IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 7, 2021, accepted April 14, 2021, date of publication May 13, 2021, date of current version May 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3080081

Unknown Payload Anomaly Detection Based on Format and Field Semantics Inference in Cyber-Physical Infrastructure Systems

HYUNJIN KIM[1], (Graduate Student Member, IEEE), SUNGJIN KIM[2], WOOYEON JO[2], (Graduate Student Member, IEEE), KI-HYUN KIM[3], AND TAESHIK SHON[1,4], (Senior Member, IEEE)

[1]Department of Artificial Intelligence Convergence Network, Ajou University, Suwon 443749, South Korea
[2]Department of Computer Engineering, Ajou University, Suwon 443749, South Korea
[3]NNSP Company Ltd., Seoul 135729, South Korea
[4]Department of Cyber Security, Ajou University, Suwon 443749, South Korea

Corresponding author: Taeshik Shon (tsshon@ajou.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) (Advanced manufacturing process anomaly detection to prevent the smart factory operation failure by cyber-attacks) under Grant 2018-0-00336, and in part by the Energy Cloud Research and Development Program through the National Research Foundation of Korea (NRF) Grant by the Ministry of Science and ICT under Grant 2019M3F2A1073385.

ABSTRACT A cyber-physical infrastructure system (CPIS) is a system that controls and manages critical infrastructure such as smart manufacturing, water treatment facilities, power generation, and distribution facilities. Although these CPISs focus on the security of air-gapped network environments, strict isolation from the outside network is difficult to achieve, leading to various attacks. CPISs also comprise various devices and proprietary communication protocols that are used exclusively for each domain and site. Therefore, experts have to adopt a customized strategy to enhance security in CPIS networks after analyzing each domain, device, and protocol in advance. These methods require a significant amount of time, cost, and manpower; consequently, they are difficult to apply existing security methods in the real field. As a solution, a method is proposed herein that includes the following: 1) inferencing the CPIS protocol format and field semantics based on the characteristics of CPIS networks and protocols; 2) multilevel anomaly detection based on the meaning and values of each inferred field. The proposed method does not require knowledge of each site and protocol. In addition, the inference method can be used to analyze the payload field, including the state and measurement value, as well as the header field. Finally, we validate the proposed technique using an open-source CPIS network dataset including response injection, command injection, denial-of-service, and reconnaissance attacks. In addition, in the aspect of detection efficiency, the proposed technique exhibits comparable performance to that of existing knowledge-based anomaly detection methods.

INDEX TERMS Cyber-physical infrastructure systems, cyber security, Ethernet-based industrial protocol, industrial control systems, unknown payload anomaly detection.

I. INTRODUCTION

Cyber-physical infrastructure systems (CPISs) are systems that control and manage critical infrastructure such as smart manufacturing, water treatment facilities, power generation, distribution facilities, and buildings. In CPISs, communication is an important element for collecting, monitoring, and controlling the data of machines and systems. The traditional CPIS communication primarily comprises a fieldbus system for converting analog data to digital and managing field devices. Existing fieldbus communication systems use proprietary devices and communication protocols of specific manufacturers; as such, they lack interoperability and scalability [1]. Hence, major manufacturing and communication equipment industries have developed the International Electrotechnical Commission (IEC) 61158

The associate editor coordinating the review of this manuscript and approving it for publication was Wenyan Wu.

and 61784, an industrial communication standard that satisfies the requirements of the CPIS domain, such as reliability, availability, time deterministic, and real-time operation based on the Ethernet standard that is used extensively in Information Technology (IT) communication. The standard includes Ethernet-based industrial communication protocols such as EtherNet/Internet protocol (IP), PROFINET, EtherCAT, and Modbus/Transmission Control Protocol (TCP). These CPIS communication technologies are rapidly evolving in terms of operational benefits and management convenience, whereas security is ensured using traditional methods that rely on vendor-specific protocols and strict isolation network environments. However, some CPIS sites and the external layer are connected without considering security, and traditional vendor-specific protocols are designed without security functions. In fact, attacks such as Stuxnet [2], BlackEnergy [3], Industroyer [4], and TRISIS [5] have been deployed against major CPIS systems. These cases indicate that cyber-attacks against CPISs are gradually increasing, and attack techniques are becoming increasingly sophisticated. In addition, it has been reported that CPISs have become a major target of cyber-attacks, and the number of cyber-attacks is increasing annually [6].

Consequently, the importance of research on cyber-attack detection in CPISs has increased. In recent years, foundation studies pertaining to CPIS security have been actively conducted, such as the definition of normal and abnormal operations of CPISs [7], testbed construction considering the real operating environment, normal/anomaly data generation/collection [8]–[10], and encryption and authentication in field networks [11]–[13]. Network-based anomaly detection methods are applied extensively for monitoring network packets and traffic flows because they involve minimal change in the existing system configuration and do not require deployment on each host. Network-based anomaly detection studies often use an approach comprising a learning stage that defines normality in the source, and a detection stage that determines whether input data are normal or anomaly based on the defined normality. In the learning phase, network features are selected/extracted from network packets collected in a normal operating environment, and normal behavior is defined through a learning technique based on static rules, statistics, modeling, as well as machine and deep learning. In the detection stage, it is determined whether the result of the learning model for the target packet deviates from the defined normal criterion. These security studies often result in the inference that (i) CPISs have less variations in terms of the configuration and deployment of systems, devices, and networks compared with the IT environment, and (ii) limited physical processes are performed periodically. However, the main difficulties in network-based anomaly detection are as follows: (i) In the preprocessing stage, prior knowledge regarding each site device, protocol specification, network data, etc. is required; hence, a significant amount of time, cost, and manpower is consumed. (ii) Even if systems use the same CPIS protocol, they customize the detailed format and semantics on each site. (iii) The CPIS protocol contains binary content, and the reverse engineering of these binary protocols is difficult to perform. These limitations hinder the application of the existing methods to real sites.

To address these problems, we analyzed the characteristics of the CPIS network and the Ethernet-based CPIS communication protocol. Herein, we propose a method of inferring the format and field meaning of the protocol from the collected packets that does not require prior knowledge regarding the protocol. The proposed method is targeted to a CPIS environment that uses polling communication operations and periodically operates a process loop containing a limited number of commands. The proposed method can reduce the cost and human effort required in the preprocessing stage and offers compatibility with various CPIS systems. In addition, a network anomaly detection method based on the inferred format and field meaning is proposed. Each layer is highly scalable because it is compatible with existing knowledge-based anomaly methods.

The main contributions of this study are as follows:

- For the Ethernet-based CPIS communication protocol, the protocol format and meaning are inferred without prior knowledge of the field and protocol. In particular, the method identifies and extract fields of the payload including measured values, which is difficult to achieve using the existing reverse engineering of binary protocols.
- Through the inferred protocol format and values, we generate rules for the external information, fixed fields, traffic patterns, and payloads. Subsequently, we utilize them for multilevel anomaly detection. This can be used in the preprocessing stage of existing knowledge-based anomaly detection methods.
- We validate the proposed technique using a public CPIS network dataset that includes various types of attacks. We demonstrate that the performance of the proposed anomaly detection method is comparable to that of existing knowledge-based anomaly detection methods.

