electronics

MDPI

Article

A Two Stage Intrusion Detection System for Industrial Control Networks Based on Ethernet/IP

Wenbin Yu, Yiyin Wang and Lei Song*

Department of Automation, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai Jiao Tong University, Shanghai 200240, China; yuwenbin@sjtu.edu.cn (W.Y.); yiyingwang@sjtu.edu.cn (Y.W.)

* Correspondence: songlei_24@sjtu.edu.cn; Tel.: +86-21-34204022

Received: 25 October 2019; Accepted: 12 December 2019; Published: 15 December 2019

Abstract: Standard Ethernet (IEEE 802.3 and the TCP/IP protocol suite) is gradually applied in industrial control system (ICS) with the development of information technology. It breaks the natural isolation of ICS, but contains no security mechanisms. An improved intrusion detection system (IDS), which is strongly correlated to specific industrial scenarios, is necessary for modern ICS. On one hand, this paper outlines three kinds of attack models, including infiltration attacks, creative forging attacks, and false data injection attacks. On the other hand, a two stage IDS is proposed, which contains a traffic prediction model and an anomaly detection model. The traffic prediction model, which is based on the autoregressive integrated moving average (ARIMA), can forecast the traffic of the ICS network in the short term and detect infiltration attacks precisely according to the abnormal changes in traffic patterns. Furthermore, the anomaly detection model, using a one class support vector machine (OCSVM), is able to detect malicious control instructions by analyzing the key field in Ethernet/IP packets. The confusion matrix is selected to testify to the effectiveness of the proposed method, and two other innovative IDSs are used for comparison. The experiment results show that the proposed two stage IDS in this paper has an outstanding performance in detecting infiltration attacks, forging attacks, and false data injection attacks compared with other IDSs.

Keywords: intrusion detection; Ethernet/IP; industrial control networks

1. Introduction

An industrial control system (ICS) is composed of various automatic control components and real-time data acquisition components together. The main purpose of the ICS is to monitor and control industrial manufacturing to ensure the normal operation of industrial equipment. The core components of the ICS include the supervisory control and data acquisition system (SCADA) [1,2], distributed control system (DCS), programmable logic controller (PLC), remote terminal unit (RTU), human–machine interface (HMI), and a variety of communication interface technologies [3–5]. The ICS has been widely applied in the energy industry, transportation, metallurgy, electric power systems, etc.

In the traditional ICS, experienced engineers mainly focus on the physical safety of the production and ignore the information security because of the natural isolation of industrial networks, which makes it impossible for malicious hackers to interact with the traditional ICS [6–8]. With the rapid development of information technology (IT), standard Ethernet (IEEE 802.3 and the TCP/IP protocol suite) has been gradually implemented in the ICS communication interface. As a consequence,many automation companies designed the standardized industrial field bus and Ethernet. Schneider Electric also released Modbus/TCP [9] to replace the previous Modbus/RTU [10], which used the original RS-485 as a communication interface. At the same time, Profinet [11], which achieved Profibus over industrial Ethernet, was defined by PROFINET International. Rockwell

Automation proposed Ethernet/IP [12], which supports data communications over industrial Ethernet, which enlargen the bandwidth of the communication. Ethernet/IP was first introduced in 2001 and now is the most developed, mature, and complete industrial Ethernet solution available for manufacturing automation, with rapid growth as users are eager to take advantage of the open technologies and Internet. Ethernet/IP implements the common industrial protocol (CIP) over standard IEEE 802.3 and the TCP/IP protocol suite.

Implementing standard Ethernet in the modern ICS improves the interoperability of the ICS and greatly reduces the cost of application developments. However, it also breaks the natural isolation of industrial networks. The modern ICS are facing more advanced threats from the Internet outside the factory. However, the original ICS security mechanisms, such as the industrial firewall and white list, cannot handle these threats effectively enough. On the one hand, an industrial firewall cannot dissect industrial communication protocols (e.g., Ethernet/IP), which makes it impossible to inspect the application layer payloads in packets or automatically generate proper filter rules according to the specific industrial scenarios [6]. On the other hand, the white list can only function as an access control list, and it is easy to forge as a result of many brilliant penetration testing tools, such as Metasploit [13,14].

As a second line of defense, the intrusion detection system (IDS) is an effective approach to detect malicious intruders, who are trying to disrupt the ICS networks from the Internet. By analyzing the information collected from the key points in the network, the IDS can find out whether there is a violation of the security policies and decrease the probability of attack occurrence. According to the analysis and inspection of the problems, the IDS takes appropriate countermeasures, such as raising an alarm or blocking the suspicious connections [3].

Without a doubt, there has been also many innovative works in designing the IDS for industrial networks. The work in [15] designed a telemetry based IDS by measuring the statistical data about client server sessions from the traffic flow. Although it was practical to detect anomalies in the ICS networks, it had a strong precondition that the existence of time delays was introduced by spoofing. Apart from this, the data related to the network protocols, which were more valuable, were not utilized to design the IDS. The work in [16] constructed an anomaly based IDS according to the normal behaviors of function control and process data. The behavior extraction algorithm was attractive to researchers because it considered the information entropy of the function code used by the Modbus/TCP protocol. However, the IDS used the function code sequence in the time interval as the input. If a packet were fabricated, which contained the same function code in the same time interval, but at the wrong time point, it would be impossible to detect the fake packets, which could be a serious threat to the ICS. The work in [17] built a model for normal system behavior to distinguish the normal and abnormal system operations. None these methods considered the characteristics of the data traffic and the normality modeling. In order to overcome the deficiencies of previous works, it is required to construct an intrusion detection system that could reflect the behavior characteristics in the ICS networks, strongly correlated to the ICS protocols, and be able to cope with vulnerabilities. Furthermore, it should have a satisfactory overall accuracy and false alarm rate.

Above all, it is better to think like a hacker before stopping a hacker. This paper firstly considers the infiltration attacks, forging attacks, and false injection attacks. In addition to this, an Ethernet/IP structure fabricates explicit messaging that uses the TCP as the transmission protocol. To prevent the attacks mentioned above, this paper designs a two stage intrusion detection system for the ICS based on the Ethernet/IP. The two stage IDS has the ability to dissect the Ethernet/IP protocol and mainly contains a traffic prediction model and an anomaly detection model. The traffic prediction model based on the autoregressive integrated moving average (ARIMA) can protect the ICS networks from infiltration attacks. The anomaly detection model based on one class support vector machine (OCSVM) is able to detect the elegant fabricated Ethernet/IP packets and protect against the.forging attacks and false injection attacks. Compared with other creative IDSs [15,16], the proposed method gives satisfactory results in terms of overall accuracy and false alarm rate.

The rest of this paper is organized as follows: Section 2 introduces the related works. Section 3 describes the simulated industrial scenario and gives a brief introduction about the Ethernet/IP protocol. Section 4 outlines the attack models, especially fabricating malicious Ethernet/IP packets. Section 5 elaborates on the two stage intrusion detection system, which consists of a traffic prediction model and an anomaly detection model. Section 6 is the simulations and analysis. Finally, Section 7 gives the conclusions of this paper.

