Lightweight Privacy-Preserving Federated Deep Intrusion Detection for Industrial Cyber-Physical System

Imtiaz Ali Soomro, Hamood ur Rehman Khan, Syed Jawad Hussain, Zeeshan Ashraf, Mrim M. Alnfiai, and Nouf Nawar Alotaibi

***Abstract*—The emergence of Industry 4.0 entails extensive reliance on industrial cyber-physical systems (ICPS). ICPS promises to revolutionize industries by fusing physical systems with computational functionality. However, this potential increase in ICPS makes them prone to cyber threats, necessitating effective intrusion detection systems (IDS) systems. Privacy provision, system complexity, and system scalability are major challenges in IDS research. We present FedSecureIDS, a novel lightweight federated deep intrusion detection system that combines CNNs, LSTMs, MLPs, and federated learning (FL) to overcome these challenges. FedSecureIDS solves major security issues, namely eavesdropping and man-in-the-middle attacks, by employing a simple protocol for symmetric session key exchange and mutual authentication. Our Experimental results demonstrate that the proposed method is effective with an accuracy of 98.68%, precision of 98.78%, recall of 98.64%, and an F1-score of 99.05% with different edge devices. The model is similarly performed in conventional centralized IDS models. We also carry out formal security evaluations to confirm the resistance of the proposed framework to known attacks and provisioning of high data privacy and security.**

***Index Terms*—Federated learning, industrial cyber-physical systems, Internet of things, intrusion detection system, symmetric key.**

I. Introduction

INDUSTRIAL cyber-physical systems (ICPS) aim to establish remote connections between industrial physical systems and control systems by incorporating cyber components [1]. These systems consist of coupled sensors and actuators. ICPS provides real-time monitoring, control, and automation of industrial operations, boosting efficiency and productivity and enabling diverse and versatile services and capabilities. These include smart transportation, intelligent manufacturing, medical systems, and other kinds of industrial automation [2], [3]. Despite advantages, the increased interconnectivity of ICPS exposes them to significant security threats [4], such as unauthorized access, denial of service (DoS) attacks, man-in-the-middle (MITM) attacks, and data breaches.

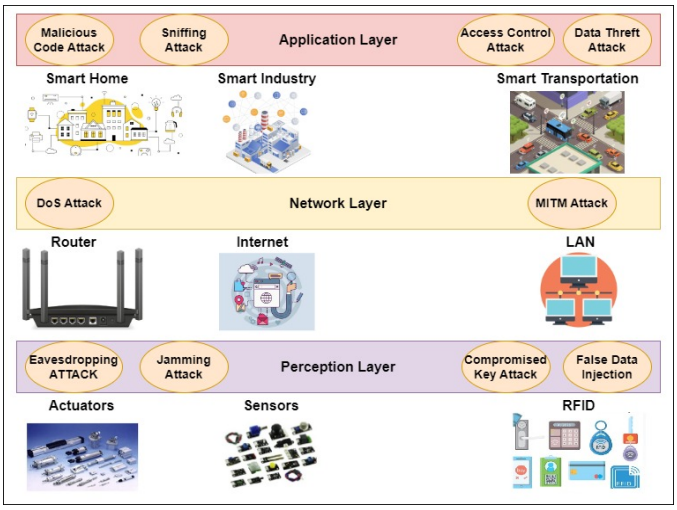

Fig. 1. ICPS architecture and attack types.

IDS plays a crucial role in identifying and alleviating potential security threats while safeguarding the integrity and dependability of industrial operations [5]. IDS systems typically operate by centralized learning and information sharing. Conventional information-sharing methods dealing with raw data can impede the viability of centralized learning, particularly when it comes to sensitive data. This constraint is more pronounced when confronted with the difficulties linked to sharing raw data with external entities. Sharing raw data can present challenges due to various data management regulations and privacy considerations [6]. Individuals may also be hesitant to disclose their information, further complicating the situation [7].

ICPS fundamentally operates at three layers: Perception, transmission, and application [8]. Each layer is defined by the type of devices within it and the related functions that should be implemented [9]. Based on the functions achieved at each layer, this paper considers a three-level CPS architecture with different attack types as shown in Fig. 1.

Federated learning (FL) provides a potential solution to ad-

Manuscript received April 1, 2024; revised September 19, 2024; approved for publication by Paek, Jeongyeup Division 3 Editor, September 26, 2024.
This research was funded by Taif University, Saudi Arabia, Project No. (TU-DSPP-2024-41).
I. A. Soomro, S. J. Hussain, and H. Khan are with the Sir Syed CASE Institute of Technology, Islamabad, Pakistan, email: imtiazalisoomro@gmail.com, Jawad.hussain@case.edu.pk, hamood.rehman@carepvtltd.com.
Z. Ashraf is with the Department of Computing and Information Technology, IISAT, Gujranwala, Pakistan, email: zeeshan.ashraf@ieee.org.
M. M. Alnfiai is with the Department of Information Technology, College of Computers and Information Technology, Taif University, Taif P.O. Box 11099, Taif, 21944, Saudi Arabia, email: m.alnofiee@tu.edu.sa.
N. N. Alotaibi is with the Department of Special Education, College of Education, Najran University, Saudi Arabia, email: nnalotaibi@nu.edu.sa.
Z. Ashraf and H. Khan are corresponding authors.
Digital Object Identifier: 10.23919/JCN.2024.000054

1229-2370/24/$10.00 © 2024 KICS

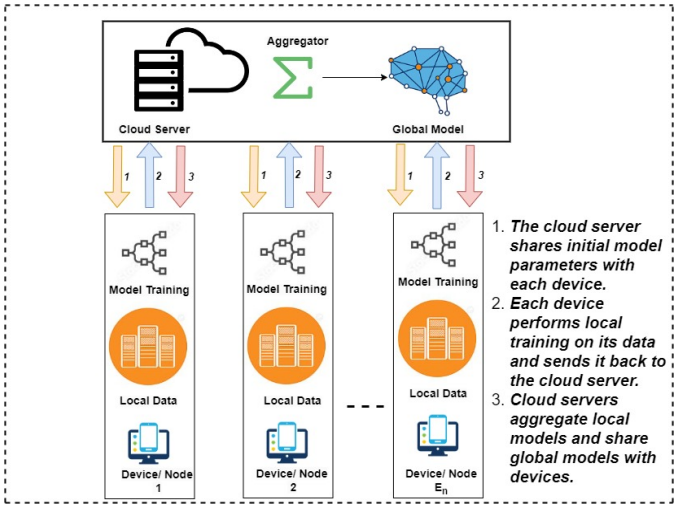

Fig. 2. Federated fearning training process.

dressing data privacy and confidentiality concerns by serving as a decentralized platform for machine learning (ML). FL allows for the creation of an effective IDS for IoT devices through training while maintaining privacy. Clients conduct local training and send their models to a central server, which combines them to form a global IDS model. The FL training procedure consists of various stages, such as client selection, local model training, and global model aggregation, as illustrated in Fig. 2.

Although FL-based IDS has been widely adopted for its privacy-enhancing features, it is vulnerable to data leakage attacks launched by malicious clients [11]. Recent research shows that a malevolent device can introduce backdoors or replace the global FL model [12]–[14]. The issue of potential private user data extraction from gradients in FL has garnered more attention in the data security and AI ethics field, commonly known as the data leakage problem [15]–[17]. It is becoming more and more crucial to tackle possible security risks as the adoption of FL-based IDS is growing in popularity. One notable threat is the MITM attack, which presents a considerable risk to the integrity of FL-based IDS. In this attack, the attacker strategically places themselves between the communication channels of the device and the central server, as illustrated in Fig. 3. The attacker can exploit vulnerabilities to intercept, manipulate, or infiltrate malicious data and thus jeopardize the quality and data protection guarantees of FL-based IDS. Therefore, addressing the potential security threats is crucial for the success of FL-based IDS in ensuring data privacy and security.

To address the challenges of safeguarding ICPS, we propose FedSecureIDS, a novel FL-based IDS. FedSecureIDS enhances security by enabling multiple distributed entities to collaboratively train a global intrusion detection model without sharing sensitive data, thereby preserving data privacy.

Imagine a smart manufacturing factory where FedSecureIDS operates within an ICPS environment. The factory employs various interconnected devices, including sensors on machinery, actuators controlling robotic arms, and a central control system managing the entire production line. During normal operation, sensors continuously monitor machinery performance, collecting data on temperature, vibration, and operational speed. Actuators adjust machinery parameters based on control system commands to ensure optimal performance, while the control system processes sensor data in real-time, optimizing operations and preventing potential issues.

