symmetry

MDPI

Article

Identification of Private ICS Protocols Based on Raw Traffic

Liang Zhai, Qiuhua Zheng *, Xu Zhang, Haizhong Hu, Weihao Yin, Yingpei Zeng and Ting Wu *

School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou 310018, China; zhailiang125@hdu.edu.cn (L.Z.); zhangxu@hdu.edu.cn (X.Z.); huhaizhong@hdu.edu.cn (H.H.); yinweihao@hdu.edu.cn (W.Y.); yingpeizeng@hdu.edu.cn (Y.Z.)
* Correspondence: zqh@hdu.edu.cn (Q.Z.); wuting@hdu.edu.cn (T.W.)

Abstract: With the development of the Industrial Internet in recent years, security issues have been a hot topic of the industrial control system (ICS) network management. Identifying the protocol traffic in the communication process of the ICS is an important prerequisite to avoid security problems, especially in ICSs that use many private protocols. The private protocols cannot be analyzed due to the unknown internal structure of the protocols, which makes the ICS protocol identification work more difficult. However, the Internet-oriented protocol identification method is not applicable to the scenario of the private ICS protocols network environment. With this problem in mind, this paper proposes a method of ICS protocol identification based on the raw traffic payload. The method firstly performs data preprocessing such as data selection, interception, cleaning conversion, and labeling on the raw traffic of the protocol based on the characteristics of the industrial control protocol. Then it uses an AM-1DCNN + LSTM deep learning model to extract temporal and spatial features of the ICS raw traffic, and performs protocol identification. This method can effectively extract ICS protocol features in scenarios where protocol parsing is impossible compared with existing methods. We constructed a dataset for ICS protocol identification based on open-source data and tested the proposed method for experiments, and the identification accuracy rate reached 93%.

Keywords: industrial control system; raw traffic; payload; 1D-CNN; LSTM

Citation: 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. https://doi.org/10.3390/sym13091743

Academic Editor: Kuo-Hui Yeh

Received: 12 August 2021
Accepted: 17 September 2021
Published: 19 September 2021

1. Introduction

The Supervisory Control and Data Acquisition (SCADA) in ICS is inseparable from communication protocols. With the development of the times, from RS232/485 to Industrial Ethernet to Industrial Real-Time Ethernet, Ethernet has been introduced in a large number of ICSs and transmitted using TCP/IP or ISO standard encapsulation, but since there is no unified standardized specification from equipment manufacturers, there are a large number of private communication protocols in the industry, such as Modbus, DNP3, S7comm, etc. The Modbus protocol, for example, is a common language used in electronic controllers. Through this protocol, controllers can communicate with each other and with other devices via a network (e.g., Ethernet). It is commonly used for communication in the oil and gas industry, for providing flow and pressure data to PLCs via RTUs and sensors, for PLC operation of safety protection systems and well control systems, etc. The ICS protocol identification can be used for network user asset discovery, Quality of Service (QoS) management, and network traffic composition analysis [1], which plays an important role in the ICS network.

Current methods for ICS network protocol identification are the same as application-layer protocol identification [2]. They can mainly be divided into rule-based methods and methods based on machine learning [3]. The former includes port-based, payload-based, deep packet inspection (DPI), etc. These methods are manually configured based on empirical knowledge, and then deep decoding and feature matching are performed at L2–L4. In addition to the previous layer analysis, DPI adds application layer analysis to identify protocols which are simple and easy to deploy. However, the methods cannot

identify the protocols of encrypted and unknown traffic. The latter include machine learning (ML) and deep learning (DL) methods. They use optimization engines to increase accuracy and can automatically fit new patterns. The methods mainly include algorithms such as support vector machine (SVM), random forest (RF), and K nearest neighbor (KNN). However, the performance of the ML-based method depends mainly on the features of the manual design. It limits the universality of the ML-based method. Especially in unknown protocols, features are difficult to extract. Deep learning methods have avoided the disadvantages of ML methods. Deep learning methods do not need to use professional knowledge to design data features. In contrast, they have a fair ability to learn these highly complex patterns. Now, methods based on deep learning have been the most popular traffic identification techniques [4]. However, the common application-layer protocol identification methods mainly focus on multimedia traffic, internet application software protocols, etc. Unlike Internet protocols, the ICS traffic is periodic and stable and has a relatively shorter payload. Therefore, we can take advantage of the symmetry between the Internet-oriented and the ICS-oriented identification methods. The asymmetry between the structure and characteristics of both is then exploited to design identification methods suitable for ICS traffic. It does not directly transplant Internet-oriented methods to the ICS protocol identification [5]. Aiming at this problem, we propose a preprocessing method and a deep learning model of protocol identification based on the raw ICS traffic payload. For ICS raw traffic preprocessing, the proposed method combines ICS traffic data sequence characteristics and data structure distribution to build four preprocessing steps. The DL model refers to automatic learning and effective feature representation of data. The proposed model utilizes attention mechanism (AM), one-dimensional convolutional neural network (1DCNN), and long short-term memory network (LSTM) to learn high-level data representations of raw data. The network model extracts high-level features from the sequence structure of space and time. The proposed model is implemented and then an experimental evaluation is based on the ICS traffic data. The results demonstrate the superiority of the proposed model in the identification of the ICS protocols.

The rest of the paper is organized as follows. Section 2 introduces the related works of protocol traffic identification. Section 3 proposes the method model and introduces the data processing and the proposed model in detail. Section 4 presents the experimental results and analyzes them. Section 5 discusses the model data parameters. Finally, Section 6 concludes this paper.

2. Related Works

2.1. Methods Based on Rules

The port-based methods are based on the port-protocol comparison table provided by the Internet Assigned Numbers Authority (IANA) [6]. The port-based methods infer the protocol type of network traffic by analyzing the port number of the data packet. However, with the increase of network protocols, particularly the emergence of many private protocols and the customization of protocol ports, it is difficult for the port protocol identification method to be effective. Moore et al. [7] and Madhukar et al. [8] respectively verified that the identification accuracy of the port-based method had been reduced from 70% to less than 20%. Another method is based on deep packet inspection (DPI) [9,10], which uses the regular expression to match the packet's character string and perform protocol identification. Sen et al. [11] realized the classification of P2P traffic by checking the data packet payload to identify application characteristics. The accuracy of this DPI method was three-times greater than the port-based method on the same dataset. However, the method identification effect is gradually reduced with the emergence of encrypted and private protocol traffic and the encoding data feature.

