Intrusion Detection and Identification Using Tree-Based Machine Learning Algorithms on DCS Network in the Oil Refinery

Kyoung Ho Kim, Byung Il Kwak, Mee Lan Han, and Huy Kang Kim, *Member, IEEE*

***Abstract*—Recently, Critical Infrastructures (CI) such as energy, power, transportation, and communication have come to be increasingly dependent on advanced information and communication technology (ICT). This change has increased the connection between the Industrial Control System (ICS) supporting the CI and the Internet, resulting in an increase in security threats and allowing a malicious attacker to manipulate and control the ICS arbitrarily. On the other hand, ICS operators are reluctant to install security systems for fear of adverse effects on normal operations due to system changes. Therefore, new research is needed to detect anomalies quickly and identify attack types while ensuring the high availability of ICS. This study proposes a host-based method to detect and identify abnormalities in an Oil Refinery's Distributed Control System (DCS) network using DCS vendor-proprietary protocols using a proposed method based on the tree-based machine learning algorithm. The results demonstrate that the proposed method can effectively detect an abnormality with the eXtreme Gradient Boosting (XGB) classifier, with up to 99% accuracy. Taken together, the results of this study contribute to the accurate detection of abnormal events and identification of attack types on the network without disrupting the normal operation of the DCS in the Oil Refinery.**

***Index Terms*—Industrial control system, distributed control system, intrusion detection, attack identification.**

I. Introduction

In the era of globalization, where the world is interconnected with the same supply chains, cyberattacks targeting Critical Infrastructure (CI) are a significant global threat. While the damage of cyberattacks targeting Information Technology (IT) systems is limited to financial losses, if cyberattacks against the ICS, which is at the core of the CI, are successful, the damage will not be limited to financial losses. An example of a recent Colonial Pipeline ransomware attack in the United States illustrates the severity of cyberattacks targeting ICS. A ransomware infection of a pipeline company, which accounts for nearly 50% of fuel supplies in the eastern United States, has caused not only financial losses to Colonial Pipeline but also social disruptions due to gas shortages and rising gasoline prices. If a cyberattack on a major oil refinery causes an explosion, the damage will be catastrophic, including the loss of human life, environmental pollution, and disruption of the global energy supply chain system. Accordingly, in recent years, there has been a steady increase in ICS security interest and academic research.

In the past, ICS has been relatively securely managed from cyber threats due to the ICS vendor's proprietary Operating System (OS) and network protocol in an isolated network. However, ICS started to connect to the corporate network and the Internet to improve productivity by deriving optimal values from process operation information. In addition, ICS manufacturers started to use general-purpose OS such as Windows OS in Ethernet and TCP/IP network environments for cost-saving. As a result, the security threats in the ICS environment are inherited from the IT environment.

To aggravate the situation, the ICS environment has a long-life cycle of systems, so it is impossible to respond to the latest security threats because many devices use the OS whose service support period has expired. In addition, critical security patches are not applied on time due to availability priorities, making them more dangerous than systems operated in the IT environment. Furthermore, it is not easy to install security programs on ICS because installing the latest security programs on the aging ICS equipment can adversely affect normal process operations due to the lack of resources.

Therefore, this paper studied a method to detect attacks by analyzing packets in the ICS network, particularly the DCS network operated in the Oil Refineries. The proposed approach is one of the most practical methods that can be immediately applied and used in the real production environment because it does not directly affect the performance of the operating equipment and requires no changes due to the installation of security programs.

Major contributions of our paper can be summarized as follows.

- We propose a machine learning-based framework to achieve intrusion detection and attack identification for the DCS network using DCS vendor-proprietary protocols.

Manuscript received 2 August 2021; revised 17 December 2021; accepted 5 February 2022. Date of publication 10 February 2022; date of current version 20 October 2022. This work was supported by the Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00624), Development of Intelligence Cyber Attack and Defense Analysis Framework for Increasing Security Level of C-ITS. Paper no. TPWRS-01211-2021. *(Kyoung Ho Kim, Byung Il Kwak, and Mee Lan Han contributed equally to this work.) (Corresponding author: Huy Kang Kim.)*

Kyoung Ho Kim, Mee Lan Han, and Huy Kang Kim are with the School of Cybersecurity, Korea University, 02841 Seoul, Republic of Korea (e-mail: quixote@korea.ac.kr; blosst@korea.ac.kr; cenda@korea.ac.kr).

Byung Il Kwak is with the School of Software, Hallym University, 24252 Gangwon-do, Republic of Korea (e-mail: kwacka12@hallym.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2022.3150084.

Digital Object Identifier 10.1109/TPWRS.2022.3150084

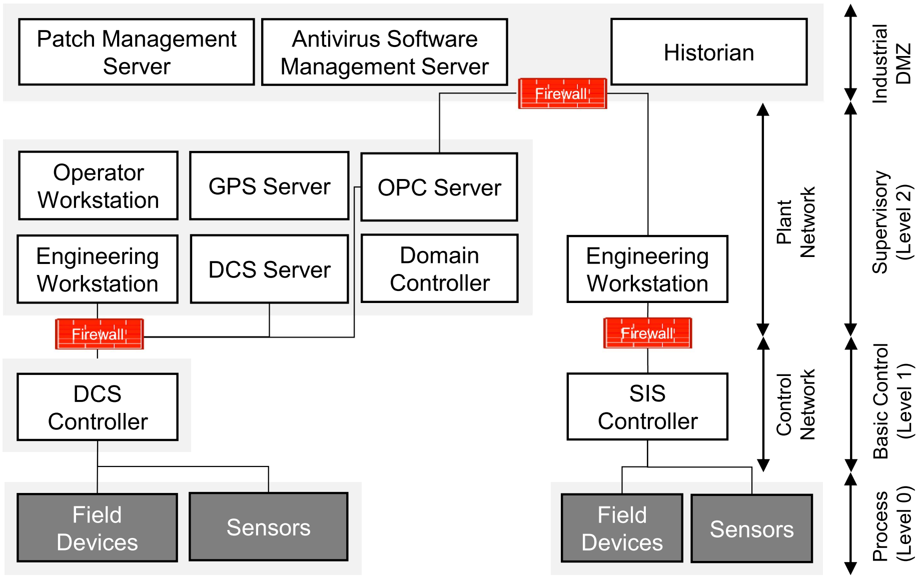


Fig. 1. DCS Network Architecture.

This framework focuses on the host-based DCS network traffic properties.

- We collected normal traffic (i.e., without the attacks) and abnormal traffic (i.e., after performing the attack scenarios) in an environment very similar to the actual operating environment of the DCS network to efficiently design our proposed algorithm and to support its evaluation.
- We also conducted extensive evaluations of the proposed methods' performance, such as the accuracy metrics for intrusion detection and a multi-class classification for several attack types. The results revealed that the proposed method can effectively detect an abnormality with the tree-based machine learning models in the DCS.

