IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 September 2024, accepted 8 October 2024, date of publication 11 October 2024, date of current version 29 October 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3478830*

RESEARCH ARTICLE

# Detecting Cybersecurity Threats for Industrial Control Systems Using Machine Learning

**WOOHYUN CHOI[1], SUMAN PANDEY[2], AND JONGWON KIM[1], (Senior Member, IEEE)**
[1]AI Graduate School, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea
[2]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Jongwon Kim (jongwon@gist.ac.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government [Ministry of Science and ICT (MSIT)], (No.2019-0-01842, Artificial Intelligence Graduate School Program, Gwangju Institute of Science and Technology (GIST)).

**ABSTRACT** Industrial control systems (ICS) are vital for ensuring the reliability and operational efficiency of critical infrastructure across various industries. However, due to their integration into modernized network environments, they are inadvertently exposed to a variety of cybersecurity threats that can compromise the reliability of critical infrastructure. This study aims to enhance ICS security by introducing a Zero Inflated Poisson (ZIP) based GRU Learning model to detect anomalies of ICS traffic in conjunction with the MITRE ATT&CK framework. The model's effectiveness was validated through experiments simulating two major cyberattack scenarios: the 'Stuxnet' attack and the 'Industroyer' attack, achieving over 95% success in attack detection. By mapping the anomalies to the MITRE ATT&CK framework, we were able to lay the groundwork for an efficient response strategy to the attacks. These findings are expected to make a meaningful contribution to assessing and strengthening the security posture of ICS.

**INDEX TERMS** Cybersecurity, industrial control system, MITRE ATT&CK.

## I. INTRODUCTION

Industrial control systems (ICS) are the backbone of modern industry, responsible for the smooth operation of a wide range of critical infrastructure, including energy, water, and traffic management [1], [2]. By precisely controlling complex machines and processes, these systems play a vital role in ensuring the reliable operation of our infrastructure and supporting our nation's economic prosperity and public safety [3], [4], [5]. The reliability and security of these critical systems is more than a technical issue; it is a serious national security concern, which means that ICS goes beyond mere industrial function and is linked to societal and national resilience [6], [7], [8], [9]. In this research, we propose a machine learning-based threat detection model to enhance the security of these crucial infrastructures.

To understand the ICS infrastructure and underlying security concerns, we can refer to the Purdue reference model [25]. The model was developed at Purdue University in the early 1990s and was designed for the integration of IT(Information Technology) and OT(Operational Technology) in industrial environments. Fig 1. shows the hierarchical structure of the Purdue reference model. Typically ICS systems are divided into five levels. Among these levels, the lowest level is composed of field I/O devices such as sensors, actuators, robots, etc. This is where the actual manufacturing process takes place. At level 1 the PLC (Programmable Logic Controller) and RTU (Remote Terminal Unit) are located. Level 2 is the control system layer, where the EWS (Engineering Work Station) and HMI (Human-Machine Interface) are located. Level 3 includes the application server, which integrates the monitoring and control of physical processes with the historian server. Level 4 is the enterprise zone, including process monitoring, and general IT system applications servers. This segmentation and isolation of the network into smaller, more manageable sections help in reducing the attack surface. It also limits the exposure of critical assets to external threats. For this study, the Purdue model provides a structured framework for simulating ICS environments and analyzing security measures.

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

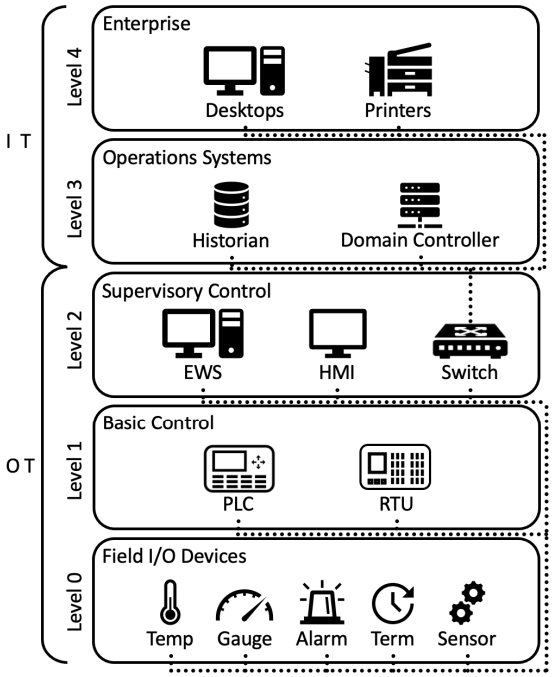


**FIGURE 1.** Structure of purdue reference model.

The technical details of the Purdue Model, such as specific roles of each level and the infrastructure used for threat simulation and model testing, are further discussed in the subsequent sections.

Purdue Model isolates and segments the network environment, which was viewed as a natural defense against external cyber threats [10], [11]. However, recent events have forced us to re-examine the notion that closed systems are secure [12]. The Stuxnet incident demonstrated that even closed networks can be breached by targeted cyber attacks [13], [14], [15], [16], [17], [18]. The cyber attack on the Ukrainian power grid showed that these threats are not abstract concepts, but can happen in the real world [19], [20], [21], [22], [23]. This means that security within closed networks cannot simply rely on physical isolation, and new approaches are needed to protect complex networks and systems.

The development of fast and accurate detection and response mechanisms to these threats is essential to ensure the reliability of ICS. There are relatively few studies that address the combined threat of both internal and external intrusions that can occur in a closed network [24]. The goal of this research is to present a method to accurately detect possible cyber threats within closed networks through a combination of real-time network packet analysis and advanced threat modeling frameworks. To accomplish this, we closely analyze network traffic in an ICS environment and integrate it with the MITRE ATT&CK framework. MITRE ATT&CK framework is a de-facto standard that covers a variety of cyber threats in ICS and systematically identifies and classifies attack techniques and attacker behaviors [26].

This research facilitates rapid and accurate threat detection by mapping network anomalies to the MITRE ATT&CK framework. Based on data collected in an experimental environment simulating a real-world ICS traffic, we developed a threat detection algorithm using the Gated Recurrent Unit (GRU) model. We used the Zero-Inflation Poisson(ZIP) distribution to set more sophisticated detection thresholds. In many ICS environments, data sets can be sparse because certain types of cyber threats occur infrequently. Zero-inflated Poisson is particularly adept at handling this sparsity, improving the model's ability to detect anomalies in data where non-events are common. This is especially important in ICS, where computational resources are limited and system efficiency is critical. We also evaluated the performance of the model with cyber-attack scenarios simulating the Stuxnet and Industroyer attacks. With this approach, we have demonstrated that the proposed algorithms and techniques can detect and discriminate cyber threats in ICS environments with high accuracy. The contributions of this paper are as follows.

- We have provided mapping of network flow data with attack tactics and techniques of MITRE ATT&CK framework. This is an important contribution for interpreting ICS-specific threats that are not addressed by traditional network security solutions.
- Our work applies a filtering technique based on the ZIP model to accurately capture critical cyber threat signals that occur infrequently in ICS network traffic. This method accounts for the sparsity and temporal characteristics of the ICS systems and contributes to improved security.
- Our research evaluates the effectiveness of new security techniques through empirical testing against historically significant cyber attacks on ICS targets. These tests demonstrate the practicality of the research in countering complex real-world attacks.

This paper consists of four parts. The Related Work describes previous research that is relevant to this paper, specifically the structure of ICSs and threat detection techniques in ICSs. The third section, Proposed Method describes the methodology and experimental design of this paper. The Implementation section describes the implementation modules. The Experiments and Results section details the results of our experiments. Finally, the Conclusions section summarizes the main conclusions of the study and provides directions for future research.

## II. RELATED WORK

In this section, we will lay the foundation for our research by explaining the de facto industry standards and ongoing research in the field of ICS security. This section specifically focuses on previously published studies that are directly relevant to ICS cybersecurity.

### A. MITRE ATT&CK FRAMEWORK FOR ICS

The MITRE ATT&CK was developed in 2013 by researchers at MITRE Corporation with the goal of organizing

publicly available cyber threat information. The framework is designed to systematically categorize attackers tactics, techniques, and procedures to help security researchers and professionals understand threats and formulate countermeasures [26]. Attack techniques in the framework include network intrusion, system sabotage, and data exfiltration, with detailed descriptions of each technique, how to identify them, and how to respond to them. The relevance of the MITRE ATT&CK framework to ICS cybersecurity is that it provides structured methodologies to identify and mitigate specific threats unique to ICS environments, which are critical for enhancing security measures in industrial systems.

For example, using the MITRE ATT&CK framework, past attacks on ICS, such as the Stuxnet and Industroyer attacks, can be analyzed to understand the methods and strategies used, enabling organizations to develop targeted defense mechanisms. Thus, MITRE ATT&CK is becoming an essential tool for enhancing the security of industrial systems, and one of the main objectives of this paper is to help organizations leverage this framework to proactively manage and respond to cyber threats.

Table 1 shows the MITRE ATT&CK framework with tactics, techniques, and corresponding data sources used in this paper. Each column and row in the table provides the following information.

