesian principle and uses the knowledge of probability and statistics to classify the samples. The process of calculation is simple and the time of calculation is small, but it assumes that the features are independent of each other, which has a bad impact on the accuracy. The RF is composed of multiple decision trees, and the features selected when building each decision tree randomly. Finally, the results of the decision sub-trees are aggregated by voting, so it can achieve high accuracy, but the cost of training time is high.



(**a**) Comparison of accuracy of classifiers (**b**) Comparison of time cost of classifiers

Figure 8. Accuracy and time cost comparison of classifiers based on different ML methods.

Table 7. Protocol classification precision and recall based on SVM.

Protocol Category	Precision	Recall
IEC 61850-MMS	99.26%	98.95%
Ethernet/IP	99.40%	98.01%

Table 8. Protocol classification precision and recall based on NB.

Protocol Category	Precision	Recall
IEC 61850-MMS	92.26%	92.27%
Ethernet/IP	93.94%	92.01%

Table 9. Protocol classification precision and recall cased on RF.

Protocol Category	Precision	Recall
IEC 61850-MMS	99.26%	98.95%
Ethernet/IP	99.99%	99.23%

6.2. Classification Effect of Different Methods in the Literature

To better demonstrate the superiority of our method, we compare our results with the latest research on traffic classification. The performance comparison is shown in Table 10. The training time and predicting time are related to the number of training samples and the number of prediction samples. In order to facilitate comparison, unit processing is performed, and the training time and predicting time occupied by each sample are calculated. The result shows that the method in this paper has the highest accuracy, average precision, and average recall, and has the lowest training time and predicting time.

Table 10. Comparison between different methods.

Methods	Classification Model	Accuracy	Training Time	Testing Time
Literature [15]	DT	82.31%	36.64 µs	35.64 µs
Literature [20]	Tree-RNN	98.98%	999.7 µs	185 µs
Literature [22]	1D-CNN+LSTM	93%	18.3 ms	\
Our method	DT	99.99%	3.04 µs	9.43 µs

The authors in [15] use the linear correlation coefficient (cor) with value ranges from −1 to 1 to represent the linear correlation between two random features, and remove the

feature when $|cor| = 1$, then use the feature dataset to train a decision tree model. However, the two features are usually regarded as strongly correlated when $|cor| > 0.5$. Therefore, this method causes a part of the interference features to be retained, so the performance of this method is not as good as that of the method in this paper in terms of accuracy and time consumption. The authors in [20,22] both use DL to automatically extract features and classify traffic. Both of them achieve high classification accuracy. However, the deep learning algorithm is complicated, which causes a high time cost.

To sum up, the feature extraction method proposed in this paper can better extract features conducive to classification, with lower feature dimensions and no need for manual participation. It is superior to the existing algorithms in terms of classification accuracy and time consumption.

7. Conclusions

This paper proposes a feature extraction method of industrial control protocol, which avoids the shortcomings of traditional machine learning methods that require expert knowledge in feature extraction, and uses the dataset after extraction to train a decision tree classifier, so that it can quickly and accurately classify the protocol traffic in the industrial control network. We use training time, predicting time, accuracy, precision, and recall as evaluation metrics. The experiments are carried out on SWAT dataset using the ethernet/IP protocol and the EPIC dataset using the IEC 61850-MMS protocol. The results show that the method in this paper has good practicability and it has a higher classification accuracy and lower time consumption than the existing traffic classification methods.

However, the method in this paper also has limitations. First, the feature extraction method designed in this paper is based on the periodicity and stability of industrial data flow, so it cannot extract protocol features from aperiodic traffic with irregular features, such as attack traffic. Second, this paper uses a supervised classification algorithm. When training the classifier, the protocol category corresponding to all data needs to be known in advance, so it cannot classify unknown protocol traffic. In the follow-up research, the unsupervised classification technology will be combined, which can improve the ability to classify the protocols with unknown categories and irregular features, and further improve the practicability of ICS traffic classifiers.

Author Contributions: Conceptualization: C.Y.; data curation: Z.Z.; investigation: Z.Z.; methodology: C.Y.; software: Z.Z.; validation: Z.Z.; writing—original draft: Z.Z.; writing—review and editing: C.Y. and M.G. All authors have read and agreed to the published version of the manuscript.

Funding: The project is supported by the National Natural Science Foundation of China (grant No.61871468); Zhejiang Province key R&D Program (grant 2017C01G2050953).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: (https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/ (accessed on 1 November 2022)).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
ICS	Industrial control system
SCADA	Supervisory control and data acquisition
DPI	Deep packet inspection
ML	Machine learning
DL	Deep learning
TCP/IP	Transmission control protocol/internet protocol
SVM	Support vector machines
NB	Naive bayes
RF	Random forest
DT	Decision tree