The remainder of this paper is organized as follows. Section II presents anomaly detection studies related to CPIS networks. Section III summarizes the characteristics of the CPIS network and Ethernet-based CPIS communication protocol. Section IV describes the proposed method comprehensively. Section V describes the normal and attack datasets used to verify the proposed technique. Section VI presents a discussion regarding the proposed technique. Finally, the conclusions and future work are provided in Section VII.

II. RELATED WORK

Studies pertaining to CPIS network-based anomaly detection methods often involve network packets collected in normal operations or field values preprocessed from packets as a source of information. Subsequently, normality is defined by selecting/extracting suitable features and learning them in various learning models. In general, features used in network anomaly detection studies can be classified into packet header, packet payload, and flow features.

The packet header feature focuses on the meaning of the packet field. Typical single packet features include the media access control (MAC) address of the L2 layer, the IP address of the L3 layer, and the port number of the L4 layer. The Ethernet-based CPIS protocol comprises a standard header field, an encapsulation header field, a CPIS header field, and a CPIS payload field. The related studies were organized based on the header field feature and the payload field feature. Anomaly detection methods based on static-rule-based [14]–[18] and modeling-based [19]–[24] learning of header field features have been proposed in various studies. The static-rule-based studies are similar to firewall rule generation studies in the IT network environment. However, based on the deep packet inspection (DPI) technique, the main signature field of the target CPIS protocol was analyzed in detail and used for rule generation. Yang *et al.* [14] proposed a signature-based intrusion detection system (IDS) for the IEC 60870-5 protocol using DPI techniques to identify anomalous behaviors and provide a security-model-based mechanism for unknown attacks. Wong *et al.* [15] proposed a signature-based IDS for the EtherNet/IP protocol and integrated it into Suricata, an open-source IDS tool. Jung *et al.* [16] proposed a whitelist generation method for traffic patterns based on the header information of a packet using each command. Nivethan *et al.* [17] and Li *et al.* [18] focused on detailed protocol fields for firewall rule generation. Studies regarding modeling- and statistics-based anomaly detection often involve command codes, address values, or transaction identification (ID) of CPIS header fields. The main assumption in these studies is that the operation of a CPIS involves a process loop and a periodic network traffic, enabling the next packet to be predicted based on the previous packet. Therefore, Goldenberg *et al.* [19] built a model by tracking actual traffic using a deterministic finite automaton (DFA) for the command field and packet length of the header field of the Modbus/TCP protocol. Yoon *et al.* [20] focused on Modbus but modeled instructions and data using a dynamic Bayesian network and probabilistic suffix tree. Kleinmann *et al.* [21] proposed a DFA-based IDS for human–machine interface (HMI)–programmable logic controller (PLC) channel traffic using the S7 protocol. Caselli *et al.* [22] proposed a method to detect anomalous packets through modeling the packet order by extracting three-tuples <ID, Function Code, Data address> from each Modbus protocol packet. Zhou *et al.* [23] demonstrated industrial anomalies and multi-model based IDSs based on the hidden Markov model to filter attacks. Kwon *et al.* [24] proposed an anomaly detection method based on a bidirectional recurrent neural network for learning traffic patterns by extracting the information of each network packet, such as the TCP flag, header function code, object type, and data point information.

Recently, some studies focusing on the CPIS payload feature have been conducted [25]–[28]. The CPIS payload primarily contains measurement, status, and result values for command processing. In these anomaly detection studies, each measurement or state value was extracted for use as a feature, assuming that the structure and meaning of the target protocol were known in advance. Secure water treatment (SWaT) datasets have been used in many studies to detect payload target attacks in CPIS networks. Inoue *et al.* [25], Goh *et al.* [26], Kravchik *et al.* [27], and Kim *et al.* [28] proposed anomaly detection methods that learn the normal operating values of each sensor and actuator based on deep neural networks, long short-term memory models, convolutional neural networks, and autoencoders, respectively.

However, anomaly detection based on header and payload features require the target protocol as well as the payload structure and meaning of fields for each site to be analyzed in advance. This consumes a significant amount of time and manpower; furthermore, the compatibility and scalability are low because the detection models must be customized for each site.

Hence, an anomaly detection method that utilizes packet traffic features, such as the order and quantity of packets, has been developed. Representative examples of packet traffic features include the features of the NSL-KDD dataset, which are used to select/extract network features for intrusion detection in IT environments and evaluate the performances of detection models [29]. Because of the lack of open network datasets for CPISs, the abovementioned detection method has been used extensively in CPIS anomaly detection studies. The packet-traffic-feature-based anomaly detection method is effective in preventing specific attacks such as denial of service (DoS); however, it does not effectively detect other attack types.

As another solution, protocol reverse engineering that infers protocol parameters, format, and semantics in the absence of specifications can be applied [30]. However, most protocol reverse engineering studies target text-based IT protocols. It is noteworthy that the CPIS communication protocol is known to be a binary communication protocol. Binary protocols (i) have no delimiters, and (ii) the binary field value contains categorical data; therefore, the value may not be meaningful. (iii) For the same protocol, the structure and meaning of the fields are different at each site. (iv) For the same message format and meaning, fields can have different values depending on the connection status, time, and order. Therefore, in current studies pertaining to protocol reverse engineering for CPIS protocols, the distribution values of each field position are used based on the n-gram technique [31]–[33].

III. CHARACTERISTICS AND ASSUMPTIONS FOR CPIS NETWORK AND PROTOCOLS

An overview of the CPIS network and Ethernet-based CPIS communication protocols is provided in this section. Because CPISs are deployed in various domains and fields, the components and topologies of the network are different. In general, however, CPIS networks exhibit a hierarchical structure based on Purdue models, such as international society

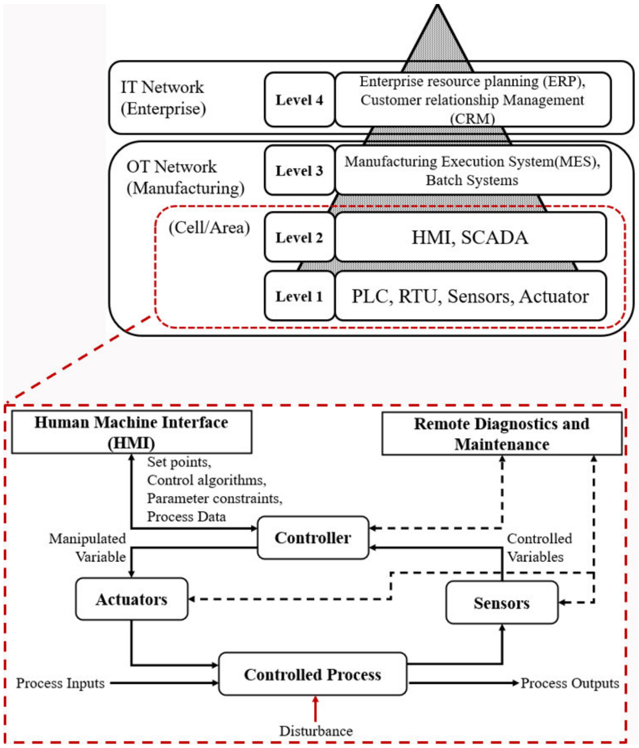


FIGURE 1. CPIS architecture and process loop [35].

