Enhancing MITM Attack Detection Mechanism for ICS using LSTM-based Hybrid Ensemble Learning

DOI: https://doi.org/10.54654/isj.v3i26.1137

Nguyen Tuan Anh, Le Van Dong, Dao Viet Cuong, Nguyen Dinh Nghia, Tran Quang Duc*

***Abstract*— With the rapid development of Information Technology (IT), the integration of IT with Industrial Control System (ICS) makes it susceptible to cybersecurity threats, including Man-in-the-Middle (MITM) attacks. Many studies focus on MITM attack detection approaches that include rule-based methods and those using Machine Learning (ML). However, these approaches suffer from two main limitations: a lack of a dataset for MITM attack detection in ICS networks and an effective MITM attack detection method due to the ever-increasing complexity of ICS networks. In this paper, we propose a novel MITM attack detection framework using an ensemble learning algorithm for large-scale ICS networks. Concretely, we propose a novel ICS simulation framework for large-scale networks using Software-Defined Networking to facilitate ICS studies. Moreover, a novel lightweight MITM attack detection mechanism using an enhanced pre-processing technique and a hybrid ensemble learning algorithm using Long Short-Term Memory (LSTM) is proposed to detect MITM attacks with high accuracy while requiring suitable processing time. Experimental results show that the proposed MITM attack detection mechanism can achieve an f1 score of 91.91% while requiring only 8.91 microseconds for inference time.**

***Tóm tắt*— Với sự phát triển nhanh chóng của công nghệ thông tin (Information Technolog - IT), việc tích hợp IT với hệ thống điều khiển công nghiệp (Industrial Control System - ICS) khiến nó dễ bị đe dọa bởi các mối đe dọa an ninh mạng, bao gồm các cuộc tấn công xen giữa (Man-in-the-Middle - MITM). Do đó, nhiều nghiên cứu tập trung vào các phương pháp phát hiện tấn công MITM, bao gồm các phương pháp dựa trên luật và các phương pháp sử dụng học máy (Machine Learning - ML). Tuy nhiên, các phương pháp này gặp phải hai hạn chế chính: thiếu bộ dữ liệu để phát hiện tấn công MITM trong mạng ICS và một phương pháp phát hiện tấn công MITM hiệu quả do tính phức tạp ngày càng tăng của mạng ICS. Trong bài báo này, nhóm tác giả đề xuất một khuôn khổ phát hiện tấn công MITM mới, sử dụng thuật toán học tổ hợp, cho các mạng ICS quy mô lớn. Cụ thể, nhóm tác giả đề xuất một khuôn khổ mô phỏng ICS mới cho các mạng quy mô lớn, sử dụng Mạng được định nghĩa bằng phần mềm (Software-Defined Networking - SDN) để tạo điều kiện thuận lợi cho các nghiên cứu ICS. Hơn nữa, một cơ chế phát hiện tấn công MITM nhẹ mới sử dụng kỹ thuật tiền xử lý nâng cao và thuật toán học tập kết hợp lai sử dụng Bộ nhớ dài Ngắn Hạn (Long Short-Term Memory - LSTM) được đề xuất để phát hiện các cuộc tấn công MITM với độ chính xác cao trong khi vẫn yêu cầu thời gian xử lý hợp lý. Kết quả thử nghiệm cho thấy cơ chế phát hiện tấn công MITM được đề xuất có thể đạt được điểm F1 91,91% trong khi chỉ cần 8,91 micro giây cho thời gian suy luận.**

***Keywords*— Man-in-the-Middle Attack, industrial control system, software-defined networking, ensemble learning.**

***Từ khóa*—Tấn công xen giữa, hệ thống điều khiển công nghiệp, mạng được định nghĩa bằng phần mềm, học tổ hợp.**

I. Introduction

Historically, Industrial Control System (ICS) denotes the interconnected systems and devices utilized for the control and automation of industrial processes. The rapid progression of Information Technology (IT) makes the integration of IT with ICS inevitable, facilitating the sustained functioning and monitoring of processes in numerous manufacturing facilities.

This manuscript was received on September 12, 2025. It was reviewed on November 25, 2025, revised on December 11, 2025 and accepted on December 17, 2025.

* Corresponding author.

Despite its benefits, ICS is susceptible to cybersecurity threats, including Advanced Persistent Threats and Man-in-the-Middle (MITM) attacks and so on. In 2024, FrostyGoop malware was designed to exploit the Modbus protocol, resulting in a widespread power outage affecting thousands of buildings in Ukraine. The subsequent research disclosed that the Modbus protocol was employed by more than 46,000 ICS devices, presenting numerous potential security vulnerabilities. In 2021, the ICS network of Colonial Pipeline in the United States was targeted, resulting in the disruption of oil delivery to numerous states, including Florida, Georgia, Alabama, Virginia, and the Carolinas.

The MIMT attack [1] is a well-known cyberattack that results in significant consequences for both IT and ICS networks. A MIMT attack seeks to insert itself into the data exchange between two parties to eavesdrop, intercept, modify, or impersonate the data. ICS protocols, such as Modbus, DNP3, and IEC-61850, lack encryption and authentication measures, rendering them susceptible to MITM attacks. Consequently, the scientific community has thoroughly examined many methods for detecting MITM attacks in the past. Conventional detection methods for MIMT attacks depend on either established rules [2]–[5] or Machine Learning (ML) techniques [6]–[8]. Nevertheless, these approaches are hindered by two main drawbacks. Firstly, MITM attacks happen unexpectedly in ICS networks, resulting in conventional detection methods lacking ICS datasets for the establishment of predefined rules and training models. In this context, simulation frameworks (e.g., miniCPS [9], etc.) represent a promising approach for simulating ICS protocols and gathering ICS data. Nonetheless, these frameworks are intended for small-scale ICS networks, whereas contemporary ICS networks are more complicated, incorporating several devices (e.g., HMI, PLC, etc.). Secondly, rule-based and ML-based approaches can achieve good performance in the considered scenarios; however, they tend to underperform in ICS-specific MITM detection, especially on minority attack classes and under imbalanced traffic, due to limited feature extraction and classification capacity tailored to ICS protocols.