In the above scenario, a potential threat could arise when an attacker attempts an MITM attack to intercept and alter sensor readings, causing the control system to make incorrect decisions that could damage machinery or halt production. FedSecureIDS counteracts such threats through a multi-step process. First, data is collected by sensors on the factory floor and continuously sent to local edge devices for initial processing. Local intrusion detection modules analyze the data using long short-term memory (LSTM) and convolutional neural networks (CNN) models to identify any anomalies indicative of an MITM attack. The local models on each edge device are periodically updated and sent to the FL Coordinator on the cloud server. This coordinator aggregates the local models to update the global intrusion detection model without accessing raw data, thus preserving data privacy. The updated global model is then distributed back to the edge devices, enhancing their capability to detect sophisticated cyber threats. Upon detecting anomalies, the system triggers alerts to the control system, which initiates predefined countermeasures such as isolating affected components and verifying data integrity through secure communication protocols.

FedSecureIDS ensures resilience and privacy-preservation by leveraging FL, allowing collaborative training without sharing sensitive data. This approach not only enhances detection accuracy but also ensures data confidentiality, protecting the factory's operations from evolving cyber threats.

A. *Motivation*

- Motivated by the need to address the existing limitations of data leakage attacks on FL, our research endeavors to enhance the security landscape by proposing innovative countermeasures.
- Our research is driven by the complex threat scenario in ICPS, where the presence of diverse cyber threats demands an advanced IDS to effectively tackle challenges such as denial-of-service, reconnaissance, tampering, exploitation, and weaponization attacks.
- In ICPS, mobile devices have to operate with limited computational resources. Our research is prompted by the imperative for an efficient security model. Thus, we strive for a lightweight solution to the problem.
- The need to strengthen security systems is brought into further consideration by the increasing number of MITM and eavesdropping attacks being reported.

B. *Contributions*

Our contributions main contributions are as follows:

- We propose advanced countermeasures in the FedSecureIDS model to enhance data protection during FL, addressing the vulnerabilities associated with data leakage.
- We introduce a novel, FL-based hybrid IDS designed for ICPS (FedSecureIDS). This model seamlessly incorporates diverse components to proficiently identify

cyber threats, including DoS attacks, reconnaissance attacks, tampering incidents, exploitation attempts, and weaponization attacks, demonstrating a noteworthy increase in detection accuracy relative to centralized ML-based methods.
- Our study introduces a lightweight and self-validated symmetric session key exchange algorithm tailored for resource-constrained devices. The algorithm effectively produces session keys that ensure secure communication while meeting these constraints.
- Our research develops a robust cryptographic algorithm that uses self-authentication and a pre-shared session key with advanced encryption standard (AES) to ensure secure communication and data confidentiality, effectively countering eavesdropping.
- With the help of BAN logic and the AVISPA tool, we provide both formal and informal security proofs for our proposed model.

The rest of the paper is structured as follows. Section II delivers a comprehensive review of existing research on IDS designed for ICPS, with a particular focus on IDS techniques leveraging FL. Section III outlines the system model assumptions and the threat model employed in this work. Section IV then delves into the methodology underpinning the proposed framework. The experimental setup is outlined in Section V, detailing the implementation of our proposed model, the datasets used for evaluation, and the metrics employed to assess the system's performance. A thorough security analysis of this framework is subsequently provided in Section VI. Section VII presents the experimental results and a discussion of their implications. Finally, Section VIII concludes the paper by summarizing the key findings and outlining potential avenues for future research.

II. Related Works

A. Anomaly-based IDS

In their work [18], the authors proposed a framework for identifying anomalies in cyber-physical systems (CPS). This framework involves a group of agents representing virtual digital shells of assets in the production line. These agents collect data from the assets, and a central agent, functioning as middleware, employs a learning algorithm to detect anomalous behaviour in the others. The framework is specifically designed for tasks related to anomaly identification, where intelligent agents oversee real-time data collection, analysis, and processing, while ML models, termed predictive models, are integrated into the multi-agent system to forecast the typical state of the cyber-physical system. The study presented in [19] introduces a methodology for detecting anomalies in cyber-physical systems through the application of long short-term memory recurrent neural networks (LSTM-RNN). By employing LSTM-RNN to forecast data series, the approach addresses the temporal nature of anomalies and cyber-attacks, allowing for the correlation of time-series data over intervals. The model characterizes normal behaviour, serves as a predictor, and integrates the cumulative sum approach to identify anomalous behaviour with a low false positive rate. This method is beneficial in real-world CPS applications, where instances of unusual behaviour are infrequent. Notably, the research leverages a modest dataset collected from a single component of the system for training and validation, a practical consideration dictated by limited resources and infrastructure.

In the research detailed in [20], the authors put forth a fresh perspective on identifying replay attacks within cyber-physical systems. This involves a comprehensive assessment of system stability coupled with the implementation of lossless watermarking techniques. Each agent in this framework is equipped with both a local estimator and an anomaly detector. The paper also introduces a method for detecting adversaries by incorporating a watermarked control approach, distributing the watermarking signal among the various agents in the network. However, it's noteworthy that the incorporation of watermarked signals has been observed to potentially diminish the overall performance of the network.

In [21], [22] researchers proposed an ML-based approach for detecting and mitigating communication threats in cooperative autonomous car applications. The methodology revolves around constructing models with the ability to comprehend typical behaviour patterns linked to benign V2X communication. It aims to discern abnormal behaviour, enabling the identification of potentially hazardous communication instances. The method is utilized in various cooperative autonomous vehicular situations, such as platooning, cooperative adaptive cruise control, intelligent intersection detection, and dynamic cooperative route management. However, the architecture only takes into account attacks that are carried out through a single communication channel.

To detect potential dangers of fake data injection in smart grid networks, the authors of [23] put up a real-time detection method. An analytical model was created using the adaptive CUMSUM algorithm to build a detection system that adheres to essential criteria for ensuring performance assurance. This method is capable of detecting fake data attacks even in cases where the probability density function post-change is not known. In [24], the authors described an alternative approach where a real-time detection method was implemented to combat denial-of-service (DoS) attacks. The fuzzy IDS effectively distinguishes between regular and malicious network traffic under uncertainty. The system demonstrates an impressive detection rate of 99.9% over a dataset of over 5 million test sessions, with a low false alarm rate of approximately 1600. The method is based on a limited number of features to identify DoS flooding attacks. In [25], a comprehensive analysis of distributed denial of service attacks (DDoS), intrusion tolerance, and various mitigation techniques is provided. The paper then explores the impact of DDoS attacks on cloud-based services, highlighting how attackers can leverage the distributed nature of the cloud to amplify the effects of their attacks. Additionally, the authors examine the characteristics of different types of DDoS attacks, such as the use of botnets to generate large volumes of traffic. The core of the paper focuses on evaluating various mitigation techniques, including rate limiting, traffic filtering, and intrusion tolerance strategies. The authors provide a detailed assessment of the strengths and

weaknesses of each approach due to a need for multifaceted defense against the evolving threat of DDoS attacks.

B. FL for Anomaly-based IDS

To distribute the workload and improve scalability, FL uses a server-client design that allows computing at both ends. Within this paradigm, [26] presents FL models for IDS that use multi-layer perceptron (MLP) and autoencoders. The efficacy of these frameworks was assessed by conducting tests with FL, distributed, and centralized systems and then by using rigorous statistical analytics.

A unique mitigation architecture called FLEAM, which leverages FL, was presented in [28] to combat massive DDoS attacks within the context of the industrial Internet of things (IIoT). Through the integration of FLEAM, fog computing, and cloud computing, the workload is distributed, which increases the effectiveness of the mitigation process. Each fog node in the network receives a pre-trained model, does local retraining, and then uploads the modified model parameters to the cloud as part of the operational workflow. In the research conducted by [29], a semi-supervised learning technique grounded in disagreement is introduced for collaborative IDS, employing FL to address limitations related to data size. Iterative model updates are made following each labeling phase in the suggested method, which entails training three classifiers and using a majority vote mechanism to give labels to unlabeled data. In terms of prediction accuracy and false alarm minimization, comparative evaluations show that their approach performs better than supervised ML techniques. A unique technique for deep learning (DL) training was suggested by the researchers in [30]. This approach utilized a privacy-preserving federated architecture, which capitalized on the heterogeneity of real-time intrusion data. Researchers in this study used a method called MT-DNN-FL to identify outliers using the same model and data. By using this method, it became easier to pinpoint and classify VPN traffic. This method not only lowered the amount of energy required for communication and the expenses associated with training, but it also demonstrated superior performance in comparison to other single-task learning methods.

C. Cryptographic based IDS

The most efficacious measures for countering a diverse range of threats, including MITM attacks, unauthorized access to agents, provenance attacks, and tampering with agent logs, are represented by the encryption and authentication of sensitive data. In [31], a message-oriented middleware was proposed to improve communication performance in multi-agent systems. The authors introduced a new component called the certification authority service, which generates certificates for agents to authenticate their identities and securely transmit messages in the architecture for cyber-physical systems. However, the technique falls short of addressing the concern regarding an agent's discretion in placing trust in a particular message sender, thereby leaving room for potential vulnerabilities. In [32]–[38], the researchers used asymmetric key-based cryptosystems to provide security services such as privacy-preservation and data confidentiality. Most of these researches used asymmetric key-based Pallier cryptosystems for data confidentiality. Authentication services were also provided by asymmetric key-based cryptosystems. This results in increased computation and communication costs, making these cryptosystems heavyweight for smart devices. In asymmetric cryptosystems, the size of the secret keys is large, increasing key generation time.