of automation (ISA) 99 [34], national institute of standards and technology (NIST) 800-82 [35], and industrial control systems-cyber emergency response team (ICS-CERT) reference models [36]. The reference model segregates the CPIS network into four layers. Level 4 is the IT network, whereas Levels 1 through 3 are defined as the Operational Technology (OT) network. The main components of the OT network layer include sensors, actuators, remote terminal units (RTUs), PLCs, HMI, and data historian. Levels 1 and 2 directly affect the physical process. These layers involve a process loop where the Level 1 sensor transmits control variables to the Level 1 or 2 management device; subsequently, the management device interprets them and transmits commands to the Level 1 actuators, as shown in Fig. 1. Therefore, the change in sensor instrumentation values is attributed to the state of other elements, such as a specific actuator, thereby resulting in the actuator being manipulated based on a specific sensor value [37].

The CPIS communication protocol is used to read the sensor/state values described above or send commands to the actuator. Traditional CPIS communication uses vendor-specific protocols; however, Ethernet-based CPIS protocols have been used recently to achieve interoperability and scalability. These Ethernet-based CPIS communication protocols reprocess existing fieldbus communication protocol messages for compatibility with existing equipment and software and then place them in the data area of the standard protocol. For example, for Modbus/TCP, the Slave ID of the Modbus RTU packet is removed, whereas the Transaction ID, Protocol ID, Legacy, and Unit ID are added and placed in the data area of the public standard protocol. The Ethernet-based CPIS protocol comprises an open standard header field, an encapsulation header field, a CPIS protocol header field, and a CPIS protocol payload field. The CPIS protocol header contains commands, session IDs, object address values, and register values; the CPIS payload includes sensor values, measurements, and processing result values. The header and payload structure of major Ethernet-based CPIS communication protocols can be categorized into four types, as shown in Fig. 2 [38]. The default format structure of the CPIS protocol is defined in each standard, but the detailed format and field meaning are predefined and used in each site. However, the types of message formats in specific device-to-device communications are limited because lower-layer devices, which primarily operate the Ethernet-based CPIS communication protocol, repeatedly handle specific commands that are limited for their own purposes. Moreover, the format similarity of each message used for specific devices in the same site is extremely high. Because CPIS domains require high availability, it is difficult to update/replace existing devices and software. This implies that instead of using new commands and response message formats, existing message formats are also reused.

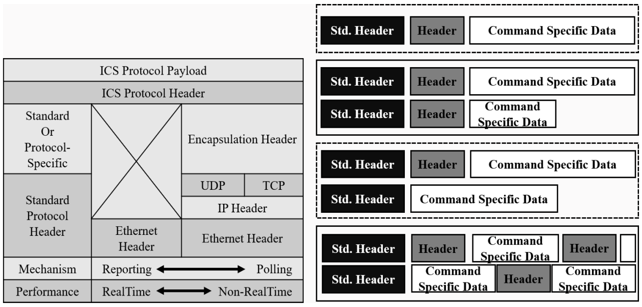


FIGURE 2. CPIS protocol stack and format [38].

Polling and reporting operation methods exist for the CPIS communication protocol. Polling is a method in which the master device (controller) delivers the command message to the slave device (sensor or actuator), and then the slave device processes the command and returns the result value to the master. In the reporting method, the slave device periodically sends messages in predefined formats that contain control and variable values/alarm messages to the master device. The reporting method renders the structure easy to understand because the slave devices deliver packets with a regular predefined message format to the master device at the same time interval. In contrast, the polling method requires preprocessing because the format and meaning of the response packet of the slave device differ based on the command message delivered by the master device. However, the number of command messages for the specific process loop is limited, and the response messages for processing them are predefined and used repeatedly. In addition, the format of the response message of the corresponding command message and the meaning of each field are predefined and used repeatedly. In addition, the major Ethernet-based CPIS protocol using the polling method exhibits similar header structures and values between the command and response messages. Based on the characteristics of the CPIS network and protocol described in this section, the assumptions of this study are as follows:

- *Assumption 1:* Compared with the IT environment, the CPIS exhibits less variability in terms of network configuration and topology, and master (HMI, PLC) devices communicate with multiple slave (sensor, actuator) devices in a hierarchical structure. In addition, the formats and sequence order of messages for managing processes are defined because the role of each device is determined based on the predefined process loop.
- *Assumption 2:* The structure and detailed formats of messages generated in a specific site are highly similar; additionally, the structure, detailed formats, and values of the header field between the command and response messages are similar in the polling network operation system.
- *Assumption 3:* In the header fields of various CPIS protocols, the field names are different, but fields signifying the command code, length, transaction ID, object address, and object count are commonly included. In addition, the CPIS protocol payload during communication between the lower layer devices primarily includes the state and the measured value; additionally, a correlation exists between the state and measured values.

The proposed method for inferring the packet field structure and semantics from the collected packets was based on the three assumptions above.

IV. PROPOSED METHOD

The proposed method infers the protocol field structure and semantics without prior knowledge regarding each site and protocol. The field inference includes external signature grouping, message field inference, header field inference, and payload format inference stages, as shown in Fig. 3. Rules are generated based on the result of each inference stage. The details of each inference stage are provided in this section.

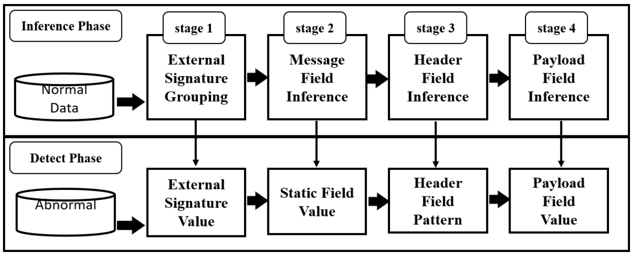


FIGURE 3. Proposed protocol format and field semantics inference method.

A. EXTERNAL SIGNATURE GROUPING

In external-signature-based grouping, packets are grouped through the standard header field information for achieving a high probability of the same message format structure. Each field device in the CPIS has a predefined role in managing the process loop; therefore, the similarity and periodicity of messages sent between the same devices are high. Hence, we leveraged the MAC address of the layer 2, a unique identification field for the device, and the IP address of the layer 3, a device identification field in the network. Additionally, we leveraged the EtherType field value of layer 2 and the port number of layer 4 to identify the CPIS protocol and service types. In addition, packets were classified based on a length pair comprising a command message and a response message because this information is the minimum requirement to identify the same message. Finally, we used nine-tuples {destination MAC address, source MAC address, EtherType value, destination IP address, source IP address, destination port number, source port number, command message length, response message length} to group the collected packets, as shown in Fig. 4.