The remainder of this paper is organized as follows. Section II reviews the background and related work. In Section III, we describe the experimental setup and attack scenarios. Section IV presents the functionalities of the DCS network, design features, and algorithms of the proposed scheme. In Section V, we explain the performance metrics and report our experimental results. Finally, conclusions are drawn in Section VI.

II. Background and Related Work

In Section II-A, we provide a detailed information on the characteristic and mechanism of DCS network. In Section II-B, we describe various attack vectors and attack pathways mainly used for DCS attacks. Finally, in Section II-C, we review existing security systems and solutions to reinforce DCS security.

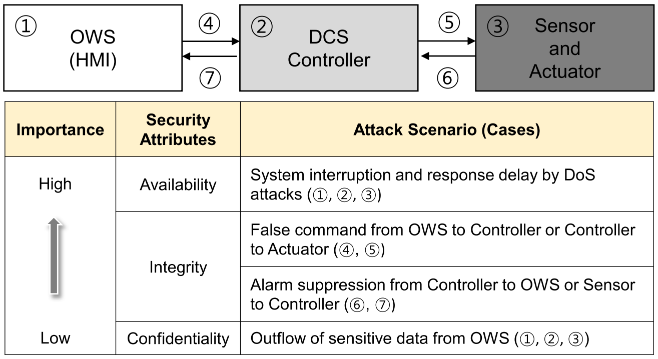
A. DCS Network

A DCS is one of the ICS and is mainly used to control production systems within the same geographical location of industries such as refineries, water treatment, chemical manufacturing plants, and pharmaceutical treatment facilities [1]. As shown in Fig. 1, this system consists of several components that perform different roles, such as Operator Workstation (OWS) also referred to as Human Machine Interface (HMI), Engineering Workstation (EWS), DCS Controller, DCS Server, Domain Controller (DC) server and OLE for Process Control (OPC) server. The HMI sends set points to and directly or indirectly requests data from the DCS Controllers through the DCS Servers. EWS is used to manage controller setting information and system configuration. DCS server provides screen values, user profile information to HMI, and gateway functions between Control Network (CN) and Plant Network (PN). The DC plays a role in managing user account, authentication, and group policies centrally. The DCS controllers control their process actuator based on control command and sensor feedback from process sensors.

Each component is connected to the DCS network to exchange information among them. According to the Purdue Enterprise Reference Architecture (PERA), the de-facto standard of the ICS architecture, the DCS systems are located at Level 1 and 2. DCS controllers are in the CN and supervisory and monitoring systems such as OWS, EWS and DCS servers are in the PN. Communication between components located on two networks using different network protocols is via a DCS server with the gateway function. The DCS server communicates with subordinates located below Level 1 through the CN. These data communications leverage the vendor's proprietary protocol, making it difficult to detect cyberattacks using an IT-based Intrusion Detection System (IDS).

B. Attack Vectors

In the past, ICS operated relatively securely, separated from the corporate network through Air Gap Defense Technology. Recently, many companies have started to analyze operation information and log information generated from sensors and actuators located on the field to improve predictive maintenance performance to secure availability and to increase productivity



Importance	Security Attributes	Attack Scenario (Cases)
High	Availability	System interruption and response delay by DoS attacks (①, ②, ③)
↑	Integrity	False command from OWS to Controller or Controller to Actuator (④, ⑤)
		Alarm suppression from Controller to OWS or Sensor to Controller (⑥, ⑦)
Low	Confidentiality	Outflow of sensitive data from OWS (①, ②, ③)

Fig. 2. Major security concerns of ICS.

through improving yield. To meet these needs, network connections between ICS and the Internet, corporate networks have started to grow, resulting in a significant increase in attack vectors for ICS.

The first attack vector is to use a phishing attack. This attack is the most recently used attack method targeting both IT and operational technology (OT), and it is a method of penetrating the corporate network in the first stage through a phishing attack, going through the reconnaissance, and then the invading OT networks in the second stage. The attack vectors were used for cybersecurity incidents such as the *BlackEnergy 3* attacks in Ukraine Power Grid in 2015, malware infection at German Steel Mills in 2015, and ransomware infections at Honda Motors in 2017 and 2020 [2]–[4].

The second attack vector is to use removable media. After first breaching into the corporate system and network through social engineering attacks and physical access to targets, such attack becomes possible. The most representative attack case using this type of attack was the *Stuxnet* attack on Iran's nuclear power plants in 2010, which made us interested in ICS cybersecurity [5].

The third attack vector is to use a misconfiguration for security equipment. An attacker finds and uses the misconfiguration of equipment located at the OT network perimeter, such as firewalls and Virtual Private Network (VPN), after breaching into the corporate network through the first stage attack. A representative attack case using this was the *Triton* attack on Safety Instrumented System (SIS) at a chemical plant in Saudi Arabia in 2014 [6]. In addition, there are various types of attack vectors such as system and network protocol vulnerability attacks, field equipment attacks, communication hijacking, and Man-In-The-Middle (MITM) attacks, database attacks, and others [1].

C. Security for ICS

The security objective for the DCS is to safely operate without an unplanned shutdown. To achieve this objective, availability must be guaranteed to transfer commands or responses within the given time period. Integrity must be ensured so that accurate values can be transmitted to field devices without tampering. Finally, confidentiality must be guaranteed so that the process operation and product recipe know-how are not accessed by unauthorized persons (see Fig. 2).

ICS is a system that prioritizes availability and is reluctant to change the system during normal operation. This is so because the latest security patches and antivirus updates to improve cybersecurity can affect the system's operation or require rebooting. Therefore, it is often limited to the maintenance period during the process is shut down for overhaul.

Furthermore, ICS has a longer replacement cycle compared to general IT systems. Thus, if a state-of-the-art security program is installed in an existing system, there is a risk that the response from the sensor or the delivery of commands to the actuator is delayed due to the lack of system resources and not delivered on time. Therefore, in an ICS environment, a passive response security solution that quickly detects an attack through system log or network traffic analysis is more effective than an active response security solution that blocks communication and operation processes between systems. Many previous studies sought to detect abnormalities and intrusions of ICS network traffic by reflecting these ICS characteristics [7], [8].

Since the approaches proposed in these studies are difficult to experimentally test in real environments or environments similar to real environments, most previous studies have performed many IDS tests on Cyber Physical System (CPS) testbeds, emulation-based testbeds, software-based testbeds, and virtual environment-based testbeds. In addition, due to the difficulty of generating vendor-proprietary protocols, many previous studies focused on detection of known and standardized protocols such as Distributed Network Protocol (DNP3), Modbus TCP, IEC 61850 series, and EtherNet/IP [9]. Research on the IDS technique is also using the latest technology, and the number is steadily increasing. Signature-based detection techniques, statistical detection techniques using time-series data [10], [11], knowledge-based technologies such as expert systems [12], [13], behavior-based techniques [14], [15], and, more recently, machine learning-based technologies are increasingly being studied [16]–[18].

This study focuses on attack detection and attack type identification using tree-based machine learning algorithms in an environment similar to the actual operating environment using vendor-proprietary protocols.