Consequently, this research proposes an innovative MITM attack detection framework utilizing an ensemble learning algorithm for large-scale ICS networks to address these shortcomings. Initially, in contrast to current studies that utilize small-scale simulation frameworks (e.g., miniCPS, etc.), we propose an innovative simulation framework for large-scale ICS networks employing Software-Defined Networking (SDN). Employing SDN, an innovative network architecture aimed at enhancing network operation and management, the proposed simulation framework can concurrently simulate a large number of network devices (e.g., switches, PLCs, etc.) to create large-scale ICS networks. Secondly, we present an innovative lightweight MITM attack detection mechanism via a hybrid ensemble learning algorithm using Long Short Term Memory (LSTM) to improve detection efficacy and facilitate real-time application. The LSTM-based hybrid ensemble learning algorithm integrates both bagging and boosting techniques, with LSTM as the fundamental classifier. Concretely, the proposed mechanism systematically gathers network traffic from the simulation framework and preprocesses it to remove flow identifiers (e.g., IP addresses, application ports, timestamps, etc.). Subsequently, the cleaned network traffic is put into an improved feature selection algorithm which integrates low-variance filtering and ANOVA F-test, to select appropriate features because several features are not effective for MITM attack detection. These appropriate features are then analyzed in an LSTM-based hybrid ensemble learning algorithm to identify MITM attacks effectively. Moreover, the proposed mechanism facilitates a real-time detection application. A real-time detection mechanism is crucial for ICS networks. The longer the duration required to identify the attack, the greater the economic losses incurred by the manufacturers. The contributions of this research are listed as follows:

- A novel ICS simulation framework for large-scale networks using SDN.
  A novel lightweight MITM attack detection mechanism using an enhanced pre-processing mechanism and an LSTM-based hybrid ensemble learning algorithm.
- A comprehensive analysis of ML, DL and Transformer algorithms in the MITM attack

detection mechanism.

Outline: The structure of the paper is listed as follows. Section II presents the background about miniCPS and the related work on MITM attack detection. In section III, the large-scale ICS simulation and MITM attack detection mechanism using ML are presented. Section IV presents the experimental setup and experimental results of the proposed MITM detection mechanism. The paper concludes with section V which highlights our future work.

II. Background and Related Work

A. Background

MiniCPS [9] is a compact simulation framework intended to replicate real-time physical interactions and network operations within Cyber-Physical Systems (CPS). Components inside MiniCPS, including HMI, PLC, and SCADA, are developed in Python to facilitate data transmission and reception over standard industrial protocols such as Ethernet/IP or Modbus. They are interconnected via a virtual networking layer established on Mininet, which provides adaptable configuration options for latency, bandwidth, and packet loss, while also facilitating the development of intricate network topologies, including star networks, high-speed Ethernet rings, and intermediary routing devices.

A significant characteristic of MiniCPS is the segmentation and abstraction of the physical layer, managed by a collection of JSON-based APIs, enabling simulated components such as PLCs and sensors to read and write data effectively. Physical parameters, including water level, pressure, and flow rate, are perpetually refreshed in real-time by physical process simulation modules, rendering MiniCPS exceptionally adaptable for integration with other tools or physics and chemistry simulation software. This study presents a large-scale simulation framework inspired by MiniCPS.

B. Related Work

The MIMT attacks are among the most prevalent methods recognized in ICS. These attacks are generally executed by disrupting the functionality of conventional industrial communication protocols (e.g., DNP3 [10], Modbus [11], IEC-61850, etc.) utilizing tools like Scapy, Metasploit, and so on. Consequently, attackers can steal confidential information, inject control instructions, or manipulate data to interfere with system functionality.

In the past, IDSs, namely Snort, Zeek, and Suricata [2]–[4] were designed to detect MITM attacks in the network system. Although functioning proficiently with IT networks, these IDSs reveal incompatibility with ICS networks.

In recent years, the swift growth of ML and Deep Learning (DL) has introduced novel approaches in the examination of anomaly detection in ICS networks. Many studies have been presented to efficiently identify prevalent attack types in ICS environments. Elrawy et al. [6] developed a hybrid network IDS for detecting and classifying MITM attacks in smart grid networks, achieving 97 % detection accuracy. Raja et al. [7] proposed a unified Random Forest model enhanced by hybrid bat optimization, while Eigner et al. [8] ultilized K-Nearest Neighbors combined with Bregman divergence, to identify deviations from normal operational patterns in ICS.

Yang et al. [12] proposed a multidimensional IDS to identify the characteristics of MiTM attacks, although it is limited to the IEC-61850 protocol. Wlazlo et al. [5] examined network traffic to find network-specific indications, including abnormal latency, packet retransmission rates, and packet order, for the detection of MiTM attacks. Various research have utilized DL algorithms [13], [14] (e.g., Convolutional Neural Network, LSTM, Autoencoder, etc.) to identify anomalies (including MITM attack) based on contextual and behavioral factors.

Despite their impressive performance in the considered environments, the above approaches are hindered by two drawbacks. Firstly, these approaches are executed and assessed inside a small-scale simulation context. Secondly, the efficacy of these methodologies is inadequate for identifying MITM attacks in ICS networks due to constrained feature extraction and classification capabilities. Consequently, this paper introduces an innovative MITM attack detection framework employing an LSTM-based ensemble learning method for large-scale ICS networks.