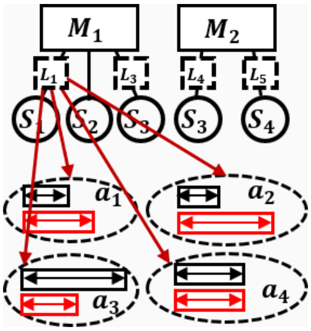


FIGURE 4. Overview of external signature grouping.

B. MESSAGE FIELD INFERENCE

The external-signature-based grouping method identifies a communication channel based on network addresses and then groups the messages. However, CPIS physical devices often do not have a network address, and they communicate with intermediate devices such as PLCs. Therefore, one communication channel can transmit multiple messages for measurement and control processes. However, for detailed classification, it requires detailed information of header fields and payloads such as the function code, object address, and object count. As such, the hex values of the protocols must be compared to identify the patterns. Therefore, we used sequence alignment techniques utilized in deoxyribonucleic acid (DNA) and protein sequence comparison analysis in bioinformatics, instead of various Natural Language Processing (NLP) techniques that primarily focus on text-based sequences. Sequence alignment is a method of arranging the DNA or protein sequences to identify regions of similarity that may arise from functional, structural, or evolutionary relationships among the sequences [39]. Sequence alignment can be classified as (i) pairwise sequence alignment for comparing two sequences, and (ii) multiple sequence alignment (MSA) for comparing three or more sequences. Pairwise sequence alignment can be classified into global alignment and local sequence alignment. MSA involves various methods such as progressive sequence alignment and iterative sequence alignment [40]. Sequence alignment is similar to n-gram methods, except that it considers GAP.

Because the number of types of messages is not known at this stage, we used Clustal Omega MSA techniques to compare and rearrange the hex messages [41]. The Clustal Omega technique sequentially executes the k-tuple method for pairwise alignment, mbed and k-means for sequence clustering, unweighted pair group method with arithmetic mean (UPGMA) packages for generating a guide tree, and HHalign packages for progressive alignments. Clustal Omega contains six parameters; however, in this study, we used all default values for reproducibility [42]. Although the complexity of Clustal Omega is lower than that of other MSAs, it is difficult to process all the data collected because CPIS environments generate a significant amount of data in a short duration. Hence, message preprocessing and data sampling are required, as shown in Fig. 5.

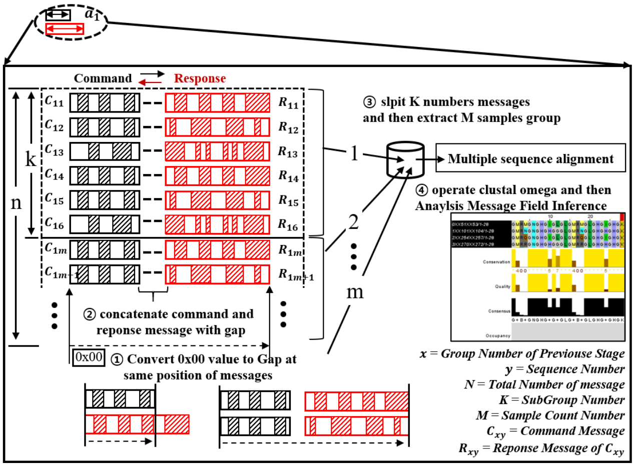

a_1
Command
Response
C_{11}
C_{12}
C_{13}
C_{14}
C_{15}
C_{16}
C_{1m}
C_{1m+1}
R_{11}
R_{12}
R_{13}
R_{14}
R_{15}
R_{16}
R_{1m}
R_{1m+1}
k
n
1
2
m
③ slpit K numbers messages and then extract M samples group
Multiple sequence alignment
④ operate clustal omega and then Anaylsis Message Field Inference
Conservation
Quality
Consensus
Occupancy
② concatenate command and reponse message with gap
0x00
① Convert 0x00 value to Gap at same position of messages
x = Group Number of Previouse Stage
y = Sequence Number
N = Total Number of message
K = SubGroup Number
M = Sample Count Number
C_{xy} = Command Message
R_{xy} = Reponse Message of C_{xy}

FIGURE 5. Message field inference based on multiple sequence alignment.

Message preprocessing converts 0 × 00 values to gap characters if 0 × 00 values are fixed at the same position of the 1) overall message, 2) command and response message groups, and 3) {a1, a2, a3, and a4} groups. A fixed 0 × 00 value in the same position of all messages implies no information at the position; however, Clustal Omega uses this value to compare and sort the hex sequences. Therefore, we preprocessed the 0 × 00 value to reduce the time/space complexity by preventing the MSA from affecting sequence comparison and rearrangement. Furthermore, the gap character was used as the delimiter in the hex sequence. Subsequently, 1) response messages corresponding to command messages were connected and arranged sequentially. 2) The N messages were grouped in terms of K messages in sequence, and M groups were randomly selected to perform MSA. Based on the inference presented in Section 3, the slave device processed the command message delivered by the master device and provided the resulting value in the response message. Therefore, the response message was determined by the command message. Conversely, command messages can be identified through the response messages. Therefore, we concatenated the command message and corresponding response messages and then performed MSA to separate the messages based on format differences. Moreover, because the types of messages are limited and periodic, if K is greater than the number of message types, then the result of performing MSA on K sorted messages is the same as that of performing MSA on all messages. However, the variability in the variable fields of messages collected within a short duration can be extremely low and hence can be recognized as fixed fields when the messages are compared. Therefore, we extracted additional M groups of random locations to identify the variable fields. Because MSA was performed for each group with the same length of command and response messages, we assumed K and M values of 10 and 20, respectively. Finally, the hex value distribution was analyzed based on the location of rearranged messages to distinguish the field types (constant, categorical, and variable).

C. HEADER FIELD INFERENCE

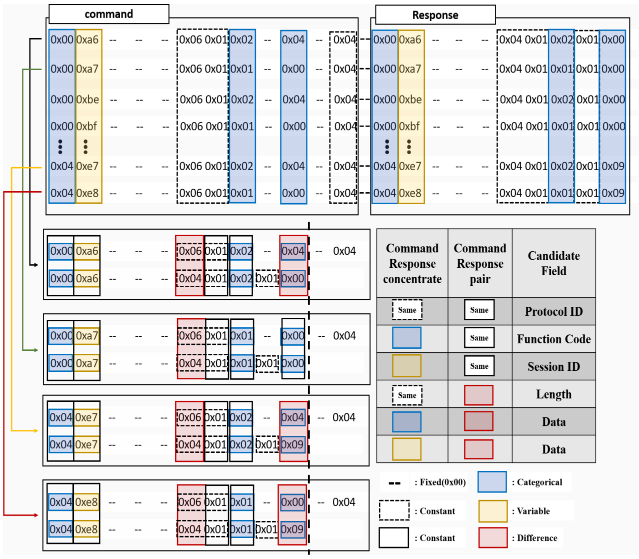
In the header field inference, the meaning of the header field format and semantics is inferred by analyzing the pattern of the hex value for the rearranged message. In the CPIS protocol, the header field contains important information for classifying messages, such as the function code, length, transaction ID, object address, and object item count. Therefore, identifying the protocol header boundary and format in the message and inferring the meaning of each field are prioritized. Because each command and response message are compared in the previous step, in this step, the command and response message pairs are compared and rearranged through pairwise sequence alignment.