III. Preliminaries

In Section III-A, we describe the experiment environment, which is nearly identical to an actual operating environment. We collect normal traffic and abnormal traffic while performing various attacks in the Factory Acceptance Test (FAT) environment, which is performed before Site Acceptance Test (SAT). Section III-B introduces two attack scenarios used in the experiment and two or more attack tools for each scenario. The attack scenarios used in the experiment are most frequently used in the ICS environment.

A. Experimental Setup

The experimental environment consisted of a total of 17 units. It consists of 1 ES, 2 DCs, 5 DCS servers, and 9 HMIs, the same size as the environment of one refinery process over a medium

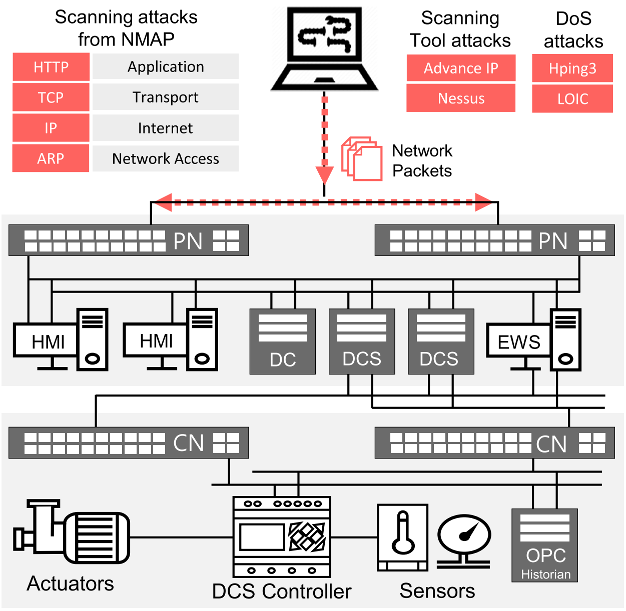


Fig. 3. Attack Scenarios.

TABLE I
DCS Network Dataset

Attack Types	Attack Tools		Normal Traffic Volume	Attack Traffic Volume
Scanning	Nessus		82,591	10,255
	Advanced IP		50,893	10,248
	NMAP	ARP	58,263	147,965
		IP	60,973	14,648
		TCP	60,543	147,989
		HTTP	59,287	80,077
DoS	Hping3		58,930	52,286
	LOIC		54,945	976,846

size. As shown in Fig. 3, the DCS network consists of a PN running over a TCP/IP network and a CN using vendor-proprietary protocols. In order to collect normal traffic and abnormal traffic including attack packets, mirroring was set on each network switch connected to PN and CN, and traffic was extracted from each network through a laptop installed with the Wireshark. Before collecting the attack traffic volumes, we collected traffics for one minute and five minutes in a normal state without connecting the attack host to the DCS network. After connecting the attack host to the DCS network, we collected traffic for one minute and second minutes while generating attack traffic. Information on the size of attack traffic (the number of the packets) employed in our proposed method is summarized in Table I.

B. Attack Scenarios

The attack tools used in the ICS environment investigated by FireEye in 2020 are identified in the following order: network discovery, software exploitation, wireless communications, and fuzzers [19]. In this study, we selected SCAN attacks using network scanners, which are the most widely used network discovery, and Denial of Service (DoS) attacks that affect availability, the most important factor in the ICS environment among CIA triad elements. We also verified how the proposed methodology detects attacks and identifies attack types. In addition, when performing a scan for each TCP/IP layer, we also checked whether our proposed methodology is good at distinguishing which layer is being scanned.

1) Network Scan Attack: Scanning is a technique used to gather computer network information, including IP address, port, version, and so forth before launching sophisticated attacks. The network scanning attack corresponds to the discovery tactic on the *MITRE ATT&CK framework for ICS matrix* [20]. For the scan attack, we selected Nessus[1] and Advanced IP,[2] which are among the most used tools by attackers [21], [22].

2) DoS Attack: DoS is a technique used to shut down or degrade a service, making it inaccessible to the intended user. Therefore, Dos attacks can adversely affect availability by impairing control over the ICS. The DoS attack corresponds to the Inhibit Response Function, Impair Process Control, and Impact tactics on the *MITRE ATT&CK framework for ICS matrix*.We chose Hping3[3] and LOIC,[4] which are frequently used tools in real-world attacks, as DoS attack [23], [24].

3) Network Scan Attack by TCP/IP Layer: In addition, we performed ARP, IP, TCP, and HTTP scans according to each TCP/IP protocol stack, adjusting the options of the NMAP[5] network scanner so that the proposed detection algorithm can accurately detect the scan attacks by TCP/IP 4-layer [25].

IV. Proposed Scheme

This section explains the proposed scheme for intrusion detection in the ICS network. This section consists of two subsections: Data Preprocessing and Detection Algorithm. In Data Preprocessing, the raw traffic data of ICS PN are transformed into the feature vector. The processed feature vector is then applied with the machine-learning classification algorithm as input values. Fig. 4 depicts an overall flowchart of intrusion detection.

A. Data Preprocessing

In this section, to apply the network traffic with the classification algorithm, we operated the time window (TW) process setting and feature vector construction.

1) Time Window and Sliding: To extract the feature vector, we set the time window from 0.1 sec to 1.0 sec. Then, the sliding window size equal to the time window not to overlap the time was selected. For example, if the time window size was 1.0 sec, we set the sliding window size as 1.0 sec. There was no overlapped time between $time\ window_{t-1}$ and $time\ window_t$. In the same size between the time window and the sliding window, the detection algorithm's operation can decrease and reduce the calculation of the feature vector.

2) Construction of Feature Vector: We constructed the feature vector to execute the intrusion detection in the ICS network as Table II. The composed features were the sum of bytes,

[1] The world's most used commercial vulnerability scanner
[2] The network scanner to analyse LAN
[3] Hping3 command can be used to perform port scans.
[4] LOIC is a DDoS program developed by the Anonymous Hacker Group.
[5] NMAP is a port scanning tool for scanning hosts or networks.