Our proposed model, FedSecureIDS, differs from existing FL for anomaly-based IDS and cryptographic-based IDS in the following ways. Traditional FL models often rely solely on a complex single architecture, such as an MLP, an autoencoder, or some other semi-supervised learning method that faces challenges like data leakage and high computational overhead. In contrast, FedSecureIDS integrates CNNs, LSTMs, and MLPs to enhance detection accuracy while utilizing a lightweight symmetric cryptographic approach to ensure data privacy. Also, unlike conventional cryptographic-based IDS that rely on symmetric key systems with high computational costs due to complex key exchange algorithms, FedSecureIDS utilizes a lightweight symmetric key exchange mechanism to deliver security services, encompassing authentication and data confidentiality. Symmetric key cryptosystems utilize smaller secret keys in comparison to asymmetric keys while ensuring an equivalent level of security. As per the National Institute of Standards and Technology (NIST), the strength of a 128-bit symmetric key is deemed comparable to that of a 3072-bit asymmetric key [39]. This approach thus reduces overhead and delivers robust security for resource-constrained devices within ICPS. The robustness of the symmetric key-based cryptosystem for mobile server-client communication has been formally and informally proved against several known attacks [40].

Table I presents a comparative analysis of the different IDS methodologies cited above, detailing their core techniques and strengths. This systematic comparison highlights the trade-offs and effectiveness inherent in various IDS approaches.

III. System Model, Assumptions, and Threat Model

The system model, assumptions, and threat model used in our proposed collaborative learning framework for ICPS are presented. The system model includes edge devices, cloud servers/aggregators, and a registration authority (R.A.) We assume that the edge devices have partial trustworthiness and that the R.A. and cloud servers are completely trustworthy. The threat model accounts for potential risks in ICPS and FL, including attacks such as MITM attacks, DDoS attacks, eavesdropping, membership inference attacks, and unintended data leakage attacks.

A. ICPS based System Model

We describe both the ICPS and our proposed FL framework within ICPS. The ICPS model involves remote users, IoT-based sensor nodes, gateways, and access points collaborating to ensure secure data exchange, thereby optimizing industrial operations. Our FL framework, illustrated in Fig. 4, strategically integrates edge devices, cloud servers/aggregators, and the R.A. This tailored system model addresses inherent challenges and risks in FL within ICPS,

TABLE I
COMPARISON OF VARIOUS IDS APPROACHES.

Study	Approach	Key techniques	Strengths	Limitations
[18]	Anomaly-based IDS	Multi-agent system	Real-time data collection and analysis, adaptability	Limited to specific types of anomalies, potential scalability issues
[19]	Anomaly detection	LSTM RNN	Low false positive rate, effective in sequential data forecasting	Requires large datasets for optimal performance, training complexity
[20]	Replay attack detection	Stability analysis, watermarking	Robust against replay and other forms of attack	Potential performance degradation from watermarking, implementation overhead
[21]	Signature-based IDS	Pattern matching	High detection rates for known threats	Ineffective against zero-day attacks, dependence on signature updates
[22]	Hybrid IDS	ML and rule-based methods	Combines the strengths of both techniques for improved detection	Higher computational cost, configuration complexity
[23]	FL for IDS	Homomorphic encryption	Strong privacy guarantees, decentralized model training	High computational and communication overhead, potential bottlenecks
[26]	FL for IDS	MLP, autoencoders	Superior accuracy compared to centralized systems, data privacy	Scalability not addressed, communication efficiency could be improved
[27]	Mitigation of DDoS attacks	FLEAM architecture	Significant reduction in mitigation delay, enhanced accuracy	Complexity in integration with existing systems, requires robust infrastructure
[28]	Blockchain-based IDS	Distributed ledger technology	Provides transparency and traceability for IDS actions	Overhead in terms of latency and resource usage, complex implementation
[29]	Semi-supervised learning	Disagreement-based FL	Better performance than traditional supervised methods in low-label scenarios	Relies on adequate initial labeled data, might not generalize across domains
[30]	Federated reinforcement learning	Reinforcement learning frameworks in a FL setting	Adaptive learning in dynamic environments	Complexity in policy convergence, high communication costs
[31]	Cryptographic-based IDS	Asymmetric key cryptosystems	Strong privacy and data confidentiality	High computational costs and key management complexity
Proposed model (FedSecureIDS)	Lightweight federated deep IDS	CNN, LSTM, MLP	Enhanced detection accuracy, privacy-preserving mechanisms, lightweight cryptography	Scalability in large networks not explicitly addressed, further testing needed in diverse environments

specifically focusing on privacy and security threats, providing a holistic approach to a solution.

1) *Edge devices*: The proposed FL framework consists of several edge devices, each representing a node that participates in the FL process. These devices collect local data from their respective sensors and use it to build a local machine-learning model.
2) *Cloud server*: The FL framework incorporates a cloud server with dual functions. In the beginning, it will initialize the parameters of the global model and then send them to the edge devices. Next, it takes the parameters that were provided by the edge devices and combines them until the model converges. After that, it sends the updated model back to the edge devices. These operations play a pivotal role in constructing an IDS model through FL.
3) *Registration authority*: Besides its role as a central aggregator, the cloud server serves as a registration authority that maintains a database of registered agents and their associated information. The R.A. facilitates secure communication between registered agents by distributing symmetric session keys. It is responsible for verifying the authenticity of registered agents and establishing a secure communication channel between them.

B. Assumptions

Our model operates under the following assumptions: Firstly, the cloud server is regarded as a trustworthy entity. The R.A. is assumed to be fail-safe in ensuring that communication between edge devices and the cloud server is safe and reliable. The edge devices are regarded as somewhat trustworthy, suggesting that they will follow the established protocol but may also have an interest in the data resources of other edge devices.

C. Adversary Model

The Dolev-Yao adversary model [46] was chosen for this research. In computer and network security, the Dolev-Yao adversary model is a popular formal model that describes the actions of an attacker who controls the entire communication network. In the Dolev-Yao model, the adversary is assumed

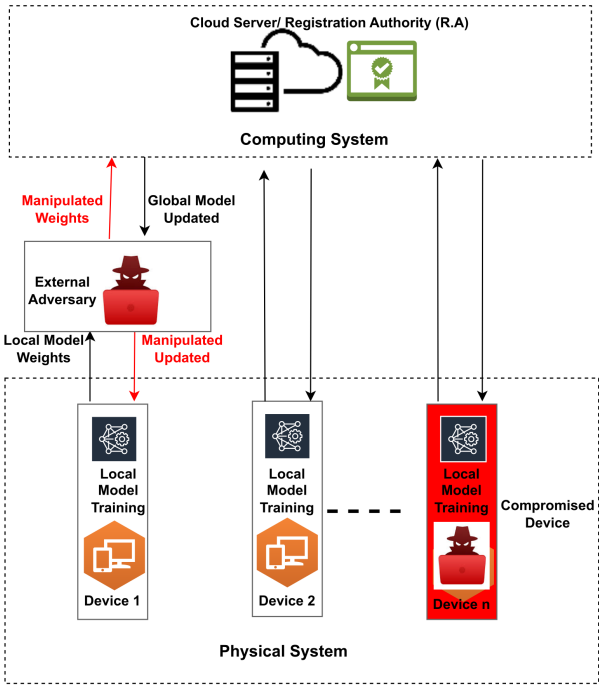

Fig. 3. System model.