- Tactic: This column represents strategies used in cyberattacks. Strategies include initial access, execution, persistence, detection, and command and control.
- Seq: This column indicates the sequence of the corresponding strategy or technique. Each combination of strategy and technique is assigned a unique sequence number.
- Technique ID: Each technique is assigned a unique identification number.
- Technique Name: Each strategy in cyberattack corresponds to multiple techniques. This column provides the name and description of the technique used to formulate an attack Tactic.
- Data Source: This column describes the data source used to detect the attack technique. Data sources are of various types, including network traffic, application logs, files, processes, logon sessions, and operational databases. This column indicates which data sources are relevant for each attack technique.

According to Table 1. Network Traffic serves as a key data source for almost all attack techniques. Packet-level network traffic data can provide detailed information essential for anomaly detection algorithms. Furthermore, in the 9. Command and Control segment, we found that network traffic plays a key role in threat detection: certain techniques such as Common Used Port, Connect Proxy, and Standard Application Layer Protocol can detect threat activity only through network traffic, which reaffirms the importance of network traffic data. Therefore, this paper underlines the need to focus on developing methods to effectively collect and analyze network traffic data. It is expected that such methods will help maximize the ability to detect and respond to cybersecurity attacks.

### B. ICS SECURITY AND PROTECTION MECHANISMS

Apart from the traditional security mechanisms outlined in the Purdue Reference Model, several advanced techniques such as honeypots, Intrusion Detection Systems (IDS), and Advanced AI techniques, such as Machine Learning (ML), can also ensure ICS security. Here is a detailed breakdown of how these mechanisms can be integrated into ICS security.

#### 1) HONEYPOT

Honeypots are used as a sophisticated cybersecurity defense in ICS [27]. Honeypots intentionally provide a point of vulnerability to attract malicious activity and entice attackers to interact with them, which can then be used to study their behavior, tactics, and techniques. They play an important role in evaluating the detection capabilities of security systems and identifying new types of attacks. Deploying honeypots in ICS environments allows for the analysis of attacker behavior, the identification of vulnerabilities, and a better understanding of attack patterns [28]. Honeypot implementations in ICS environments are mainly divided into low-interaction and high-interaction honeypots. Low-interaction honeypots are relatively easy to manage because the system provides limited services. High-interaction honeypots can collect more information from attackers by providing an environment that resembles a real system [29]. Though honeypots can mimic ICS components to lure attackers and detect, deflect, or study hacking attempts, they are complex to manage and maintain, and if implemented incorrectly, they can act as a vulnerability for real systems [30].

#### 2) INTRUSION DETECTION SYSTEM

An intrusion detection system (IDS) is a security solution that monitors and detects malicious activity and policy violations in real-time. It automatically takes countermeasures when suspicious activity is detected. In ICS, IDS can be tailored to the unique protocols and behaviors of industrial networks [31], [32]. Researchers are focusing on enhancing the capabilities of IDS to quickly detect anomalies and take appropriate countermeasures for industrial networks. Hybrid IDS models that combine anomaly detection with signature-based detection are also being studied [33], [34], [35]. However, the ICS environment requires real-time processing and minimal latency. A complex IDS solution may introduce delays that can disrupt the real-time operations of ICS. Moreover, the ICS landscape continues to change, and providing tailored IDS solutions for ICS requires continuous exploration and integration of new threats into the system [36]. Our proposed ML-based NIDS addresses the complexity issues of traditional IDS while enabling real-time operation through: (1) a lightweight GRU model structure, reducing computational complexity; (2) efficient data preprocessing using ZIP models to handle data sparsity; and (3) near

**TABLE 1.** Tactics, techniques and corresponding data sources representing the MITRE ATT&CK Matrix.

| Tactic | Seq | Technique ID | Technique Name | Data Source | | | | |
|---|---|---|---|---|---|---|---|---|
| 1. Initial Access | 1-1 | T0817 | Drive-by Compromise | Network Traffic | Application Log | File | Process | |
| | 1-2 | T0819 | Exploit Public-Facing Applications | Network Traffic | Application Log | | | |
| | 1-3 (7-2) | T0866 a) | Exploitation of Remote Services | Network Traffic | Application Log | | | |
| | 1-4 | T0822 | External Remote Sevices | Network Traffic | Application Log | Logon Session | | |
| | 1-5 | T0883 | Internet Accessible Device | Network Traffic | Logon Session | | | |
| | 1-6 (7-6) | T0886 b) | Remote Services | Network Traffic | Command | Logon Session | Network Share | Process |
| | 1-8 | T0848 | Rogue Master | Network Traffic | Application Log | Operational Databases | | |
| | 1-11 | T0864 | Transient Cyber Asset | Network Traffic | Application Log | | | |
| 2. Execution | 2-4 | T0823 | Graphical User Interface | Network Traffic | Process | | | |
| 3. Persistence | 3-2 | T0889 | Modify Program | Network Traffic | Application Log | Asset | Operational Databases | |
| 6. Discovery | 6-3 | T0846 | Remote System Discovery | Network Traffic | Command | File | Process | |
| 7. Lateral Movement | 7-2 (1-3) | T0866 a) | Exploitation of Remote Services | Network Traffic | Application Log | | | |
| | 7-4 | T0867 | Lateral Tool Transfer | Network Traffic | Command | File | Process | |
| | 7-6 (1-6) | T0886 b) | Remote Services | Network Traffic | Command | Logon Session | Network Share | Process |
| 8. Collection | 8-1 | T0802 | Automated Collection | Network Traffic | Command | File | Script | |
| | 8-5 | T0830 | Man in the Middle | Network Traffic | Command | Process | | |
| | 8-9 | T0845 | Program Upload | Network Traffic | Application Log | | | |
| 9. Command and Control | 9-1 | T0885 | Commonly Used Port | Network Traffic | | | | |
| | 9-2 | T0884 | Connection Proxy | Network Traffic | | | | |
| | 9-3 | T0869 | Standard Application Layer Protocol | Network Traffic | | | | |
| 10. Inhibit Response Function | 10-3 | T0803 | Block Command Message | Network Traffic | Application Log | Operational Databases | Process | |
| | 10-4 | T0804 | Block Reporting Message | Network Traffic | Application Log | Operational Databases | Process | |
| | 10-8 | T0814 | Denial of Service | Network Traffic | Application Log | Operational Databases | | |
| 11. Impair Process Control | 11-1 | T0806 | Brute Force I/O | Network Traffic | Application Log | Operational Databases | | |

real-time analysis through 10-minute data aggregation. This approach allows for complex network traffic analysis without compromising ICS operational efficiency.

### 3) MACHINE LEARNING IN ICS SECURITY

Machine learning is currently positioned as a key technology to significantly improve the security and efficiency of ICS [37], [38]. Recent research trends show that it plays a crucial role in solving complex problems within ICS, with significant advances in anomaly detection, predictive maintenance, and network intrusion detection [39], [40], [41]. Anomaly detection uses machine learning algorithms to identify activity that deviates from normal system operating patterns. In particular, these algorithms use unsupervised and semi-supervised learning techniques to extract meaningful insights from unlabeled data. Deep learning techniques, particularly neural networks, are highly effective at processing complex data structures and predicting changes over time [42], [43]. In predictive maintenance, machine learning analyzes historical operational data and fault records to predict potential equipment failures. This helps reduce maintenance costs and minimize downtime. Applying AI techniques, specifically ML, to network intrusion detection systems enables the identification of security threats in real-time. In this research, we propose an extensive methodology to filter network traffic data and train ML models for detecting cybersecurity threats, contributing to contemporary research efforts.

We see that the Purdue reference model provides us with standards for organising and managing ICS, followed by the MITRE ATT&CK framework, which provides us with a comprehensive, detailed matrix of tactics, techniques, and procedures used by attackers to compromise and manipulate ICS environments. We adopt an ML, specifically deep learning-based approach. ML was chosen for its ability to adapt to evolving threats and efficiently process large-scale data, making it more effective than traditional honeypots or IDSs [44]. Particularly, the GRU model excels at capturing temporal dependencies and learning complex features, making it suitable for detecting sophisticated cyber threats in ICS. In this research, we closely mapped the network traffic matrix with the MITRE ATT&CK tactics and techniques and leveraged advances in machine learning to develop an effective defence mechanism to combat the most common threats. This is achieved through efficient data processing and decision-making algorithms. Our approach uses a streamlined GRU model, which is optimized for sequential data analysis, and ZIP preprocessing to reduce data dimensionality. These techniques allow for rapid processing of network traffic patterns without the need for complex rule matching, enabling real-time threat detection suitable for ICS environments.

## III. PROPOSED METHOD

We opted for a machine learning approach over traditional honeypots or IDS due to its superior adaptability to evolving

cyber threats, pattern recognition capabilities in complex network traffic, and scalability for large ICS networks. Specifically, our GRU-based model was chosen for its efficacy in processing temporal patterns in network traffic, memory efficiency, and interpretability - crucial factors in ICS environments. This research collects network traffic data with anomalies and processes it to align with the MITRE ATT&CK framework. The resulting time series data, which contains an abundance of zero events, is processed using the ZIP model. Additionally, GRU is employed to capture the sequential dependencies of these input features. Ultimately, the model accurately detects the occurrence of specific attack events.