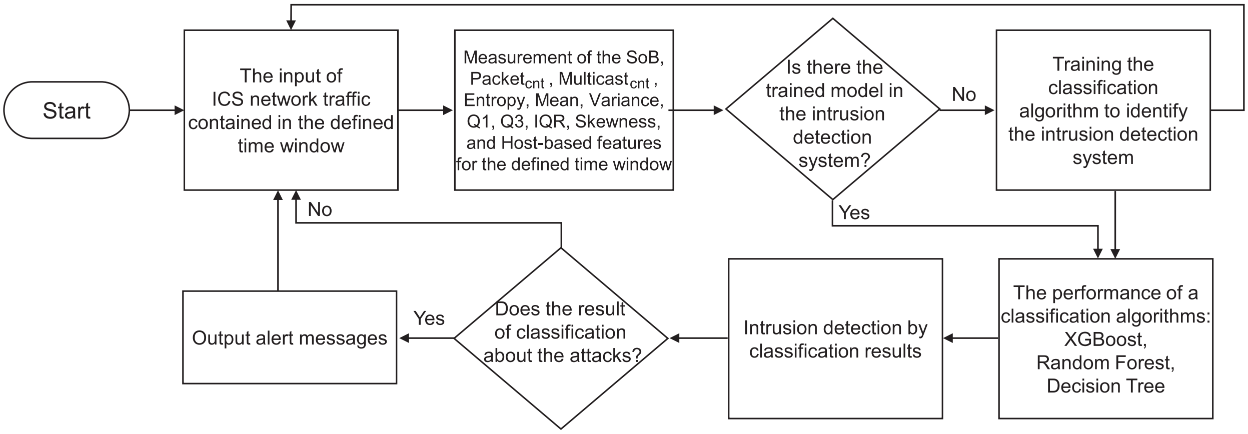
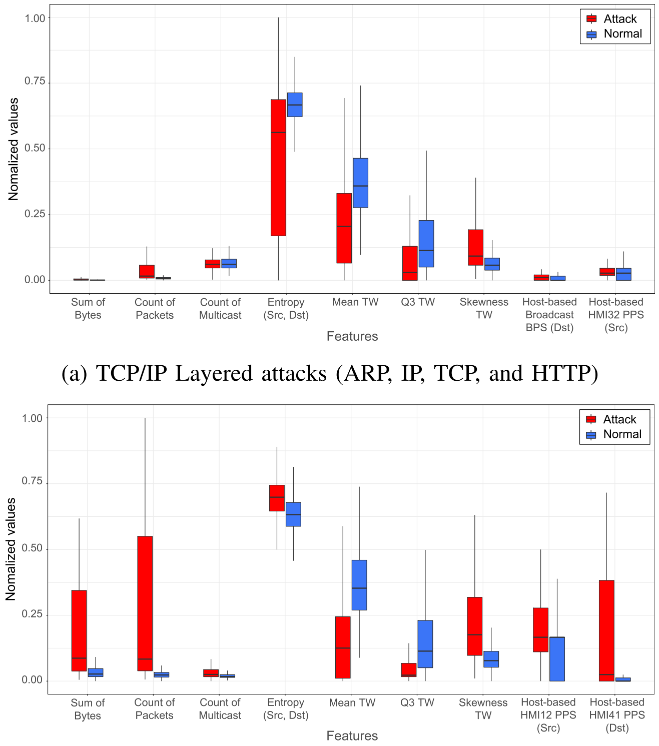

Start
The input of ICS network traffic contained in the defined time window
Measurement of the SoB, Packet$_{cnt}$, Multicast$_{cnt}$, Entropy, Mean, Variance, Q1, Q3, IQR, Skewness, and Host-based features for the defined time window
Is there the trained model in the intrusion detection system?
No
Training the classification algorithm to identify the intrusion detection system
Yes
The performance of a classification algorithms: XGBoost, Random Forest, Decision Tree
Intrusion detection by classification results
Does the result of classification about the attacks?
Yes
Output alert messages
No

Fig. 4. Flowchart of Intrusion Detection; The IDS classification model is regularly retrained depending on the network nodes changes and time.

counts of packets, multicast count, entropy between source and destination MAC address, time window related statistical value (mean, first quartile (Q1), third quartile (Q3), and skewness), and host-based features (sum of bytes and count of packets). The mean of the time window was used to measure a central tendency of the time window of the network packets. The variance of the time window showed the distribution of the time window. In quartile deviation, the Q1 and the Q3 were top 25% value and top 75% value in sorted values of the time window. The skewness was a degree of bias about the tail of time window distribution, showing the degree of the lean about the mean value between the negative side and the positive side. In the ICT networks, the sum of bytes and counts of packets, multicast count, entropy between source and destination MAC address could be different from the ICS network due to users' service usage. However, in the ICS network, the hosts periodically communicated with other linked nodes to inform the machine's status. Moreover, the number of hosts was fixed to control the linked nodes and machines. Based on these characteristics, we composed the host-based features (the sum of bytes in the linked host and the count of packets in the linked host) such as host-based Bytes per Second (BPS) and host-based Packets per Second (PPS), respectively.

We displayed the feature values through the boxplot to check the feature values with attack and normal state (See Fig. 5). Moreover, the showed features are in Table II and include two host-based features. We showed the features that are processed the min-max normalization ranged from 0 to 1. In Fig. 5(a), the general statistical features and host-based features (i.e., host-based BPS and PPS) show differences between attack and normal. However, the host-based features did not show the differences between attack and normal than statistical features in Fig. 5(b).



(a) TCP/IP Layered attacks (ARP, IP, TCP, and HTTP)

(b) Scan and DoS attacks (Advanced IP, Nessus, Hping3, and LOIC)

Fig. 5. The boxplot of normalized feature value for intrusion detection in time window 0.5 sec.

B. Detection Algorithm

As shown in Fig. 4, the intrusion detection module was operated to detect the attack scenarios' intrusions. Our proposed method measures the features such as sum of bytes, count of packets, multicast count, entropy between source and destination MAC address, time window related statistical value (i.e., mean, Q1, Q3, and skewness), and host-based features (host-based BPS and PPS) through the collected the network packets and timestamp in the defined time window. The feature vector was applied to the tree-based machine learning models with binary classification. A description of three tree-based machine learning models is provided in Table III. The results from the classification algorithm revealed the accuracy and the F1-score values as a metric of the algorithm. Algorithm 1 describes the running process of our intrusion detection module. It is crucial to detect intrusion and attacks in the PN because the PN is located in front of the CN for the ICS network. Therefore, in applying for our proposed algorithm, we focused on the intrusion detection of the PN about the ICS network. Moreover, we

Algorithm 1: Intrusion Detection.

```
Input: 'Packet' and 'Timestamp'
Output: 'Alert' log
$TW_{start}$ and $TW_{end}$ are the start time and the end time in the Time window, respectively
Timestamp is the Unix time of the received packet on the receiver
$Pkt_{buffer}$ is the packet buffer
L1 is the fixed host list
SoB and $Packet_{Cnt}$ are sum of bytes and count of packets, respectively
Model is trained on the 'Test' mode
Mode is the process state
    ▷Feature Generation
if $TW_{start} \leq Timestamp \leq TW_{end}$ then
  Store the Packet and Timestamp in the $Pkt_{buffer}$
else if $Timestamp \geq TW_{end}$ then
▷The features ranging from F1 to F9 have matched features in Table 1.
  Extract the features from F1 to F9 for $Pkt_{buffer}$
  Extract features of the SoB and the $Packet_{Cnt}$ for $Pkt_{buffer}$ depending on the L1
  if Mode == 'Training' then
    Split the dataset into the training and testing
    Train the classifier models with the training dataset
    Classify the test dataset using the trained model
    Evaluate the accuracy for the results of test dataset and labels
    Store the trained model has the highest accuracy from the evaluation
  else if Mode == 'Test' then
    Classify the test dataset using the stored model
    if Classification result == 'Abnormal' then
      Return 'Alert'
    end if
  end if
end if
```