2.2. Methods Based on Traditional ML

The methods based on traditional ML identify the traffic protocol by classifying and predicting the measurable parameters, such as time, packet length, and interval in the

protocol [12]. ESTE et al. [13] proposed a traffic classification approach based on SVM, which solves multi-class problems with SVMs in statistical traffic classification, and uses a simple optimization algorithm that allows the classifier to perform correctly with as little training as a few hundred samples. The result confirms that SVM-based classifiers can be very effective at discriminating traffic generated by different applications. Zhou et al. [14] proposed a traffic detection method that uses a feedforward neural network. The method combined with fast correlation-based feature selection filters and elimination of accessing the contents of the packets. Alshammari et al. [15] uses various supervised learning methods (such as C4.5, AdaBoost, GA, SVM, RIPPER, Naïve Bayes) to identify encrypted and non-encrypted SSH and Skype traffic with a high detection rate and low false-positive rate. However, the ML methods still require expert knowledge to design data features in advance, and will cause concept drift problems due to differences in the distribution of services carried by traffic in different time periods and in different regions [16].

2.3. Methods Based on Deep Learning

Deep learning can learn from massive amounts data and obtain high-level features directly from the data, reduce the complexity caused by feature processing and reliance on expert knowledge and solve deficiencies of the classical ML. Many deep learning methods have been applied, and the performance is beyond classical ML [17]. Wang Z. et al. [18] collected the first 1024 bytes of TCP sessions and used ANN and SAE to solve network protocol identification. This method first converts the data into a one-dimensional vector and then inputs the vector into the artificial neural network model for training. The method is evaluated based on the collected known protocol data, and the average accuracy rate is 90.9%. However, this method uses TCP sessions as the data format, ignoring the correlation between single data packets. Ma et al. [19] proposed a traffic identification method based on CNN which cannot identify known traffic and unknown traffic. The method was tested on data composed of 13 protocols. The test results are compared with the traditional ML algorithms of the SVM and Naive Bayes classification model and got higher accuracy than those methods. Still, this method only uses the payload as the test data, and discards the packet header data, which reduces the identification accuracy and only achieves an accuracy of 85%. CNN models have also been used to identify malware traffic identification. Wang W. et al. [20] regards traffic data like images and unusual patterns classified by malware traffic exhibits by representation learning. Nevertheless, its method only uses the spatial feature of the flow and completely ignores the temporal feature. The rule-based method has the advantages of being fast and convenient, but it is difficult to detect unknown and encrypted traffic. The method based on classic ML can fit the data pattern and identify and predict the traffic, but it depends on expert knowledge to design the features. Deep learning methods are widely used in traffic protocol identification problems and have achieved excellent performance. However, different methods and models are required due to the specific characteristics of different scenarios and data, and a single transplantation method cannot be used to solve various problems.

3. Protocol Identification Method

The method is mainly divided into the following steps: data collection, data preprocessing, model training, and protocol identification. As shown in Figure 1, it firstly collects the ICS raw traffic, then builds the dataset. Secondly, it performs the data preprocessing steps and then divides the preprocessed data into a training set, a validation set, and a test set. The data in the training set and the validation set need to be labeled. In addition, it is using the proposed AM-1DCNN + LSTM to train the model on the training set and the validation set.

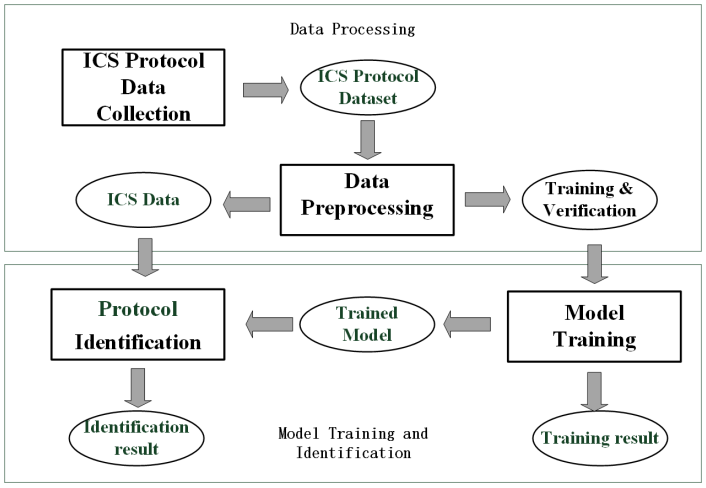


Figure 1. Method architecture.

3.1. Data Processing

The data preprocessing converts the raw traffic into an input format that the neural network can use. As shown in Figure 2, it includes four steps: selection of data format, data packet interception, data randomizing and conversion, and data labeling.

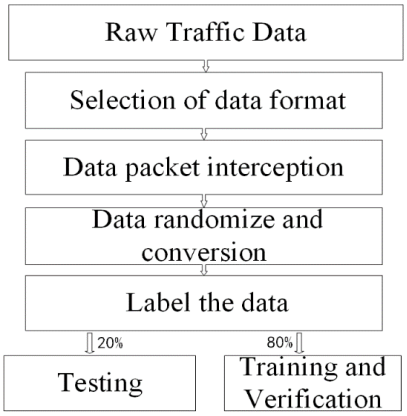


Figure 2. Data preprocessing.

3.1.1. Selection of Data Format

For identifying the ICS protocol, it is first necessary to divide and select the captured raw protocol traffic data. Now, the conventional division forms include flow, session, and data packet [21]. The flow refers to all packets with the same 5-tuple (source IP, source port, destination IP, destination port, protocol). The session refers to all packets composed of bidirectional flow (source and destination interchange), and the data packet form refers to the selected single data packet. We choose the single data packet form to combine the TCP payload to extract feature protocol identification and conveniently get the payload part of each data packet.