### A. RAW DATA SET

A typical ICS system includes a Packet Collector server, which utilizes a port mirroring technique to collect and filter raw traffic data with anomalies. This raw traffic data has several features including anomaly_seq, anomaly_type, plant_id, area_no, manufacturer_id, protocol_type, protocol_detail, src_ip, src_mac, src_port, dst_ip, dst_mac, dst_port, payload, packet_code, policy_name, packet_time, event_time, and loged_time. But, not all of them are necessary for our experiment.

**TABLE 2. Relevant Field for proposed GRU based detection model.**

| field name | description |
|---|---|
| plant_id | Power Plant ID |
| area_No | Internal area number within the Plants |
| manufacturer_Id | Manufacturer ID |
| anomaly_type | Type of anomaly (1 to 13) |
| protocol_type | Type of protocol |
| src_ip | Source IP Address |
| packet_time | Packet Transmission Time |
| port_number | Source_port |
| anomaly_count | Number of anomaly_type |

Table 2. shows the selected fields relevant to our experiment. plant_id, area_no, and manufacture_id in Table 2. are used to identify the plant, its internal area code, and the manufacturer of the equipment. anomaly_type distinguishes the type of anomaly that occurred. Packet collectors can identify 13 different types of anomaly altogether. The details of the anomaly_type are provided in Table 3. protocol_type provides specific information about the type of protocol used in the network communication process. We utilized this information to identify blocklisted protocols shown in Table 4. Moreover, packets are grouped based on the same anomaly_type, plant_id, area_no, manufacture_id, src_ip for every 10 minutes introducing an additional field called anomaly_count. anomaly_count. allows us to quantitatively analyze the frequency of anomalous behavior on the network. This adds extra depth to the observation of anomalies over time.

Table 3. provides a detailed description of the anomaly_type. Packet Collector can identify these anomalies in real-time. The anomaly_type 1 (New Asset Detection) indicates changes in network setup or access to new equipment. Anomaly_type 3 (Unauthorized Communication Detection) and 4 (Banned Communication Detection) help detect, communications that break security policies. Anomaly_type 6 (Data Anomaly Detection) finds unusual patterns that differ from the normal packet data flow. Anomaly_type 10 (Traffic Threshold Exceeded), 12 (Asset Under-Traffic), and 13 (Asset Over-Traffic) are metrics used to identify wasted network resources, overloads, or unusually low traffic by monitoring the volume of network traffic.

**TABLE 3. Anomaly_type and their details.**

| anomaly_type | discription |
|---|---|
| 1 | New Asset Detection |
| 2 | New Communication between Assets |
| 3 | Unauthorized Communication Detection |
| 4 | Banned Communication Detection |
| 5 | Unauthorized IP Detection |
| 6 | Data Anomaly Detection |
| 7 | Detect New Packet Codes |
| 8 | Detect New Packet Code Usage |
| 9 | New Protocol Packet Code Detection |
| 10 | Detect New Protocol Packet Code Usage |
| 11 | Traffic Threshold Exceeded |
| 12 | Asset Under-Traffic |
| 13 | Asset Over-Traffic |

**TABLE 4. Blocklist protocol for anomaly detection.**

| protocol | port_Number | description |
|---|---|---|
| FTP | 20,21 | File transfer |
| FTPS | 989, 990 | Secured file transfer |
| HTTP | 80 | Web communication |
| HTTPS | 443 | Secured web communication |
| RDP | 3389 | Remote GUI access |
| SMB | 445, 137-139 | File sharing, network access |
| SNMP | 161, 162 | Network monitoring |
| SSH | 22 | Secure remote access |
| TELNET | 23 | Remote access |
| VNC | 5900-5903 | Remote desktop (UNIX) |
| WELL KNOWN | 0-1023 | Reserved ports |

Table 4. organizes a list of protocols and ports that an administrator needs to be aware of in a closed network environment. We list them as a blocklist protocol in this study. Most of the ports listed in this table are for services that require communication with external networks or can be a vulnerable point for security. FTP and FTPS are commonly used for file transfer, but should not be allowed in closed networks because they are structured to allow access from outside. Similarly, HTTP and HTTPS are intended for data exchange between web servers and clients, which are also considered dangerous in closed networks due to the possibility of external communication. In addition, RDP, SMB, SNMP, SSH, and TELNET are also possible security concerns. In particular, RDP is a protocol that provides remote desktop services, which can be a serious security breach if access from outside is open. In addition, port 0-1023 is a Well-known port used by well-known services, and immediate action is required if communication using a port within this range is detected. Therefore, the raw data provided by the packet collector gives us the count

of anomalies and occurrence of blocklisted protocols every 10 minutes.

### B. ALIGNING DATA WITH MITRE ATT&CK FRAMEWORK

Our research uses the MITRE ATT&CK framework to protect ICS from security threats, as shown in Table 1. We have associated each of the 24 technique_ID from Table 1 with a set of anomaly_type from Table 3 and blocklist protocol from Table 4. These associations form the basis of 24 input data points (13 anomaly_type and 11 blocklist protocol) that are fed into our model. Our model analyzes network traffic to identify these anomalies and prohibited protocols. The packets identified in this way are then mapped to the MITRE ATT&CK framework to pinpoint specific threats. To achieve this, our team carefully linked each technique_id from MITRE ATT&CK with a Mapped_Rule. As shown in Table 5, each Mapped_Rule consists of a set of anomaly_type, anomaly_count, and protocol. For example, let's assume the following situation is observed during a 10-minute packet collection period:

- anomaly_count for anomaly_type 1 (New Asset Detection) is 3
- anomaly_count for anomaly_type 4 (Banned Communication Detection) is 5
- anomaly_count for anomaly_type 5 (Unauthorized IP Detection) is 3
- Detection of blocklisted protocols such as SSH, RDP, and TELNET

In this case, we map these observations to attack technique_id T0822 (External Remote Services). We have constructed a total of 22 such Mapped_Rules to cover various attack techniques, excluding 2 duplicates.

**TABLE 5. MITRE ATT&CK-based rule matrix environments.**

| Mapped_Rule | anomaly_type : anomaly_count | protocol |
|---|---|---|
| T0817 | "1":1, "2":2, "5":1 | - |
| T0819 | "4":1 | WELL_KNOWN |
| T0866 | "3":4 | - |
| T0822 | "1":3, "4":5, "5":3 | SSH, RDP, TELNET |
| T0883 | "4":2 | HTTP, HTTPS |
| T0886 | "4":1 | SSH, RDP, TELNET |
| T0848 | "3":5 | - |
| T0864 | "1":1, "3":5, "5":2 | - |
| T0823 | "4":1 | RDP, VNC |
| T0889 | "3":4 | - |
| T0846 | "2":5 | - |
| T0867 | "4":1 | FTP, FTPS, SMB |
| T0802 | "4":1 | SNMP |
| T0830 | "1":4, "5":4 | - |
| T0845 | "4":1 | FTP, FTPS, SMB |
| T0885 | "3":5, "4":2 | WELL_KNOWN |
| T0884 | "3":6 | - |
| T0869 | "4":5 | WELL_KNOWN |
| T0803 | "7":1, "9":1 | - |
| T0804 | "7":2, "9":2 | - |
| T0814 | "13":100 | - |
| T0806 | "13":30 | - |

This mapping results in our time series input data in the form of a Rule_Matrix as shown in Table 6. For each 10-minute interval, our model is supplied with a Rule_Matrix that includes the accumulated packet anomalies and blocklist protocol aligned with MITRE ATT&CK framework. Data collection occurred from January 30, 2023, to February 15, 2023. The input dataset comprises 880 of these Rule_Matrices. The model is now trained with this time series input matrix to detect the occurrence of an attack with high accuracy.

**TABLE 6. Part of time series training data in the form of Rule_Matrix after ZIP.**

Time_Stamp: 00:00:00

| Mapped_Rule | anomaly_type | | | ... | protocol | | Labels |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | | HTTP | HTTPS | |
| T0817 | 0.43 | 0.73 | -1.0 | ... | -1.0 | -1.0 | 1 |
| T0819 | -1.0 | -1.0 | -1.0 | ... | -1.0 | -1.0 | 2 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| T0806 | -1.0 | -1.0 | -1.0 | ... | -1.0 | -1.0 | 22 |

Time_Stamp: 00:10:00

| Mapped_Rule | anomaly_type | | | ... | protocol | | Labels |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | | HTTP | HTTPS | |
| T0817 | 0.43 | 0.73 | -1.0 | ... | -1.0 | -1.0 | 1 |
| T0819 | -1.0 | -1.0 | -1.0 | ... | -1.0 | -1.0 | 2 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| T0806 | -1.0 | -1.0 | -1.0 | ... | -1.0 | -1.0 | 22 |

### C. ZERO-INFLATED POISSON

We observe that anomaly_count and protocol data in Rule_Matrix have excessive 0. This is an underlying nature of the attack data set of ICS. The occurrence of anomalous behavior is intermittent rather than continuous resulting in an excessive 0 in the dataset. To handle the problem of excessive 0 we adopted the ZIP model. The ZIP model is a variant of the generalized Poisson distribution model designed to address the problem of excess zeros.