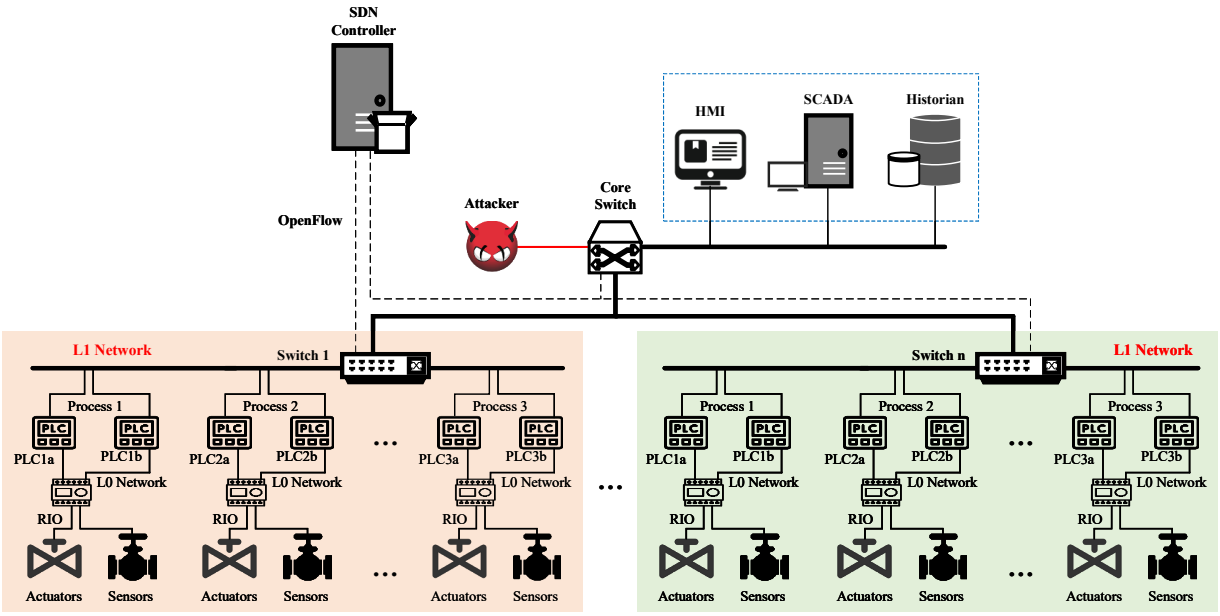


Figure 1. System architecture for large-scale ICS simulation

III. Methodology

This section details the simulation of a large-scale ICS and the proposed MITM attack detection method.

A. *Large-scale Industrial Control System Simulation*

The proposed large-scale ICS simulation framework, inspired by MiniCPS and the standard SWaT system process, is depicted in Figure 1. The proposed framework is divided into multiple independent zones. Each zone will contain six PLC devices that gather data from sensors and regulate actuators, including pumps and valves. The PLC will process sensor input and synchronize it with the central management system, which contains the SCADA and historian servers. Concurrently, PLC devices receive control commands from the HMI to execute tasks such as opening or closing valves and initiating or halting pumps. HMI devices will have the responsibility of interpreting the water level data from the storage tanks, as gathered by the PLC. Utilizing the gathered water level status data, it will determine whether to transmit control orders to the PLC devices for the adjustment of pump or valve statuses. The switches in each partition will connect to the high-speed central switch centrally, facilitating communication between the HMI and the PLCs in each partition. All switching devices inside the system will be controlled according to regulations set forth by the SDN Controller via the OpenFlow protocol. In this architecture, POX functions as a controller, facilitating bandwidth configuration on the central switch to control congestion and ensure smooth traffic flow across the network.

The data interchange mechanism between the HMI and the underlying PLCs adheres strictly to the procedures established by the CIP protocol. The HMI utilizes the receive and send functions offered by MiniCPS to read or write a PLC tag value. Communication between the HMI and PLC transpires in three primary stages. The HMI establishes a TCP connection with the PLC via the conventional three-way handshake procedure (SYN, SYN/ACK, ACK). The HMI subsequently transmits a Register Session Request in accordance with the EtherNet/IP standard and receives a response from the PLC to initiate an official communication session. Upon session establishment, the HMI transmits CIP (Common Industrial Protocol) queries, generally read or write requests for a specified tag, utilizing CIP CM Request packets. The CIP protocol transmits data between the HMI and the PLC device without encryption, facilitating interception and decryption of the captured packets by attackers. Simultaneously, with techniques such as Scapy, attackers might illegally alter the content of

packets transferred between the HMI and PLC without the awareness of either party involved.

In the proposed framework, we assume that the attacker can exploit vulnerabilities and effectively connect to the ICS network. The attacker can thereafter target the core switch, HMI, and PLC. In MITM attack, there are two scenarios: data stealing and data spoofing. In order to complete these scenarios, the attacker need a high-speed monitoring tool to monitor and collect the network traffic in the context of a large-scale ICS network. In this case, *ntopng* [15] is taken into account as a monitoring tool which offers many properties like flow exports (NetFlow, sFlow, IPFIX) from routers/switches and various OS support (e.g., Window, Ubuntu).

In order to study and research about ICS attacks, ICS protocols, building a real system is needed, but it is costly and time-comsuming. Therefore, in this paper, we proposed a novel ICS simulation framework for large-scale networks. This framework is a simulation framework to simulate ICS attacks (data sniffing, data manipulation, etc.), which leverages SDN and monitoring tools (e.g., ntopng). Unlike traditional network architecture, which combines the control and data layer in a network device, SDN is a novel network architecture that separates these two layers. This allows us to manage a large number of network devices and easily extend the network infrastructure. In SDN, the data layer (switches, etc.) is responsible for data forwarding, while the control layer contains SDN controllers, which control the network management via application layers. This separation reduces the burden on data-plane devices and optimizes network performance. In the research, the proposed framework is utilized not only for dataset building, but also for simulating ICS attacks, ICS protocols, etc. Regarding the lightweight MITM attack detection mechanism, it is an use-case for the detection mechanism that can work effectively with the proposed framework.

MiniCPS is a small-scale framework which is composed of a few sensors and actuators. With a large number of sensors and actuators, MiniCPS is inefficient. In contrast, the SDN-based proposed framework leverages SDN architecture to programmatically manage flows and simulate a larger number of devices (switches, PLCs, actuators, etc.) with dynamic traffic control.

B. MITM Attack Scenarios

Presuming that the attacker possessed access to the internal network of manufacturing. The attacker can employ ARP Spoofing techniques to execute a MITM attack, thereby intercepting and unlawfully gathering traffic exchanged between the HMI device and the PLC devices. To execute a MITM attack that intercepts communication between the HMI and PLC1, the attacker undertakes the following steps:

- Step 1: Send a spoofed ARP response packet to compromise the HMI device. The attacker transmits a fake packet that includes the source IP address of the PLC1 device and the source MAC address of the attacker to the HMI device, tricking the HMI into seeing the packet as originating from the trusted PLC1 device.
- Step 2: Send a spoofed ARP response packet to compromise the PLC1. The attacker transmits a fake packet with the source IP address of the HMI and the source MAC address of the attacker to the PLC1 device, fooling the PLC1 into thinking the PLC1 is communicating with the legitimate HMI device.
- Step 3: All data transmitted between the HMI device and PLC1 will be redirected to the computer of the attacker. The attacker could steal or alter the data transmitted between the HMI and PLC1. To keep up the attack, it is essential to continuously transmit spoofed ARP response packets during steps 1 and 2 to the targets.