Most ICS protocols are based on TCP/IP communication. During the process, many TCP establishment packets, confirmation establishment packets, and connection confirmation packets are generated, and a large number of repeated packets and bad packets. Because these data packets have a little positive impact on feature extraction and protocol identification [22], we have removed them.

3.1.2. Uniform Data Length

In general, the length of the ICS protocol data is different. We investigated some of the ICS protocols and found that the length of the ICS protocol header is mostly concentrated

at 10–40 bytes. The remaining is the payload data part of the industrial control protocol. Figure 3 is a schematic diagram of the ICS packet's structure.

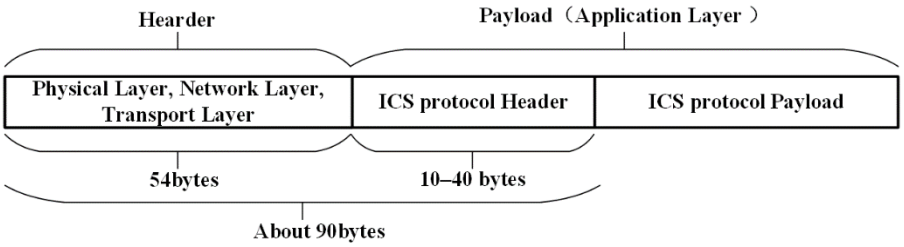
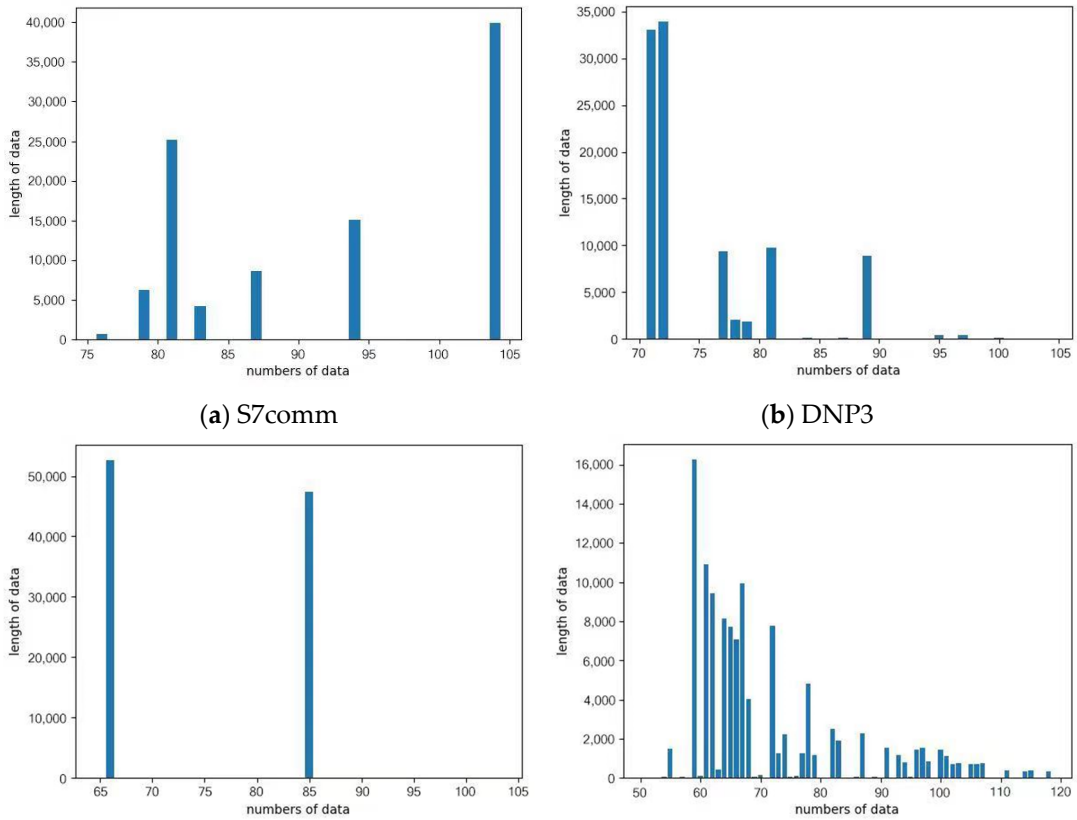


Figure 3. Packet's structure diagram.

To further verify the basis of the effective length of the data interception, we analyze each protocol packet's length and the number of traffic data packets in our dataset. As shown in Figure 4c, while the lengths of the Modbus packet are relatively uniform, concentrated at 66 and 85 bytes, the BACnet (Figure 4d) packet lengths are widely distributed. From the overall perspective of the experimental dataset, the length distributions in the four figures are different. Still, they concentrate on the 60–120 bytes interval, according to the above ICS protocol length division schematic and the statistical results of the experimental dataset length. This paper finally intercepts the TCP header, the ICS protocol header, and part of the ICS protocol payload, and the first 90 bytes of raw traffic data packets to more accurately and completely characterize the traffic protocol. Data packets longer than 90 bytes are truncated, and for the data packet of less than 90 bytes they are filled with 0.



(**a**) S7comm (**b**) DNP3

(**c**) Modbus (**d**) BACnet

Figure 4. Packet numbers distribution diagram.

3.1.3. Data Randomization and Conversion

Randomization and data conversion are important steps in data preprocessing. Unlike Internet protocol identification, training data for ICS protocol identification models are usually collected from a limited number of specific ICS environments. Protocol-specific traffic is usually collected from a limited number of specific devices. Thus, the MAC and IP addresses of particular protocol traffic usually vary less. If traffic data is directly used for training, it may lead to the over-fitting phenomenon in model training, and the protocol MAC address and IP address are relied on for the protocol. This paper proposes a solution to randomize the device's MAC address and IP address to avoid this problem. This method replaces the first 8 bytes and the 18th to 26th bytes of the packet with 16 bytes generated randomly. Through randomization, we obtained 90 bytes of hexadecimal character data, including randomization. This data needs to be converted to 180-bit numeric data bit by bit. e.g., 2C→2 12.

3.1.4. Label the Training Data

When constructing the training data set, protocol labels should be added to each row of data to represent the protocol type in the row after data cleaning and parsing.