First, the header boundaries are identified through local sequence alignment, which implies matching the command and response messages, i.e., two sequences with the greatest difference, to obtain the header region, i.e., the sequence fragment with the highest similarity. In this case, the similarity between the format and content of the header field of the CPIS protocol command message and those of the corresponding response message is utilized. The detailed format within the header field is then inferred by performing a global sequence alignment that fully aligns two sequences that are considered similar to the identified header field boundary.

Finally, as shown in Fig. 6, the meaning of the header field is inferred by comparing and analyzing the types of fields identified in the previous stage and in the current stage. In the previous stage, we categorized the field types into three types. Subsequently, these types were combined with the two types from the comparison between the command and response messages for matching the important field structure and semantics in the header area.

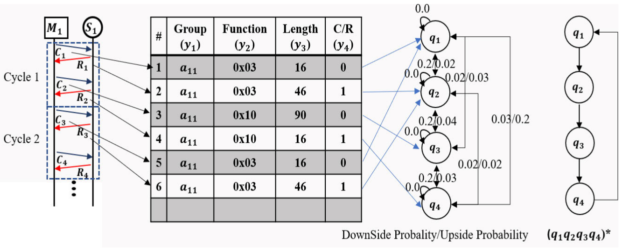
D. PAYLOAD FIELD INFERENCE

In the payload field inference stage, the information in the header field is used to determine the period of all messages in a 1:1 communication channel to infer the measured and status values of the payload. We referenced the protocol ID, session ID, function code, and length from the header field in the previous stage; however, the additional header and


command
Response
0x00 0xa6 -- -- -- 0x06 0x01 0x02 -- 0x04 -- 0x04
0x00 0xa6 -- -- -- 0x04 0x01 0x02 0x01 0x00
0x00 0xa7
0x00 0xbe
0x00 0xbf
0x04 0xe7
0x04 0xe8
0x04 0x01 0x01 0x01 0x09
0x04

Command Response concentrate	Command Response pair	Candidate Field
Same	Same	Protocol ID
🟦	Same	Function Code
🟨	Same	Session ID
Same	🟥	Length
🟦	🟥	Data
🟨	🟥	Data

FIGURE 6. Message comparison method for header field inference.



#	Group (y_1)	Function (y_2)	Length (y_3)	C/R (y_4)
1	a_{11}	0x03	16	0
2	a_{11}	0x03	46	1
3	a_{11}	0x10	90	0
4	a_{11}	0x10	16	1
5	a_{11}	0x03	16	0
6	a_{11}	0x03	46	1

FIGURE 7. Traffic cycle search based on inferred header field.

payload details (e.g., object address) must be determined to distinguish the messages. The message sequence exhibited periodicity in the 1:1 communication; therefore, we classified the message by identifying the traffic cycle based on limited header information. First, we extracted and listed {group ID, length, function code, command/response} tuple values using the header field information based on the message from the 1:1 communication channel, as shown in Fig. 7. Subsequently, a state for unique tuples was generated, and the probability of transferring states was calculated. Finally, we removed the direction with a low probability and used a cycle detection algorithm to obtain the traffic cycle.

The messages from the same cycle sequence were grouped to infer the payload field. In this study, a payload field is located after the header field boundary, and the payload field contains the state and/or measurement values. First, we identified variable fields in the payload field using a sliding window of 1-byte units (two hex codes), which is the minimum data field size unit of the CPIS protocol field, as shown in Fig. 8. Subsequently, we separated them into constant, categorical, and variable fields. We observed a few cases where the categorical field was 1) a combination of state data, or 2) the largest number field of digits in the measurement (if big-endian). To classify the two, we investigated whether a variable field existed from the categorical field to the four-byte field. If a variable field exists in an adjacent

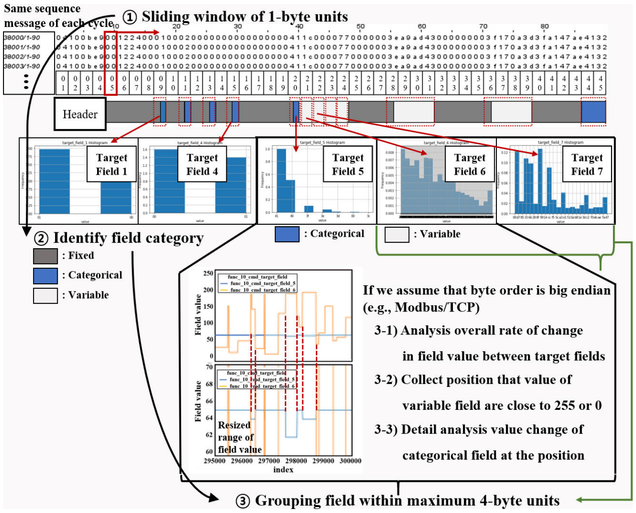

Same sequence message of each cycle
① Sliding window of 1-byte units
Header
Target Field 1
Target Field 4
Target Field 5
Target Field 6
Target Field 7
: Categorical
: Variable
② Identify field category
: Fixed
: Categorical
: Variable
Field value
Resized range of field value
index
If we assume that byte order is big endian (e.g., Modbus/TCP)
3-1) Analysis overall rate of change in field value between target fields
3-2) Collect position that value of variable field are close to 255 or 0
3-3) Detail analysis value change of categorical field at the position
③ Grouping field within maximum 4-byte units

FIGURE 8. Payload field format and semantic inference.

field, observe that the value of the variable field is constantly increasing, and when it is close to the value of 255 or the value of the variable field is constantly decreasing, and is close to zero. If the categorical field value of the associated data increases or decreases after a certain time, then the fields are combined and regarded as a numeric region. Hence, adjacent fields are investigated equally, even in the presence of numerical values, and grouped.

V. EXPERIMENTS

A. DATASET DESCRIPTION

An anomaly detection method based on the inferred protocol format and semantics is proposed herein. Therefore, the original network data (e.g., pcap file) of the CPIS containing labeled anomaly data must be used to evaluate the anomaly detection performance. The protocol field inference and anomaly detection methods were investigated using the public dataset proposed by Morris *et al.* [43]. The Morris dataset is a log of traffic data captured by an actual network of a laboratory-scale gas pipeline system and includes both normal operation and cyber-attack data. The system comprises a closed pipeline connected to a compressor, a solenoid-controlled discharge valve, and a pressure gauge.

The entire system maintains the internal pressure of the pipeline using a control scheme known as the productional integrative (PID). The systems use the Modus/TCP communication protocol for communication at the application layer. A raw dataset that stores the entire hex string values of Modbus frame and an attribute relationship file format (ARFF) dataset that extracts and stores 20 attributes from raw data exist. The two sets of data are recorded with timestamps and can be mapped. Table 1 lists some of the characteristics of the raw dataset used in this study.

The Morris testbed sends a legitimate command or launches a cyber-attack at random via an AutoIt automation and scripting language [44]. Four main attack categories are considered in the dataset: command injection, response

TABLE 1. Features of Morris dataset.