$\lambda$ represents the mean Poisson rate of occurrence of events. Higher lambda values mean more events are happening. $p$ represents the probability of seeing zero events in the data. These two metrics (lambda and zero ratio) are used to calculate the probability of an event occurring on the network using a ZIP distribution.

$$P(X = x) = \begin{cases} p + (1-p)e^{-\lambda} & \text{if } x = 0, \\ (1-p)e^{-\lambda}\dfrac{\lambda^x}{x!} & \text{if } x > 0. \end{cases} \quad (1)$$

Where $X$ represents the number of times an event happens. This distribution model provides a better representation of the attack data with a high number of zero events. If the calculated probability of an event is 0 or less than 0, we mark it as a nonoccurrence of the event, and the value is changed to 0. Our input data to the GRU model is now shaped by the ZIP distribution.

### D. GRU BASED DETECTION MODEL

We initially chose the Gated Recurrent Unit (GRU) model for its ability to efficiently handle long-term dependencies in sequential data, which is crucial for analyzing network traffic patterns in ICS environments. Unlike Long Short-Term Memory (LSTM) networks, GRUs have a simpler structure

with fewer parameters, making them computationally less intensive while maintaining comparable performance in capturing temporal dependencies. The GRU-based detection model starts with an 'Input' layer, followed by a 'Sequential' layer. The sequence layer expands input data dimensions. The 'GRU' layer, is a type of RNN, that learns time-related characteristics and manages long-term dependencies. 'Dense' layers expand data dimensions for recognizing complex patterns, and 'ReLU' activation functions add nonlinearity to improve detection accuracy. The 'Output' layer uses a sigmoid activation function to convert detections into probabilities for classification tasks. The 'Early stopping' callback prevents overfitting, and the 'Categorical Crossentropy' loss function with 'Adam (0.005)' optimization ensures effective learning. We employed the Adam optimization algorithm with a learning rate of 0.005 to ensure effective training of the GRU model. To prevent overfitting, early stopping was implemented during training with a patience of 10 epochs. We set the number of neurons in the initial Dense layer to twice the input feature size, followed by a GRU layer comprising 64 units, which effectively captured sequential dependencies in the data. Training was conducted with a batch size of 16, over a maximum of 500 epochs, ensuring robust model performance through early stopping. Fig. 2 shows the model structure in detail.

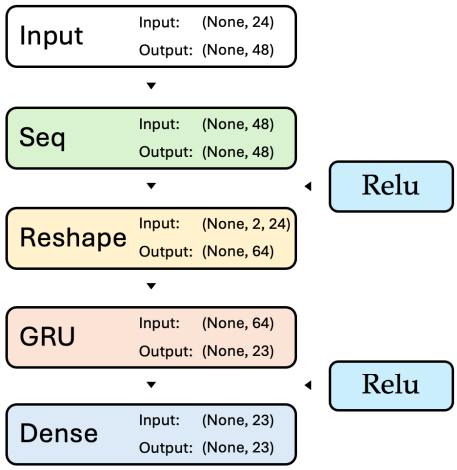


**FIGURE 2.** Proposed GRU based model architecture.

The model's architecture is designed to process a comprehensive set of 24 input features, comprising 13 anomaly_type features and 11 protocol features. This diverse input allows the model to capture a wide range of network behaviors and potential threats. The output layer consists of 23 nodes, each representing a distinct classification. Of these, 22 nodes correspond to specific threat or attack scenarios, carefully curated to eliminate any redundancies from the input features. The 23rd node serves a crucial function by representing a ''non-threat'' or normal state, allowing the model to identify benign network activity. This thoughtful design enables the model to effectively distinguish between 22 specific attack types and normal network behavior. By incorporating a dedicated node for non-threatening states, the model can accurately classify incoming data as either a known threat or as part of regular network operations. This approach enhances the model's ability to detect a broad spectrum of threats while minimizing false positives in normal network conditions.

Our study used a confusion matrix to check how well the detection model works. The confusion matrix sorts the detections into four types: False Positive, False Negative, Missing, and True Positive. This helps us see how accurate the model is and where it makes mistakes.

- True Positive (Tp) : The number of correctly identified anomalies.
- True Negative (Tn): The number of correctly identified normal instances.
- False Positive (Fp): The number of normal instances incorrectly identified as anomalies.
- False Negative (Fn): The number of anomalies incorrectly identified as normal instances.

We use several key metrics to fully evaluate our model, including Accuracy, Precision, Recall, and F1Score. Each metric gives us unique information about how well the model can detect anomalies.
Accuracy measures the proportion of correct detections and provides an overall measure of the model's performance:

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{2}$$

Precision (positive predictive value) evaluates the proportion of true positives among the positive detections and highlights the model's ability to avoid false positives:

$$Precision = \frac{Tp}{Tp + Fp} \tag{3}$$

Recall (sensitivity) evaluates the proportion of correctly identified positive detections and highlights the model's ability to detect all relevant instances:

$$Recall = \frac{Tp}{Tp + Fn} \tag{4}$$

F1Score is the harmonic mean of Precision and Recall, which balances the two and is critical for scenarios where both metrics are important:

$$F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5}$$

These metrics are critical to understanding the effectiveness of the model in detecting anomalies in real-time ICS and ensuring robustness and reliability in operational environments.

### E. MODEL PERFORMANCE COMPARISON

Four performance metrics (accuracy, precision, recall, and F1 score) are used to evaluate the performance of each model on a scale of 0 to 1. The proposed model scored very high, with a

score of 0.99 across all metrics. It outperformed other models, including LSTM, Naïve Bayes, LightGBM, SimpleRNN, SVM and Logistic regression models as presented in Table 7.

**TABLE 7. Performance comparison of machine learning models.**

| Methods | Accuracy | Precision | Recall | F1Score |
|---|---|---|---|---|
| **GRU Model** | **0.99** | **0.99** | **0.99** | **0.99** |
| LSTM | 0.98 | 0.98 | 0.98 | 0.97 |
| Naive Bayes | 0.97 | 0.97 | 0.97 | 0.97 |
| LightGBM | 0.94 | 0.93 | 0.94 | 0.93 |
| SimpleRNN | 0.91 | 0.91 | 0.91 | 0.90 |
| SVM | 0.91 | 0.84 | 0.91 | 0.88 |
| Logistic Regression | 0.84 | 0.84 | 0.84 | 0.83 |

Accuracy shows the percentage of correct predictions, precision shows the percentage of true positive predictions, recall measures the true positive rate, and the F1score reflects the harmonic mean of precision and recall, indicating balanced performance. The high performance of the GRU model is also clear when compared to a recent algorithm, LightGBM.

As evident from Table 7, our GRU-based model outperforms not only traditional machine learning algorithms like SVM but also other RNN variants such as LSTM and SimpleRNN in our specific ICS security context. The GRU model achieves the highest scores across all metrics, demonstrating its superior ability to detect and classify cyber threats in ICS network traffic. This performance can be attributed to GRU's efficient handling of long-term dependencies and its ability to capture complex temporal patterns in network data, which are crucial for identifying sophisticated cyber attacks in ICS environments

The proposed GRU model is superior to other models for several reasons. First, GRUs, a type of RNN, excel at processing sequential data by capturing long-term dependencies with their gating mechanism, enabling them to learn crucial temporal patterns for anomaly detection. Second, effective preprocessing of input data using ZIP to handle non-occurrence of events enhanced model training. Finally, the well-suited model architecture and hyperparameter settings, including model depth, number of units, normalization techniques, and optimization algorithms, were carefully tuned to further improve performance.

## IV. IMPLEMENTATION

We implemented the proposed model in four modules, as illustrated in Figure 3.

The first module is the 'Data Collection' module, where a one-way port mirroring technique is used to collect data packets from network traffic. This secure, low-impact method sends mirrored packets to an analysis server known as the Packet Collector. The Packet Collector filters these mirrored packets to identify anomalies, which can represent various network situations and attacks, forming the basis for further analysis and processing.

The second module is the 'Pre-Processing' phase. Initially identified anomalies are analyzed to find specific Mapped_Rules. The detected Mapped_Rules, identified

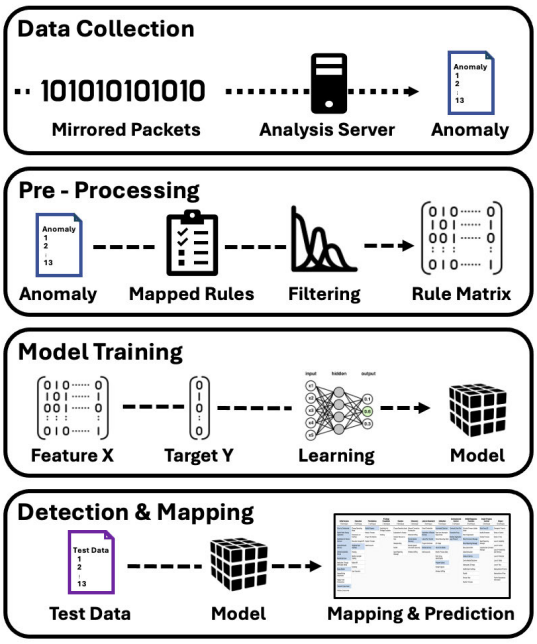


**FIGURE 3. Implementation details of proposed method.**

through this statistical method, are organized into a Rule_Matrix. During this process, the ZIP model is used to capture the temporal properties of the anomalies. The sequences of these Rule_Matrix serve as crucial input data for subsequent analysis and model training.