During model training, the discrete-type protocol labels are encoded in a one-hot way so that the feature can be mapped to Euclidean space [23], making subsequent protocol identification more convenient and data easier to process.

3.2. ICS Protocols Identification Model

Kelvin Xu et al. [24] pointed out that the ICS networks are usually constructed for specific production businesses, unlike the Internet. Therefore, it has periodic and stable temporal characteristics. At the same time, the form of a single data packet selected in this paper is a one-dimensional sequence, which is similar to the common text data form. Therefore, to extract the spatial text features of a single data and the periodic and stable time series features between multiple data frames, we choose a one-dimensional convolutional neural network and a long short-term memory neural network as the core of the architecture. Furthermore, we proposed a one-dimensional convolutional neural network model combined with attention mechanism + long and short-term memory network (AM-1DCNN + LSTM). The model focuses on fine-grained spatial-temporal feature extraction for training and identification. The model is shown in Figure 5 in detail.

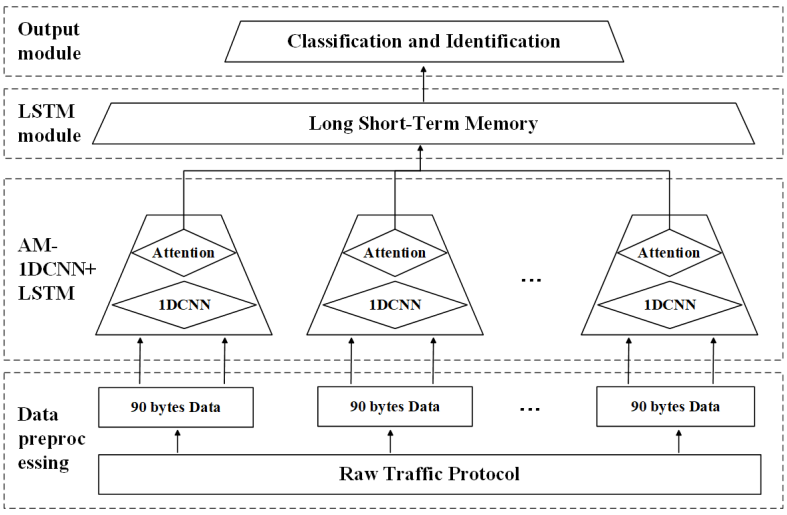


Figure 5. Model illustration.

This model comprises a convolutional neural network combined with an attention mechanism and a long and short-term memory network. The model is divided into an Input unit, AM-1DCNN unit, LSTM unit, and an Output unit.

1. Input unit. It directly inputs the preprocessed data into the model network.
2. AM-1DCNN unit. CNN mainly includes the convolutional layer and pooling layer. The convolutional layer is composed of several convolutional units. The parameters of the convolution unit are all optimized through the back-propagation algorithm. The purpose of the convolution operation is to extract different features of the input. The pooling layer performs a subsampling operation on the convolution output to retain strong features and remove weak features while reducing the parameters to prevent overfitting. 1DCNN can be better applied to one-dimensional sequence data. The raw traffic data selected in this paper also is a one-dimensional sequence, so we extracted the spatial correlation feature of the data through 1DCNN. Combining the Attention Mechanism can make the 1DCNN network have the ability to focus on its input subset features. Select a specific input and weight all input features one by one so that 1DCNN can extract features with emphasis, thereby improving the effectiveness of 1DCNN for extracting spatial features.
3. LSTM unit. It uses the output of the previous unit as the input of this unit. The multiple consecutive raw traffic packets of the ICS have very strong periodicity, persistence, and other time-related features. Therefore, it is equally important to extract the relationship between the multiple data packets. Introducing the LSTM gate function (input gate, forget gate, output gate) can mine the relatively long intervals and delays in the time series changes [25] and extract the related time series feature relationships from multiple input data. Effectively improving the accuracy of time feature extraction.
4. Output unit: It uses the SOFTMAX activation function to classify the hidden layer features of the raw traffic data extracted from the AM-1DCNN + LSTM model and output the classification and identification results.

4. Experiment

4.1. Experiment Data Collection

Although some ICS network traffic data is publicly available, there is still a lack of available ICS traffic data sets for protocol identification due to the closed and limited nature of the ICS proprietary protocol. Therefore, we collected some industrial protocol traffic data from data sharing sites like Netresec, 4SIC, Shodan, and GitHub [26]. We set the criteria for data collection, preferably real industrial control business data, followed by simulation data of large experiments. Its data are collected in the same period, so that the data have strong temporal and spatial correlation. Finally, the data were collected for the last five years. The collected data are sourced from the ICS village at 4SICS, Digital Bond S4 $\times$ 15 ICS Village CTF PCAPs, and OT and IT protocols used in Industrial Control System (by ICS Defense/ICS Savunma), etc. Where the data are dated between 2015 and 2018. (https://www.netresec.com/?page=pcapfiles (accessed on 15 September 2021). A portion of the data were simulated by ICS simulation software. The researchers generated network traffic packets by creating master and slave station programs to simulate real industrial equipment environments, then captured by Wireshark. Part of the traffic data is collected in the real industrial control environment. In this experiment, protocol traffic data of Modbus, DNP3, S7Comm, and BACnet protocols were collected, totaling about 400,000 pieces of raw traffic data, merging data of the same category through Wireshark software. In this paper, the collected data of four public ICS private protocols are treated as unknown private protocols without analyzing the ICS protocol data.