2. Related Works

As a hot research topic, many researchers are focusing on the two stage intrusion detection approaches. A classical chemical process and went over all the attack vectors were studied in [18]. A data driven approach to detect anomalies was designed for early indicators of malicious activity. The model of the attack process was built to profile some kinds of fault disturbances. A machine learning based approach was proposed in [19], which could reduce the amount of manual customization required for different ICS networks. Furthermore, a series of features for the machine learning input was selected, and the structure was used for a real industrial process control network. A SCADA based IDS was designed in [20], which could deal with the denial of service attacks. The network was investigated to resist response injection and denial of service attacks. A periodicity characteristic analyzing algorithm was designed for SCADA networks, and this feature was used for intrusion detection. Similarly, these papers all considered detecting the intrusion by modeling the normal operation features, which is also the main idea of this paper. Differently, this paper focused on the traffic of the ICS network based on the control and data stream. Furthermore, the application scenario in this paper was basic data transmission for the industrial network.

After the normal transmission traffic is modeled, a classifier needs to be designed to judge whether the receiving data are normal or attack data. Various machine learning algorithms were compared to reduce the false alarm rate and maintain high accuracy for industrial intrusion detection [21]. The work introduced A machine learning based classifier was introduced to reduce the false alarms and guarantee the precision for intrusion detection in the industrial area [22]. The work in [23] focused on the IDS and compared different techniques for the Industrial Internet of Things including machine learning and non-machine learning methods. The machine learning structure is a widely used method for industrial intrusion detection. However, as is known to all, the training of the structure requires a large quantity of labeled data. In this paper, an SVM based structure was designed, and an optimization problem was formulated to build the classifier, which could make full use of the labeled data.

3. Industrial Scenario

In order to make the proposed approach more effective and practical in the real ICS, an ICS demo platform was established based on Ethernet/IP. As shown in Figure 1, the platform consisted of three layers: the process monitor layer, the process control layer, and the plant-floor layer. The process monitor layer contained the operation station, where engineers could monitor the entire manufacturing procedure and modify the program or parameters in the PLC. This layer also contained the malicious attacker and the intrusion detection system (IDS). The process control layer in the middle constituted a set of devices that served the production, including the PLC, industrial router, and variable frequency drive (VFD). The PLC was the Allen-Bradley Micro 850 series, and the VFD was the PowerFlex 525 from Rockwell. The plant-floor layer consisted of a whole auto-guided vehicle (AGV) system, which included an optical-electricity encoder and three phase asynchronous motor encoders.

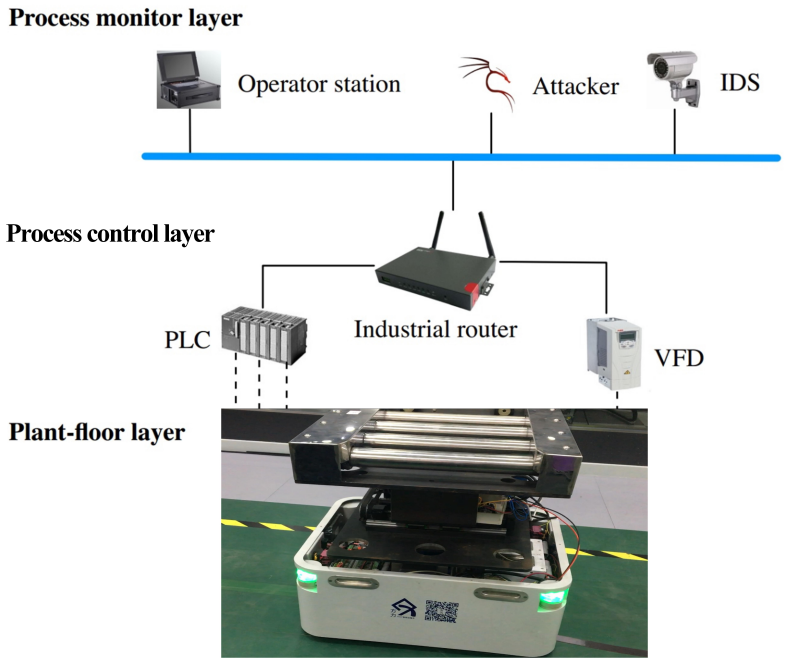


Figure 1. Simulated industrial control system (ICS) based on Ethernet/IP. PLC, programmable logic controller; VFD, variable frequency drive.

In this ICS, the AGV system was regarded as the controlled object, and the controller was the AB Micro 850 PLC. The feedback from the optical-electricity encoder was used to determine the current position of the encoder and the speed of the motors. The motions of the AGV system were controlled by the motors, which were driven by the PowerFlex 525 VFD. The entire process can be summarized as follows: according to the feedback from the optical-electricity encoder, the PLC made decisions and sent Ethernet/IP control packets to the VFD, and the VFD adjusted the motors speed and position by running the motors. This was a precision linear motion mechanism that converted the motor rotation into linear motion, and it is widely used in various linear motion applications.

The utilized communication protocol was Ethernet/IP. The physical layer and data link layer were based on Ethernet, while the transport layer and network layer were based on the TCP/IP protocol suite including the transmission control protocol (TCP), user datagram protocol (UDP), Internet protocol (IP), address resolution protocol (ARP), etc. The CIP was used as the application layer, and it defined two primary types of communications: implicit and explicit messaging. Implicit messaging is often used to transfer real-time control data from a remote I/O device with UDP, while explicit messaging, which is mainly discussed in this paper, is utilized with the TCP for request/reply transactions [24].

4. Attack Models

In this section, some kinds of infiltration attacks will be modeled. Besides that, a technique to fabricate an Ethernet/IP packet containing explicit messaging with the help of scapy, which is a powerful interactive packet manipulation program, is creatively proposed. The attack platform was based on Kali Linux, which is an advanced penetration testing Linux distribution. All the Ethernet/IP packets were parsed out by using the Python scripts. However, this paper mainly focuses on cyber-attacks and safety protection after entering the network.

4.1. Infiltration Attacks

Port scanning is usually used to identify some services and systems in the traditional network. By sending a TCP SYN packet to establish a connection at each port, it can be recognized that the port was opened according to the response. Through the open situation of the port, it was possible to

understand what services and operating system were running. Utilizing port scanning in the target ICS system, the results showed that the TCP port number (e.g., port 44818) was open, which indicated that the Ethernet/IP service was running on this ICS. However, the information collected by port scanning was incomplete, and device enumeration was needed to identify the ICS devices.

Device enumeration was used to identify the device information in the target ICS, and it was achieved by sending an Ethernet/IP packet to the remote device that had some TCP port number open. The packet was a request, using the Ethernet/IP list identity command, whose function code was, e.g., 0x63. Once a response was received, the information was parsed out, including the vendor ID, device type, product name, device IP, etc. Having achieved device enumeration in the ICS, the PLC, VFD, and the corresponding address could be identified.