The third module is the 'Model Training' phase. Here, a GRU-based machine learning model is trained using the pre-processed Rule_Matrix data. The features (Feature X) represent the network traffic characteristics influenced with the ZIP distribution model, while the target variable (Target Y) represents the Mapped_Rules mapped to attack techniques in MITRE ATT&CK framework.

Finally, in the 'Detection & Mapping' module, the trained model is applied to datasets. The identified attacks undergo a 'Detection & Mapping' process for follow-up. During which we identify if the combination of attacks will result into specific threats like Stuxnet or Industroyer. In such scenario a notification is send to system administrator and attacks are logged in database and log files.

## V. EXPERIMENT AND RESULT

The MITRE ATT&CK framework is a comprehensive, globally accessible knowledge base of adversary tactics and techniques based on real-world observations. It is designed to help organizations understand and defend against cyber threats. Figure 4. provides a comprehensive list of attack techniques categorized into 12 different tactic categories. We integrated this framework into our model with the help of Mapped_Rules. The highlighted pink cells indicate the attack techniques our model could detect using the time series traffic data.

The major attack techniques detected through our model includes the following:

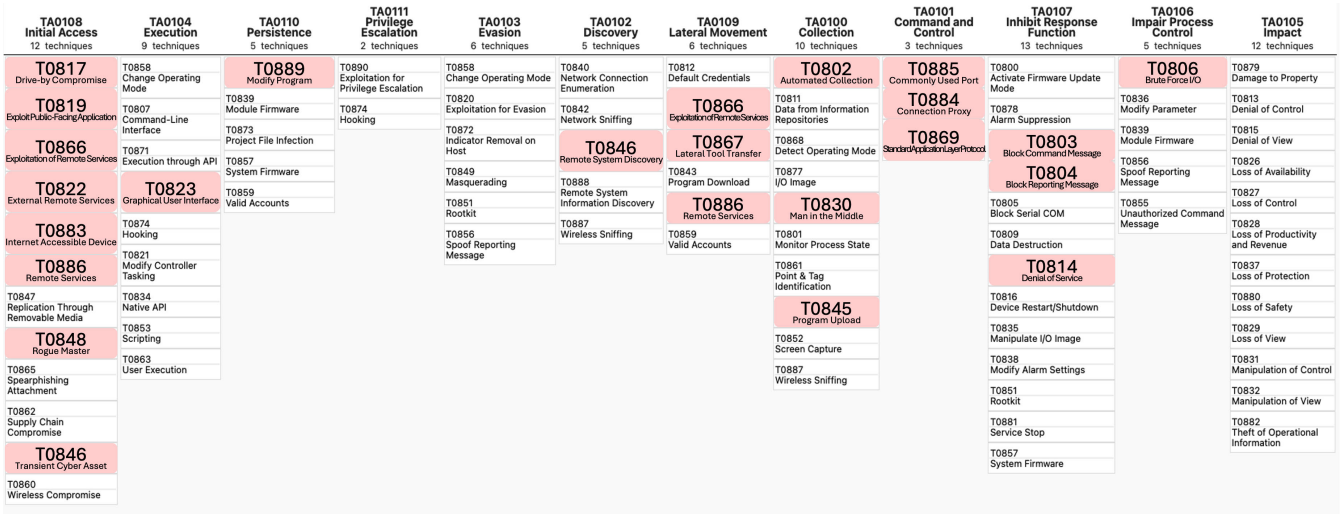


**FIGURE 4.** Detectable anomaly type through proposed model in 'MITRE ATT&CK for ICS'.

Drive-by Compromise [T0817],
Exploit Public-Facing Applications [T0819],
External Remote Services[T0822],
Internet Accessible Device [T0883],
Remote Services [T0886],
Graphical User Interface [T0823],
Modify Program [T0889],
Remote System Discovery [T0846],
Exploitation of Remote Services [T0866],
Lateral Tool Transfer [T0867],
Remote Services [T0886],
Brute Force I/O [T0806],
Exploitation of Remote Services [T0866],
Transient Cyber Asset [T0864],
Rogue Master [T0848],
Automated Collection [T0802],
Standard Application Layer Protocol [T0869],
Connection Proxy [T0884],
Denial of Service [T0814],
Commonly Used Port [T0885],
Block Reporting Message [T0804],
Program Upload [T0845],
Man in the Middle [T0830],
Block Command Message [T0803].

To validate our proposed method, we simulated two major cyber threats Stuxnet and Industroyer against ICS [45]. Seeing how serious these attacks were, MITRE Corporation also developed MITRE ATT&CK guidelines of tactics and techniques used in Stuxnet and Industroyer as shown in Fig. 6 [46] and Fig. 8 [47] respectively. Through our traffic simulation and testing, our proposed model could detect these two most notorious threats with over 99% accuracy.

### A. STUXNET

Stuxnet was a very advanced computer worm that targeted ICS, mainly to destroy Iran's uranium enrichment facilities in year 2010. The worm spread through USB drives and used a Windows vulnerability to control PLCs using Siemens's Step7 software. Our research simulated the occurrence of Stuxnet and aimed to detect Stuxnet using network packets.

#### 1) ATTACK SCENARIO

The Stuxnet attack simulation involves a three-step process: 1) USB penetration, 2) network propagation and PLC control, and 3) malware dissemination and PLC damage.

First, a USB penetration happens on one of the devices in the network, producing packets labeled with anomaly_type 1 and 5. The second phase, involving network propagation and PLC control, generates traffic labeled by anomaly_type 2, 3, 4, 6, 7, and 8. The final phase, consisting of malware dissemination and PLC damage, results in traffic data labeled with anomaly_type 9, 10, 11, 12, and 13. The detailed description of these anomalies is provided in Table 8.

**TABLE 8.** Attack scenario, anomaly type and anomaly description for Stuxnet.

| Attack scenario | anomaly Type | anomaly Description |
|---|---|---|
| USB Penetration | 1 | New Asset Detection |
| | 5 | Unauthorized IP Detection |
| Network propagation and PLC control | 2 | New Communication between Assets |
| | 3 | Unauthorized Communication Detection |
| | 4 | Banned Communication Detection |
| | 6 | Data Anomaly Detection |
| | 7 | Detect New Packet Codes |
| | 8 | Detect New Packet Code Usage |
| Malware dissemination and PLC damage | 9 | New Protocol Packet Code Detection |
| | 10 | Detect New Protocol Packet Code Usage |
| | 11 | Traffic Threshold Exceeded |
| | 12 | Asset Under-Traffic |
| | 13 | Asset Over-Traffic |

To simulate the Stuxnet attack scenario, we generated a dataset of 388 attacks over a 2-hour period at 15-minute intervals. Each sample contains 24 features, including 13 anomaly types and 11 blocklisted protocols. The data is divided into three phases: USB penetration, network propagation and PLC control, and malware dissemination and PLC damage, with anomaly frequencies shown in Figure 5. We created traffic data with these anomaly events, including additional features like timestamp, IP addresses, ports, protocols, plant ID, area, and manufacturer. This dataset reflects the characteristics of an actual Stuxnet attack and is suitable for training and evaluating our detection model.

Figure 5 shows the frequency of anomaly events over time in the Stuxnet attack scenario. The x-axis represents time, with intervals of approximately 15 minutes, while the y-axis shows the count of anomaly events in each phase of the three-step Stuxnet attack model. Each phase is marked in different colors: blue circles represent USB penetration, red X's represent network propagation and PLC control, and green squares represent malware dissemination

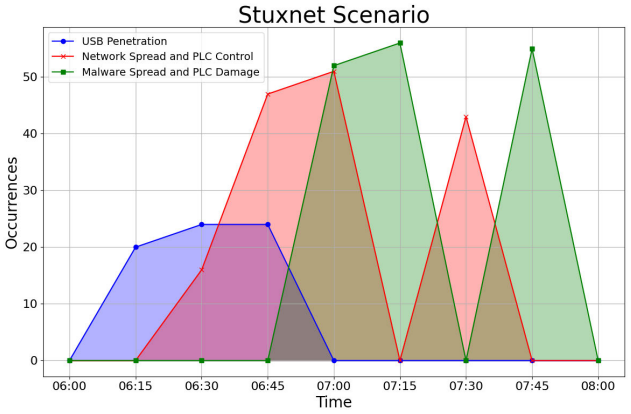


**FIGURE 5.** Stuxnet attack scenario by occurrence.

and PLC damage. The USB penetration phase starts with a low frequency and gradually increases, peaking at about 24 events. This shows the process of Stuxnet initially infiltrating the system via USB. The network propagation and PLC control phase exhibits two distinct peaks, with approximately 51 and 43 events respectively. These peaks indicate the initial spread of malware and subsequent attempts at PLC control. The final malware dissemination and PLC damage phase also shows two peaks, with the second peak reaching the highest frequency of over 56 events. This indicates the widespread dissemination of malware across the system, followed by intensive attacks on the PLC system.