4.2. Experimental Settings and Metrics

Our experiment implemented all the programming work in Anaconda (Python 3.7) and Keras and TensorFlow deep learning framework using PyCharm as IDE. The computer comprises an Intel core i-5 processor@ 3.20 GHz, 8 GB RAM, and a 64-bit Windows 10 based OS. The graphics card is GeForce GTX 1060 6 GB. The evaluation's metrics mainly include

the overall metrics accuracy (Accuracy) and single category identification performance metrics (*Precision*, *Recall*, *F1 value F1-Measure*). The metrics are defined as follows:

$$Accuracy = (TP + TN)/(TP + FN + FP + TN), \quad (1)$$

$$Precision = TP/(TP + FP), \quad (2)$$

$$Recall = TP/(TP + FN), \quad (3)$$

$$F1 - score = 2P \cdot R/(P + R) \quad (4)$$

4.3. Model Parameter Tuning

The training parameters need to be tuned according to the experience and expert knowledge in the training process. These parameters will not change with the iteration of the algorithm model:

- Learning rate: The learning rate determines whether the loss function can converge appropriately and at convergence speed. The learning rate is adjusted to 0.001 in this paper.
- Mini-batch: In the model row training, the data is divided into multiple identical blocks, and the weight and bias are updated for a single mini-batch each time. We set a mini-batch to 100.
- Epoch: If the epoch setting is too small, the network will not have enough data and time to train to obtain the optimal parameters. If the epoch is set too large, it will make the network training over-fitting. Due to the large amount of training data in this experiment, the epoch is set to 20.

4.4. Experimental Design and Analysis

We were aiming at the feasibility of the length interception and the data randomization in data preprocessing. In the Figure 3 Packet's structure diagram, we had confirmed the data statistics basis of this processing method, but this method is not persuasive. Therefore, this paper is to prove the realistic correctness of the method. We design related experiments to prove the experimental basis of this method further. The experiment designs four scenarios for comparing the existence of TCP/IP payload and the influence of randomization of some data in the header on the experimental classification results. The experimental design scenarios are as follows:

- Scenario 1: It uses the TCP/IP payload and randomizes part of the header.
- Scenario 2: It uses the TCP/IP payload and not randomizes part of the header.
- Scenario 3: It does not use the TCP/IP payload but randomizes part of the header.
- Scenario 4: It does not use the TCP/IP payload and randomizes part of the header.

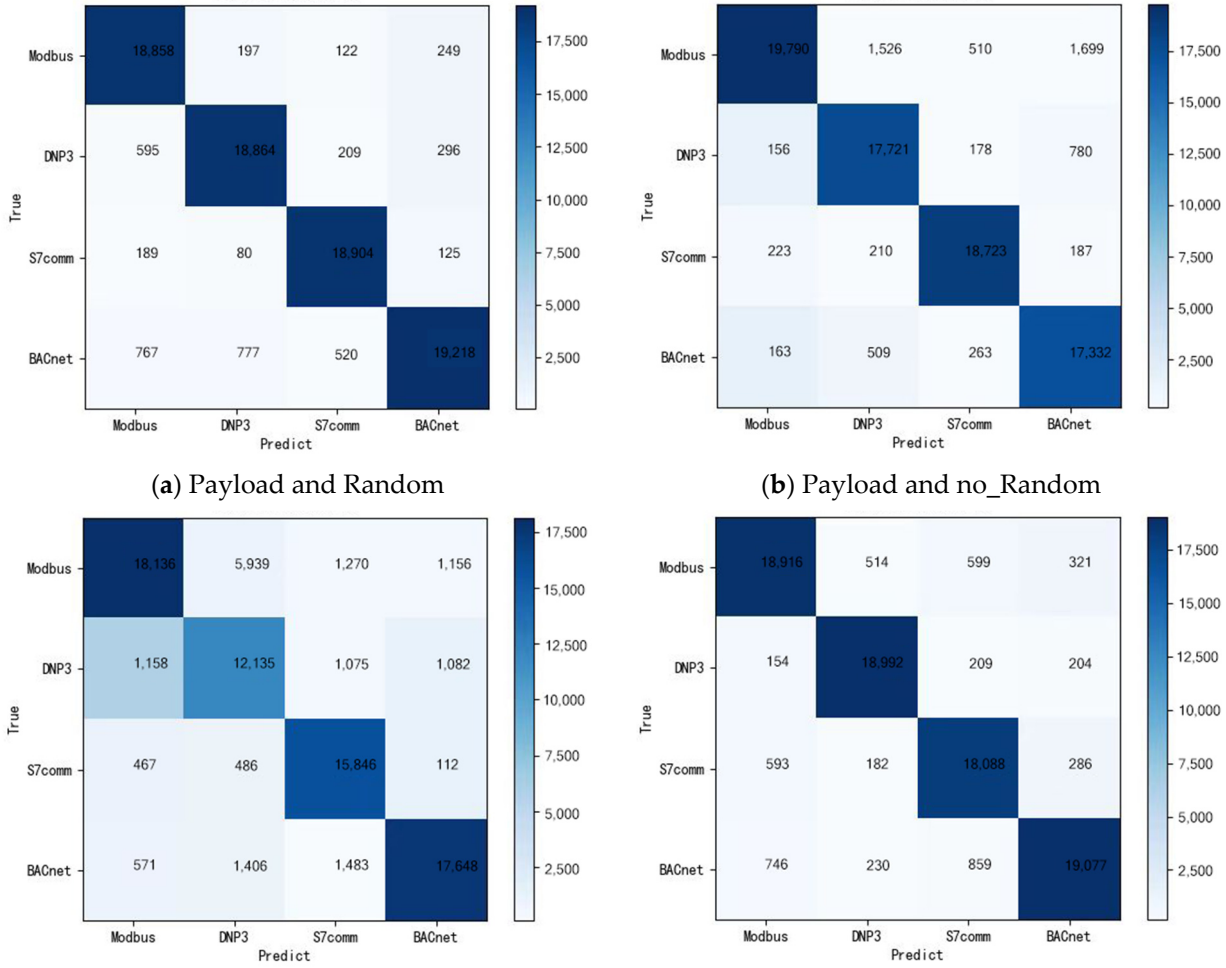
The AM-1DCNN + LSTM model proposed in this paper is used for experiments under four different programs for the above-designed experimental program.