It is similar to executing a MITM attack between the HMI and other PLC devices. For the communication between the HMI and PLC devices, there are three scenarios: normal (Figure 2a), data manipulation (Figure 2b) and data sniffing (Figure 2c):

- Scenario 1 (normal): In this case, the network traffic is transmitted from HMI devices through core switches to PLC devices as usual.
- Scenario 2 (data manipulation): The attacker uses Scapy to modify control commands and device status values in order to deceive the victim.

- Scenario 3 (data sniffing): The attacker uses Scapy to create copies of the packets and forward them. To execute this scenario, the attacker must disable the `ip_forward` feature on the Linux operating system.

The proposed detection mechanism is not limited to ARP-based attacks. In LHEL, pre-processing and feature-selection stages take into account flow-level statistics (e.g., duration, byte, packet rates, and inter-arrival-time, etc.). Moreover, other ICS attacks (e.g., DNS poisoning, command injection, DoS attacks, etc.) are expected to cause characteristic deviations in such flow metrics in comparison with normal samples. Therefore, LHEL can be generalized to detect other kinds of ICS attacks.

C. MITM Attack Detection Mechanism using Machine Learning

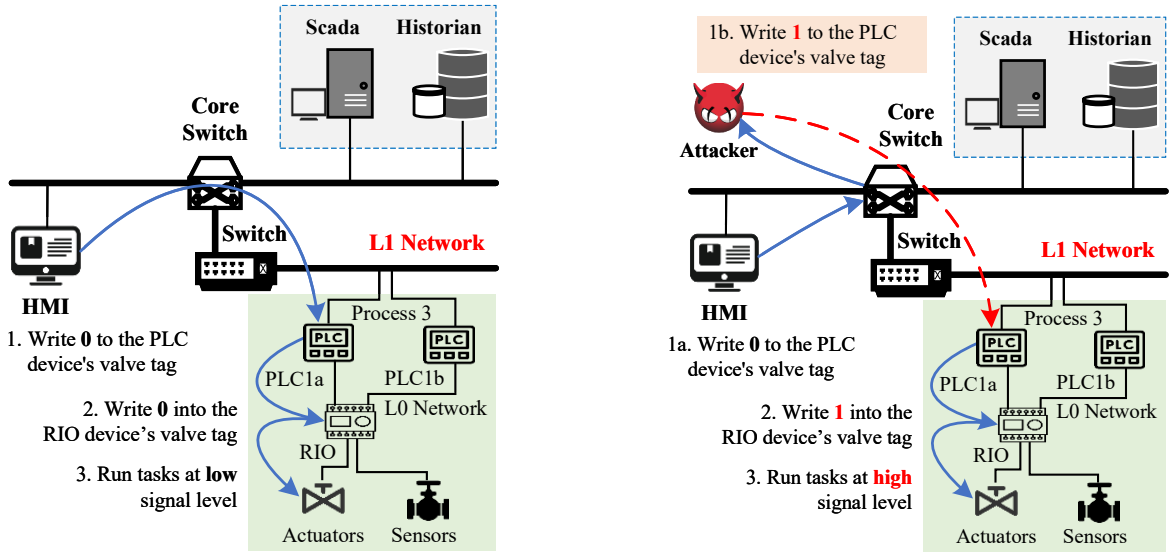
The ML-based method for identifying MITM attacks is described in this subsection, with a focus on a robust pre-processing pipeline and an LSTM-based Hybrid Ensemble Learning Mechanism (LHEL). To maintain consistency and stop information leakage, the pre-processing step standardizes data by eliminating identifiers, normalizing labels, and imputing missing values. Accuracy and computing efficiency are balanced with high-capacity and lightweight configurations in the pre-processing step through a two-stage attribute selection procedure that eliminates low-variance features and ranks the remaining features using ANOVA F-statistics. Then, the selected features are examined in LHEL, which integrates XGBoost and LSTM algorithm utilizing ensemble learning, to capture both temporal and tabular patterns for MITM attack detection. The following provides an illustration of these phases in depth.

a) Enhanced Pre-processing

The pre-processing phase standardizes semantics, cleans measurements, and selects appropriate attributes prior to model training in order to guarantee comparability between trials and ease deployment. In order to avoid models learning erroneous correlations linked to particular hosts or capture sessions instead of protocol behaviour, fields that uniquely identify flows or endpoints (e.g., IP addresses, application ports, timestamps, etc.) are eliminated. To ensure consistency across sources and scenarios, ground-truth annotations are normalized to a binary scheme (*benign* $=$ 0, *attack* $=$ 1), eliminating case and spelling variations. The empirical mean calculated over the training data for each attribute is used to impute infinite values to missing values. In features that subsequently feed variance-sensitive selectors, this decision maintains scale and prevents diminishing dispersion. To prevent information leaking, all ensuing changes are first applied to the training split and then to the validation and test data.

We use a two-phase approach to attribute selection. First, as they lack a discriminative signal, properties with zero empirical variance over the training split are eliminated. The remaining characteristics are then ranked by class separability using a univariate relevance screen based on the ANOVA F-statistic. The top-k are kept. The details of the feature selection algorithm, which integrates low-variance filtering and ANOVA F-test.

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote the feature matrix with n samples and d candidate features, and let $\mathbf{y} \in \mathbb{R}^n$ represent the corresponding class labels. The algorithm reduces dimensionality through four main operations. First, for each feature j ($1 \leq j \leq d$), the empirical variance $s_j^2 = \mathrm{Var}(\mathbf{X}_{:,j})$ is computed across all samples, and features with variance lower than a threshold τ are discarded because they contribute little discriminative power. The surviving features are then standardised to zero mean and unit variance: each value X'_{ij} is rescaled by subtracting the feature mean μ_j and dividing by its standard deviation σ_j. After this rescaling, every feature is on a comparable scale, preventing dominance by attributes with larger numerical ranges. Next, each feature is evaluated using the one-way analysis of variance (ANOVA), which computes an F-score measuring how well that feature separates the two classes. A larger F-score indicates stronger relevance for distinguishing benign traffic from MITM attacks. Finally, features are ranked by descending F-scores, and the top-k are selected to form the reduced feature matrix $\mathbf{Z} \in \mathbb{R}^{n \times k}$, with the indices of the chosen features stored in $\mathcal{S}$. In effect, the procedure discards uninformative features, rescales the rest to a common basis, scores them for class-separation ability, and retains only the most relevant subset. This ensures that the model



(a) Normal communication

(b) Data manipulation

(c) Data sniffing

Figure 2. An illustration of normal, manipulated, and sniffed data communication between HMI and PLC devices

trains on compact yet highly informative representations of network flows, improving both efficiency and robustness.