Dataset name	features	Description	Feature type
Raw dataset	Frame	Hex string of Modbus Frame	Numeric
	Categorized attack	Seven categories of attacks executed against system, including all four types of attacks.	Nominal
	Specific attack	Specific attacks (0–35)	
	Time stamp	Timestamp of Modbus packet	Numeric
	Source	Source device address	Nominal
	Destination	Destination device address	Nominal
	Specific attack	Specific attacks (0–35)	Nominal

TABLE 2. Description of attacks.

Types	Label	Abbreviation	Description of attacks
Normal	0	Normal	N/A
Response Injection Attacks	1	NMRI	Naïve Malicious Response Injection
	2	CMRI	Complex Malicious Response Injection
Command Injection Attacks	3	MSCI	Malicious State Command Injection
	4	MPCI	Malicious Parameter Command Injection
	5	MFCI	Malicious Function Code Injection
Denial of Service Attacks	6	DoS	Denial of Service
Reconnaissance	7	Recon.	Reconnaissance

injection, DoS, and reconnaissance. These four categories are further subcategorized into seven types of attacks, as shown in Table 2.

First, response injection attacks provide two types of operation. The first is naïve malicious response injection (NMRI), which performs sporadic, out-of-scope operations that do not exist in normal operations. These attacks typically occur when a malicious attacker lacks information regarding a physical system process. The second type of response injection is complex malicious response injection (CMRI). These attacks use state and physical process information to design attacks that falsify normal operations. CMRI attacks offer more sophistication than NMRI attacks. They imitate certain operations that occur within normal limits. Command injection attacks include malicious state command injection (MSCI), malicious parameter command injection (MPCI), and malicious function code injection (MFCI) attacks. These three attacks modify the system state and operation by injecting control configuration commands. MSCI attacks are designed to modify the state of the current physical process. MPCI attacks modify set points and parameters that determine the PID configuration. The MFCI attack uses network protocol commands to inject commands that change the network operation. DoS attacks attempt to disrupt communication between control and processes by interrupting or abusing the network. The final attack category is reconnaissance attack. Reconnaissance attacks are designed to collect information regarding a system either by passive collection or forcing information to be collected from a device. A total of 274,628 data exists in the dataset comprising 214,580 normal network and 60,048 attack data. The datasets can be classified as normal and attack datasets through labels. However, the response data to the attack data are labeled as normal, although they cannot occur under normal operating conditions. In this study, this dataset was removed for inferring and learning. The datasets can be accessed from the following web page: https://sites.google.com/a/uah.edu/tommy-morris-uah/CPIS-data-sets.

B. EVALUATION

The performance of anomaly detection is evaluated herein. In most anomaly detection studies, the indicators used to evaluate the performance are defined as follows and in Table 3.

- True positive (TP): When the proposed method classifies an attack and the data are attack data.
- True negative (TN): When the proposed method classifies an attack but the data are not attack data.
- False positive (FP): When the proposed method classifies as normal data and the data are normal data.
- False negative (FN): When the proposed method classifies as normal data but the data are attack data.

TABLE 3. Confusion matrix setting for Morris dataset.

Symbol	Positive (Attack packet: 1-7/1-5)	Negative (Normal packet: 0)
Positive	True Positive – Attack Correctly classified	False Positive – Attack Incorrectly classified
Negative	False Negative – Normal Incorrectly classified	True Negative – Normal Correctly classified

For the performance evaluation, we used precision, recall, and F1 scores that consider the indicators above for the final result obtained using the multilayer detection method. The precision is defined in Equation (1), where TP and FP are the sizes of true and false positives, respectively.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{1}$$

The recall is defined as shown in Equation (2), where FN is the size of false negatives.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{2}$$

The F1-measure is defined in Equation (3).

$$\text{F1 score} = 2^{*}\text{Precision}^{*}\text{Recall}/(\text{Precision} + \text{Recall}) \tag{3}$$

However, in the case involving the Morris dataset, the generation and transmission of attack data were performed by an external device; therefore, several attack data were detected in the external signature detection stage. As this renders it difficult to evaluate the performance of subsequent stages, the performance was evaluated after the static field-based detection stage. Fig. 9 shows the performance of the proposed

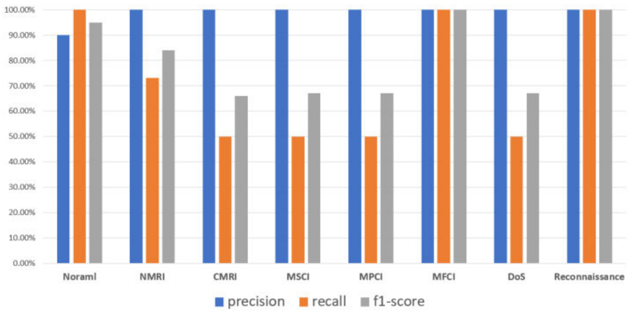


FIGURE 9. Performance of anomaly detection based on format and semantic inference method.

TABLE 4. Performance comparison with other studies.

Attack types	Number	17' Feng [45]	18' Khan [46]	This study
Normal Data	214580	Not specified	Not specified	0.95
NMRI	7753	0.88	0.93	0.84
CMRI	13035	0.67	0.76	0.66
MSCI	7900	0.62	0.68	0.67
MPCI	20412	0.80	0.85	0.67
MFCI	4898	1.00	1.00	1.00
DoS	2176	0.94	0.98	0.67
Recon.	3874	1.00	1.00	1.00

method for the Morris dataset. The proposed method demonstrated a precision of 100% for all attacks; this implies that the detected packets were all real attacks. Therefore, the proposed method is suitable for systems requiring high availability and reliability domains, such as CPISs. However, the recall rates were lower in the CMRI, MSI, and MPCI attacks. This is because, in the case of the Morris dataset, normal response data were labeled as anomaly in case of the response to MSCI and MPCI attacks. This may vary depending on the definition of anomaly. However, in this study, the detection rate was not changed for the case above to achieve an accurate evaluation. Finally, it was confirmed that DoS attacks were detected in the header field pattern-based detection stage. However, the proposed method could not detect attacks when the entire cycle pattern attack messages from a normal operating environment were injected simultaneously. For DoS attacks, detection performance can be improved using the packet flow feature-based detection method rather than using protocol field and semantics inference methods. Nonetheless, the overall F1 scores were similar to those of existing knowledge-based anomaly detection methods.

Table 4 shows a comparison of the F1-scores obtained in this study using the proposed method with those of Feng *et al.* [45] and Khan *et al.* [46], which were obtained using the proposed anomaly detection method based on the same Morris dataset. The results obtained in this study using the proposed method was comparable to those of Feng but worse than those of Khan. However, both Feng and Khan used ARFF files, in which the values of header and payload fields were preprocessed and extracted based on prior knowledge. However, in this study, anomalies were detected after inferring and extracting a field from the raw data, i.e., the hex value of the protocol. In particular, it is encouraging that for the MSCI and MPCI attacks, which manipulated the state and measurement values included in the payload field, the performance was similar to those of comparative studies.