According to the results of the device enumeration, the ARP spoof can be implemented to hijack the Ethernet/IP session between the PLC and VFD. The ARP was used to convert an IP address (network layer address) to the MAC address (link layer address). By sending a malicious ARP reply, attackers can be disguised as a VFD in front of a PLC, as shown in Figure 2, and spoof the VFD in the same way. With the ARP spoof, attackers can hijack the Ethernet/IP session and monitor the data flow between the PLC and VFD. The communication packets between the PLC and VFD were based on explicit messaging in Ethernet/IP. It mainly included two parts: periodic maintenance and control instructions. The periodic maintenance packets refer to the read and write messaging for device parameters, which used the class code (e.g., 0x93) and the whole period including the request. The control instructions contained rotate clockwise with function code (e.g., 0x2A), rotate anticlockwise with, e.g., 0x29, stall clockwise with, e.g., 0x01, and stall anticlockwise with, e.g., 0x19. The specific rotate speed was appended after the function code.

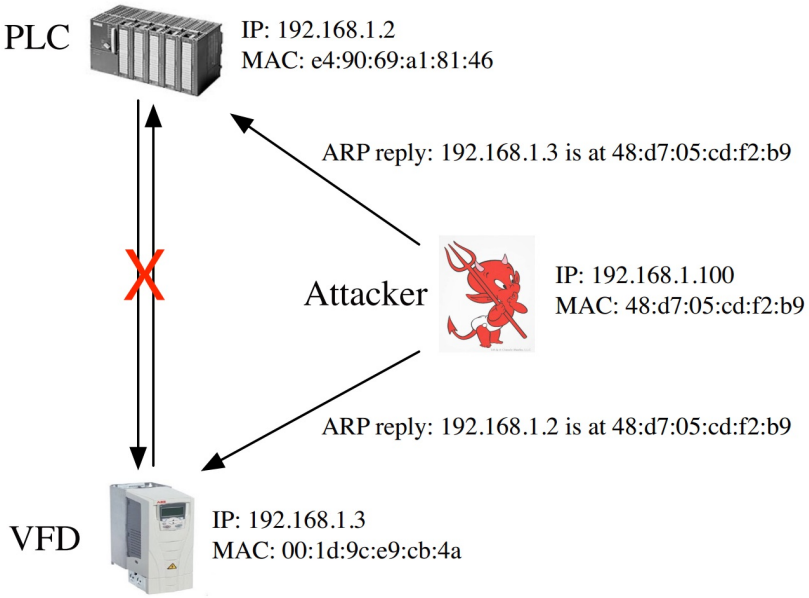


Figure 2. ARP spoof hijack Ethernet/IP session.

In order to avoid raising an alarm and disturbing the production process, it was better to select the forwarding periodic maintenance packets and discard the control instruction packets at the same time. Selective forwarding maintenance messages could ensure that the interface of the VFD had Ethernet/IP traffic and it would not report errors. Dropping the control messages would prevent the PLC from controlling the VFD and destroy the production process.

4.2. Forging Attack: Fabricate Ethernet/IP Explicit Message

As mentioned in Section 3, Ethernet/IP defines two kinds of communications: implicit and explicit messaging. The communication between the PLC and VFD in the ICS was based on explicit messaging because it used TCP as the transport layer protocol. The TCP provided reliable, ordered, and error checked delivery of a stream of octets between the PLC and VFD, and it was impossible to inject fake data [25–27].

However, this paper proposes a technique to inject fabricated Ethernet/IP explicit messaging. Algorithm 1 shows the principle of the proposed mechanism. The algorithm was realized by writing a tailored Python script, and the third party library used here was scapy, which is an elegant packet manipulation tool for networks. The inputs contained the IP addresses of the PLC and VFD and specific control instruction, which were acquired by the infiltration attacks mentioned above.

The forging attack consisted of three steps. First was capturing any Ethernet/IP packets to get the session handle in the transaction between the PLC and VFD, where the session handle could be obtained by dissecting the session handle field in the encapsulation header of Ethernet/IP. Second was capturing the TCP ACK packet sent by the PLC to obtain some key fields (seq, ack, TCP source port, IP identification). Third was to forge and send the Ethernet/IP packet according to the information previously obtained. The forged packet contained certain control instructions (e.g., 0x2A00DC05), which indicated rotating clockwise at a specific rotation speed.

Algorithm 1 Fabricate Ethernet/IP packet using scapy.

function FORGE ENIP PACKET(*plc ip, vfd ip, control*)
    enip_pkt ← sniff(filter: TCP port 44818)
    session_handle ← *enip_pkt*[session]
    ack_pkt ← sniff(filter: ip src *plc_ip* and dst port 44818 and length 64)
    seq← *ack_pkt*[TCP][seq]
    ack← *ack_pkt*[TCP][ack]
    sport← *ack_pkt*[TCP][sport]
    ip_id← *ack_pkt*[IP][id] + 1
    enip_data← unhexlify(*control, session handle*)
    forged_pkt← IP(src: *plc_ip*, dst: *vfd_ip*, id: *ip_id*) + TCP(sport: *sport*, dport: 44818, seq: *seq*, ack: *ack*, flags: PSH and ACK) + *enip_data*
    send(*forged_pkt*)
end function

4.3. False Data Injection Attack

As mentioned in Section 3, the Ethernet/IP based network structure would transmit several kinds of data including an optical-electricity encoder and three phase asynchronous motor encoders. The false data injection attacks tried to damage the sampled data and change the control instructions by injecting false data. The attack vector was defined as $a = [a_1, a_2, ..., a_m]^T$, and the sampled data would be changed as:

$$s^a = s + a = Hx + e + a \tag{1}$$

When the false data injection attack was operating, the attack vector a was non-zero and the state error vector was c, while the sampled data value would be changed to $s + a$. The proposed attack model focused on the data collecting and the motion control instructions. When the attack

vector satisfied $a = Hc$, the attack model could bypass the classical attack detection approaches [28]. The attack model is defined as:

$$\begin{aligned} r_a =& ||s^a - H(\hat{x} + c)|| \\ =& ||s - H\hat{x} + a - Hc|| \\ \leq& ||s - H\hat{x}|| + ||a - Hc|| \\ =& r + \tau_a, \end{aligned} \tag{2}$$

in which r_a indicates the error sampled after the false data injection attack worked. τ_a indicates the error change caused by the attack vector. When $\tau_a = 0$, the attack could not be detected because no change would be caused on the real sample data value. Furthermore, the attack strategy could perform the attack at the data collecting control by changing the sensor ID or even the sensor type.

5. Two Stage IDS

The attack models proposed in Section 4 were feasible and practical because Ethernet/IP does not define any explicit or implicit security mechanisms. In order to protect the ICS from these threats, a two stage IDS for the ICS networks based on Ethernet/IP is proposed. The two stage IDS was located in the process monitor layer, as shown in Figure 1, and its workflow is shown in Figure 3. The inputs were captured packets in one time slot, which could be modified by users. The two stage IDS had the ability to capture and analyze all communication packets from the monitoring port on the industrial router. Besides that, it used the libpcap technique to capture packets and dissected all the Ethernet/IP packets by running our Python scripts. In the Python scripts, the Ethernet/IP packets layer was dissected by layer and obtained necessary information according to the key fields, such as the control instructions in Section 4.