There are three options for feature selection mechanism as follows:

- S1 (without using feature selection): This option uses original features after pre-processing containing identifier removal, sanitisation, constant-feature removal and flow direction.
- S2 (using selected features): To isolate the value of timing and rate statistics, we select compact and protocol-agnostic subset of features (e.g., flow duration, byte and packet rates, inter-arrival moments in both directions) and skip flow directions. The objective is to be deployed on resource-constrained hardware.

- S3 (using selected features with positive-class augmentation): In the begining, the proposed MITM attack detection mechanism uses all features and then removes the features which create so much noise for negative-class (normal samples), reducing the number of normal samples are incorrectly classified to samples of MITM attacks.

b) LSTM-based Hybrid Ensemble Learning Algorithm

To achieve robustness under non-stationarity and reduce estimator variances, we ensemble two complementary classifier families of recurrent neural networks (via LSTM) for sequential dependencies and gradient-boosted trees (via XGBoost) for non-linear tabular structure in LHEL. Then, the results of these classifiers are aggregated using a simple late-fusion rule. LHEL starts by making several bootstrap copies of the training data. This makes sure that each base learner in the ensemble see a slightly different view of the same distribution. For the recurrent branch, a two-layer LSTM network with a hidden width of 64 is trained on each replica, followed by a linear classifier. The Adam optimizer updates the parameters at a learning rate of η, and model selection is based on early stopping with patience P epochs, which stops overfitting. A dropout rate of 0.2 adds more regularization. After training, only the parameter set that gives the best validation performance is kept. This is done N_{LSTM} times to make a diverse ensemble $\mathcal{M}_{\text{LSTM}}$.

At the same time, the tabular branch trains gradient-boosted decision trees on bootstrap replicas taken from the union of training and validation examples. Each model is fit using a logistic objective that has T boosting rounds, a shrinkage parameter η_{xgb}, and regularization terms λ and γ. The process saves N_{XGB} models in $\mathcal{M}_{\text{XGB}}$, each of which shows a different way that the bootstrap process has split and partitioned the data. These two groups of learners make up a bagged ensemble with inductive biases that work well together.

Inference is also defined by Algorithm 1. For a test input X_{test}, each LSTM model outputs a probability for the positive (attack) class, and these probabilities are aggregated across $\mathcal{M}_{\text{LSTM}}$ to form the bagged recurrent posterior. A similar averaging is done over $\mathcal{M}_{\text{XGB}}$ to get the boosted-tree posterior. To get a single ensemble score, the two family-level posteriors are combined using an unweighted arithmetic mean. This score is compared to a fixed threshold of 0.5 to make predictions. For deployments that need more accuracy, higher thresholds are available.

Algorithm 1: LSTM-based Hybrid Ensemble Learning Algorithm

Data: Test data X_{test}, trained models $\mathcal{M}_{\text{LSTM}}$, $\mathcal{M}_{\text{XGB}}$
Result: Predictions y_{pred}, probabilities p_{ens}
$p_{\text{LSTM}} \leftarrow \emptyset$
for $m_i \in \mathcal{M}_{LSTM}$ **do**
  Compute probabilities
  $p_i \leftarrow \text{softmax}(m_i(X_{\text{test}}))[:, 1]$
  Append p_i to p_{LSTM}
end
$p_{\text{LSTM}} \leftarrow \frac{1}{|\mathcal{M}_{\text{LSTM}}|} \sum_{p_i \in p_{\text{LSTM}}} p_i$
$p_{\text{XGB}} \leftarrow \emptyset$
for $m_i \in \mathcal{M}_{XGB}$ **do**
  Compute probabilities
  $p_i \leftarrow m_i(\text{flatten}(X_{\text{test}}))[:, 1]$
  Append p_i to p_{XGB}
end
$p_{\text{XGB}} \leftarrow \frac{1}{|\mathcal{M}_{\text{XGB}}|} \sum_{p_i \in p_{\text{XGB}}} p_i$
$p_{\text{ens}} \leftarrow \frac{p_{\text{LSTM}} + p_{\text{XGB}}}{2}$
$y_{\text{pred}} \leftarrow \begin{cases} 1 & \text{if } p_{\text{ens}} \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$

In other words, the LHEL framework combines the piecewise-constant decision power of boosted trees with the temporal sensitivity of recurrent networks. While late fusion balances each family's unique strengths in non-stationary scenarios, bagging reduces estimator variance within each family. The architecture is useful for intrusion detection tasks because it is easy to parallelize in practice, GPU-friendly for the recurrent branch, and computationally efficient for the boosted branch.

IV. Experimental Results

A. Experimental setup

The proposed mechanism LHEL and benchmarks are evaluated with a synthesized dataset which contains 164457 samples of normal flows and 20778 samples of MITM attacks. The synthesized dataset is built using the large-scale

ICS simulation in section IV.A. The dataset is split into three parts for training, validation, and testing using 70:15:15 splitting ratio.

The benchmarks contain various ML algorithms: Multilayer Perceptron (MLP), Decision Tree, Random Forest, XGBoost and Bagged XGBoost (Decision Tree in Random Forest algorithm is replaced by XGBoost). Besides, the benchmarks also contains DL algorithms: LSTM, Bagged LSTM (Decision Tree in Random Forest algorithm is replaced by LSTM), and bidirectional LSTM (BiLSTM). In benchmarks, Transformer-based approaches are considered with TabTransformer and SecBERT. The proposed mechanism is denoted as LHEL which integrates both LSTM and XGBoost using ensemble learning algorithm. These algorithms are evaluated using performance metrics containing: precision, recall, F1-score (F1), accuracy and ROC curve.

All experiments are implemented on a workstation running Ubuntu 20.04, equipped with an Intel Core i5-12400F CPU, 16 GB RAM, and an NVIDIA RTX 3080 GPU (The source code and dataset are published to Github for the research community https://github.com/Daocuong-main/ICS-Detectio).