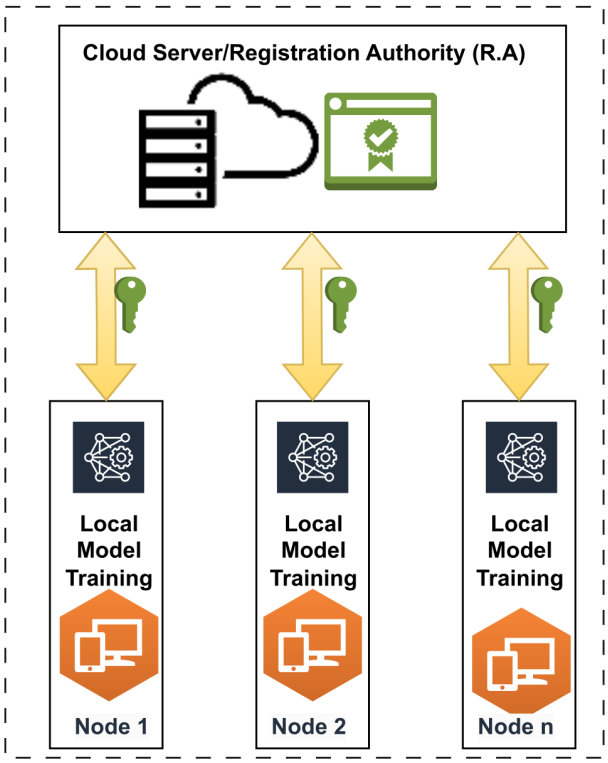

Fig. 4. Secure federated learning work flow diagram.

to be an attacker who can intercept, modify, delete, and insert messages in the communication channels between the participants of the protocol. In addition to this, it is presumed that the adversary possesses a limitless amount of processing power and is familiar with the whole protocol specification. The Dolev-Yao model operates under the presumption that messages are sent via a secure channel that is guarded by encryption methods. In an MITM attack, the adversary may gain access to encryption keys by intercepting the communication between parties or through other means, allowing them to decrypt and potentially manipulate the exchanged data. The adversary can also eavesdrop on the communication channels and observe the behaviour of the participants. The Dolev-Yao model is widely used in the analysis of security for cryptographic protocols, providing researchers with a means to identify and address potential vulnerabilities before they can be exploited by real attackers. Furthermore, we have assumed that the adversary would be unable to retrieve the data contained in the server's database due to the secure nature of the R.A.'s database.

IV. Methodology

A. Data Collection and Pre-processing

In our study, we utilized the X-IIoTID dataset to ensure the generalizability of our proposed methods. The dataset includes various industrial-related data, enabling us to validate our FL-based IDS across different industrial settings with diverse processes, devices, and network configurations.

For model training and testing, we partitioned the dataset into 80% for training and 20% for testing. The training data was evenly distributed among the smart devices using it to train their local models. To address variations in data strengths across clients, a scaling factor was incorporated into the federated framework to ensure fair consideration of all client data while accounting for uneven data distribution.

We employed a multi-stage data preprocessing pipeline to ensure the quality and consistency of the dataset used to train and evaluate FedSecureIDS. This involved addressing missing values, handling categorical features, normalizing numerical features, and addressing class imbalance. To prepare data for ML algorithms, all features must be converted into numerical values. One approach to accomplish this involves label encoding for features in string format. Once the label encoding is done, the next step is to use the min-max normalization technique to normalize all feature values [47]. This technique scales the minimum feature value to 0 and the maximum value to 1. Normalization ensures that all feature values fall within a similar range, aiding in faster model convergence. After normalization, the feature values lie within a range of 0 and 1. In the proposed model, the pre-processing unit serves as the first stage, where the data is transformed for the input layer. This approach enhances the robustness and applicability of our IDS framework, ensuring its effectiveness in real-world industrial scenarios. These preprocessing steps are in line with established practices in [48]–[50].

B. Proposed FedSecIDS Framework

The proposed framework, FedSecureIDS, is designed to enhance cybersecurity within ICPS by leveraging FL to develop a privacy-preserving IDS. This FL-based IDS enables multiple distributed entities to collaboratively train a global anomaly detection model without sharing sensitive data, ensuring robust data privacy. The framework comprises several key

components, including data collection agents, local intrusion detection modules, an FL coordinator, and a security manager. The proposed framework comprises two integral components. Firstly, we develop an FL-based IDS capable of addressing various types of attacks, including DDoS attacks. In the second phase, we focus on designing a robust communication protocol to ensure secure and resilient data exchange within the system. This dual-phase approach aims to enhance the overall cybersecurity posture by proactively detecting and mitigating potential threats, coupled with the implementation of a secure communication infrastructure. The complete FL framework technique, which permits collaborative learning among multiple agents, is described in Algorithm 1. The system enables the building of a strong and comprehensive FL-based IDS model that is trained on numerous data resources. This approach enhances the performance of DL models and enables the building of a highly effective IDS. The proposed privacy-preserving FL framework comprises several phases, as shown in Fig. 4, and details are described below.

1) In the first phase of the proposed system, the registered agents connect to the cloud server which also works as a registration authority. The agent first exchanges a session symmetric key with the server described in Section IV-E. Then, in order to confirm the validity of the registered agent, an authentication procedure is carried out on both ends using a pre-shared session key. If the authentication completes and the connection is established, an agent will send or receive secure model parameters to the server. If the authentication fails, the connection immediately terminates.
2) The cloud server sends the first set of model parameters, which include batch size, learning rate, and loss function, to the edge devices. At the same time, the server receives data size information from every edge device to determine their respective contribution ratios. The weight given to the gradient updates from each edge device during model training is determined through the utilization of the contribution ratio.
3) The IDS model parameters acquired from the cloud server are utilized to train the IDS model using the local dataset D_i from each edge device E_n. The IDS model utilized for training is a hybrid CNN, LSTM, and MLP model, as detailed in Section IV-D. Algorithm 1 explains the stages for local model training, gradient calculation, and uploading to the cloud server. The technique also outlines the cloud server's mechanism for collecting model gradients and updating global model parameters.
4) The edge devices in the suggested architecture use symmetric key encryption to secure the model's gradients W_r after training it on their local data D_i. It is represented by W_r that the model's gradients are in the Rth round, following training at edge device E_n. The whole global model is then constructed by aggregating the generated encrypted gradients $E(W_r)$ in the cloud.
5) Each local device updates its model and sends it back to the central server once training is complete. The server then compiles all of the modifications and uses them to create a new global model.

Algorithm 1 Federated learning algorithm

Input: Number of clients devices E_n, number of training rounds r, learning rate η
Initialize global model w_0
for each round $r = 1, \cdots, R$ **do**
  Sample a subset S_r of devices from E_n
  for each client $i \in S_r$ **do**
    Rx global model w_{r-1}
    Initialize local model $w_{i,r} = w_{r-1}$
    for each local epoch $j = 1, \cdots, E$ **do**
      Sample a mini-batch of data $B_{i,j}$
      Update local model: $w_{i,r} = w_{i,r} - \eta \nabla f_i(w_{i,r}, B_{i,j})$
    end for
    Tx local model $w_{i,r}$ to the server
  end for
  local model Aggregation: $w_t = \frac{1}{|S_r|} \sum_{i \in S_r} w_{i,r}$
end for
Output: Final global model w_R

C. Federated Learning Algorithm

Algorithm 1 assumes a synchronous, non-IID setting where each client has access to a local dataset D_i, and $f_i(w_{i,t}, B_{i,j})$ is the loss function for client i at round t with local model $w_{i,t}$ and mini-batch $B_{i,j}$. At each round, the algorithm employs random selection to choose a subset of clients, and each client conducts local training for a predetermined number of epochs E. Following this, the updated model is transmitted back to the server for aggregation. The final global model is generated by the aggregation process, which involves averaging the local models of the selected clients after executing T rounds in the algorithm.

D. Proposed Deep Learning-based Intrusion Detection Model

This section presents the introduction of the proposed FedSecureIDS model, which combines elements from CNN, LSTM, and MLP neural networks to form a hybrid architecture. Before finalizing this particular ML model, we conducted comprehensive testing and evaluation of multiple DL methodologies, including CNN, LSTM, and MLP. The proposed model comprises four distinct units, i.e., a pre-processing unit, CNN unit, LSTM unit, and MLP unit. The integration of CNN and LSTM networks enables precise extraction of time-series patterns in network traffic data, which is vital for ensuring robust IDS. The success of the model can largely be attributed to CNN's effective capability in extracting high-level feature representations. The exceptional performance of this ML model has led to its selection as the IDS model within the proposed FL architecture, which aims to identify cyber threats in ICPS. Each constituent element of the proposed FL model will be thoroughly explored and analyzed in subsequent sections of the research paper.