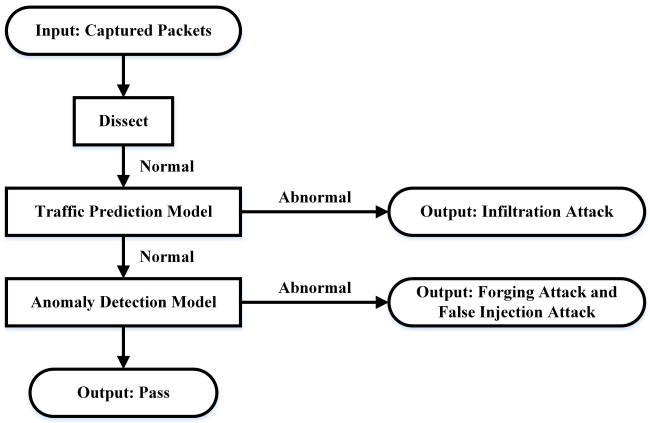


Figure 3. The workflow of two stage intrusion detection system (IDS).

The two stage IDS mainly consisted of two parts: the traffic prediction model and anomaly detection model. The traffic prediction model based on the ARIMA model was designed to estimate the number of packets in next time slot according to the captured packet flow previously. It was feasible to detect the infiltration attacks listed in Section 4 because the infiltration attacks would lead to an abnormal increase or decrease of the ICS network traffics at a specific time slot. If abnormal instances did not occur, the packets would be delivered to the anomaly detection model. That was because it was still possible for a forging attack, which did not lead to traffic changes. The anomaly detection model based on OCSVM was practical to detect the proposed forged Ethernet/IP packet because it could acquire essential control instruction by analyzing the Ethernet/IP packets in depth and detect abnormal behaviors by comparing with the normal pattern.

5.1. Traffic Prediction Model

A well designed traffic prediction model can precisely reflect the traffic characteristics of the ICS network. Since infiltration attacks lead to abnormal traffic fluctuations, it is possible to be detected with the traffic prediction model. The traffic flow data are a kind of time series, and the ARIMA model is the most commonly utilized model for time series prediction. The ARIMA model is used for short term forecasting, and the fundamental patterns of the time series should not be changed, which means the ICS networks should be immutable and the production process stable.

The raw input time series of the ARIMA, which is the count of traffic packets in one time slot, is not stable and fluctuates periodically. In order to use the ARIMA model, it is required to preprocess the raw input by using logarithmic transformation and differentiating. Logarithmic transformation is mainly done to reduce the vibration amplitude of the sequence, making the linear rule more obvious. Differentiating is able to make the series stable, and the difference periods are the periods of the ICS. The augmented Dickey–Fuller test is used to test the stationarity of the time series. Furthermore, the Ljung–Box test for autocorrelation is essential to ensure that the time series is not white noise. After preprocessing and testing, a stable and non-white noise time series $\{x_t\}_{t=1}^{n}$ is obtained, and it is suitable for the ARIMA model.

The prediction function of ARIMA can be depicted as:

$$\hat{x}_t = \psi_1 x_{t-1} + \psi_2 x_{t-2} + ... + \psi_p x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}, \tag{3}$$

where x_t is the stable and non-white noise time series and p and q are the order of the autoregressive (AR) model and moving average (MA) model. ψ_i and θ_i are the parameters of the ARIMA model. The prediction error must be uncorrelated and obey a normal distribution $\epsilon_t \sim N(0, \sigma^2)$.

The orders p and q determine the accuracy of the ARIMA model, and they can be estimated by calculating the autocorrelation function (ACF) and partial autocorrelation function (PACF) [29].

In addition to the ACF and PACF, enumerating many ARIMA models with different orders and using some criterion can also determine the proper model, i.e., the optimal parameters p and q. The criterion contains the Akaike information criterion (AIC), Bayes information criterion (BIC), and Hannan–Quinn information criterion (HQIC). The statistical ideas of these criteria are the same, that is they consider the fitting of the residuals and imposing punishments related to the number of variables at the same time. After calculating the predicted value x_t, the traffic prediction values are recovered by the inverse difference and exponentiation. The results also need to be rounded to the nearest integer.

5.2. Anomaly Detection Model

The anomaly detection model acts as a second line of defense after the traffic prediction model. Malicious attackers may drop the original Ethernet/IP control packet and replace it with the fabricated one that contains the wrong control instructions. The forging attack and false data injection attack cannot be detected by the traffic prediction model because it has little impact on traffic flow. Therefore, it is necessary to establish an anomaly detection model for the forging attack and false data injection attack.

The anomaly detection model firstly filtered out the Ethernet/IP control packets according to the field of service. The service name of the control packets was set single attribute, whose code was, e.g., 0x10. After obtaining the control packets, specific control instructions should be extracted from the packets. The control instructions had four features, i.e., relative time, action, direction, and speed. The relative time refers to the packet time stamp relative to the control period, which was a control cycle for the application. The action refers to rotate, whose value was, e.g., 0x02, or stall (0x01), and the direction refers to clockwise (0x0A) or counterclockwise (0x09). The speed simply refers to the rotation speed. After feature selection, the feature samples (control instructions) were obtained $\{\mathbf{x}_i\}_{i=1}^{N}$ and each sample $\mathbf{x}_i$ with the four features.

In order to detect the forged packets with the wrong control instructions, an OCSVM was constructed with the collected samples, which were all normal data. The OCSVM was a modified algorithm based on the SVM and has been widely used for one class classification problems, such as anomaly detection. According to [30–32], the OCSVM firstly mapped a sample $\mathbf{x}_i$ from the input space to the feature space F using the kernel function. The feature space had a higher dimension, and the separation may be easier in the feature space. Secondly, the OCSVM considered the origin as abnormal and the training samples as normal and constructed an optimal hyperplane between normal and abnormal by maximizing the margin.