TABLE II
DESCRIPTION FOR THE FEATURES OF INTRUSION DETECTION IN THE ICS NETWORK TRAFFIC

#	Name	Description
1	Sum of bytes	$SumOfBytes = \sum_{i=1}^{N} (byte_{x(i)})$
2	Count of packets	$PacketCnt_i = n_i$
3	Count of multicast	$MulticastCnt_i = n_i^{Multicast}$
4	Entropy of source and destination MAC address	$H = \sum_{i=1}^{N} (p_i \cdot ln(p_i)),$ $where\ p_i = \frac{x_m(i)}{\sum_{i=1}^{n} x_m(i)}$
5	Mean of TW	$\mu = \frac{1}{N} \sum_{i=1}^{N} x(i)$
6	Variance of TW	$Var = \frac{1}{N} \sum_{i=1}^{N} (x(i) - \mu)$
7	First quartile of TW (Q_1)	$Q_1 = \frac{(n+1)}{4}$
8	Third quartile of TW (Q_3)	$Q_3 = \frac{3(n+1)}{4}$
9	Skewness of TW	$Skew = \frac{1}{N} \sum_{i=1}^{N} (\frac{x(i) - \mu}{\sigma})^3$

TABLE III
TREE-BASED MACHINE LEARNING MODELS

Classifier	Ref	Description
Decision Tree	[26]	- Easy to understand, interpret, and visualize - Able to solve both regression and classification problems - Less effort for data preparation during pre-processing - Overfitting when the training data are excessively learned.
Random Forest	[26]	- Works well with both categorical and continuous values. - Overcomes the overfitting problem by creating a random forest through multiple decision trees - Requires much time for training as it combines a lot of decision trees to determine the class.
XGBoost	[27]	- Supports parallel processing of tree structure created by boosting - Combines less accurate weak models to eventually derive higher-accuracy results - Fast classification and high classification accuracy than the Gradient Boosting

TABLE IV
CONFUSION MATRIX

		Predicted class	
		Intrusion (class = positive)	Normal (class = negative)
Actual class	Intrusion (class = positive)	TP	FN
	Normal (class = negative)	FP	TN

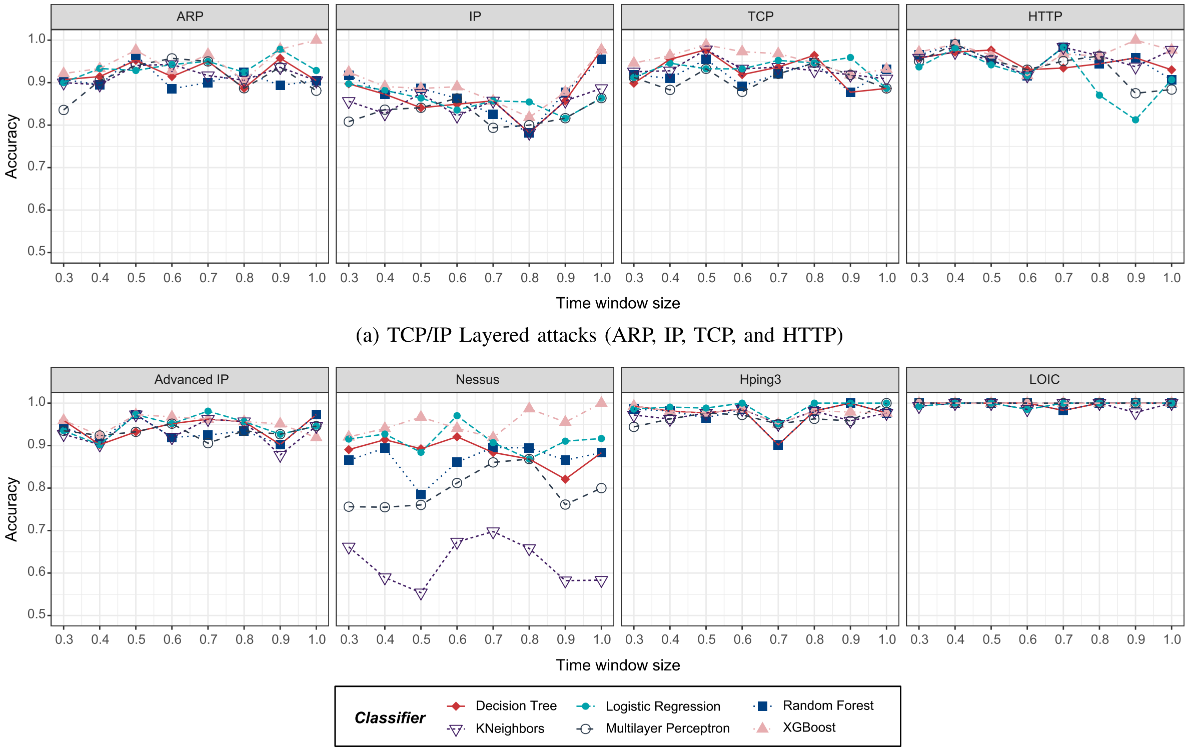
experimented with the various attack scenarios that are possible to cause anomalies of linked nodes on TCP/IP 4-layer.

V. PERFORMANCE EVALUATION

In this section, we present the results of the overall performance evaluation of the proposed method. Section V-A briefly describes several metrics to evaluate the results and summarizes the relationships between the predicted and actual classes via the confusion matrix, an indicator used to evaluate modeling performance. In Section V-B, we present the experimental results for anomaly detection and attack identification as metrics (i.e., Accuracy and F1-score).

A. Performance Metrics

We measured the accuracy for anomaly detection and attack identification by considering the following two factors: the defined time window and the attack scenario based on the TCP/IP 4-layer, which is the basic hierarchical structure of the network. This was performed through the most representative metrics in the performance evaluation (e.g., Accuracy, Precision, Recall, etc.). In the case of anomaly detection, the result was evaluated by the metrics of Accuracy, Precision, Recall, F1-score depending on the defined time window. The results of attack identification were evaluated by the following



(a) TCP/IP Layered attacks (ARP, IP, TCP, and HTTP)

(b) Scan and Dos attacks (Advanced IP, Nessus, Hping3, and LOIC)

Fig. 6. Accuracy of Intrusion Detection.