1) CNN unit: The CNN unit's architecture in the proposed IDS model is the focus of this section. CNNs are employed for feature extraction due to their proven effectiveness in

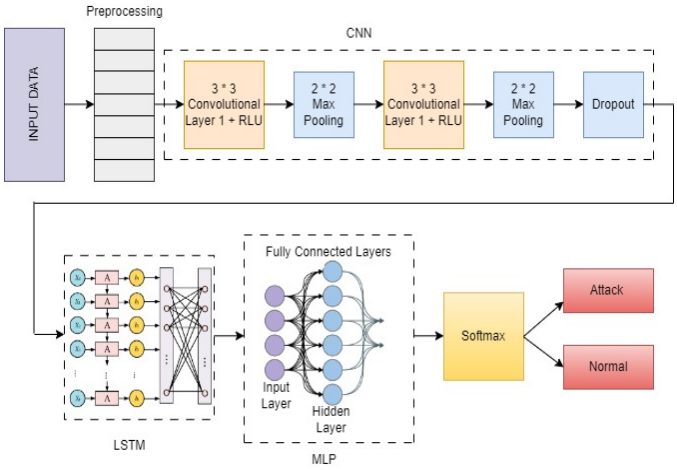

Fig. 5. CNN, LSTM, and MLP architecture diagram.

identifying spatial hierarchies in data. They can automatically and adaptively learn spatial hierarchies of features from input data, making them highly suitable for analyzing network traffic data to detect anomalies indicative of cyber threats. According to [51], the CNN unit consists of five components: An input layer, a convolutional layer, a pooling layer, a fully connected (FC) layer, and an output layer. The CNN architecture in this study was customized to meet the specific needs of the use case. This involved the utilization of convolutional layers, pooling layers, and FC layers, as visually depicted in Fig. 5. The convolutional layer plays a pivotal role in the primary task of extracting features from the input data. It consists of multiple convolution kernels, each with a weight and deviation coefficient. During the convolution process, the weight coefficient and deviation quantity for each kernel are denoted as w_k and a_k, respectively, and the input of convolutional layer k is represented as Y_{k-1} in (1).

$$Y_k = f(w_k \oplus Y_{k-1} + a_k). \tag{1}$$

In the proposed CNN architecture, the convolutional layer is responsible for feature extraction from the input data. It is comprised of multiple convolution kernels, each with a corresponding weight and deviation coefficient. The operation of the convolution kernel k is represented by Y_k, and it applies the activation function $f(Y)$, where RELU is chosen in this study as shown in (2). This activation function has advantages over others, such as sigmoid and tanh in terms of easy derivation, faster model training, and preventing gradient disappearance. The output result of the convolutional layer is obtained through regular sweeping of the input data by the convolution kernels to extract essential information.

$$ReLU(Y_k) = \begin{cases} Y_k, & (Y_k > 0) \\ 0, & (Y_k \leq 0) \end{cases} \tag{2}$$

The pooling layer in a CNN is used to reduce the complexity of the network by downsampling and removing unnecessary information. In this research, max pooling was employed as the selected type of pooling due to its capability of preserving essential information by selecting the highest value within a specific area to represent that area. Max pooling can be expressed as in (3):

$$G^{(m)} = Max(g_1^m, g_2^m, \cdots, g_k^m)^T, \tag{3}$$

In the max pooling operation, $G^{(m)}$ represents the output result of the pooling region k, and g_k^m denotes the element of the pooling region k. In the CNN model used for this research, the FC layers act as classifiers to evaluate the features extracted by convolutional and pooling layers. These features are then mapped to a hidden layer space and then again to a sample-marker space. To prevent the model from becoming too specialized during training and performing poorly on unseen data, a dropout operation is added to the FC layer. Dropout randomly removes some of the neurons to create a more generalized model.

2) LSTM component description: The LSTM network stands out in ML for its efficacy in handling sequential data, overcoming the challenges of gradient vanishing or exploding found in traditional RNNs. Thanks to its memory function, LSTM units adeptly retain both long-term and short-term information. Integral to this architecture are three gate structures: Forget, input, and output gates, which manage the retention or discard of information. The forget gate, particularly significant, determines the extent of information to discard based on the previous hidden state and current input, generating a value between 0 and 1, indicating the amount of information to retain or discard from the current cell state.

$$g_t = \sigma(W_{g'}[h_{t-1}, x_t] + b_{g'}). \tag{4}$$

In (4). g_t represents the gate value at time step t, σ denotes the sigmoid function, $W_{g'}$ and $b_{g'}$ are the weight and bias parameters, respectively. h_{t-1} is the previous hidden state, and x_t is the current input.

The input gate g_t is pivotal in LSTM networks for determining how to update the cell state based on new input data. It undergoes two main steps: Firstly, it assesses the significance of information to retain or discard using the sigmoid function, as depicted in (5). Secondly, it generates alternative information to adjust the cell state via the hyperbolic tangent function, as illustrated in (6). The cell state is then updated from its previous state C_{t-1} to C_t by considering the current input gate i_t and the candidate values $\tilde{C}_t$. This update involves discarding irrelevant information from the previous state and integrating relevant information from the current candidate state, as shown in (7).

$$it = \sigma(W_{i'}[h_{t-1}, X_t] + b_{i'}) \tag{5}$$

$$Ct = \tanh(W_{c'}[g_{t-1}, x_t] + b_{c'}) \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{7}$$

In these equations, $W_{i'}$ and $b_{i'}$ represent the weight and bias parameters for the input gate, respectively. Similarly, $W_{c'}$ and $b_{c'}$ denote the weight and bias parameters for the candidate cell state. The current time step's forgotten gate is labeled as

f_t, while the input gate is denoted by i_t, and the candidate cell state is represented as $\tilde{C}_t$. The forget gate primarily discards specific information from the previous cell state, whereas the input gate integrates new candidate values and updates relevant information. Additionally, the output gate in an LSTM neural network serves as both an input for the subsequent cell state and a key determinant in establishing the final output based on the current cell state. A fundamental role of the output gate is to utilize the sigmoid function to identify which portion of the cell state should be outputted. Moreover, the final output is determined by the content identified by the output gate and involves the utilization of the hyperbolic tangent function. The equation for the output gate, denoted as (8), is presented as follows:

$$g_t = \sigma(W_{o'}[h_{t-1}, x_t] + b_{o'}), \tag{8}$$

$$h_t = g_t * \tanh(C_t). \tag{9}$$

Additionally, the updated hidden state h_t, indicated by (9), is computed by element-wise multiplication of the output gate value g_t and the hyperbolic tangent of the cell state C_t.

LSTMs are chosen over other RNNs due to their capability to capture long-term dependencies and temporal patterns in sequential data without suffering from the vanishing gradient problem. This makes LSTMs particularly effective for recognizing patterns over time, which is crucial for identifying complex attack behaviours in network traffic.

3) Multilayer perceptron layer unit: The final component of the proposed model MLP is due to their robust classification capabilities. They can learn complex mappings from input features to output classes, making them suitable for distinguishing between normal and malicious network activities. MLP unit includes a dropout layer added between the dense layer and the output layer to counteract overfitting. The LSTM output, denoted as O in (10), serves as input to the MLP unit, undergoing processing as follows, integrating the dropout layer:

$$M = \text{Dense}(O), \tag{10}$$

$$D = \text{Dropout}(M), \tag{11}$$

$$Y = \text{Output}(\text{sigmoid}(D)). \tag{12}$$

In (11), the output of the dropout layer is represented as D, while Y denotes the output of the MLP with dropout, as shown in (12). Considering the predicted class output can be either "attack" or "normal," the sigmoid function is applied to pass the output of the MLP with dropout to the output layer. The loss function for the proposed ML model is binary cross-entropy, expressed in (13):

$$\eta = -\frac{1}{B}\sum_{i=0}^{1} t_i \log \hat{t}_i + (1 - t_i) \log\left(1 - \hat{t}_i\right). \tag{13}$$

In the proposed IDS model, edge devices labeled as E_n train local models on their respective local data D_n. During each communication round, an edge device receives a ciphertext C, which is decrypted to obtain new gradients $\hat{w^r}$. These gradients are then used to update the local model weights W^r. This iterative process continues until convergence is achieved. The model parameters are adjusted using the binary cross-entropy loss function, as both predicted and target values evolve, where B denotes the batch size, t_i represents the target value, and $\hat{t_i}$ represents the predicted value.

Algorithm 2 Secure symmetric key exchange

Require: S: Variable storing an initially 128-bit size value. Server's stored device identity (HID_D) and secret value (HSV).

The device sends its identity (HID_D) to the server.
if HID_D matches the stored identity **then**
  The server generates two secret values R_1 and R_2 and sends them securely to the device as X^* and Y^* respectively.
  The device generates a random number R_D.
  if Server's random number $R_S = 0$ **then**
    Go to step 4.
  end if
  Compute $D = (R_D \times R_1) + R_1$.
  Compute $R_D = (D \times R_2) + R_1 + R_2$.
  Send R_D to the server.
  Calculate $r_D = R_D - (R_1 + R_1)$.
  Calculate $R_D = (r_D/(R_1 \times R_2)) - 1$.
end if
The server generates a random number R_S.
if $R_S = 0$ **then**
  Go to step 13.
end if
Compute $D = (R_S \times R_1) + R_1$.
Compute $R_S = (D \times R_2) + R_1 + R_2$.
Send R_S to the device.
Both server and device repeat steps 11 to 12.
Compute the shared key as $K_{DS} = (R_D \oplus R_S) \mod S$.
if $K_{DS} = 0$ **then**
  Go to step 4.
end if
if Identities do not match **then**
  Terminate the connection.
end if

E. Secure Communication Protocol

This section describes the secure communication protocol designed to establish and maintain secure communication between the cloud server and a variety of devices, employing symmetric session keys described in Algorithm 2. The protocol follows several stages: Device registration, login, secret key generation, and mutual authentication, as detailed below.