The OCSVM mainly resolved the following quadratic programming optimization problem:

$$\begin{aligned} \min_{w \in F} \quad & \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{vn}\sum_{i=1}^{n}\xi_i - \rho \\ s.t. \quad & \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle \geq \rho - \xi_i, \xi_i > 0, i = 1...n, \end{aligned} \tag{4}$$

in which $\mathbf{w}$ is the normal vector in the feature space; n is the number of training samples; ξ_i is the slack variable to handle outliers; and $\phi(\cdot)$ is the mapping function mentioned above. ρ is the compensation parameter, and v defines the upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

By utilizing the Lagrangian method, the dual formulation and transformation of the original problem were obtained calculate the Lagrangian operator $\{\alpha_i\}_{i=1}^{n}$. The Lagrangian operator could be resolved by sequential minimal optimization (SMO) [33–35]. Finally, the decision function could be obtained by using the kernel method:

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - \rho = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i, \mathbf{x}) - \rho, \tag{5}$$

where $\Phi(\mathbf{x}_i, \mathbf{x}_j) == \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is the kernel function. For any new input sample $\mathbf{x}$, if $f(\mathbf{x}) \geq 0$, then $\mathbf{x}$ was labeled as normal. Otherwise, if anomaly instances were detected, this indicated that the input control instruction was different from the normal behavior and that it was likely to be a fake packet. The anomaly detection model was implemented based on OCSVM using Python. The kernel function selected was the Gaussian kernel, as shown in the following formulation:

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\gamma}\right). \tag{6}$$

6. Simulations

6.1. Scenarios

The two stage IDS was tested in the ICS network based on Ethernet/IP. As mentioned in Section 3, the ICS platform was built for experiments to testify to the effectiveness of the proposed approach. The platform consisted of three layers: process monitor layer, process control layer, and plant-floor layer. The process monitor layer contained the malicious attacker and IDS. The process control layer was constituted by the PLC, industrial router, and VFD. The PLC used was the Allen-Bradley Micro 850 series, and the VFD used was the PowerFlex 525 from Rockwell. Moreover, the network structure was used to transmit the AGV localization data including an optical-electricity encoder and three phase asynchronous motor encoders. In this paper, according to the technological requirements, the time slot was selected to be 1 s, and the ICS ran for one day including 86,400 time slots.

The packets extracted from each time slot were delivered to the IDS for real-time inspection, which lasted for less than 10^{-4} A. One-thousand infiltration attacks, 1000 forging attacks, and 1000 false data injection attacks were randomly launched during the day. Furthermore, the corresponding 3000 normal instances were used for the simulation. The simulation results were displayed using a confusion matrix,

which is shown in Table 1. The confusion matrix was designed for a two class classifier and was useful to evaluate the performance of the IDS [36].

Table 1. Confusion matrix for the IDS system.

Actual Class	**Predicted Class**	
Class	**Normal**	**Attack**
Normal	True negative (TN)	False Positive (FP)
Attack	False Negative (FN)	True positive (TP)

6.2. Simulation Results

6.2.1. Metrics

After calculating the predicted value x_t, the traffic prediction values were recovered by inverse difference and exponentiation. The results also needed to be rounded to the nearest integer. The traffic prediction model in the real ICS network was implemented, and the results of the rolling prediction are depicted in Figure 4. The root mean squared error (RMSE), which is calculated by:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(x_t - \hat{x}_t)^2}{n}}, \tag{7}$$

was used to determine the threshold between normal and abnormal status. At time slot t, if the difference $d_t = x_t - \hat{x}^t$ was larger than $1.5 * RMSE$, the ICS network was suspected to be infiltration attacked, and the two stage IDS would notify the engineer to check the traffic log. If no anomaly was detected, the packets would be delivered to the anomaly detection model for the next step of inspection. The kernel coefficient was set to be 0.1 according to the validation. The training samples (control instructions) were extracted from the ICS network in normal operation, and the smallest training error was guaranteed.

To evaluate the performance of the proposed method, four metrics were selected, the overall accuracy decision rates, false positive rate, false negative rate, and precision rate [36,37].

The overall accuracy is defined as:

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

which demonstrates the accuracy of the behavior of the attack detection. Literally, the overall accuracy denotes all the correct classifications against all the classifications.

The false positive rate (FPR) is defined by:

$$FPR = \frac{FP}{TN + FP}, \tag{9}$$

which describes the rate of the wrong predictions for the normal instances. The FPR can also be denoted as the fallout rate [37].

Conversely, the false negative rate (FNR) is defined as:

$$FNR = \frac{FN}{TP + FN}, \tag{10}$$

which describes the rate of the wrong predictions for the normal instances. The FNR can also be denoted as the miss rate [37].

Furthermore, the precision rate (PR) is defined as:

$$PR = \frac{TP}{TP + FP}, \tag{11}$$

which describes the precision of the prediction when an attack prediction is made [37].

All the above metrics were used for simulations comparing other IDS methods.

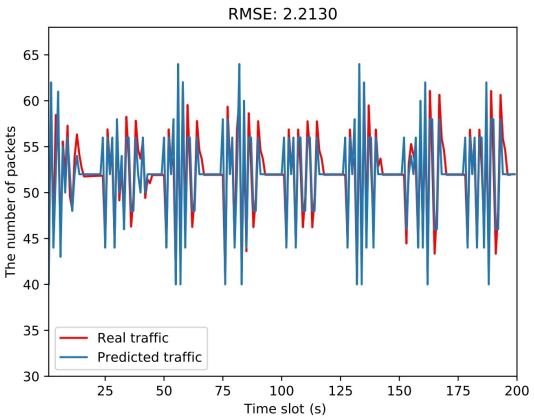


Figure 4. The result of the traffic prediction model based on ARIMA.

6.2.2. Performance of the Traffic Prediction Model

The orders p and q determined the accuracy of the ARIMA model, and they could be estimated by calculating the autocorrelation function (ACF) and the partial autocorrelation function (PACF) [29]. According to the results, both the ACF and PACF had trailing characteristics, and they both had obvious first order correlations. Therefore, we set $p = 1$ and $q = 1$.

The ARIMA model using Python and the fit model by the exact maximum likelihood via Kalman filter was simulated. After comparing the models by the criteria, the selected model was ARIMA(3,1,1), where $p = 3$, $q = 1$, and $d = 1$, which indicated the first difference. The given orders p and q were different from the orders observed by the ACF and PACF. The results of the observation were partly influenced by the real operation data. Furthermore, the ARIMA model needed to be updated periodically, and it was more convenient to select the proper model by using the criteria than observing the ACF and PACF. The criteria of the selected model were small enough and satisfactory.

$$AIC = -867.42, BIC = -887.15, HQIC = -875.41$$

The final parameters were as follows:

$$\psi_1 = -0.8561, \psi_2 = 0.6794, \psi_3 = -0.446, \theta_1 = -0.7998$$

Then, the prediction function can be calculated by the following equation.

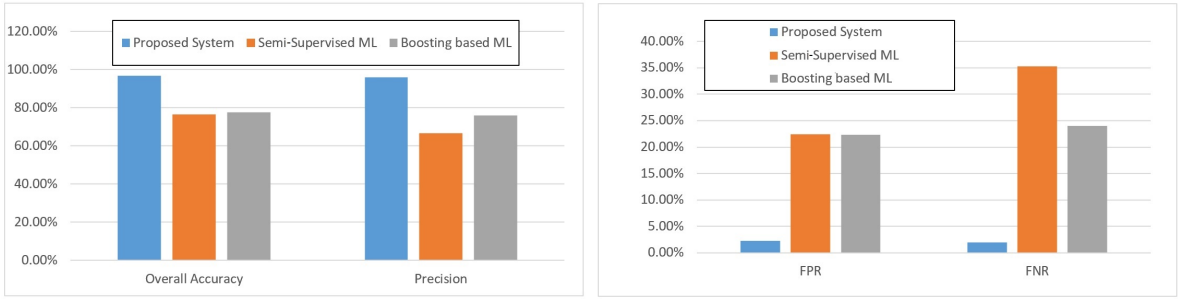
$$\hat{x}_t = -0.8561x_{t-1} + 0.6794x_{t-2} - 0.446x_{t-3} + \epsilon_t - 0.7998\epsilon_{t-1}, \tag{12}$$