This traffic data, embedded with anomalies and distribution patterns, undergoes a pre-processing phase to produce the Rule_Matrix, which is subsequently used as test input for testing and validation of the proposed model.

### 2) RESULTS AND ANALYSIS

The GRU-based learning model is provided with the simulated Stuxnet scenario data. Our proposed model demonstrates exceptionally high accuracy in detecting the Stuxnet attack with this simulated data.

Figure 6 shows the techniques used in the Stuxnet attack highlighted in MITRE ATT&CK framework. The light blue color represents techniques that can be seen in Stuxnet's network packets but are not detected by our model. The purple color represents techniques that can be found in Stuxnet's network packets and are detected by our model.

The following techniques are detected by our model:

- **Exploitation of Remote Services (T0866)** This is a technique where an attacker uses an insecure remote access service to gain access to a system. Our model can detect unusual remote service attempts.
- **Lateral Tool Transfer (T0867)** This technique involves transferring malicious tools or scripts from one infected system to another. Our model can detect unusual file transfers or tool usage in internal network packets.
- **Standard Application Layer Protocol (T0869)** This technique uses protocols like HTTP and FTP to transfer data or execute commands. Our model can detect the use of protocols that are restricted on the internal network.
- **Commonly Used Port (T0885)** An attacker tries to gain access through commonly used ports like TCP 80, 443, and 23. Our model can detect abnormal traffic through these specific ports.
- **Remote Services (T0886)** This technique involves controlling the system or accessing data through a remote service. Our model can detect anomalous access through network flow analysis.
- **Modify Program (T0889)** This technique involves malware modifying a legitimate program or process within a system to inject its own code. Our model can detect changes or modifications to programs. This matrix helps us respond quickly and accurately to real-world security incidents and enables predictive analysis for similar future attacks.

Specific techniques that could not be detected through the proposed model include the following.

- **Monitor Process State (T0801)** and **Remote System Information Discovery (T0888)** involve activities that monitor the state of processes inside a system or gather information from remote systems. Because these activities often look similar to legitimate administrative tasks, it is often difficult to distinguish them as malicious based on network packets alone.
- **Modify Parameter (T0836)** is a technique that involves changing system settings over the network. When analyzing network packets, these changes are often disguised as normal traffic or lack specific context, making it difficult to detect malicious intent.
- **Program Download (T0843)** and **User Execution (T0863)** are related to the act of a user downloading and executing a program. While these behaviors can be observed directly in network traffic, a deeper analysis of the contents of the packets is required to distinguish between legitimate and malicious behavior.

**TABLE 9.** Analsis_score of Stuxnet with the proposed model.

| Stuxnet Detectable Technique ID | Anomaly_Score |
|---|---|
| (T0866) Exploitation of Remote Services | 0.9999 |
| (T0867) Lateral Tool Transfer | 0.9979 |
| (T0869) Standard Application Layer Protocol | 0.9998 |
| (T0885) Commonly Used Port | 0.9999 |
| (T0886) Remote Services | 0.9989 |
| (T0889) Modify Program | 0.9998 |
| **Average_Scores** | **0.9993** |

Table 9 shows the Analysis_Score of detectable techniques for Stuxnet using the proposed model. Our study showed a high detection rate of more than 99% for various attack techniques, especially the Exploitation of Remote Services (T0866) technique was detected with 99.99% accuracy and the Standard Application Layer Protocol (T0869) was detected with 99.98% accuracy. The Analysis_Score provides a confidence measure that indicates the extent to which the detected anomaly matches each technique described in the

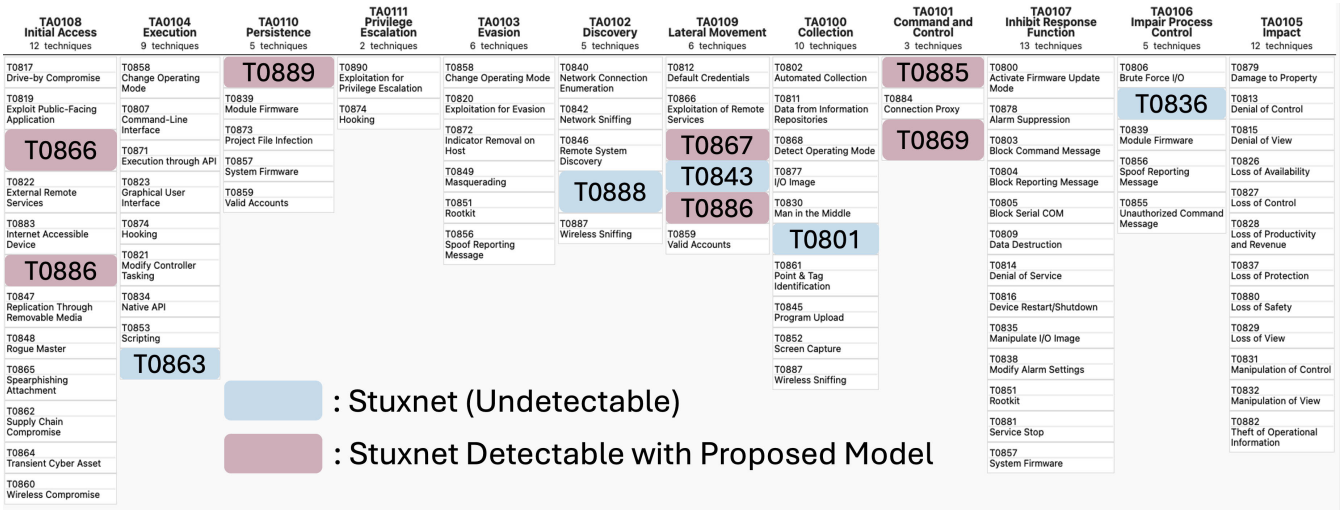


**FIGURE 6.** Stuxnet detection through proposed model.

MITRE ATT&CK framework. Scores closer to 1 indicate a closer match to the detected technique. The Average_Score represents the overall confidence level for the detected techniques, suggesting that the identification of multiple techniques signals a significant anomaly within the network.

### B. INDUSTROYER

Industroyer was an advanced malware attack on Ukraine's power grid in 2016. It targeted ICS equipment like SCADA and PLCs to disrupt power operations and cut off the power supply. The malware spread through email spearphishing and moved across networks, using specialized ICS communication protocols to issue commands and disrupt the system [8].

#### 1) ATTACK SCENARIOS

The Industroyer attack simulation involves a three-step process: 1) email spearphishing infiltration, 2) network spread and SCADA control, and 3) PLC detection and control.

First, an email spearphishing infiltration occurs, allowing Industroyer to enter the network through a malicious email. This phase generates packets labeled with anomaly_type 1 and 5. The second phase, network spread and SCADA control, produces traffic labeled by anomaly_type 2, 3, 4, 6, and 11. The final phase, consisting of PLC detection and control, results in traffic data labeled with anomaly_type 7, 8, 9, 10, 12, and 13. The detailed description of these anomalies is provided in Table 10.

To simulate the Industroyer attack scenario, we generated a dataset of 412 attacks over a 2-hour period at 15-minute intervals. Each sample contains 24 features, including 13 anomaly types and 11 blocklisted protocols. The data is divided into three phases: email spearphishing infiltration, network spread and SCADA control, and PLC detection and control, with anomaly frequencies shown in Figure 7. This dataset reflects the characteristics of an actual

**TABLE 10.** Attack scenario, anomaly type and anomaly description for industroyer.

| Attack scenario | Anomaly Type | Anomaly Description |
|---|---|---|
| Email Spearphishing Infiltration | 1 | New Asset Detection |
| | 5 | Unauthorized IP Detection |
| Network Spread and SCADA Control | 2 | New Communication between Assets |
| | 3 | Unauthorized Communication Detection |
| | 4 | Banned Communication Detection |
| | 6 | Data Anomaly Detection |
| | 11 | Traffic Threshold Exceeded |
| PLC Detection and Control | 7 | Detect New Packet Codes |
| | 8 | Detect New Packet Code Usage |
| | 9 | New Protocol Packet Code Detection |
| | 10 | Detect New Protocol Packet Code Usage |
| | 12 | Asset Under-Traffic |
| | 13 | Asset Over-Traffic |

Industroyer attack and is suitable for training and evaluating our detection model.