B. Performance Analysis

1. What is the impact of feature selection to the MITM attack detection?

Table I indicates the performance of three kinds of feature selection in LHEL. With the option S1 (without using feature selection), LHEL can detect benign samples with 96.9% of F1. However, the recall of the MITM attack collapses to 49.8%, yielding a low MITM F1 of 66.5%. The reason is that the option S1 use all features while some of them are not effective for the MITM attack detection. Besides, the dataset is imbalanced, so the proposed mechanism learns conservative decision boundaries that protect benign precision at the cost of minority sensitivity.

With the option S2, LHEL selects appropriate features to create a proper subset using low-variance filtering, followed by ANOVA F-test for MITM attack detection, allowing for achieving good performance. In this case, the recall of the MITM attack in LHEL rises to 80.25%, and the overall accuracy increases to 98.61%. This improvement shows that curating timing and rate statistics in the option S1 reduces spurious correlations that bias the classification model, allowing the classifier to respond more consistently to attack–driven perturbations. Compared to the option S1 without using feature selection, the performance of the option S2 improves by over 4% accuracy in the considered dataset.

As for the option S3, LHEL removes the features which create so much noise for negative-class (normal samples) compared to the option S1. Removing these features can increase the performance of benign samples, and F1 of benign samples in the option S3 is slightly lower than the figure for the option S2, resulting in a slightly lower overall accuracy of 97.04%.

2. Can LHEL accurately detect MITM attacks compared to benchmarks?

Table II evaluates the performance of LHEL with benchmarks in terms of performance metrics. In this case, the feature selection with the option S2 is taken into account. The performance of ML algorithms (MLP, Decision Tree, Random Forest, XGBRF, XGBoost and Bagged XGBoost) for MITM attack detection is good, achieving over 98.3% of F1 for benign samples and above 85% of F1 for MITM attacks. As a result, the accuracy of ML algorithms is approximately 97%. Besides, there is no significant difference about performance between different ML algorithms. Although DL algorithms (Bagged LSTM, LSTM and BiLSTM) proves its effectiveness in many applications (e.g., web attack detection, vulnerability detection, etc.), the performance of DL algorithms is nearly as the same as the figure for ML algorithms. The DL algorithm achieves approximately 91.8% of average F1 and 97% of accuracy.

A noticeable feature from Table II is that the performance of Transformer-based approaches are not good. SecBERT can detect normal samples with over 97% of F1 while it cannot detect MITM attacks, resulting in low average F1 of 48.99%. As for TabTransformer, it can detect MITM attacks, but its performance is not good with 66.51% of F1 for MITM attacks. This shows a severe inductive-bias mismatch between language-pretrained transformers and low-level flow statistics without domain-appropriate

TABLE I. IMPACT OF THREE KINDS OF FEATURE SELECTION IN THE DATA PRE-PROCESSING PHASE OF LHEL.

Option	Benign			MITM Attack			Avg			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
S1	0.9403	1.0000	0.9693	1.0000	0.4979	0.6648	0.9702	0.7490	0.8170	0.9437
S2	0.9882	0.9976	**0.9929**	0.9230	0.7079	0.8025	0.9556	0.8527	0.8977	**0.9861**
S3	0.9725	0.9948	0.9835	0.9498	0.7770	0.8548	0.9611	0.8859	0.9191	0.9704

TABLE II. PERFORMANCE OF LHEL AND BENCHMARKS FOR MITM ATTACK DETECTION.

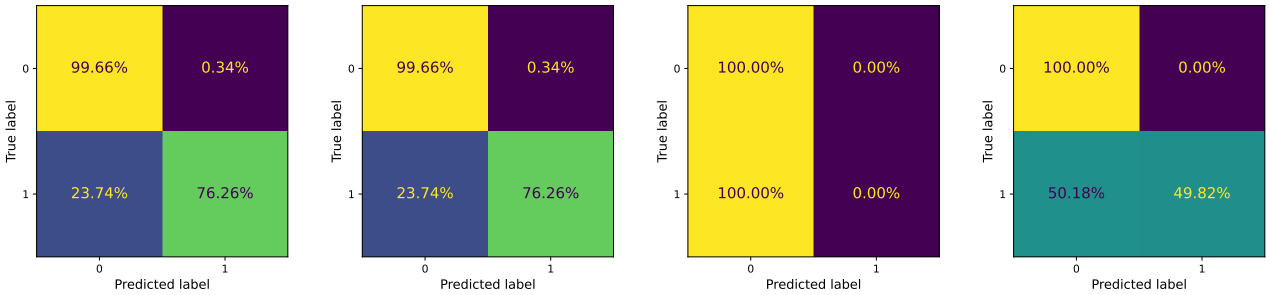
Model	Benign			MITM Attack			Avg		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
MLP	0.9710	0.9961	0.9834	0.9617	0.7648	0.8520	0.9664	0.8805	0.9177
Decision Tree	0.9709	0.9965	0.9835	0.9651	0.7632	0.8524	0.9680	0.8799	0.9179
Random Forest	0.9709	0.9965	0.9835	0.9651	0.7632	0.8524	0.9680	0.8799	0.9179
XGBoost	0.9708	0.9965	0.9835	0.9651	0.7629	0.8522	0.9680	0.8797	0.9178
Bagged XGBoost	0.9835	0.9963	0.9898	0.9651	0.7629	0.8522	0.9680	0.8797	0.9178
Bagged LSTM	0.9709	0.9965	0.9835	0.9651	0.7636	0.8526	0.9680	0.8800	0.9181
LSTM	0.9709	0.9965	0.9835	0.9651	0.7636	0.8526	0.9680	0.8800	0.9181
BiLSTM	0.9710	0.9967	0.9834	0.9100	0.7642	0.8305	0.9405	0.8805	0.9170
SecBERT	0.9606	1.0000	0.9799	0.0000	0.0000	0.0000	0.4803	0.5000	0.4899
TabTransformer	0.9404	1.0000	0.9693	1.0000	0.4982	0.6651	0.9702	0.7491	0.8172
LHEL	**0.9725**	**0.9948**	**0.9835**	**0.9498**	**0.7770**	**0.8548**	**0.9611**	**0.8859**	**0.9191**

pretraining or adaptation.