References

1. Zhou, S.; Wang, S.J. Research on classificati-on method of private industrial control protocol. *Inf. Technol. Netw. Secur.* **2021**, *40*, 19–24.
2. Dainotti, A.; Pescape, A.; Claffy, K.C. Issues and future directions in traffic classification. *IEEE Netw.* **2012**, *26*, 35–40. [CrossRef]
3. Moore, A.W.; Papagiannaki, K. Toward the accurate identification of network applications. In Proceedings of the International Workshop on Passive and Active Network Measurement, Boston, MA, USA, 31 March–1 April 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 41–54.
4. Khandait, P.; Hubballi, N.; Mazumdar, B. Efficient keyword matching for deep packet inspection based network traffic classification. In Proceedings of the 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bangalore, India, 7–11 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 567–570.
5. Gu, Y.; Li, D.; Gao, K.H. Research on network traffic classification based on machine learning and deep learning. *Telecommun. Sci.* **2021**, *37*, 105–113.
6. Pacheco, F.; Exposito, E.; Gineste, M.; Baudoin, C.; Aguilar, J. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1988–2014. [CrossRef]
7. Dong, S. Multi class SVM algorithm with active learning for network traffic classification. *Expert Syst. Appl.* **2021**, *176*, 114885. [CrossRef]
8. Li, J.; Pan, Z. Network traffic classification based on deep learning. *KSII Trans. Internet Inf. Syst. (TIIS)* **2020**, *14*, 4246–4267.
9. Shapira, T.; Shavitt, Y. FlowPic: A generic representation for encrypted traffic classification and applications identification. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1218–1232. [CrossRef]
10. Wang, X.; Lv, K.; Li, B. IPART: An automatic protocol reverse engineering tool based on global voting expert for industrial protocols. *Int. J. Parallel Emergent Distrib. Syst.* **2020**, *35*, 376–395. [CrossRef]
11. Ni, J.; Yin, W.; Jiang, Y.; Zhao, J.; Hu, Y. Periodic mining of traffic information in industrial control networks. In Proceedings of the International Conference on Advanced Information Networking and Applications, Caserta, Italy, 15–17 April 2020; Springer: Cham, Switzerland, 2020; pp. 176–183.
12. Nguyen, T.T.T.; Armitage, G. A survey of techniques for internet traffic classification using machine learning. *IEEE Commun. Surv. Tutor.* **2008**, *10*, 56–76. [CrossRef]
13. Cao, J.; Wang, D.; Qu, Z.; Sun, H.; Li, B.; Chen, C.-L. An Improved Network Traffic Classification Model Based on a Support Vector Machine. *Symmetry* **2020**, *12*, 301. [CrossRef]
14. Jiang, J.R.; Chen, Y.T. Industrial Control System Anomaly Detection and Classification Based on Network Traffic. *IEEE Access* **2022**, *10*, 41874–41888. [CrossRef]
15. Aouedi, O.; Piamrat, K.; Parrein, B. Performance evaluation of feature selection and tree-based algorithms for traffic classification. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
16. Mokhtari, S.; Abbaspour, A.; Yen, K.K.; Sargolzaei, A. A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electronics* **2021**, *10*, 407. [CrossRef]
17. Lan, H.; Zhu, X.; Sun, J.; Li, S. Traffic data classification to detect man-in-the-middle attacks in industrial control system. In Proceedings of the 2019 6th International Conference on Dependable Systems and Their Applications (DSA), Harbin, China, 3–6 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 430–434.
18. Ling, J.; Zhu, Z.; Luo, Y.; Wang, H. An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit. *Comput. Electr. Eng.* **2021**, *91*, 107049. [CrossRef]
19. Lin, K.; Xu, X.; Gao, H. TSCRNN: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of IIoT. *Comput. Netw.* **2021**, *190*, 107974. [CrossRef]
20. Ren, X.; Gu, H.; Wei, W. Tree-RNN: Tree structural recurrent neural network for network traffic classification. *Expert Syst. Appl.* **2021**, *167*, 114363. [CrossRef]

21. Mendonca, R.V.; Silva, J.C.; Rosa, R.L.; Saadi, M.; Rodriguez, D.Z.; Farouk, A. A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms. *Expert Syst.* **2022**, *39*, e12917. [CrossRef]
22. Zhai, L.; Zheng, Q.; Zhang, X.; Hu, H.; Yin, W.; Zeng, Y.; Wu, T. Identification of Private ICS Protocols Based on Raw Traffic. *Symmetry* **2021**, *13*, 1743. [CrossRef]
23. Dai, H.; Li, H.; Chen, C.S.; Shang, W.; Chen, T.H. Logram: Efficient log parsing using N-Gram dictionaries. *IEEE Trans. Softw. Eng.* **2022**, *48*, 879–892. [CrossRef]
24. Lei, Q.; Li, H.; Wei, R. Leveraging Zipf's Law to Analyze Statistical Distribution of Chinese Corpus. In Proceedings of the 2021 IEEE International Conference on Software Engineering and Artificial Intelligence (SEAI), Xiamen, China, 11–13 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
25. Yu, T.T.; Xu, P.N.; Jiang, Y.E. Text similarity method based on the improved Jaccard coefficient. *Comput. Syst. Appl.* **2017**, *26*, 137–142.
26. Cui, Y.; Bao, Z.Q. Survey of association rule mining. *Appl. Ions Res. Comput.* **2016**, *33*, 330–334.
27. Charbuty, B.; Abdulazeez, A. Classification based on decision tree algorithm for machine learning. *J. Appl. Sci. Technol. Trends* **2021**, *2*, 20–28. [CrossRef]
28. Mathur, A.P.; Tippenhauer, N.O. SWaT: A water treatment testbed for research and training on ICS security. In Proceedings of the 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), Vienna, Austria, 11 April 2016; pp. 31–36.
29. Singapore University of Technology and Design (SUTD). Electric Power and Intelligent Control (EPIC) Testbed. Available online: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_epic/ (accessed on 13 January 2021).