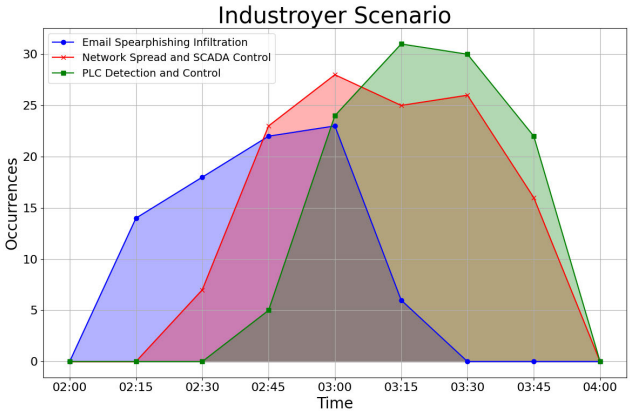


**FIGURE 7.** Industroyer attack scenario by occurrence.

Figure 7 shows the frequency of events over time in the Industroyer scenario. Each event is marked with a different color and symbol to help distinguish them. The x-axis represents time. Each time interval is separated by about 15 minutes. The y-axis shows the number of times the event

occurred. The legend indicates that blue circles represent email spearphishing infiltration, red X's represent network spread and SCADA control, and green squares represent PLC detection and control. The first peak mainly represents the email spear phishing phase, where the attacker initially enters the network. The second peak occurs during the network spread and SCADA control phase, where the attacker tries to expand the attack through the network and take control of the SCADA system. The third peak happens in the PLC detection and control phase, where the malware attempts to disrupt the core functions of the system by manipulating the PLC.

Figure 7 shows the Industroyer attack scenario, which can be compared with the Stuxnet scenario depicted in Figure 5. While both attacks show a similar three-phase structure, there are notable differences in their temporal patterns and intensity. Unlike Stuxnet, which showed two distinct peaks in its second and third phases, Industroyer exhibits a more gradual increase in anomaly events across all phases. This difference reflects the distinct propagation and attack strategies employed by each malware.

Following the Stuxnet model, we generated traffic data for Industroyer that incorporates the anomalies and patterns depicted in Figure 7. Each packet in this simulation contained details of transmission time, source and destination IP addresses, protocol type, anomaly type, and the number of occurrences. This traffic data, embedded with anomalies and distribution patterns undergoes a pre-processing phase to produce the Rule_Matrix, which is subsequently used as test input for testing and validation of the proposed model.

### 2) RESULTS AND ANALYSIS

In our study, we evaluated the performance of the proposed model in the Industroyer scenario and found that it can detect attacks with very high accuracy.

Figure 8 provides a visual representation of the techniques in the network packet field used in the Industroyer attack and the techniques that are detectable in our model. The matrix follows the structure of the MITRE ATT&CK framework, which allows us to identify attack techniques and intuitively determine their detectability. The mauve color represents techniques that are visible in Industroyer's network packets but are not detected by our model. The purple color indicates techniques that are visible in Industroyer's network packets and can be detected by our model. The following technologies are detectable with our model

- **Automated Collection(T0802)** A technique that automatically collects data from a system, our model can detect unusual data traffic or unexpected increases in the use of system resources to capture this behavior.
- **Block Command Message (T0803)** A technique that blocks specific command messages being delivered to the network or system. Our model detects anomalies by monitoring interruptions in command flow or system response.
- **Block Reporting Message (T0804)** A technique to block reporting messages about security incidents or events in a system or network. Our model can check the communication between the security system and monitoring tools.
- **Brute Force I/O (T0806)** A technique that attempts to gain access to input/output devices through repeated attempts. Our model detects these attacks by monitoring failed login attempts.
- **Denial of Service (T0814)** A denial of service attack, a technique that overloads network resources to prevent normal service usage. Our model can detect these attacks early by using network performance monitoring tools.
- **Remote System Discovery (T0846)** A technique in which an attacker explores and gathers information from other systems within a network. Our model can detect network scan attempts or unusual network queries.
- **Connection Proxy (T0884)** A technique in which an attacker hides the source of his traffic by diverting it through a proxy server or other relay device. Our model detects by monitoring unusual proxy usage or connections to unknown proxy services.

Specific techniques that could not be detected using proposed model include the following.

- **Activate Firmware Update Mode(T0800)** is a technique to activate a device's firmware update mode. This can be considered a normal maintenance activity, and it is difficult to distinguish if this activity is performed for malicious purposes based on network packet analysis alone.
- **Monitor Process State(T0801)** and **Remote System Information Discovery(T0888)** are techniques for monitoring the state of a system or gathering information from a remote system. These information gathering activities are similar to normal system administration, and it can be difficult to determine malicious intent based on packet analysis alone.
- **Block Serial COM(T0805)** and **Device Restart Shutdown(T0816)** relate to techniques that block communication ports or restart or shut down equipment. These behaviors are difficult to detect directly in network traffic and can only be verified through physical manipulation of the device or analysis of internal logs.
- **Unauthorized Command Message(T0855)** involves techniques that send unauthorized command messages. These unauthorized commands can be similar to legitimate commands and are sometimes difficult to distinguish through packet analysis. Detecting them requires more advanced analysis techniques, specific signatures, and behavior-based detection.

Table 11 shows the Analysis_Score of detectable techniques for Industroyer using the proposed model. Our study showed a high detection rate of over 99% for attack techniques against the Indestroyer. Our model was able to detect Automated Collection, Block Command Message, Block Reporting Message, Brute Force I/O, Denial of Service, Remote System Discovery, and Connection Proxy techniques with

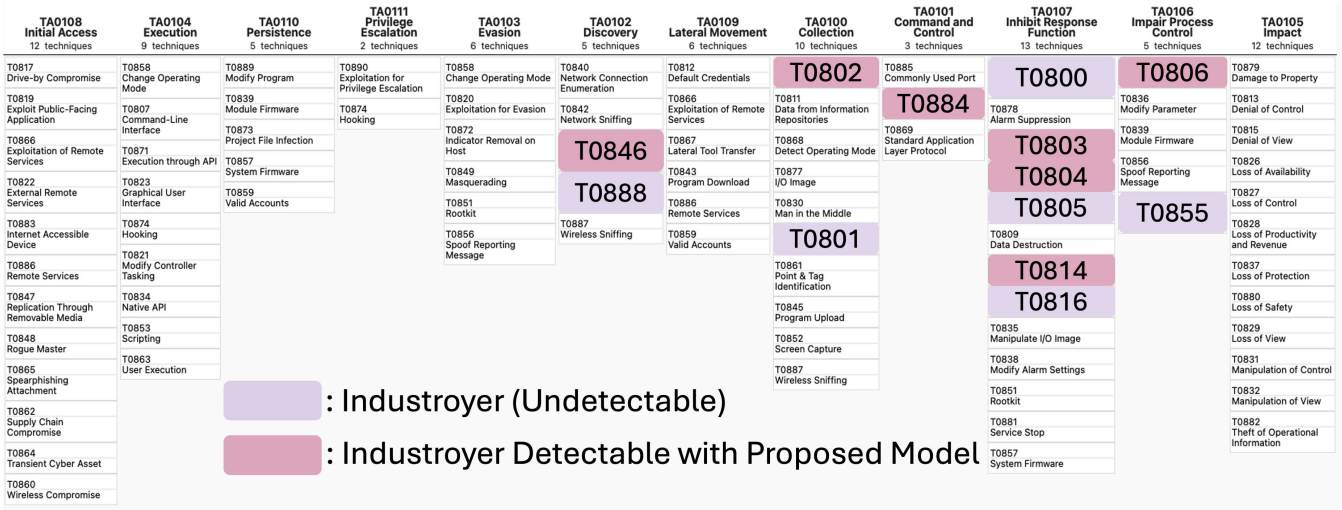


**FIGURE 8.** Industroyer detection through proposed model.

**TABLE 11.** Analsis_score of Industroyer with the proposed model.

| Industroyer Detectable Technique ID | Anomaly_Score |
|---|---|
| (T0802) Automated Collection | 0.9999 |
| (T0803) Block Command Message | 0.9997 |
| (T0804) Block Reporting Message | 0.9998 |
| (T0806) Brute Force I/O | 0.9999 |
| (T0814) Denial of Service | 0.9989 |
| (T0846) Remote System Discovery | 0.9998 |
| (T0884) Connection Proxy | 0.9996 |
| **Average_Scores** | **0.9996** |

high accuracy. In particular, Automated Collection (T0802) was detected with 99.99% accuracy, and Block Command Message (T0803) was detected with 99.97% accuracy.

## VI. CONCLUSION

In this paper, we introduce an innovative approach that employs GRU-based machine learning techniques for detecting cyber attacks in ICS network. Our model integrates the standard attack techniques outlined by the MITRE ATT&CK framework with raw input data. To address the temporal characteristics of infrequent anomaly events, we utilized a zero-inflated Poisson distribution model. We validated our methodology by simulating major cyberattack scenarios, including Stuxnet and Industroyer. Our experiments demonstrated that our model can detect attacks with a high success rate exceeding 95%.

Our current research provides a foundation for our future endeavors. While we presently focus on network traffic data, we plan to incorporate log files and other operating system event data to identify a wider range of attack patterns and complex scenarios. Currently, the proposed model is designed for attack detection, but future developments will aim to predict imminent attacks, enabling proactive measures. Furthermore, the effectiveness of our model requires further validation through application and testing in real industrial environments. In conclusion, this research is anticipated to significantly contribute to the security of ICS and the advancement of related research and technology.