LHEL is composed of LSTM and XGBoost algorithms which are integrated using the bagging technique of ensemble learning. Concretely, the output of these algorithms is then averaged to detect MITM attacks. Therefore, LHEL can combine the advantages of both LSTM and XGBoost algorithms. As a result, the performace of LHEL is slightly higher than the figure for other algorithms. LHEL achieves 91.91% of F1, improving approximately 43, 10 and 0.2% of F1 compared to SecBERT, TabTransformer and other ML and DL algorithms.

Figure 3 shows the confusion matrix for LHEL and three representative baselines (XGBoost, SecBERT, and TabTransformer). The matrices are normalized per row, so diagonal entries correspond to recall for each class. SecBERT (Figure 3c) fails to detect any MITM traffic, assigning all flows to the benign class. The recall of beign class is 100% while the recall for MITM attacks is 0%. This explains its degeneration in F1 for MITM attacks in Table II. Despite obtaining good performance for benign samples, SecBERT is unusable in ICS contexts where minority-class detection is paramount. Concerning TabTransformer (Figure 3d), this algorithm balances performance more evenly. It accurately classifies benign instances (100% recall for beign samples) but splits MITM samples nearly evenly between the two classes ($\approx 49.8\%$ and 50.2%). This is consistent with its relatively high precision but limited recall in Table II, leading to a lower F1 for MITM attacks in comparison with LHEL and ML algorithms. As for XGBoost (Figure 3b), this algorithm achieves a slightly higher benign recall of 99.66% (false positive is 0.34%), but this algorithm sacrifices MITM sensitivity, with recall dropping to 76.26%. This visualizes the precision–recall tradeoff reported in Table II where XGBoost maximizes benign discrimination but overlooks more attacks than LHEL. Regarding LHEL (Figure 3a), the true negative rate for benign traffic reaches 99.66%, with only 0.34% false positives. In the MITM class, recall is 76.26%, yielding the lowest false negative rate (23.74%) among all models. This proves that LHEL can maintain high benign accuracy and detect more minority-class attacks, which is crucial in imbalanced ICS environments.

Figure 4 presents Receiver Operating Characteristic (ROC) curves for LHEL and benchmarks, illustrating the trade-off between the true positive rate (sensitivity) and the false positive rate at various threshold settings. Area Under Curve (AUC) values indicate the ability of these mechanisms to distinguish between benign samples and samples for MITM attacks.



(a) LHEL (b)) XGBoost (c) SecBERT (d) TabTransformer

Figure 3. Confusion matrices for LHEL and three representative baselines (XGBoost, SecBERT, and TabTransformer)

TabTransformer achieves an AUC of 0.980, and the AUC for SecBERT is approximately 0.522, indicating limited effectiveness in separating the classes. For the other algorithms (including LHEL), the AUC is impressive, achieving over 0.986.

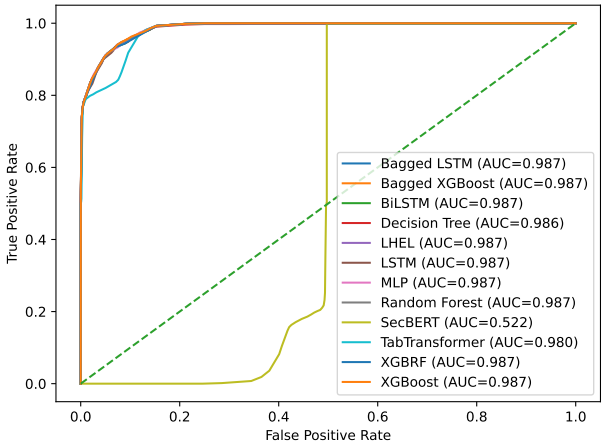


Figure 4. Receiver operating characteristic (ROC) curves for LHEL and benchmarks

Regarding the computational efficiency of LHEL, training and inference times were measured during experimentation. The total training time for the ensemble, comprising 20 LSTM models and 20 XGBoost models on the bootstrapped replicas, was approximately 733.68 seconds. This duration reflects the parallelizable nature of bagging across models, with the recurrent branch leveraging GPU acceleration for LSTM training and the boosted-tree branch executing efficiently on the CPU. For inference, the prediction time per sample-encompassing forward passes through all base models, intra-family averaging, and late fusion-was measured at 8.91 microseconds.

In this research, we evaluate the proposed MITM attack detection mechanism with different ML algorithms to select an appropriate one. LHEL improved slightly compared to other ML and DL algorithms. This improvement is not significant, but it is needed in the large-scale network. In ICS networks, there are a few thousand to a few hundred thousand network flows per second. If there are a number of incorrectly-classified samples, it can negatively impact the ICS networks (FrostyGoop malware, malware targeting Colonial Pipeline, etc.). The misclassification can cause LHEL to ignore the samples from ICS attacks or block the normal samples. Therefore, the slight improvement is meaningful for the large-scale network.

V. CONCLUSIONS

In this paper, we propose an innovative MITM attack detection framework utilizing ensemble learning algorithm to deal with two limitations: lacking of MITM attack dataset for ICS network and ineffective MITM attack detection mechanism. Concretely, we propose a novel ICS simulation framework for large-scale networks using SDN to create synthesized datasets for the research community. Besides, we propose a new lightweight MITM attack detection mechanism using an enhanced pre-procesing mechanism and a LSTM-based hybrid ensemble learning algorithm. The proposed mechanism can achieve 91.91% of f1-score and outperform benchmarks in considered dataset.

In the future, we will investigate the unlearning techniques (exact and approximate approaches, etc.) to unlearn samples which are incorrectly labeled to improve the performance of MITM attack detection.

ACKNOWNLEDGMENT

This work is supported by the Ministry of Science and Technology of Vietnam (MOST) under Grant No. KC.01.04/21-30. Cuong Dao was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2024.ThS.02.

REFERENCES

[1] MITRE, "Adversary-in-the-Middle Technique - T0830," https://attack.mitre.org/techniques/T0830/, 2020, accessed: 2025-06-06.

[2] Z. Hill, J. Hale, M. Papa, and P. Hawrylak, "Using bro with a simulation model to detect cyber-physical attacks in a nuclear reactor," in *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*. IEEE, 2019, pp. 22–27.

[3] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. Wang, "Intrusion detection system for iec 60870-5-104 based scada networks," in *2013 IEEE power & energy society general meeting*. Ieee, 2013, pp. 1–5.