TABLE V
PRECISION, RECALL, AND F1-SCORE FOR XGBOOST CLASSIFIER (0.5 SEC TIME WINDOW)

Condition	Attack Types	**Decision Tree**			**Random Forest**			**XGBoost**			**MLP**			**KNeighbors**			**Logistic Regression**		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
NMAP	ARP	0.938	0.938	0.938	0.960	0.946	0.953	0.984	0.955	0.968	0.840	0.955	0.894	0.905	0.864	0.884	0.864	0.864	0.864
	IP	0.774	0.774	0.774	0.858	0.803	0.825	0.858	0.803	0.825	0.667	0.600	0.632	0.765	0.650	0.703	0.682	0.750	0.714
	TCP	0.958	0.985	0.971	0.953	0.924	0.938	0.993	0.977	0.985	0.808	0.955	0.875	0.955	0.955	0.955	0.833	0.909	0.870
	HTTP	0.962	0.962	0.962	0.973	0.875	0.915	0.979	0.906	0.938	1.000	0.750	0.857	1.000	0.750	0.857	1.000	0.688	0.815
ExNMAP	Advanced IP	0.742	0.871	0.789	0.986	0.800	0.868	0.986	0.800	0.868	0.500	0.800	0.615	1.000	0.600	0.750	0.800	0.800	0.800
	Nessus	0.892	0.882	0.886	0.776	0.783	0.778	0.974	0.958	0.965	0.673	0.771	0.718	0.429	0.375	0.400	0.815	0.917	0.863
	Hping3	0.972	0.972	0.972	0.964	0.953	0.958	0.972	0.972	0.972	1.000	0.923	0.960	1.000	0.923	0.960	1.000	0.962	0.980
	LOIC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

[5]* MLP: Multilayer Perceptron, * P: Precision, R: Recall, F1: F1-score

two metrics: accuracy and the area under the curve (AUC) of the receiver-operating characteristic (ROC) curve. The anomaly detection and attack identification were evaluated by considering attacks from NMAP Scanner based on the TCP/IP 4-layer, other scan and DoS attacks. The performance metrics based on the confusion matrix (shown in Table IV) were computed using Eq. (1)-(5):

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \tag{1}$$

$$Precision = TP/(TP + FP) \tag{2}$$

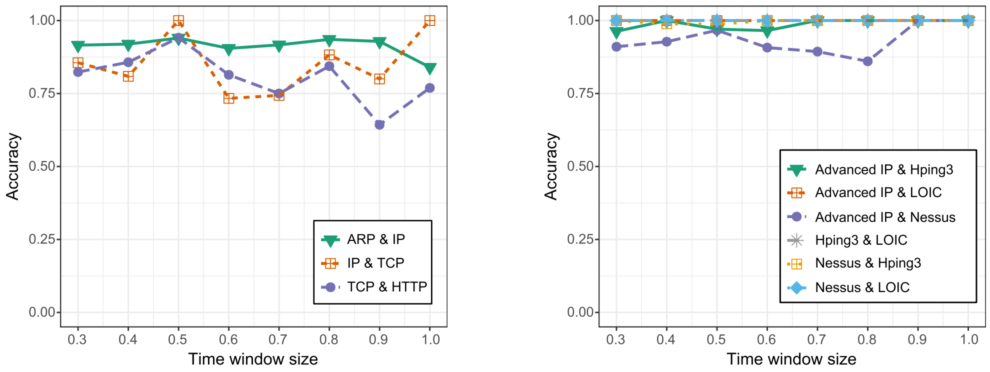
$$Recall = TP/(TP + FN) \tag{3}$$

$$F1 - score = 2(Precision \times Recall)/(Precision + Recall) \tag{4}$$

$$AUC = \int_0^1 Pr\left[TP\right](v)dv,$$

$$where\ Pr\left[TP\right]\ is\ a\ function\ of\ v = Pr\left[FP\right] \tag{5}$$

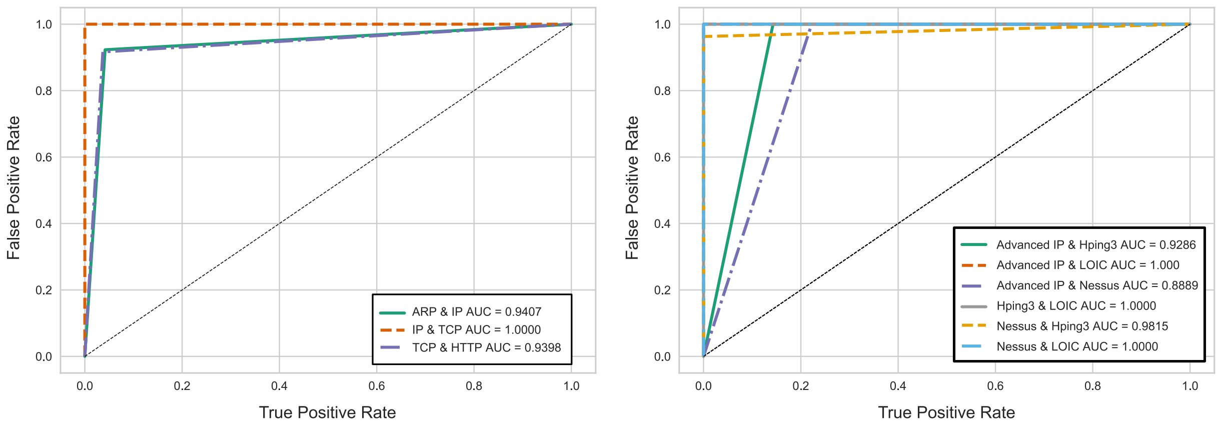
To demonstrate the applicability and scalability of the proposed scheme, we employed three well-known tree-based machine learning models for anomaly detection and attack identification. The proposed method was found to work the best on the decision tree classifier (DTC), the random forest classifier (RFC), and the XGBoost for anomaly detection. Among them, XGBoost is suitable for attack identification for multi-class classification. Accordingly, the machine learning model with a tree structure and classification usually exhibit a better anomaly detection and attack identification as compared to those without such a structure.



(a) TCP/IP Layered attacks

(b) Scan and DoS attacks

Fig. 7. Accuracy of Attack Identification for XGBoost Classifier.



(a) TCP/IP Layered attacks

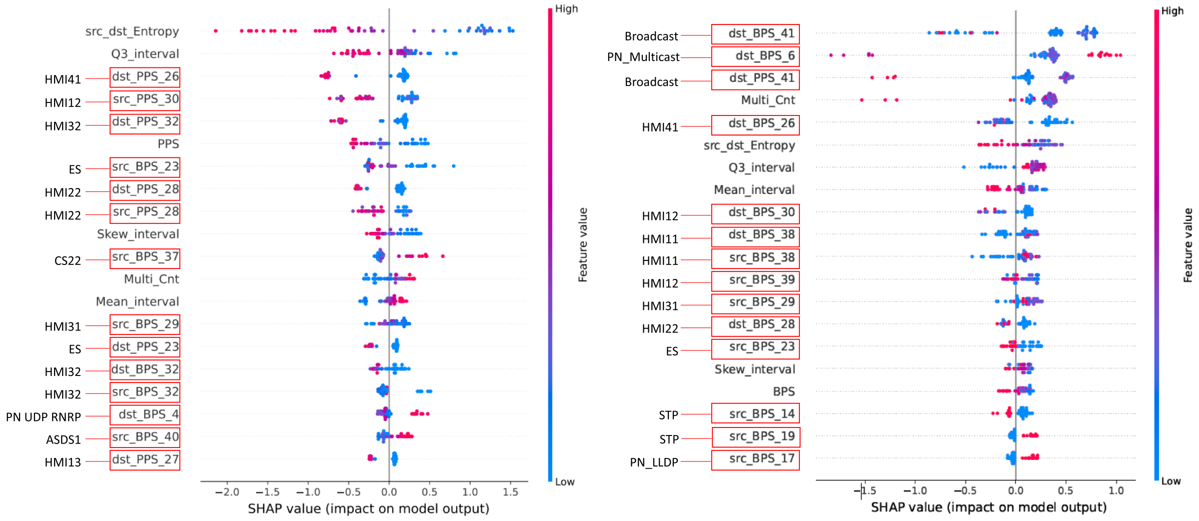
(b) Scan and DoS attacks

Fig. 8. ROC curve for XGBoost Classifier (0.5 sec time window).