## REFERENCES

[1] S. K. Das, K. Kant, and N. Zhang, "Introduction," in *Handbook on Securing Cyber-Physical Critical Infrastructure*. Amsterdam, The Netherlands: Elsevier, 2012, pp. 357–359.
[2] A. V. Gheorghe, M. Masera, M. Weijnen, and L. De Vries, "Critical infrastructures at risk," in *Securing the European Electric Power System*, 1st ed. Dordrecht, The Netherlands: Springer, 2006.
[3] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Syst. Mag.*, vol. 21, no. 6, pp. 11–25, Dec. 2001.
[4] H. Kagermann, "Change through digitization-value creation in the age of Industry 4.0," in *Management of Permanent Change*. Wiesbaden, Germany: Springer, 2015, pp. 23–45.
[5] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *J. Intell. Manuf.*, vol. 31, no. 1, pp. 127–182, Jan. 2020.
[6] T. Macaulay, B. L. Singer, and H. Plc, *Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS*. Boca Raton, FL, USA: CRC Press, 2011.
[7] G. Bush, *National Security Strategy of the United States*. Washington, DC, USA: White House, 1991.
[8] J. R. Lindsay, "Stuxnet and the limits of cyber warfare," *Secur. Stud.*, vol. 22, no. 3, pp. 365–404, Jul. 2013.
[9] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, Aug. 2014.
[10] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.
[11] W. Knowles, D. Prince, D. Hutchison, J. F. Disso, and K. A. Jones, "A survey of cyber security management in industrial control systems," *Int. J. Crit. Infrastruct. Protection*, vol. 9, pp. 52–80, Jun. 2015.
[12] J. Ohlin, *Cyber War: Law and Ethics for Virtual Conflicts*. Oxford, U.K.: OUP, 2015.
[13] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, Feb. 2011.
[14] T. Rid and B. Buchanan, "Attributing cyber attacks," *J. Strategic Stud.*, vol. 38, nos. 1–2, pp. 4–37, Jan. 2015.
[15] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet under the microscope," ESET LLC, Bratislava, Slovakia, Tech. Rep. 6, Sep. 2010.
[16] T. Rid, "Cyber war will not take place," *J. Strat. Stud.*, vol. 35, no. 1, pp. 5–32, 2012.
[17] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special Publication*, vol. 800, no. 82, p. 16, 2011.
[18] Y. Mo, T. H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, Jan. 2012.
[19] D. Case, "Analysis of the cyber attack on the Ukrainian power grid," *Electr. Inf. Sharing Anal. Center (E-ISAC)*, vol. 388, p. 3, Mar. 2016.

[20] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *Int. J. Electr. Power Energy Syst.*, vol. 99, pp. 45–56, Jul. 2018.

[21] A. N. Jahromi, H. Karimipour, A. Dehghantanha, and K. R. Choo, "Toward detection and attribution of cyber-attacks in IoT-enabled cyber–physical systems," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13712–13722, Sep. 2021.

[22] N. Bhusal, M. Abdelmalak, M. Kamruzzaman, and M. Benidris, "Power system resilience: Current practices, challenges, and future directions," *IEEE Access*, vol. 8, pp. 18064–18086, 2020.

[23] E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. The Netherlands: Syngress, 2014.

[24] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[25] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial Internet of Things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018.

[26] O. Alexander, M. Belisle, and J. Steele, *MITRE ATT&CK for Industrial Control Systems: Design and Philosophy*, vol. 29. Bedford, MA, USA: The MITRE Corporation, 2020.

[27] M. H. Almeshekah and E. H. Spafford, "Cyber security deception," in *Cyber Deception*. Cham, Switzerland: Springer, 2016, pp. 23–50.

[28] S. Maesschalck, V. Giotsas, B. Green, and N. Race, "Don't get stung, cover your ICS in honey: How do honeypots fit within industrial control system security," *Comput. Secur.*, vol. 114, Mar. 2022, Art. no. 102598.

[29] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for Internet of Things, industrial Internet of Things, and cyber-physical systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2351–2383, 4th Quart., 2021.

[30] N. Dutta, N. Jadav, N. Dutiya, and D. Joshi, "Using honeypots for ICS threats evaluation," in *Recent Developments on Industrial Control Systems Resilience*. Cham, Switzerland: Springer, 2020, pp. 175–196.

[31] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 8, Aug. 2018, Art. no. 155014771879461.

[32] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based IDS: A review and open issues of an anomaly detection system in IoT," *Future Gener. Comput. Syst.*, vol. 133, pp. 95–113, Aug. 2022.

[33] E. J. M. Colbert and S. Hutchinson, "Intrusion detection in industrial control systems," in *Advances in Information Security*. Cham, Switzerland: Springer, 2016, pp. 209–237.

[34] Y. Otoum and A. Nayak, "AS-IDS: Anomaly and signature based IDS for the Internet of Things," *J. Netw. Syst. Manage.*, vol. 29, no. 3, p. 23, Jul. 2021.

[35] K. Hwang, M. Cai, Y. Chen, and M. Qin, "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes," *IEEE Trans. Dependable Secure Comput.*, vol. 4, no. 1, pp. 41–55, Jan. 2007.

[36] A. M. Y. Koay, R. K. L. Ko, H. Hettema, and K. Radke, "Machine learning in industrial control system (ICS) security: Current landscape, opportunities and challenges," *J. Intell. Inf. Syst.*, vol. 60, no. 2, pp. 377–405, Apr. 2023.

[37] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[38] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of big data and machine learning in smart grid, and associated security concerns: A review," *IEEE Access*, vol. 7, pp. 13960–13988, 2019.

[39] J.-P.-A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, "Cyber-physical systems security: Limitations, issues and future trends," *Microprocess. Microsyst.*, vol. 77, Sep. 2020, Art. no. 103201.

[40] L. Rosa, T. Cruz, M. B. D. Freitas, P. Quitério, J. Henriques, F. Caldeira, E. Monteiro, and P. Simoes, "Intrusion and anomaly detection for the next-generation of industrial automation and control systems," *Future Gener. Comput. Syst.*, vol. 119, pp. 50–67, Jun. 2021.

[41] Y. Wu, H.-N. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9214–9231, Jun. 2022.

[42] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.

[43] A. K. Ozcanli, F. Yaprakdal, and M. Baysal, "Deep learning methods and applications for electrical power systems: A comprehensive review," *Int. J. Energy Res.*, vol. 44, no. 9, pp. 7136–7157, Jul. 2020.

[44] D. Javaheri, M. Fahmideh, H. Chizari, P. Lalbakhsh, and J. Hur, "Cybersecurity threats in FinTech: A systematic review," *Exp. Syst. Appl.*, vol. 241, May 2024, Art. no. 122697.

[45] G. M. Makrakis, C. Kolias, G. Kambourakis, C. Rieger, and J. Benjamin, "Industrial and critical infrastructure security: Technical analysis of real-life security incidents," *IEEE Access*, vol. 9, pp. 165295–165325, 2021.

[46] *Stuxnet*. Mitre.Org. Accessed: Oct. 2024. [Online]. Available: https://attack.mitre.org/software/S0603/

[47] *Stuxnet*. Mitre.Org. Accessed: Oct. 2024. [Online]. Available: https://attack.mitre.org/groups/G0064/

**WOOHYUN CHOI** received the B.S. degree in computer of education from Gyeong-Sang National University, Jinju, Republic of Korea, in 2012, and the M.S. degree in information security from Soong-Sil University, Seoul, Republic of Korea, in 2015. He is currently pursuing the Ph.D. degree with the AI Graduate School, Gwangju Institute of Science and Technology (GIST), Gwangju, Republic of Korea. From 2023 to 2024, he was a Visiting Scholar at the Aerospace Systems Design Laboratory (ASDL), Georgia Tech. His research interests include artificial intelligence (AI), industrial control system (ICS), cyber-physical system (CPS), security for ICS, and abnormal behavior detection using AI.

**SUMAN PANDEY** received the M.S. degree in computer science from POSTECH and the Ph.D. degree from Kangwon National University, in 2020. From 2020 to 2022, she was a Postdoctoral Fellow at POSTECH. She currently works as an Assistant Professor with the Department of EECS, GIST. Prior to joining GIST, she was an Assistant Professor at the Department of Computer Science, Daegu Catholic University, from 2011 to 2019. Before transitioning to academia, she gained experience in multinational software companies in India, from 2004 to 2007. Her research interests include network management, artificial intelligence, and streaming technologies.

**JONGWON KIM** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1987, 1989, and 1994, respectively. From 1994 to 2001, he was a Faculty Member at Kongju National University, Gongju, South Korea, and the University of Southern California, Los Angeles, CA, USA. In 2001, he joined Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, where he is currently working as a Full Professor. Since 2008, he has been directing the GIST Super Computing Center. Since 2019, he has been the Dean of the AI Graduate School, GIST. He is also leading Networked Computing Systems Laboratory, where he is involved in dynamic and resource-aware composition of media-centric service employing programmable/virtualized computing/networking resources. His recent research interest includes agile and visible p+v+c function-leveraged composition of SmartX the IoT-cloud services employing programmable/sliced/hyper-converged (computing/storage/networking) resources.