1) Device registration phase: Each authorized device must undergo registration with the R.A. Throughout this registration process, the R.A. assigns a distinct identity, denoted as ID_D, to each device and generates a confidential secret value. For security purposes, the device's

hash-based identity, denoted as HID_D, is computed as outlined in (14), while the secret value, termed HSV, is determined according to the procedure specified in (15).

$$HID_D = hash(ID_D) \tag{14}$$

$$HSV = hash(ID_D||MacAddress) \tag{15}$$

The server stores the device's confidential credentials in its database. If a device tries to register with an identity already on record, the attempt is rejected.

2) Login phase: During the login phase, the device sends HID_D to the server for validation. After receiving the login details from the remote device, the server looks for the corresponding HID_D in its database. If HID_D is found in the database, the server retrieves the associated HSV for that device. Next, the server generates two large random numbers, $R1$ and $R2$. Using (16) and (17), the server computes X^* and Y^*, respectively, and sends them to the remote device.

$$X^* = (R_1 \oplus HSV) \tag{16}$$

$$Y^* = (R_2 \oplus HSV) \tag{17}$$

The device obtains the values of X^* and Y^* from the server and calculates R_1 and R_2 using (18) and (19)

$$R_1 = (X^* \oplus HSV) \tag{18}$$

$$R_2 = (Y^* \oplus HSV) \tag{19}$$

3) Key exchange phase: The system produces a 128-bit random number R_D. It then multiplies R_D by R_1, adds R_1 to the product, and calculates the final result, as shown in (20).

$$Res_D = (R_D * R_1) + R_1 \tag{20}$$

The device calculates a final result and sends it to the server as FR_D. It is important to note that FR_D, the output value, does not directly reveal the key. Therefore, if a hacker intercepts FR_D, they will not be able to obtain the key.

$$FR_D = (Res_D * R_2) + R_1 + R_2 \tag{21}$$

Upon receiving the final result FR_D from the device, the server infers the hidden number R_D by utilizing both numbers R_1 and R_2 as shown in (21). Then, the server calculates the difference between R_1 and R_2, yielding Res_S as outlined in (22).

$$Res_S = FR_D - (R_1 + R_2) \tag{22}$$

The server extracts R_D, as presented in (23)

$$R_D = (Res_D/(R_1 * R_2)) - 1 \tag{23}$$

When both parties possess shared secret numbers, denoted as R_D and R_S, each side performs a bitwise XOR operation between R_D and R_S, followed by a modulus calculation with respect to the variable M. The resultant value, referred to as the final session key K_S, is clandestinely derived on both ends. This process is illustrated in (24), where M represents the larger value of 128 bits. Initially, it limits the size of the key to 128 bits. We can increase the size of the key easily.

$$K_s = (R_D \oplus R_S)\ Mod\ M \tag{24}$$

4) Authentication phase: After successfully exchanging symmetric session keys, the cloud server, and registered agent will authenticate their identities to prevent MITM attacks. Both parties utilize HMAC with a symmetric session key for authentication. The HMAC calculation involves using their IP addresses, combining them with random numbers and symmetric session keys, as illustrated in (25) and (26), and exchanging the results. SHA-2-256 hashing algorithm is employed to produce HMAC, generating a 64-bit hexadecimal fixed-length code. HMAC values are computed on both ends for verification against the received value. Authentication is deemed successful if the calculated and received HMAC values match on both ends. In case of a mismatch, the connection is promptly terminated. In equation (25) and (26), IA represents the IP address.

$$HMAC_{D/A} = SHA-256(IA_{D/A}||N_{D/A}, K_S) \tag{25}$$

$$HMAC_{S/A} = SHA-256(IA_{S/A}||N_{S/A}, K_S) \tag{26}$$

5) Encryption/decryption phase: After mutual authentication on both sides, the cloud server and agent proceed to exchange encrypted data. The AES algorithm is used for encryption and decryption to achieve this. AES works on a block-by-block basis, with each block being 128 bits in size. The encryption and decryption procedures follow the equations presented in (27) and (28) correspondingly.

$$M_{Enc} = AES(PlainText, K_S) \tag{27}$$

$$M_{Dec} = AES(M_{Enc}, K_S) \tag{28}$$

Each agent securely transmits encrypted data to the cloud server. Upon receiving the encrypted data, the cloud server decrypts it using the symmetric session key. Subsequently, it aggregates all the data and securely sends it to all agents in an encrypted format. This approach ensures that the data remains protected against any potential adversary.

V. Experimental Setup

In this section, we detail the experimental setup used to evaluate the effectiveness and performance of our proposed FedSecureIDS scheme within the context of a privacy-preserving FL-based IDS. The evaluation focuses on several key aspects, including the implementation details, the datasets employed, and the metrics used for performance assessment.

A. Dataset Description

The X-IIoTID dataset, comprising 68 attributes related to network traffic and system activities, was utilized to evaluate the proposed framework. It covers data on nine types of attacks, including reconnaissance, weaponization, lateral movement, C&C, tampering, RDoS, exfiltration, and crypto-ransomware. The diversity and breadth of this dataset support the generalizability of the proposed framework across different industrial contexts, as it encapsulates a wide range of scenarios and potential threats commonly encountered in industrial environments. However, this research focused solely on distinguishing between normal requests and attacks by classifying them under the "attack" category. Future work will explore methods for categorizing multiple attack types. The X-IIoTID dataset comprises 820,834 cases categorized as normal or attack. Among these, 421,417 were labeled as normal and 399,417 as attacks.

B. Implementation Detail

The FL-based IDS model presented in this study was developed using TensorFlow, an open-source Python library for DL, and the Keras API. The implementation and evaluation of the framework were conducted on a system with a 4-core CPU, 16 GB RAM, and 512 SSD using Python 3.0.

C. Evaluation Metrics

Four common evaluation metrics, classification accuracy, precision, recall, F1-score, and RoC, were employed to evaluate the efficacy of the suggested framework. These metrics depend on values like true positive, true negative, false positive, and false negative.

$$\text{Accuracy} = \left(\frac{C_P + C_N}{C_P + C_N + I_P + I_N}\right) \tag{29}$$

$$\text{Recall} = \left(\frac{C_P}{C_P + I_N}\right) \tag{30}$$

$$\text{Precision} = \left(\frac{C_P}{C_P + I_P}\right) \tag{31}$$

$$\text{F1-score} = 2 \times \left(\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}\right) \tag{32}$$

In (29) to (32):
- C_P represents the count of true positive predictions.
- C_N represents the count of true negative predictions.
- I_P represents the count of false positive predictions.
- I_N represents the count of false negative predictions.
- Accuracy measures the overall correctness of the predictions.
- Recall measures the proportion of actual positive cases that were correctly identified.
- Precision measures the proportion of positive predictions that were correctly identified.
- F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.

VI. Security Analysis

We demonstrate the robustness of our proposed lightweight FL-based IDS model through informal and formal security analysis.

A. Informal Security Analysis

The informal security analysis covers protection against known attacks like replay attacks, impersonation attacks, MITM attacks, side-channel attacks, DoS attacks, and perfect forward secrecy discussed in [40], [45]. Different mechanisms and standards have been used to protect against known attacks.

B. Security Analysis through AVISPA

For formal security analysis, we utilize the AVISPA tool based on the Dolev-Yao adversary model. The AVISPA tool code can be found on GitHub [52]. Validation tests using OFMC and CL-AtSe were conducted to evaluate the resilience of our security model against severe assaults. The test results, shown in Fig. 6, validate the security and efficacy of our model.

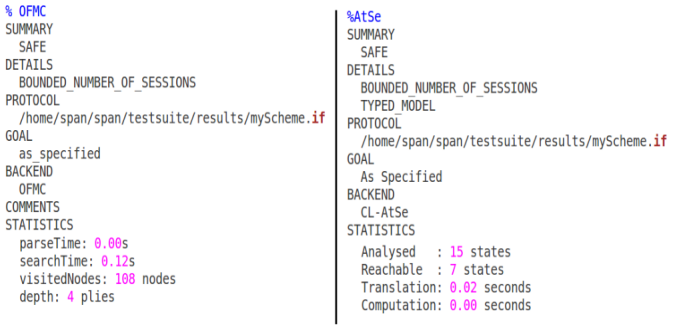

Fig. 6. Security analysis through OFMC and AtSe.

C. Security Analysis through BAN Logic

In this section, we showcase the resilience of our proposed scheme against known attacks by leveraging Burrows-Abadi-Needham (BAN) logic. BAN logic, a symbolic logic system devised for scrutinizing security protocols, is utilized to validate the adherence of a protocol to various security properties. A comprehensive BAN logic analysis for the user registration phase of our proposed protocol, complete with equations and explanations, is provided below:

– **Initial assumption:**

– Device D places trust in server S.
– Device D selects a distinctive identity HID_D along with a secret value HSV.
– Utilizing a hash function, both HID_D and HSV are generated to facilitate secure communication.
– Server S securely stores the confidential credentials of devices, encompassing device type and serial number.