B. Experimental Results

1) Intrusion Detection: In this experiment, we found that the proposed method could be a dynamic mechanism to detect abnormal states of the DCS system. First, we defined a time window to most correctly detect an abnormality. The time window was determined at from 0.3 to 1.0 sec and was increased on a 0.1 sec timescale. DCS traffic extracted by time was analyzed individually by division into one unit of the defined time window. Next, based on the three well-known tree-based machine learning models, the anomaly detection for the attacks from NMAP Scanner based on the TCP/IP 4-layer, other Scan attacks, and DoS attacks was conducted. Finally, we evaluated the detection accuracy based on the metrics of Accuracy, Precision, Recall, F1-score presented in Section V-A. As shown in Fig. 6, the experimental results showed that abnormality of DCS was detected at high accuracy for all defined attack scenarios in all other classifier except the Kneighbors and Multilayer Perceptron classifier. The detection accuracy was generally high for all defined attack scenarios, except for the scanning attacks by Nessus of ExNMAP, regardless of the defined time window. Among the three machine learning models, XGBoost classifier detected all attack types with high accuracy. Table V shows three metrics of detection accuracy for all attack types based on the 0.5 sec time window. The proposed method reached high Precision, Recall, and F1-score of over 90% in all attacks, except for scanning attacks by the Advanced IP and scanning attacks against the IP layer of NMAP. In particular, the accuracy metrics were high at the XGBoost classifier. Overall, all of the metrics were high in the scanning attacks against ARP and TCP among the TCP/IP Layered attacks of NMAP. Meanwhile, in the case of ExNMAP, all of the metrics were high in the scanning attacks against Hping3 and LOIC. These results show that our method can detect anomalies in DCS with high accuracy for diverse attack types.

2) Attack Identification: We evaluated the identification accuracy for attack types based on the accuracy and the ROC curve. The performance was conducted for the attack types of multi-class classification (two types of attack) in diverse defined attack scenarios. The defined attack scenarios were as follows. The first attack types of multi-class classification consisted of four attacks (i.e., ARP, IP, TCP, and HTTP) from the NMAP scanner. On the assumption that an attack on the adjacent layer among the TCP/IP 4-layer is possible, we evaluated whether our proposed method could identify two attack types for the ARP and IP layer, the IP and TCP layer, and the TCP and HTTP layer. The second attack types of multi-class classification consisted



(a) TCP/IP Layered attacks

(b) Scan and DoS attacks

Fig. 9. Feature Importance.

of the two types of scanning tools (i.e., Nessus and Advanced IP) and the two types of DoS tools (i.e., Hping3 and LOIC). The evaluation of the identification accuracy based on the XGBoost classifier with the highest detection accuracy among the three machine learning models was performed by dividing the training data set and the test dataset in a 9:1 ratio. Similarly to the evaluation procedure of the detection accuracy, the measurement of the identification accuracy was performed by dividing the DCS packet data into time windows of 0.1 sec.

Fig. 7 shows the identification accuracy for XGBoost Classifier evaluated through the attack types of multi-class classification (two types of attack) concerning the defined attack scenarios. The results are reported as the mean values of the attack identification accuracy for two mixed attack scenarios. In the case of the attack types of the ARP and IP layer among the attacks by the NMAP scanner tool, our proposed method identified particularly well, regardless of the defined time window. In the IP and TCP layer, the TCP and HTTP layer, our proposed method identified well at the specific time window such as 0.5 sec and 1.0 sec. In the case of the attacks by the scanning tools and the DoS tools, our proposed method identified well regardless of the defined time window, except for an attack by the Nessus and Advanced IP scanning tool. Specifically, our proposed method identified well at the 0.5 sec time window. Fig. 8 shows the results of identification performance for ROC curve for XGBoost Classifier (0.5 sec time window) by concerning the attacks from NMAP Scanner based on the TCP/IP 4-layer, other Scan attacks and DoS attacks.

Fig. 9 shows the results of the feature significance that indicate the more significant feature in the dataset used for the attack identification module. Significance of the features was measured based on the two types of attack, i.e., the Scanning and the DoS attack. In the case of the scanning attack, since it is performed by broadcasting to the DCS network, the related BPS and PPS values mainly affect the attack identification. On the other hand, in the DoS attack, since it is performed by targeting a specific host, the BPS and PPS values related to the specific host mainly affect the attack identification.

VI. Conclusion

In this study, we developed and verified a novel attack detection and identification method based on the changes of BPS value and PPS value of each host, which are components of the DCS. Since DCS uses its own protocols developed by product vendors, it is difficult for existing IT-based IDS to understand the DCS traffic. In addition, the execution of host-based IDS entails a cumbersomely additional task of changing each system installed in DCS. In contrast, the proposed method can promptly apply in the DCS operational environment without changing the configuration or settings of the DCS in operation. Moreover, our method enables the IDS to effectively detect and identify abnormality of the system. To efficiently design and evaluate our proposed algorithm, normal traffic and abnormal traffic were collected in an environment almost identical to the actual operating environment. Whether or not the traffic is normal was detected depending on the defined time window. Attack types of the traffic were identified by considering the TCP/IP 4-layer, i.e., the basic hierarchical structure of the network.

References

[1] K. Stouffer, J. Falco, and K. Scarfone, “Guide to Industrial Control Systems (ICS) Security,” *NIST Special Publication 800-82 Revision 2*, vol. 2, no. 3, pp. 1–171, May 2015, doi: 10.6028/NIST.SP.800-82r2.

[2] R. M. Lee, M. J. Assante, and T. Conway, “German steel mill cyber attack,” *Ind. Control Syst.*, vol. 30, no. 62, pp. 1–15, Dec. 2014.

[3] M. Geiger, J. Bauer, M. Masuch, and J. Franke, “An analysis of black energy 3, crashoverride, and trisis, three malware approaches targeting operational technology systems,” in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Automat.*, 2020, vol. 1, pp. 1537–1543.

[4] Q. Chen and R. A. Bridges, “Automated behavioral analysis of malware: A case study of wannacry ransomware,” in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl.*, 2017, pp. 454–460.

[5] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” *IEEE Secur. Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[6] T. Alladi, V. Chamola, and S. Zeadally, “Industrial control systems: Cyberattack trends and countermeasures,” *Comput. Commun.*, vol. 155, pp. 1–8, 2020.