VI. DISCUSSION

In existing studies, it is assumed that the structure of the protocol and the meaning of each field are known through prior knowledge of the site where data were collected as well as the protocol. However, this method requires a significant amount of time and manpower and must be performed repeatedly at each site. Moreover, many CPIS sites are critical infrastructures that require the strictest confidentiality; therefore, information is difficult to acquire. This can result in extremely low efficiency and scalability when applying the existing methods to the real site. To solve this problem, anomaly detection methods based on packet flow features and protocol reverse engineering have been proposed; however, the IT environment rather than CPISs was targeted in existing studies. Therefore, in this study, the characteristics of the CPIS network and protocol were observed, and the main characteristics were summarized. Subsequently, a method to infer the format and meaning of the header and payload fields of the protocol without using prior knowledge of each site was proposed. Because the proposed method infers the payload field, it can compensate for the limitations of previous protocol reverse engineering studies conducted based on the CPIS protocol. Finally, a multiple-level anomaly detection method was proposed, and the experimental results showed that the method performed similarly to existing studies.

To compare and evaluate the performances of anomaly detection methods based on the protocol reverse engineering of this study, 1) raw network data, 2) normal/anomaly labeling, and 3) public data are required. However, public data for anomaly detection in most CPIS fields are provided by preprocessing data from the original network and extracting values by classifying each field. Therefore, in this study, the Morris dataset, which provides both the original data of the network and the preprocessed data, was used for performance evaluation. In the case of iTrust's SWaT dataset, a network dataset is provided; however, it was not used because it is difficult to accurately evaluate the detection performance because only the attack period information is described. However, in the case of the Morris dataset, the entire dataset is provided with a combination of normal/anomaly data. This rendered it difficult to learn the normal data because they must be extracted from the entire dataset. For example, when learning a traffic pattern, an abnormal pattern appears in the area from which the attack data are extracted. Therefore, we proposed a method to derive a pattern through a probabilistic state transition sequence, instead of using the DFA, which is used in many existing studies. In addition, in cases involving attack data that manipulate status and measurement values, it is difficult to apply learning methods for sequential status and measurement values of the normal data because the previous message has been extracted as attack data. Therefore, to improve the performance of the proposed method, network data collected in a normal operating environment are required.

VII. CONCLUSION

We proposed a method for inferring the format and meaning of fields for a CPIS binary protocol that is used extensively in the CPIS domain without using prior knowledge of each site. Because the proposed method manipulates the original network data instead of the separated field values extracted through preprocessing, it can be directly applied to CPIS sites. In addition, it has been shown that not only the header field but also the payload field, which was inferred restrictedly in existing reverse engineering studies owing to the CPIS protocol, can be inferred. The payload field includes the measurement and state values. This implies that it can respond to attacks that are targeted at measurement and status values, classified as advanced attacks. Furthermore, the multilevel detection method using an inferred field is simple; however, it is designed to detect to various types of attacks.

The experimental results showed that the proposed method yielded an average F1 score of 0.75 for response injection attacks and 0.78 for command injection attacks. Furthermore, it was confirmed that the proposed method is sufficiently practical based on a comparison with knowledge-based studies. Because the proposed method does not require prior knowledge regarding each site and protocol, it can reduce the analysis time and cost. In addition, it is highly compatible because it corresponds to the preprocessing stage generally assumed in existing network anomaly detection studies pertaining to CPISs.

For future works, in a CPIS site that requires a fast real-time operation, the polling method is used in the initial setting stage or the environment change, whereas the reporting method is used for other operations. Therefore, further investigations must be performed to infer packet fields and meanings in a reporting environment.

REFERENCES

[1] T. Sauter, "The three generations of field-level networks—Evolution and compatibility issues," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3585–3595, Nov. 2010.

[2] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier," Symantec Corp., Tempe, AZ, USA, White Paper, 2011, p. 29, vol. 5, no. 6.

[3] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, and S. Sezer, "Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid," in *Proc. 4th Int. Symp. ICS SCADA Cyber Secur. Res.*, Apr. 2016, pp. 53–63.

[4] A. Cherepanov, "WIN32/INDUSTROYER: A new threat for industrial control systems," ESET, Bratislava, Slovakia, White Paper, Jun. 2017.

[5] K. E. Hemsley and E. Fisher, "History of industrial control system cyber incidents," Idaho Nat. Lab., Idaho Falls, ID, USA, Tech. Rep. INL/CON-18-44411-Rev002, 2018.

[6] Kaspersky. (Dec. 2020). *ICS Threat Predictions for 2021*. Accessed: Dec. 8, 2020. [Online]. Available: https://ics-cert.kaspersky.com/media/Kaspersky-ICS-CERT-Threat-Predictions-2021-EN.pdf

[7] J. McCarthy, "Securing manufacturing industrial control systems: Behavioral anomaly detection," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 8219, 2018.

[8] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2016, pp. 88–99.

[9] A. Lemay and J. M. Fernandez, "Providing SCADA network datasets for intrusion detection research," in *Proc. 9th Workshop Cyber Secur. Experimentation Test (CSET)*, 2016, pp. 1–8.

[10] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim, "HAI 1.0: HIL-based augmented ICS security dataset," in *Proc. 13th USENIX Workshop Cyber Secur. Experimentation Test (CSET)*, 2020, pp. 1–5.

[11] H. Yoo and T. Shon, "Challenges and research directions for heterogeneous cyber–physical system based on IEC 61850: Vulnerabilities, security requirements, and security architecture," *Future Gener. Comput. Syst.*, vol. 61, pp. 128–136, Aug. 2016.

[12] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Comput. Electr. Eng.*, vol. 52, pp. 114–124, May 2016.

[13] S. A. Chaudhry, T. Shon, F. Al-Turjman, and M. H. Alsharif, "Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems," *Comput. Commun.*, vol. 153, pp. 527–537, Mar. 2020.

[14] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang, "Intrusion detection system for IEC 60870-5-104 based SCADA networks," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2013, pp. 1–5.

[15] K. Wong, C. Dillabaugh, N. Seddigh, and B. Nandy, "Enhancing Suricata intrusion detection system for cyber security in SCADA networks," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–5.

[16] W.-S. Jung, J.-H. Yun, S.-K. Kim, K.-S. Shim, and M.-S. Kim, "Structured whitelist generation in SCADA network using PrefixSpan algorithm," in *Proc. 19th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2017, p. 326.

[17] J. Nivethan and M. Papa, "A Linux-based firewall for the DNP3 protocol," in *Proc. IEEE Symp. Technol. Homeland Secur. (HST)*, May 2016, pp. 1–5.

[18] D. Li, H. Guo, J. Zhou, L. Zhou, and J. W. Wong, "SCADAWall: A CPI-enabled firewall model for SCADA security," *Comput. Secur.*, vol. 80, pp. 134–154, Jan. 2019.

[19] N. Goldenberg and A. Wool, "Accurate modeling of modbus/TCP for intrusion detection in SCADA systems," *Int. J. Crit. Infrastruct. Protection*, vol. 6, no. 2, pp. 63–75, Jun. 2013.