– Server S authenticates the identity of devices by cross-referencing the stored data in its secure and protected database.
– **Idealized protocol model:**

– **Protocol description:**

– **Formal agreement analysis:**

1) BAN logic formal analysis for device registration phase::

– Device D selects a unique identity ID_D and a secret value HSV_D.
– Device D computes hash-based identity HID_D = hash(ID_D) and hash-based secret value HSV = hash($ID_D \| PSW_D$).

Verification of HID_D by server S:

- For verification, device D transmits HID_D to server S.
- Server S cross-references the received HID_D with its stored database to validate the identity.
- If the identity is confirmed, server S continues; otherwise, it terminates the connection.

2) Equations and BAN logic examination::

- **Initial assumptions (idealization):**
  – D places trust in {S, K_S} security: D | S, K_S
  – D selects a distinct identity and robust password: D | {ID_D, PSW_D}
  – D trusts the security of the hash function: D | hash(ID_D)
  – S securely maintains user credentials: S | {Secrets}
  – S verifies identities using stored data: S | {Verified}
- **Idealized protocol model (idealization):**
  – Device D dispatches HID_D to server S for verification: D → S: {HID_D}
  – Server S scrutinizes HID_D in its database, confirming identity: S → D: {Verified}
- **Protocol description (formalization):**
  – Device D assumes server S received HID_D: D | S: {HID_D}
  – Device D assumes server S verified the identity: D | S: {Verified}
- **Formal agreement analysis (inference rules):**
  – Device D believes server S verified identity with HID_D: D | S: {HID_D, Verified}
  – Server S securely preserves user credentials: S | {Secrets}
  – Server S authenticates identities using stored data: S | {Verified}
  – Device D securely registers with server S: D | S: {Registered}

3) BAN logic assessment: The device registration phase is designed to ensure the secure registration of device D with server S, utilizing a unique identity and a robust password. server S meticulously validates identities by cross-referencing with stored data, thereby permitting only authorized individuals to complete the registration process successfully. The protocol guarantees the precise utilization of the designated identity and password. This BAN logic formal analysis definitively confirms the adherence of the user registration phase to security properties, consequently diminishing the likelihood of unauthorized access. Comparable analyses can be conducted for additional protocol phases as deemed necessary.

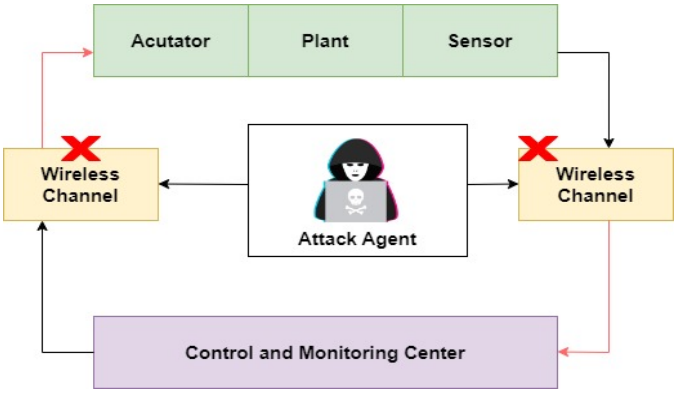

Fig. 7. ICPS DoS attack scenario.

D. Attack Scenarios and Mitigation Strategies

To further illustrate the practical implications of our security analysis, this section presents detailed scenarios for different attack types and discusses strategies for mitigating vulnerabilities. These scenarios demonstrate how FedSecureIDS detects and responds to various threats, reinforcing the findings from our formal and informal analyses. Due to the constraints of paper length, we will focus on two critical attack scenarios: DoS attacks and MITM attacks.

1) DoS attacks: In a DoS attack, an attacker aims to disrupt the communication channels within an ICPS. As illustrated in Fig. 7, the attack agent targets the wireless channels between the control and monitoring center and the actuators, plants, and sensors. The attacker floods these wireless channels with excessive traffic, preventing legitimate control signals and sensor data from being transmitted. Specifically, the attacker overwhelms the wireless channel from the control and monitoring center to the actuators, making it impossible for the center to send control commands. Simultaneously, the wireless channel from the sensors to the control and monitoring center is flooded, causing a loss of critical sensor data needed for real-time monitoring and decision-making. This disruption can result in halted operations, malfunction of the physical plant, and the control system operating blindly without sensor updates, potentially causing unsafe conditions or operational inefficiencies.

To counteract DoS attacks, FedSecureIDS employs a multi-layered approach combining real-time monitoring, anomaly detection, and dynamic response mechanisms. Firstly, FedSecureIDS continuously monitors traffic on the wireless channels using distributed edge devices, collecting and analyzing traffic patterns to establish a baseline of normal operations. CNNs are then utilized to detect anomalies in traffic patterns, examining features such as packet rate, packet size, and traffic volume to identify abnormal spikes indicative of a DoS attack. Upon

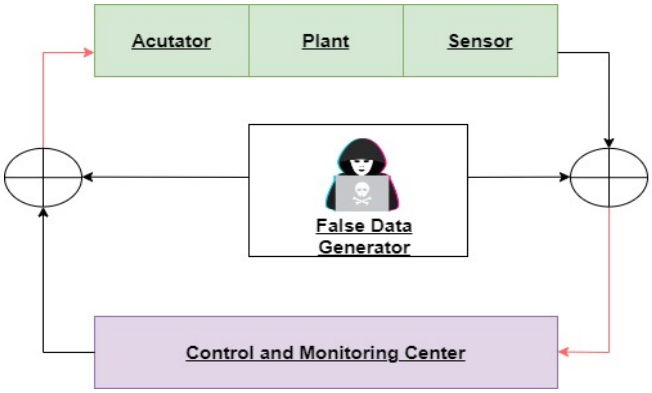

Fig. 8. ICPS MITM attack scenario.

detecting an anomaly, rate limiting is applied to control the flow of traffic, ensuring that no single source can overwhelm the wireless channels, and traffic-shaping techniques prioritize critical data such as control commands and sensor updates.

2) MITM Attacks: In an MITM attack, an attacker intercepts and manipulates the communication between components of an ICPS. As depicted in Fig. 8, the attacker, acting as a false data generator, positions themselves between the sensors, actuators, and the control and monitoring center. The attacker intercepts the data transmitted from the sensors, alters it, and sends false information to the control and monitoring center. Similarly, they intercept control commands from the control and monitoring center, modify them, and send erroneous commands to the actuators. This manipulation can cause the system to make incorrect decisions, leading to potential damage to the physical plant, inefficiencies, or unsafe operating conditions.

To mitigate MITM attacks, FedSecureIDS utilizes a combination of real-time monitoring, cryptographic protocols, and anomaly detection techniques. Firstly, FedSecureIDS implements end-to-end encryption for all communications between the control and monitoring center, sensors, and actuators using strong cryptographic protocols such as transport layer security (TLS). This ensures that even if an attacker intercepts the data, they cannot read or modify it without the encryption keys.

FedSecureIDS also continuously monitors network traffic for signs of anomalies that indicate an MITM attack. Local Intrusion Detection Modules on edge devices employ LSTM networks to analyze temporal patterns in the data, detecting subtle changes indicative of data manipulation. CNNs are used to examine features such as packet integrity, timing, and sequence to identify discrepancies that suggest the presence of an attacker.

By implementing these strategies, FedSecureIDS effectively mitigates the risk of MITM attacks, ensuring the integrity and authenticity of data within the ICPS environment. This approach enhances the resilience of the system against sophisticated cyber threats, maintaining safe and efficient operations.

TABLE II
PERFORMANCE COMPARISON OF CENTRALIZED AND PROPOSED MODEL.

Type of cyber attack	Centralized model				Proposed model			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Denial of service attack	0.981	0.991	0.922	0.948	0.979	0.992	0.951	0.993
Weaponization attack	0.979	0.991	0.934	0.963	0.98	0.991	0.964	0.992
Exploitation attack	0.989	0.993	0.947	0.982	0.986	0.995	0.973	0.997
Tampering attack	0.982	0.998	0.972	0.983	0.984	0.99	0.979	0.992
Reconnaissance attack	0.992	0.994	0.986	0.987	0.991	0.994	0.993	0.994

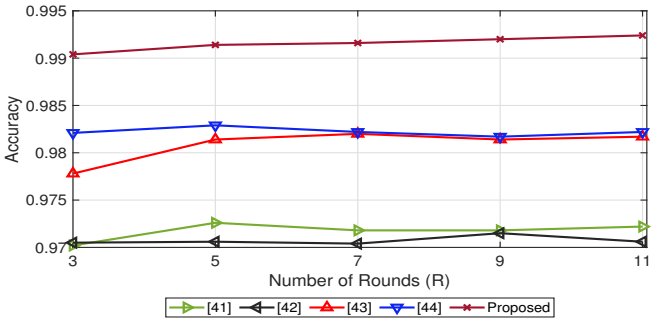

Fig. 9. Performance comparison of accuracy vs. number of rounds when edge devices = 3.