From the results in Table 1, the accuracy of Scenario 1 is 12% higher than Scenario 3. Moreover, the identification accuracy of payload data is higher than those without payload, at least 3% higher. The result shows that part of the payload data can improve the accuracy of classification and identification. However, it can also be seen that the accuracy of Scenario 4 is reduced by 13% compared with Scenario 3. After randomizing the raw data header part of the data in the experiment, the identification accuracy has decreased. The above results confirm our preprocessing method at the data randomize and conversion part. Because the raw data has only fixed device IP and MAC, it leads to a one-to-one correspondence between the MAC address, the IP address, and the various protocols, which positively affect identification. However, the MAC address, IP address, and other raw data information we get come from completely different devices in the real industrial control environment. Therefore, to be more real and close to a real ICS environment, this experiment is designed to randomize features such as MAC and IP. Figure 6 is the confusion

matrix of the identification results of the model under the four scenarios, a more intuitive display.

Table 1. Comparison of results under different scenarios.

Scenario	Accuracy	Training Time (mins)
Scenario 1	92.891%	122.5
Scenario 2	94.840%	121.9
Scenario 3	80.736%	100.08
Scenario 4	86.876%	99.77



(**a**) Payload and Random (**b**) Payload and no_Random

(**c**) no_Payload and Random (**d**) no_Payload and no_Random

Figure 6. Confusion matrix of the identification results.

It can be seen from the confusion matrix of the identification results that the model proposed in this paper achieve accurate identification results for the four different types of protocol raw traffic data. On the whole, the identification accuracy between different protocols is high, and the false alarm rate is low. However, it can be seen from Figure 6c that after Scenario 3 does not include payload data and randomizes part of the header data, the MODBUS and DNP3 classification results have higher false alarms, it predicts a large number of MODBUS as DNP3. It shows that when the payload is not included, the dimension of the data feature is reduced, and randomization will weaken the connection between data features, resulting in a reduction in the effect of the classification result.

4.5. Model Result Analysis

To more convincingly prove the method's effectiveness proposed in this paper in identifying the ICS protocol traffic, we also compare the performance of our proposed method with other existing protocol traffic identification methods. Ren J. [27] et al. proposed an autoencoder method to automatically extract the raw protocol data features in protocol traffic identification. M. Kim [28] proposed a method to classify Tor traffic using the raw data packet header and convolutional neural network model, using 1DCNN classification training. We reproduced their method and applied it to the dataset proposed in this article. Therefore, we selected the following three models for comparative experiments: Autoencoder, 1DCNN, AM-1DCNN, and AM-1DCNN + LSTM; and experiments were carried out on different data preprocessing methods. The following results are obtained.

It can be seen from Figure 7 that there are obvious differences in the experimental results under different models and different scenarios. In the two scenarios that include a payload, the randomization processing under the Autoencoder model has a greater impact on the experimental results. It leads to a 30% difference in the results of the two scenarios. The reason is that the randomization destroys the coding structure of the data to decrease the feature extraction ability of Autoencoder. The identification accuracy of 1DCNN and AM-1DCNN under this scenario has been improved compared with Autoencoder. However, when the payload is not included and randomized, the accuracy of the three models is at a low level and can only reach an accuracy of about 50%. When the payload is included and not randomized, the first three models can achieve 70–80% accuracy. Compared with the model mentioned above, the model proposed in this paper increases the extraction of time-series features. Without including the payload and randomizing, the accuracy rate is increased by 40%. In the other three scenarios, the accuracy rate also increased by 10%. The first three models are mainly used to extract the correlation between data encoding and spatial structure features. When the payload is not included and randomized, the correlation of spatial features is reduced, resulting in reduced identification accuracy. At the same time, the ICS traffic data has strong temporal characteristics and is less affected by randomization, which can effectively improve identification accuracy.

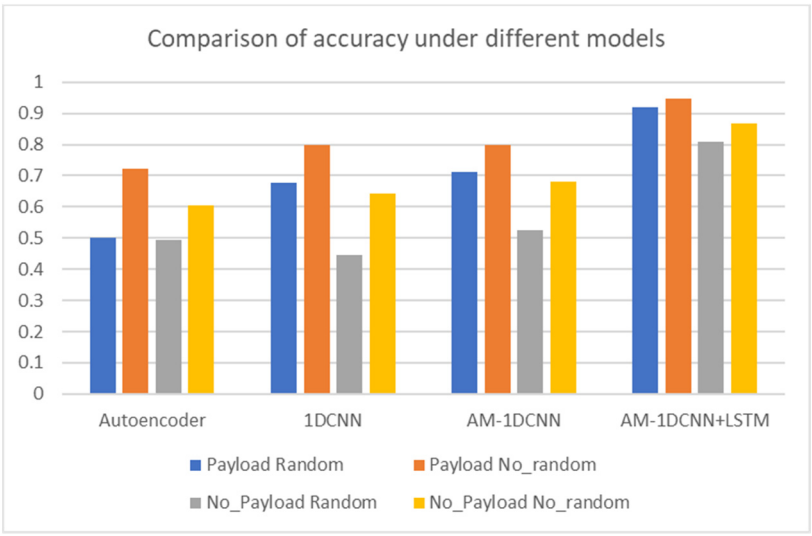


Figure 7. Comparison of accuracy under different model.

This paper also compares with other methods based on Scenario 3 (data with payload and randomization of the header field). The experiment calculated the accuracy, recall, F1-score of the data under each model and the time spent training 20 epochs. The experimental metrics are shown in Figure 8.

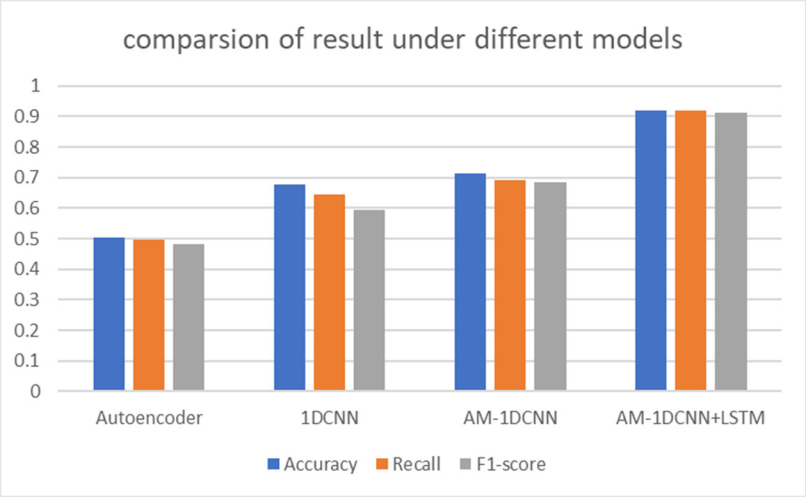


Figure 8. Comparison of results under different model.