[20] M.-K. Yoon and G. F. Ciocarlie, "Communication pattern monitoring: Improving the utility of anomaly detection for industrial control systems," in *Proc. NDSS Workshop Secur. Emerg. Netw. Technol.*, 2014, pp. 1–10.

[21] A. Kleinmann and A. Wool, "Automatic construction of statechart-based anomaly detection models for multi-threaded industrial control systems," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 4, pp. 1–21, Jul. 2017.

[22] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proc. 1st ACM Workshop Cyber-Phys. Syst. Secur.*, Apr. 2015, pp. 13–24.

[23] C. Zhou, S. Huang, N. Xiong, S.-H. Yang, H. Li, Y. Qin, and X. Li, "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 10, pp. 1345–1360, Oct. 2015.

[24] S. Kwon, H. Yoo, and T. Shon, "IEEE 1815.1-based power system security with bidirectional RNN-based network anomalous attack detection for cyber-physical system," *IEEE Access*, vol. 8, pp. 77572–77586, 2020.

[25] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 1058–1065.

[26] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2017, pp. 140–145.

[27] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proc. Workshop Cyber-Phys. Syst. Secur. PrivaCy*, Jan. 2018, pp. 72–83.

[28] S. Kim, W. Jo, and T. Shon, "APAD: Autoencoder-based payload anomaly detection for industrial IoE," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106017.

[29] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[30] B. D. Sija, Y.-H. Goo, K.-S. Shim, H. Hasanova, and M.-S. Kim, "A survey of automatic protocol reverse engineering approaches, methods, and tools on the inputs and outputs view," *Secur. Commun. Netw.*, vol. 2018, pp. 1–17, Feb. 2018.

[31] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, "N-gram against the machine: On the feasibility of the N-gram network analysis for binary protocols," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2012, pp. 354–373.

[32] J.-Z. Luo and S.-Z. Yu, "Position-based automatic reverse engineering of network protocols," *J. Netw. Comput. Appl.*, vol. 36, no. 3, pp. 1070–1077, May 2013.

[33] S. Tao, H. Yu, and Q. Li, "Bit-oriented format extraction approach for automatic binary protocol reverse engineering," *IET Commun.*, vol. 10, no. 6, pp. 709–716, Apr. 2016.

[34] *Industrial Automation and Control Systems Security*, Standard ISA99, ISA, 2018.

[35] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-82, 2011, p. 16.

[36] M. Fabro, "Recommended practice: Improving industrial control system cybersecurity with defense-in-depth strategies," DHS Ind. Control Syst. Cyber Emergency Response Team, Tech. Rep., 2016.

[37] C. Feng, V. R. Palleti, A. Mathur, and D. Chana, "A systematic framework to generate invariants for anomaly detection in industrial control systems," in *Proc. NDSS*, Feb. 2019, pp. 1–15.

[38] Y. Chang, S. Choi, J.-H. Yun, and S. Kim, "One stage more: Automatic ICS protocol field analysis," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2017, pp. 241–252.

[39] D. W. Mount and D. W. Mount, *Bioinformatics: Sequence and Genome Analysis*, vol. 1. Cold Spring Harbor, NY, USA: Cold Spring Harbor Laboratory Press, 2001.

[40] J. Daugelaite, A. O'Driscoll, and R. D. Sleator, "An overview of multiple sequence alignments and cloud computing in bioinformatics," *Int. Scholarly Res. Notices*, vol. 2013, Jun. 2013, Art. no. 615630.

[41] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, and D. G. Higgins, "Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega," *Mol. Syst. Biol.*, vol. 7, no. 1, p. 539, Jan. 2011.

[42] F. Sievers and D. G. Higgins, "Clustal omega for making accurate alignments of many protein sequences," *Protein Sci.*, vol. 27, no. 1, pp. 135–145, Jan. 2018.

[43] T. H. Morris and Z. T. I. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," in *Proc. 7th Annu. Southeastern Cyber Secur. Summit*, 2015, pp. 3–4.

[44] J. Brand and J. Balvanz, "Automation is a breeze with AutoIt," in *Proc. 33rd Annu. ACM SIGUCCS Conf. User Services (SIGUCCS)*, 2005, pp. 12–15.

[45] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2017, pp. 261–272.

[46] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, vol. 7, pp. 89507–89521, 2019.

HYUNJIN KIM (Graduate Student Member, IEEE) received the B.S. degree in information and computer engineering and the M.S. degree in computer science and engineering from Ajou University, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in artificial intelligence convergence networks. His current research interests include industrial control systems and automotive systems that apply machine learning and protocol reverse engineering for anomaly detection.

SUNGJIN KIM received the B.S. degree in information and computer engineering from Ajou University, Suwon, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in computer engineering. He is also a member of the Information and Communication Security Laboratory. His current research interests include software vulnerability analysis, machine learning, anomaly detection, and industrial control systems.

WOOYEON JO (Graduate Student Member, IEEE) received the B.S. degree in computer engineering from Ajou University, Suwon, South Korea, in 2015, where he is currently pursuing the integrated M.S./Ph.D. degree. His research interests include the development of digital forensic tools and techniques that utilize metadata from various file systems, network forensics, extensions to digital forensic science, and application of digital forensics on control systems.

KI-HYUN KIM received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1993 and 1995, respectively, and the Ph.D. degree in information security from Chungbuk National University, Cheongju, South Korea, in 2011.

He is currently a CTO with the Research and Development Center, NNSP Company Ltd., Seoul, South Korea. His current research interests include ICS network perimeter security, industrial control system security, ICS security monitoring, and intrusion detection.

TAESHIK SHON (Senior Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Ajou University, Suwon, South Korea, in 2000 and 2002, respectively, and the Ph.D. degree in information security from Korea University, Seoul, South Korea, in 2005.

From August 2005 to February 2011, he was a Senior Engineer with the Convergence S/W Laboratory, DMC Research and Development Center, Samsung Electronics Company Ltd. In 2017, he was a Visiting Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA. He is currently a Professor with the Division of Cyber Security, College of Information Technology, Ajou University. His research interests include industrial control systems, anomaly detection algorithms, and digital forensics.

He was awarded the KOSEF Scholarship to be a Research Scholar at the Digital Technology Center, University of Minnesota, Minneapolis, USA, from February 2004 to February 2005. He was awarded the Gold Prize for the Sixth Information Security Best Paper Award from the Korea Information Security Agency, in 2003; the Honorable Prize for the 24th Student Best Paper Award from Microsoft-KISS, in 2005; the Bronze Prize for the Samsung Best Paper Award, in 2006; the Second Level of TRIZ Specialist Certification in Compliance with the International TRIZ Association Requirement, in 2008; and the Silver, the Bronze, and the Excellent Publication Prize for the Ajou University Award, in 2013, 2014, and 2016, respectively. He is also a Guest Editor, an Editorial Staff, and a Review Committee Member of *Computers and Electrical Engineering* (Elsevier), *Mobile Networks and Applications* (Springer), *Security and Communication Networks* (Wiley InterScience), *Wireless Personal Communications* (Springer), the *Journal of the Korea Institute of Information Security and Cryptology*, the *IAENG International Journal of Computer Science*, and other journals.