6.2.3. Performance of the Anomaly Detection Model

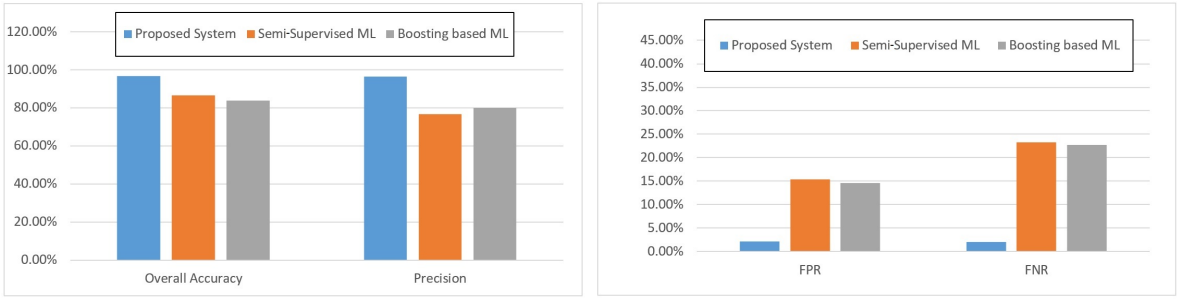
In this section, the performance of the OCSVM is tested. As the proposed OCSVM approach is a kind of classification algorithm, the other two machine learning based methods were selected for the comparisons, which were called the semi-supervised machine learning method [22] and the boosting based machine learning method [28].

The size of the training data seriously affects the performance of each classification approach. In order to simulate the methods equally, three data sizes were selected to be the labeled training data, 877 (380 min), 3323 (24 h), and 6646 (48 h).

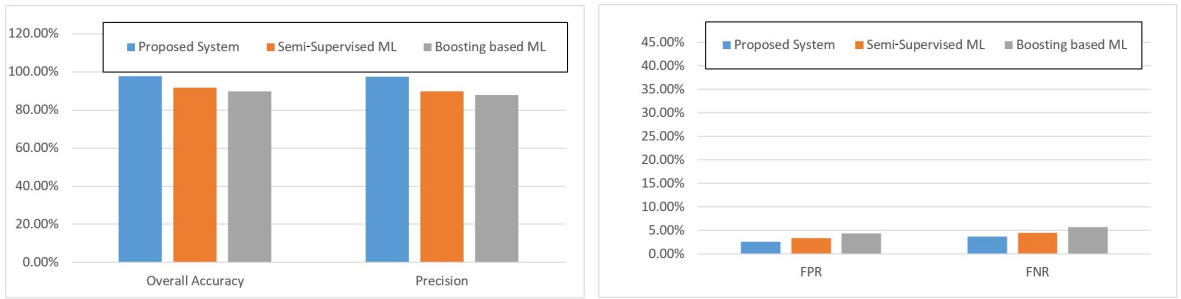
The simulation results are shown in Figure 5. It is depicted in Figure 5 that almost all the algorithms were improving as the training dataset size increased. When the data size was large, the overall accuracy and precision rate of all the methods could obtain an acceptable performance; both the overall accuracy and precision rate were more than 85%, which is shown in Figure 5c. However, when the data size was small, the proposed method performed better than the others, which is shown in Figure 5a. Furthermore, when the data size increased by about six times, the FPR of the two machine learning methods dropped from the highest of more than 20% to below 5%. Furthermore, the FNR dropped from more than 35% to 6%. This means that as the data size increased, the machine learning based performed better, and the smallest data size would be selected as 6646. Comparatively, the proposed method in this paper performed much better when the data size was small. As the data size increased, the performance even became less effective, which may be because as the training data size was enlarged, the classification model was overfitted.



(**a**)



(**b**)



(**c**)

Figure 5. The classification experiment results between OCSVM and machine learning methods. (**a**) Train data size = 877. (**b**) Train data size = 3323. (**c**) Train data size = 6646.

6.2.4. Performance of the Proposed Two Stage IDS

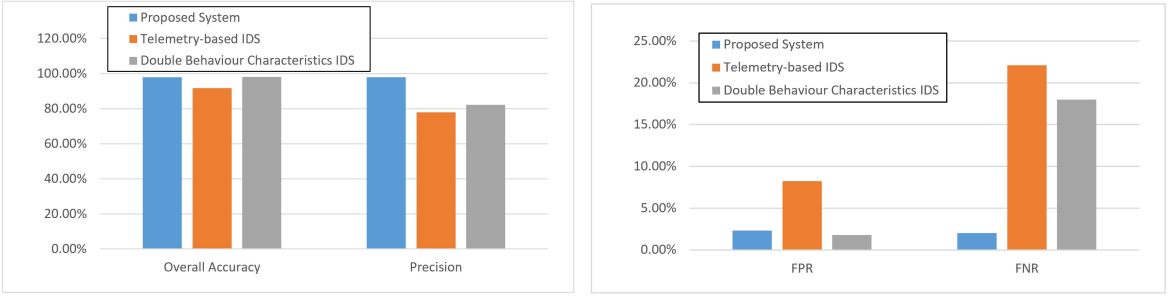
A comparison was made among our two stage IDS, telemetry based IDS [15] and double behavior characteristics IDS [16]. The simulation results of the confusion matrices are shown in Table 2.

Table 2. Confusion matrix results for the IDS system.

Attacks	Infiltration				Forging				False Data Injection			
Confusion Matrix Parameters	**TN**	**TP**	**FN**	**FP**	**TN**	**TP**	**FN**	**FP**	**TN**	**TP**	**FN**	**FP**
Proposed two stage IDS	977	**980**	**20**	23	**967**	**957**	**43**	**33**	**911**	**879**	**121**	**89**
Telemetry based IDS	918	779	221	82	927	789	211	73	768	878	122	232
Double behavior characteristics IDS	**982**	821	179	**18**	964	866	134	36	837	855	145	163

Table 2 illustrates that the performance of the proposed method was better than the other two algorithms. For the infiltration attack, the proposed method obtained the highest precision rate (higher value of TP), and the double behavior characteristics IDS had the highest TN, which reflected the overall accuracy; however, the low TP also undermined the overall accuracy. For both forging and false data injection attacks, the proposed two stage IDS resulted in having the best performances. Although the defense to the false data injection attack may not have be as good as the other two kinds of attacks, it still worked much better than the other IDSs. A more detailed analysis is as follows, and the experimental results are illustrated in Figures 6–8, which was simulated under the proposed attack model mentioned in Section 4.