From the evaluation metrics of Figure 8 different models, it can be seen that the method we propose is better than the existing protocol traffic identification methods to a large extent. Our proposed method can achieve up to 93% accuracy, 91% recall, and 92% F1-score. In the classification results of different models, the identification effect of autoencoders is poor, with an accuracy of only about 50%. Comparatively, 1DCNN extracts the spatial feature in the traffic data better than the Autoencoder, with higher accuracy, recall, and F1-score. After 1DCNN combines the attention mechanism, its classification effect is slightly improved. The attention mechanism effectively improves the feature weight of 1DCNN at the output and better reflects the importance of the feature. After adding LSTM, while extracting the spatial feature of a single-frame data packet, the temporal-relevant feature between multi-frame data packets is extracted. The fine-grained fusion of spatiotemporal features and more comprehensively described data features are achieved. We can see that the model method proposed in this article has greatly improved the accuracy rate, recall rate, and F1 score through experimental comparison [29,30].

5. Discussion

5.1. Effect of Different Packet Lengths on the Results

In data packet interception steps for experimental data preprocessing, the length of each traffic data packet is unknown and varied, and data packets cannot be intercepted according to certain standards. However, the determination of the packet length has a profound impact on the experimental results. For this purpose, this article sets data packets of different lengths: 54, 60, 70, 80, 90, and 100. Experiments are conducted to verify the influence of data packet length on model recognition. The experiment obtains the experimental results of the accuracy rate and the corresponding training time under different lengths. The results are shown in Table 2.

Table 2. Comparison table of results of different data lengths.

Data Length (bytes)	Accuracy	Training Time (mins)
54	79.736%	90.08
60	85.218%	89.19
70	88.046%	102.8
80	90.884%	110.34
90	93.091%	122.51
100	92.411%	135.74

As shown in Table 2, with the length of the data packet increases, the classification accuracy and training time also increase. When the data packet length reaches 90 bytes, the accuracy and time reach the optimal balance. As described in the section on data packet interception in the 4.2 data preprocessing section, most ICS protocols are based on the TCP/IP protocol cluster. The first 54 bytes are the IP header, TCP/IP header feature, and the remaining 36–46 bytes are TCP/IP payload, the ICSs protocol. This part contains the industrial control protocol header and part of the payload. This feature is enough to classify and identify the data. It can also be seen from the experimental results that if the packet length is too short, part of the data payload feature will be lost, resulting in lower accuracy of the classification and identification results. However, as the length gradually increased to more than 90 bytes, the accuracy did not increase but reached a threshold. Since the data packet longer than 90 bytes is the payload data of the industrial control protocol, this data has less of an impact on the data classification. Therefore, 90-byte data packets reached the highest accuracy. The interception of a longer data packet can only increase the training time and does not positively impact classification accuracy.

5.2. Effect of the Number of Different Data Packets on the Result

The raw traffic of the ICS has a strong periodicity, and there is a temporally relevant feature among multiple consecutive frames of the data packet. Therefore, when constructing the model's input dimensions, we select multiple data packets to input simultaneously to extract the feature between multiple consecutive data packets. The number of input data packets at the same time affects the feature extraction effect of the model. To verify the influence of the number of consecutive data packets, we conduct the following comparative experiments and set the number of data packets as 10, 20, 30, and 40. The experimental results are as follows.

From the experimental results in Table 3, it can be concluded that the accuracy of identification gradually increases when the number of simultaneous input data packets increases from 10 to 30, and the highest accuracy is achieved when the number reaches 30. At this time, as the number of data packets continues to increase, the accuracy rate gradually converges, but the training time continues to increase. Analysis from the experimental results: the LSTM model is used in the experimental model, so the time correlation between multiple subsequent data packets can be extracted. However, too much or too little data packet input will have a greater impact on the feature extraction of LSTM.

Table 3. Comparison table of results of different numbers of data packets.

Data Packet (Number)	Accuracy	Training Time (mins)
10	90.807%	85.45
20	92.248%	99.49
30	93.256%	123.88
40	92.172%	141.31

6. Conclusions

By comparing the structural similarity between ICS traffic and text, making full use of the periodicity and stability of ICS raw traffic, we propose a preprocessing method and a deep learning identification model based on the raw protocol traffic of the ICS. At the same time, according to the developed data collection criterion, the ICS traffic protocol data were collected and preprocessed in online ICS data resources site, and merge them into an ICS data set with four types of protocol data.

On this dataset, we conducted experiments on the method proposed in this paper. We first designed different comparative experiments to prove the correctness of the payload selection and randomization processing, and secondly, we compared our method with other existing methods. From the experimental results, our method can achieve 93% identification accuracy, which is at least 20% better when compared to other methods, establishing the superiority of our data extraction models for data with spatio-temporal

characteristics. In the end, the effect of the length and number of experimental packets on the experimental results is discussed, and the conclusion of correlation between length quantity and results is given.

Overall, deep learning methods for unknown traffic identification can achieve better results while greatly reducing the workload of manual configuration, and in the future, the processing methods and identification patterns in this paper can be applied to more general traffic identification. However, the method proposed in this paper can only identify the raw traffic and cannot identify the type of instructions in the protocol data. Moreover, it cannot identify the abnormal instructions of the traffic.

Therefore, in the follow-up, we will classify the identified protocol traffic by instruction type based on the results of this article and detect abnormal instructions. We are studying related issues in depth with the aim of building a complete end-to-end protocol identification and command identification and intrusion detection defense system based on ICS raw traffic.

Author Contributions: Conceptualization, L.Z.; Methodology, L.Z.; Software, L.Z.; Validation, L.Z.; Investigation, X.Z. and W.Y.; Resources, X.Z. and H.H.; Data curation, L.Z.; Writing—original draft preparation, L.Z.; Writing—review and editing, L.Z. and Q.Z.; Visualization, L.Z.; Supervision, Y.Z., Q.Z., and T.W.; Funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Zhejiang Province key R&D Program (nos. 2020C01078 and 2019C01012).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: (https://www.netresec.com/?page=pcapfiles (accessed on 15 September 2021) [SCADA/ICS Network Capture]).