[4] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, and H. Wang, "Multiattribute scada-specific intrusion detection system for power networks," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092–1102, 2014.

[5] P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Man-in-the-middle attacks and defence in a power system cyber-physical testbed," *IET Cyber-Physical Systems: Theory & Applications*, vol. 6, no. 3, pp. 164–177, 2021.

[6] M. F. Elrawy, L. Hadjidemetriou, C. Laoudias, and M. K. Michael, "Detecting and classifying man-in-the-middle attacks in the private area network of smart grids," *Sustainable Energy, Grids and Networks*, vol. 36, p. 101167, 2023.

[7] D. J. S. Raja, R. Sriranjani, P. Arulmozhi, and N. Hemavathi, "Unified random forest and hybrid bat optimization based man-in-the-middle attack detection in advanced metering infrastructure," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.

[8] O. Eigner, P. Kreimel, and P. Tavolato, "Detection of man-in-the-middle attacks on industrial control networks," in *2016 International Conference on Software Security and Assurance (ICSSA)*. IEEE, 2016, pp. 64–69.

[9] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," in *Proceedings of the First ACM workshop on cyber-physical systems-security and/or privacy*, 2015, pp. 91–100.

[10] A. Ashok, P. Wang, M. Brown, and M. Govindarasu, "Experimental evaluation of cyber attacks on automatic generation control using a cps security testbed," 07 2015, pp. 1–5.

[11] Y. Yang, K. Mclaughlin, T. Littler, S. Sezer, E. G. Im, Z. Yao, B. Pranggono, and H. Wang, "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid scada systems," vol. 2012, 09 2012, pp. 1–8.

[12] Y. Yang, L. Gao, Y.-B. Yuan, K. Mclaughlin, S. Sezer, and Y.-F. Gong, "Multidimensional intrusion detection system for iec 61850 based scada networks," *IEEE Transactions on Power Delivery*, vol. 32, 01 2016.

[13] A. P. Mathur and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ics security," in *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 2016, pp. 31–36.

[14] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: a water distribution testbed for research in the design of secure cyber physical systems." New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3055366.3055375

[15] L. Deri, M. Martinelli, and A. Cardigliano, "Realtime high-speed network traffic monitoring using ntopng," in *28th large installation system administration conference (LISA14)*, 2014, pp. 78–88.

ABOUT THE AUTHORS

Nguyen Tuan Anh
Workplace: People's Security Academy
Email: anh.nt240075d@sis.hust.edu.vn
Education: Nguyen Tuan Anh, M.Sc. (2024), is currently a faculty member at People's Security Academy.
His recent research interests include machine learning, and network security.

Tên tác giả: Nguyễn Tuấn Anh
Cơ quan công tác: Học viện An ninh nhân dân
Email: anh.nt240075d@sis.hust.edu.vn
Quá trình đào tạo: Nhận bằng Thạc sĩ năm 2024 và hiện là giảng viên tại Học viện An ninh nhân dân.
Hướng nghiên cứu hiện nay: Học máy, An ninh mạng.

Le Van Dong
Workplace: SoICT, HUST
Email: dong.levan1@hust.edu.vn
Education: Le Van Dong, M.Sc. (2022), is currently a faculty member at the School of Information and Communication Technology, Hanoi University of Science and Technology.
His research interests include information security, and digital forensics.

Tên tác giả: Lê Văn Đồng
Cơ quan công tác: Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội
Email: dong.levan1@hust.edu.vn
Quá trình đào tạo: Nhận bằng Thạc sĩ năm 2022 và hiện là trợ giảng tại Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội.
Hướng nghiên cứu hiện nay: Bảo mật mạng, Điều tra số.

Dao Viet Cuong
Workplace: SoICT, HUST
Email: cuong.dv241037m@sis.hust.edu.vn
Education: Cuong Dao received the B.Eng. degree in computer science from Hanoi University of Civil Engineering in 2023. He is currently pursuing the M.S. degree in Computer Engineering at the School of Information and Communication Technology, Hanoi University of Science and Technology.
His research interests include network security, artificial intelligence, and blockchain vulnerability detection.

Tên tác giả: **Đào Việt Cường**
Cơ quan công tác: Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội, Việt Nam
Email: cuong.dv241037m@sis.hust.edu.vn
Quá trình đào tạo: Nhận bằng Kỹ sư ngành Khoa học Máy tính tại Trường Đại học Xây dựng Hà Nội năm 2023. Hiện đang là học viên cao học ngành Kỹ thuật Máy tính tại Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội.
Hướng nghiên cứu hiện nay: An ninh mạng, Trí tuệ nhân tạo, Phát hiện lỗ hổng trong mạng Blockchain.

Nguyen Dinh Nghia
Workplace: People' Security Academy
Email: nghiahvan@gmail.com
Education: Nghia Nguyen Dinh received the Ph.D. degree in 2020 and became an Associate Professor in 2024. He is currently a faculty member at People's Security Academy.
His research interests include cybersecurity, network security and information security.

Tên tác giả: **Nguyễn Đình nghĩa**
Cơ quan công tác: Học viên An ninh nhân dân
Email: nghiahvan@gmail.com
Quá trình đào tạo: Nhận bằng Tiến sĩ năm 2020, được phong Phó giáo sư năm 2024, hiện công tác tại Học viện an ninh nhân dân
Hướng nghiên cứu hiện nay: An ninh mạng, an toàn thông tin.

Tran Quang Duc
Workplace: SoICT, HUST
Email: ductq@soict.hust.edu.vn
Education: Duc Tran received the Ph.D. degree in 2015 and became an Associate Professor in 2020. He is currently a faculty member at the School of Information and Communication Technology, Hanoi University of Science and Technology.
His research interests include network security, multimedia data security, biometric-based authentication systems, digital image processing, and vulnerability detection.

Tên tác giả: **Trần Quang Đức**
Cơ quan công tác: Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội, Việt Nam
Email: ductq@soict.hust.edu.vn
Quá trình đào tạo: Nhận bằng Tiến sĩ năm 2015, được phong Phó Giáo sư năm 2020, hiện công tác tại Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội.
Hướng nghiên cứu hiện nay: An ninh mạng, Bảo mật dữ liệu đa phương tiện, Hệ thống xác thực dựa trên sinh trắc học, Xử lý ảnh số, Khai thác lỗ hổng.