As shown in Figure 6, when the simulation was carried out under infiltration attacks, the proposed two stage IDS approach performed much better than the other two IDS methods, which was because the proposed method could detect the infiltration attacks by the traffic detection model. For the overall accuracy, the three selected methods all performed well, and the accuracy of the proposed method was better than 95%. Correspondingly, the precision rate performance of the proposed method was about 20% better than the other two methods, which means the proposed method could predict the attack instance more precisely, as shown in Figure 6a. Furthermore, the double behavior characteristics IDS method could perform well in FPR metrics, and not very well for FNR, which was because this method could classify more normal data into the attack group, as shown in Figure 6b. According to the simulation results, the proposed method could detect the infiltration attacks precisely.



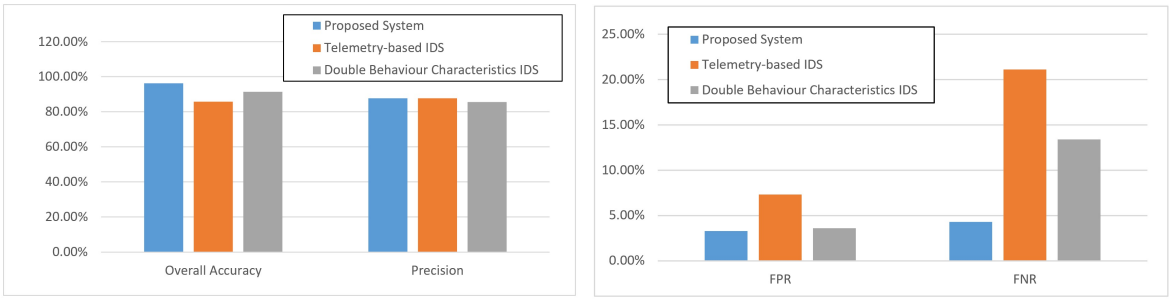
(**a**) (**b**)

Figure 6. Experimental results under infiltration attacks.

As shown in Figure 7, when the simulation was carried out under forging attacks, the proposed two stage IDS approach performed much better than the other two IDS methods, which was because the proposed method could detect the forging attacks by the traffic detection model. The three selected methods all had good performance for the overall accuracy. The proposed method performed as well as the double behavior characteristics IDS, around 90%, which is shown in Figure 7a. It is to be mentioned that the proposed method could maintain a high precision rate also, because the proposed OCSVM methods could precisely extract the features of the forging attack. On the other hand, double behavior

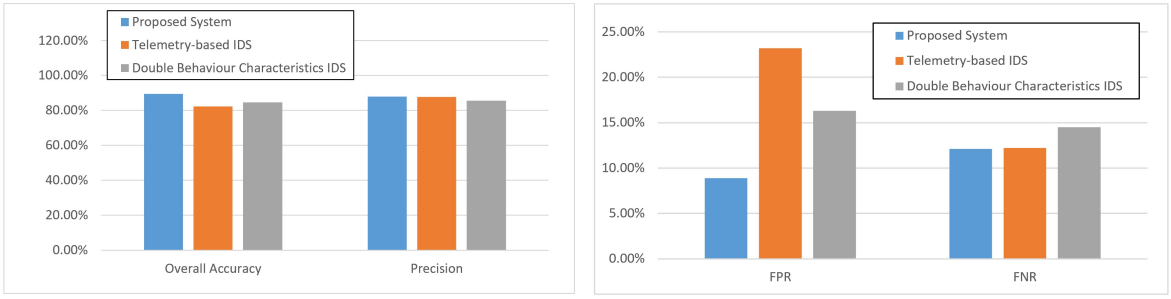
characteristics IDS had a competitive performance in terms of FPR, but the FNR was much more than that of our two stage IDS because it was impossible to detect the forging attack that was launched at a malicious time point. The proposed method had an outstanding performance in detecting the infiltration attacks and forging attack because of its models being strongly related to the ICS scenario, which is shown in Figure 7b. The two stage IDS processed the data collected from the ICS protocol and was capable of precisely reflecting the behavior characteristics in the ICS network.

As shown in Figure 8, when the simulation was carried out under false data injection attacks, it was hard to conclude that the proposed two stage IDS approach performed much better than the other two IDS methods, which was because the false data injection attacks could conceal themselves by changing their parameters, and the proposed approach, the traffic detection model, and the other two methods could not precisely detect the attacks. The proposed method could maintain the overall accuracy and precision rate over 80%, as shown in Figure 8a. Both the FPR and the FNR were a little bit higher than the former two kinds of attacks, which is shown in Figure 8b. To discuss this more, the two stage IDS processed the data collected from ICS protocol and was capable of precisely reflecting the behavior characteristics in the ICS network. If the attack models performed differently from the normal traffic, the proposed method would perform better. Otherwise, the method would not detect all the attack data or misjudge the normal data.



(**a**) (**b**)

Figure 7. Experimental results under forging attacks.



(**a**) (**b**)

Figure 8. Experimental results under false data injection attacks.

7. Conclusions

This paper proposed a two stage IDS for the ICS based on Ethernet/IP. The two stage IDS contained a traffic prediction model and an anomaly detection model. Compared with machine learning methods, the proposed method could distinguish normal data and attack data with much fewer data. Furthermore, compared with telemetry based IDS and double behavior characteristics IDS, it offered excellent performance in detecting infiltration attacks and forging attack. It is to be mentioned that the performance under false data injection attack was not as good as the above two attack models. Furthermore, the proposed approach could not cope with the situation of asynchronous protocols such

as the IEC 60870-5 series, which is mainly used in energy distribution networks and others because the transmission time must be known before two stage IDS is ready to work. Future work can be done in this specific area.

Future work can be divided into two parts: on one hand, evaluating the two stage IDS performance in a complex ICS scenario, which contains more controllers and actuators; on the other hand, refining the two stage IDS to defend I/O data transfers based on Ethernet/IP.

Author Contributions: Conceptualization, W.Y., and L.S.; methodology, Y.W. and W.Y.; system, W.Y.; validation, Y.W. and L.S.; experiment, Y.W.; analysis, Y.W. and W.Y.; writing, original draft preparation, Y.W. and W.Y.; writing, review and editing, L.S.

Funding: The research is sponsored by the National Key Research and Development Program of China (2018YFB1308304, 2017YFB1301103), the National Natural Science Foundation of China (61803261), and the Shanghai Natural Science Foundation of China (18ZR1421100).

Acknowledgments: The authors would like to thank and appreciate the support of all the scholars for helping us with this piece of work and the problems encountered.