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

Abbreviations

The following abbreviations are used in this manuscript:

ICS	Industrial control system
1D-CNN	One-dimensional convolutional neural network
AM	Attention mechanism
LSTM	Long and short-term memory network
QoS	Quality of service
DPI	Deep packet inspection
ML	Machine learning
DL	Deep learning
IANA	Internet Assigned Numbers Authority
GA	Genetic algorithm
SVM	Support vector machines
TCP/IP	Transmission control protocol/internet protocol
MAC	Media access control

References

1. Plissonneau, L.; Costeux, J.L.; Brown, P. Analysis of peer-to-peer traffic on ADSL. In *International Workshop on Passive and Active Network Measurement*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 69–82.
2. Hosseini, M.; Ahmed, D.T.; Shirmohammadi, S.; Georganas, N.D. A survey of application-layer multicast protocols. *IEEE Commun. Surv. Tutor.* **2007**, *9*, 58–74. [CrossRef]
3. Bayat, N.; Jackson, W.; Liu, D. Deep learning for network traffic classification. *arXiv* **2021**, arXiv:2106.12693.
4. Rezaei, S.; Liu, X. Deep learning for encrypted traffic classification: An overview. *IEEE Commun. Mag.* **2019**, *57*, 76–81. [CrossRef]

5. Tong, V.; Tran, H.A.; Souihi, S.; Mellouk, A. A novel quic traffic classifier based on convolutional neural networks. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; IEEE: Manhattan, NY, USA, 2018; pp. 1–6.
6. Touch, J.; Mankin, A.; Kohler, E. Service Name and Transport Protocol Port Number Registry [EB/OL]. 2019. Available online: https://www.iana.org/assignment-s/service-names-port-numbers/service-names-port-numbers.xhtml (accessed on 15 September 2021).
7. Moore, A.W.; Zuev, D. Internet traffic classification using Bayesian analysis techniques. In Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Banff, AB, Canada, 6–10 June 2005; Volume 33, pp. 50–60.
8. Madhukar, A.; Williamson, C. A longitudinal study of P2P traffic classification. In Proceedings of the 14th International Symposium on Modeling, Analysis, and Simulation, Monterey, CA, USA, 14 September 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 179–188.
9. Ma, J.; Levchenko, K.; Kreibich, C.; Savage, S.; Voelker, G.M. Unexpected means of protocol inference. In Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, Rio de Janeriro, Brazil, 25–27 October 2006; ACM: New York, NY, USA, 2006; p. 313.
10. Oveis, A.; Nima, A.; Hamidreza, Z. A new feature selection technique for load and price forecast of electrical power systems. *IEEE Trans. Power Syst.* **2017**, *32*, 62–74.
11. Sen, S.; Spatscheck, O.; Wang, D. Accurate, scalable innetwork identification of p2p traffic using application signatures. In Proceedings of the 13th International Conference on World Wide Web, New York, NY, USA, 17–20 May 2004; ACM: New York, NY, USA, 2004; pp. 512–521.
12. Hu, X.; Gong, J. Relevance analysis of network traffic classification measure. In Proceedings of the 16th Annual Academic Conference, Tianjin, China, 12–14 December 2009; CERNET: Beijing, China, 2012; pp. 221–224.
13. Este, A.; Gringoli, F.; Salgarelli, L. Support vector machines for TCP traffic classification. *Comput. Netw.* **2009**, *53*, 2476–2490. [CrossRef]
14. Zhou, W.; Dong, L.; Bic, L.; Zhou, M.; Chen, L. Internet traffic classification using feed-forward neural network. In Proceedings of the 2011 International Conference on Computational Problem-Solving, Chengdu, China, 21–23 October 2011; IEEE: New York, NY, USA, 2011; pp. 641–646.
15. Alshammari, R.; Zincir-Heywood, A.N. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection. *Comput. Netw.* **2011**, *55*, 1326–1350. [CrossRef]
16. Zhang, H.; Lu, G.; Qassrawi, M.T.; Zhang, Y.; Yu, X. Feature selection for optimizing traffic classification. *Comput. Commun.* **2012**, *35*, 1457–1471. [CrossRef]
17. Zhou, F.Y.; Jin, L.P.; Dong, J. Review of convolutional neural network. *Chin. J. Comput.* **2017**, *40*, 1229–1251.
18. Wang, Z. The Applications of Deep Learning on Traffic Identification [EB/OL]. 2019. Available online: https://www.black-hat.com/docs/us-15/materials/us-15-Wang-The-Applications-Of-Deep-Learning-On-Traffic-Identification-wp.pdf (accessed on 15 September 2021).
19. Ma, R.; Qin, S. Identification of unknown protocol traffic based on deep learning. In Proceedings of the 3rd IEEE International Conference on Computer and Communications, Chengdu, China, 13–16 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1195–1198.
20. Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 712–717.
21. Dainotti, A.; Pescape, A.; Claffy, K.C. Issues and future directions in traffic classification. *IEEE Netw.* **2021**, *26*, 35–40. [CrossRef]
22. Lotfollahi, M.; Siavoshani, M.J.; Zade, R.S.H.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Comput.* **2020**, *24*, 1999–2012. [CrossRef]
23. Harris, D.; Harris, S. *Digital Design and Computer Architecture*, 2nd ed.; Morgan Kaufmann: San Francisco, CA, USA, 2012; pp. 129–133.
24. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*; PMLR: New York, NY, USA, 2015.
25. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
26. Available online: https://www.netresec.com (accessed on 15 September 2021).
27. Available online: https://4sics.se (accessed on 15 September 2021).
28. Available online: https://www.shodan.io (accessed on 15 September 2021).
29. Available online: https://github.com (accessed on 15 September 2021).
30. Tor traffic classification from raw packet header using convolutional neural network. In Proceedings of the 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), Jeju Island, Korea, 23–27 July 2018; pp. 187–190. [CrossRef]