[7] D. Peterson, “Intrusion detection and cyber security monitoring of SCADA and DCS networks,” ISA Automation West, May 2004. [Online] Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.3420&rep=rep1&type=pdf

[8] N. Rajasinghe, J. Samarabandu, and X. Wang, "Insecs-dcs: A highly customizable network intrusion dataset creation framework," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, 2018, pp. 1–4.

[9] S. V. B. Rakas, M. D. Stojanović, and J. D. Marković-Petrović, "A review of research work on network-based scada intrusion detection systems," *IEEE Access*, vol. 8, pp. 93083–93108, 2020.

[10] J. Zhang, S. Gan, X. Liu, and P. Zhu, "Intrusion detection in scada systems by traffic periodicity and telemetry analysis," in *Proc. IEEE Symp. Comput. Commun.*, 2016, pp. 318–325.

[11] D. Wang and D. Feng, "Intrusion detection model of scada using graphical features," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Automat. Control Conf.*, 2018, pp. 1208–1214.

[12] H. Lin, A. Slagell, Z. T. Kalbarczyk, P. W. Sauer, and R. K. Iyer, "Runtime semantic security analysis to detect and mitigate control-related attacks in power grids," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 163–178, Jan. 2018.

[13] C. Wressnegger, A. Kellner, and K. Rieck, "Zoe: Content-based anomaly detection for industrial control systems," in *Proc. IEEE 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2018, pp. 127–138.

[14] Y. Kwon, H. K. Kim, Y. H. Lim, and J. I. Lim, "A behavior-based intrusion detection technique for smart grid infrastructure," in *Proc. IEEE Eindhoven PowerTech*, 2015, pp. 1–6.

[15] Y. Kwon, S. Lee, R. King, J. I. Lim, and H. K. Kim, "Behavior analysis and anomaly detection for a digital substation on cyber-physical system," *Electron.*, vol. 8, no. 3, p. 326, Mar. 2019, doi: 10.3390/electronics8030326.

[16] R. Benisha and S. Raja Ratna, "Design of intrusion detection and prevention in SCADA system for the detection of bias injection attacks," *Secur. Commun. Netw.*, vol. 2019, p. 12, Nov. 2019, doi: 10.1155/2019/1082485.

[17] A. Derhab *et al.*, "Blockchain and random subspace learning-based ids for sdn-enabled industrial iot security," *Sensors*, vol. 19, no. 14, p. 3119, Jul. 2019, doi: 10.3390/s19143119.

[18] Y. Lai, J. Zhang, and Z. Liu, "Industrial anomaly detection and attack classification method based on convolutional neural network," *Secur. Commun. Netw.*, vol. 2019, p. 11, Sep. 2019, doi: 10.1155/2019/8124254.

[19] J. Ashcraft, D. K. Zafra, and N. Brubaker, "Monitoring ICS cyber operation tools and software exploit modules to anticipate future threats," MANDIANT Threat Research, Mar. 2020. [Online]. Available: https://www.mandiant.com/resources/monitoring-ics-cyber-operation-tools-and-software-exploit-modules

[20] O. Alexander, M. Belisle, and J. Steele, "Mitre ATT&CK® for industrial control systems: Design and philosophy," Bedford, MA, USA: The MITRE Corporation, Mar. 2020. [online] Available: https://collaborate.mitre.org/attackics/img_auth.php/3/37/ATT%26CK_for_ICS_-_Philosophy_Paper.pdf

[21] D. Du, X. Li, W. Li, R. Chen, M. Fei, and L. Wu, "Admm-based distributed state estimation of smart grid under data deception and denial of service attacks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 8, pp. 1698–1711, Aug. 2019.

[22] R. Deraison, R. Gula, and T. Hayton, "Passive vulnerability scanning: Introduction to nevo," *Revision*, vol. 9, no. 7, pp. 1–13, 2003.

[23] L. Liang, K. Zheng, Q. Sheng, and X. Huang, "A denial of service attack method for an iot system," in *Proc. IEEE 8th Int. Conf. Inf. Technol. Med. Educ.*, 2016, pp. 360–364.

[24] S. Bravo and D. Mauricio, "Ddos attack detection mechanism in the application layer using user features," in *Proc. IEEE Int. Conf. Inf. Comput. Technol.*, 2018, pp. 97–100.

[25] G. F. Lyon, "Nmap network scanning: The official nmap project guide to network discovery and security scanning," *Insecure. Com LLC (US)*, pp. 1–456, 2008.

[26] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4362–4369, Jul. 2019.

[27] M. L. Han, B. I. Kwak, and H. K. Kim, "Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2941–2956, Mar. 2021, doi: 10.1109/TIFS.2021.3069171.

Kyoung Ho Kim received the B.S. degree in computer science from Hansung University, Seoul, South Korea, in 2002, and the M.S. degree in information and technology from Sogang University, Seoul, South Korea, in 2008, and the Ph.D. degree in information security from the School of Cybersecurity, Korea University, Seoul, South Korea, in 2021. He is currently a Cybersecurity Architect for more than 15 years with S-OIL Corporations and has specialties in IT and OT cybersecurity. Before joining the S-OIL corp., he was the Cybersecurity Specialist with KT Hitel and Cybersecurity Consultant in A3 Security Consulting, which is the first information security consulting company in South Korea. His research interests include security modeling, CPS and IoT cybersecurity, and intrusion and anomaly detection for industrial control system.

Byung Il Kwak received the B.S. degree in computer and science from Sejong University, Seoul, South Korea, in 2012, and the Ph.D. degree in information security from the School of Cybersecurity, Korea University, Seoul, South Korea, in 2021. In 2021, he was a Research Professor with the School of Cybersecurity, Korea University. He is currently an Assistant Professor with the School of Software, Hallym University, Chuncheon, South Korea. His research interests include vehicle security, network security, machine learning, and deep learning.

Mee Lan Han received the B.S. degree in computer and science from Dongduk Women's University, Seoul, South Korea, in 2002, the M.S. and Ph.D. degrees in information security from the School of Cybersecurity, Korea University, Seoul, South Korea, in 2014 and 2020, respectively. From 2004 to 2012, she was a Chief Researcher of the development part against Chinese-speaking countries with Nexon, and a Research Professor with the School of Cybersecurity, Korea University from 2020 to 2021. She is currently a Professor, Industry-University Cooperation with the Department of AI Cyber Security, Korea University Sejong Campus, Yeongi-gun, South Korea. Her research interests include vehicle security, network security, cyber threat intelligence, and data mining.

Huy Kang Kim (Member, IEEE) received the B.S. degree in industrial management, the M.S. degree in industrial engineering, and the Ph.D. degree in industrial and systems engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2000, 1998, and 2009, respectively. From 2004 to 2010, he was a Technical Director and the Head of Information Security Department with NCSOFT, one of the most famous MMORPG companies in the world. He is currently a Professor with the School of Cybersecurity, Korea University, Seoul, South Korea. His research focuses on solving many security problems in online games based on the user behavior analysis. In 1999, he Founded A3 Security Consulting, the first information security consulting company in South Korea. He was also a Member and the Last Leader of KAIST UNIX Society, the legendary hacking group in South Korea.