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

References

1. Almalawi, A.; Tari, Z.; Fahad, A.; Khalil, I. A Framework for Improving the Accuracy of Unsupervised Intrusion Detection for SCADA Systems. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, Australia, 16–18 July 2013; pp. 292–301. [CrossRef]
2. Oliver, E.; Philipp, K.; Tavolato, P. Identifying S7comm Protocol Data Injection Attacks in Cyber-Physical Systems. In Proceedings of the 2018 Proceedings of the 5th International Symposium for ICS & SCADA Cyber Security Research, Hamburg, Germany, 29–30 August 2018.
3. Kargl, F.; van der Heijden, R.W.; König, H.; Valdes, A.; Dacier, M.C. Insights on the Security and Dependability of Industrial Control Systems. *IEEE Secur. Priv.* **2014**, *12*, 75–78. [CrossRef]
4. Berhe, A.B.; Kim, K.; Tizazu, G.A. Industrial control system security framework for ethiopia. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 814–817. [CrossRef]
5. Paridari, K.; O'Mahony, N.; El-Din Mady, A.; Chabukswar, R.; Boubekeur, M.; Sandberg, H. A Framework for Attack-Resilient Industrial Control Systems: Attack Detection and Controller Reconfiguration. *Proc. IEEE* **2018**, *106*, 113–128. [CrossRef]
6. Cheminod, M.; Durante, L.; Valenzano, A. Review of Security Issues in Industrial Networks. *IEEE Trans. Ind. Inform.* **2013**, *9*, 277–293. [CrossRef]
7. George, G.; Thampi, S.M. A Graph-Based Security Framework for Securing Industrial IoT Networks From Vulnerability Exploitations. *IEEE Access* **2018**, *6*, 43586–43601. [CrossRef]
8. Fan, X.; Fan, K.; Wang, Y.; Zhou, R. Overview of cyber-security of industrial control system. In Proceedings of the 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, China, 5–7 August 2015; pp. 1–7. [CrossRef]
9. Meza, G.; Carpio, C.; Vinces, N.; Klusmann, M. Control of a three-axis CNC machine using PLC S7 1200 with the Mach3 software adapted to a Modbus TCP/IP network. In Proceedings of the 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 8–10 August 2018; pp. 1–4. [CrossRef]
10. Hittanagi, K.N.; Ramesh, M.; Kumar, K.N.R.; Mahadeva, S.K. PLC based DC drive control using Modbus RTU communication for selected applications of sugar mill. In Proceedings of the 2017 International Conference on Circuits, Controls, and Communications (CCUBE), Bangalore, India, 15–16 December 2017; pp. 80–85. [CrossRef]
11. Dias, A.L.; Sestito, G.S.; Turcato, A.C.; Brandão, D. Panorama, challenges and opportunities in PROFINET protocol research. In Proceedings of the 2018 13th IEEE International Conference on Industry Applications (INDUSCON), Sao Paulo, Brazil, 11–14 November 2018; pp. 186–193. [CrossRef]

12. Davies, S. Industrial ethernet—The fundamentals of ethernet/IP - Ethernet/IP has reached the million-node landmark, but what is making this protocol so attractive to industrial control engineers? *Comput. Control Eng. J.* **2007**, *18*, 42–45. [CrossRef]
13. Denis, M.; Zena, C.; Hayajneh, T. Penetration testing: Concepts, attack methods, and defense strategies. In Proceedings of the 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 29 April 2016; pp. 1–6. [CrossRef]
14. Shebli, H.M.Z.A.; Beheshti, B.D. A study on penetration testing process and tools. In Proceedings of the 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 4 May 2018; pp. 1–7. [CrossRef]
15. Ponomarev, S.; Atkison, T. Industrial Control System Network Intrusion Detection by Telemetry Analysis. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 252–260. [CrossRef]
16. Wan, M.; Shang, W.; Zeng, P. Double Behavior Characteristics for One-Class Classification Anomaly Detection in Networked Control Systems. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 3011–3023. [CrossRef]
17. Oliver, E.; Philipp, K.; Tavolato, P. Attacks on Industrial Control Systems—Modeling and Anomaly Detection. In Proceedings of the 2018 4th International Conference on Information Systems Security and Privacy, Madeira, Portugal, 22–24 January 2018.
18. Keliris, A.; Salehghaffari, H.; Cairl, B.; Krishnamurthy, P.; Maniatakos, M.; Khorrami, F. Machine learning based defense against process-aware attacks on Industrial Control Systems. In Proceedings of the 2016 IEEE International Test Conference (ITC), Fort Worth, TX, USA, 15–17 November 2016; pp. 1–10. [CrossRef]
19. Mantere, M.; Sailio, M.; Noponen, S. Network Traffic Features for Anomaly Detection in Specific Industrial Control System Network. *Future Internet* **2013**, *5*, 460–473. [CrossRef]
20. Zhang, J.; Gan, S.; Liu, X.; Zhu, P. Intrusion detection in SCADA systems by traffic periodicity and telemetry analysis. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 318–325. [CrossRef]
21. Haripriya, L.; Jabbar, M.A. Role of Machine Learning in Intrusion Detection System: Review. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 925–929. [CrossRef]
22. Wagh, S.K.; Kolhe, S.R. Effective intrusion detection system using semi-supervised learning. In Proceedings of the 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC), Delhi, India, 5–6 September 2014; pp. 1–5. [CrossRef]
23. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]
24. 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. [CrossRef]
25. Shah, M.; Soni, V.; Shah, H.; Desai, M. TCP/IP network protocols—Security threats, flaws and defense methods. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 2693–2699.
26. Bobade, S.; Goudar, R. Secure Data Communication Using Protocol Steganography in IPv6. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 275–279. [CrossRef]
27. Ponmaniraj, S.; Rashmi, R.; Anand, M.V. IDS Based Network Security Architecture with TCP/IP Parameters using Machine Learning. In Proceedings of the 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 28–29 September 2018; pp. 111–114. [CrossRef]
28. Wei, L.; Gao, D.; Luo, C. False Data Injection Attacks Detection with Deep Belief Networks in Smart Grid. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 23–25 November 2018; pp. 2621–2625. [CrossRef]
29. Wei, M.; Kim, K. Intrusion detection scheme using traffic prediction for wireless industrial networks. *J. Commun. Netw.* **2012**, *14*, 310–318. [CrossRef]
30. Xiao, Y.; Wang, H.; Xu, W. Parameter Selection of Gaussian Kernel for One-Class SVM. *IEEE Trans. Cybern.* **2015**, *45*, 941–953. [CrossRef] [PubMed]

31. Maglaras, L.A.; Jiang, J.; Cruz, T. Integrated OCSVM mechanism for intrusion detection in SCADA systems. *Electron. Lett.* **2014**, *50*, 1935–1936. [CrossRef]
32. Li, Y.; Zhang, T.; Ma, Y.Y.; Zhou, C. Anomaly Detection of User Behavior for Database Security Audit Based on OCSVM. In Proceedings of the 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, China, 8–10 July 2016; pp. 214–219. [CrossRef]
33. Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Comput.* **2001**, *13*, 637–649. [CrossRef]
34. Toyoda, K.; Okamoto, T.; Koakutsu, S. An optimal routing search method on the network routing problem using the sequential minimal optimization. In Proceedings of the 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, Japan, 19–22 September 2017; pp. 805–810. [CrossRef]
35. Sheenu; Joshi, G.; Vig, R. A multi-class hand gesture recognition in complex background using Sequential minimal Optimization. In Proceedings of the 2015 International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 24–26 September 2015; pp. 92–96. [CrossRef]
36. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]
37. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.C.; Bellekens, X.J.A. A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets. *arXiv* **2018**, arXiv:1806.03517.