VII. RESULT AND DISCUSSION

A. Performance Comparison with Baseline Studies

The FL-based IDS model's effectiveness is compared to other well-known research utilizing FL frameworks like [41]–[44]. DL models from these studies are replicated and assessed against the proposed model, as shown in Figs. 9–14. Experimental results demonstrate the superiority of the proposed IDS model in accuracy, precision, recall, and F1-score over current leading models. The FL-based IDS model's performance typically enhances as the number of communication rounds R escalates from 3 to 15, stabilizing at a sufficiently large R. Figs. 9–11 shows the accuracy of five federated learning algorithms [41]–[44] and the proposed FedSecureIDS model across varying rounds R and a number of edge device E_n (3, 7, and 15). In all the considered scenarios, the proposed method consistently achieves the highest accuracy, approaching 0.995%, demonstrating superior performance and scalability. Algorithm [44] generally ranks second, maintaining an accuracy of around 0.98%, while [43] shows moderate performance. Algorithms [41] and [42] exhibit the lowest accuracy, remaining around 0.975% and below. These findings are encapsulated in Fig. 15, which presents the same results in a bar graph format, reinforcing the superior performance of the proposed method. Similarly, for the F1-score, the proposed FedSecureIDS again leads, indicating its superior precision and recall, as shown in Figs. 12–14. This illustrates that with 3, 7, or 15 edge devices, it exhibited slight variation in F1-scores, which evidences its robustness. An algorithm in [44] remains the second-best performer, while [43] demonstrates moderate results. Algorithms [41] and [42] exhibit the lowest F1-scores, reflecting poorer performance.

These findings are summarized in Table III, which presents the same results in a tabular format, reinforcing the superior performance of the proposed method. The results of the

TABLE III
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART RESEARCH PAPERS.

k	R	[41]				[42]				[43]				[44]				Proposed			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
3	3	0.9622	0.9681	0.9441	0.9561	0.9683	0.9691	0.9454	0.9563	0.9684	0.9686	0.9438	0.9541	0.9778	0.9741	0.9482	0.9812	0.9832	0.9822	0.9574	0.9839
	5	0.9625	0.9687	0.9448	0.9558	0.9691	0.9689	0.9461	0.9569	0.9693	0.9708	0.9436	0.9549	0.9801	0.9738	0.9484	0.9811	0.9827	0.9813	0.9577	0.9842
	7	0.9627	0.9685	0.9449	0.9556	0.9698	0.9695	0.9459	0.9572	0.9701	0.9702	0.9444	0.9652	0.9811	0.9735	0.9482	0.9807	0.982	0.9838	0.9579	0.9867
	9	0.9627	0.9683	0.9447	0.9555	0.9703	0.9696	0.9463	0.9576	0.9706	0.9703	0.9449	0.9655	0.9821	0.9735	0.9489	0.9809	0.9818	0.9838	0.9582	0.9869
	11	0.9622	0.968	0.9448	0.956	0.9707	0.9696	0.9467	0.9581	0.9711	0.9704	0.9453	0.9653	0.9817	0.9726	0.9491	0.9811	0.9827	0.9842	0.9585	0.9871
7	3	0.9611	0.9672	0.9434	0.9548	0.9701	0.9687	0.9465	0.9572	0.9687	0.9689	0.9448	0.9591	0.9811	0.9719	0.9486	0.9795	0.9902	0.9865	0.9578	0.9847
	5	0.9619	0.9674	0.9437	0.9557	0.97	0.9704	0.9437	0.9564	0.9701	0.9708	0.9452	0.9589	0.982	0.9693	0.9501	0.9807	0.9908	0.9871	0.9577	0.9849
	7	0.9621	0.9679	0.9447	0.9551	0.9709	0.9715	0.9431	0.9567	0.9707	0.9712	0.9454	0.9601	0.9818	0.9705	0.9511	0.9811	0.9906	0.9869	0.9581	0.9853
	9	0.9619	0.9673	0.9443	0.9553	0.9711	0.9719	0.9435	0.9572	0.9717	0.9716	0.946	0.9638	0.9822	0.9709	0.9515	0.9818	0.9911	0.9882	0.9583	0.9865
	11	0.9624	0.9674	0.9441	0.9555	0.9708	0.9721	0.9439	0.9577	0.9719	0.9724	0.9463	0.9654	0.9824	0.9716	0.9522	0.9814	0.9914	0.9877	0.9586	0.9870
15	3	0.9616	0.9681	0.9437	0.954	0.9696	0.9697	0.9447	0.9565	0.9691	0.9698	0.9458	0.9649	0.9795	0.9721	0.9499	0.9797	0.9918	0.9865	0.9580	0.9858
	5	0.9622	0.9685	0.9432	0.9549	0.9707	0.9703	0.9449	0.9566	0.9708	0.9621	0.9451	0.9657	0.9806	0.9729	0.9497	0.9801	0.9921	0.9887	0.9577	0.9863
	7	0.9629	0.968	0.9437	0.9552	0.9705	0.9708	0.9451	0.9571	0.9714	0.9719	0.9456	0.9683	0.9814	0.9734	0.9492	0.9806	0.992	0.9901	0.9579	0.9873
	9	0.9631	0.9682	0.94	0.9558	0.9708	0.9711	0.9448	0.9576	0.972	0.9711	0.9461	0.9679	0.9819	0.9738	0.9502	0.981	0.9919	0.9894	0.9573	0.9876
	11	0.9629	0.9686	0.9445	0.9556	0.971	0.9717	0.9452	0.9579	0.9718	0.9714	0.9462	0.9688	0.9825	0.9742	0.9508	0.9812	0.9924	0.9905	0.9587	0.9879

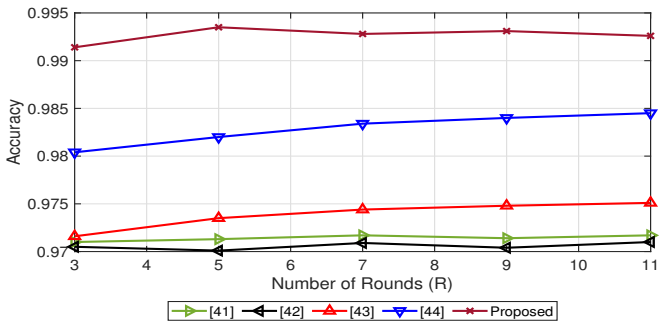

Fig. 10. Performance comparison of accuracy vs. number of rounds when edge devices = 7.

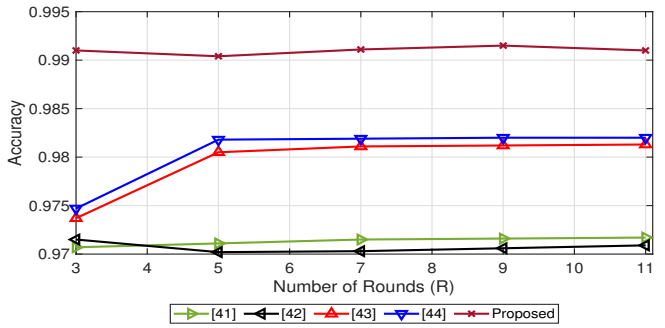

Fig. 11. Performance comparison of accuracy vs. number of rounds when edge devices = 15.

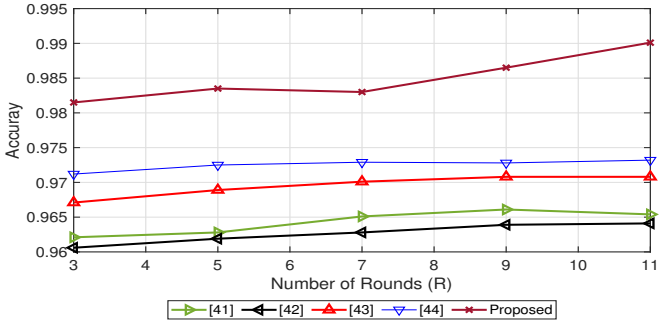

Fig. 12. Performance comparison of F1-score vs. number of rounds when edge devices = 3.

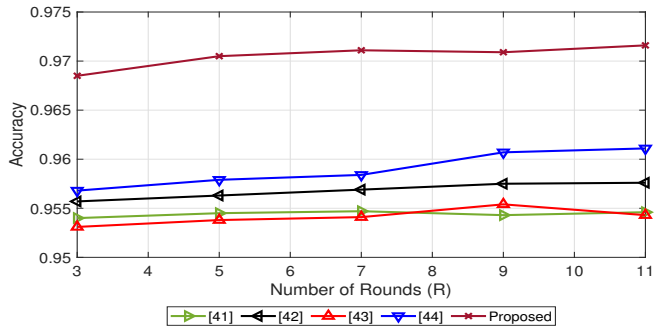

Fig. 13. Performance comparison of F1-score vs. number of rounds when edge devices = 7.

proposed FedSecureIDS confirm the approach's its robustness and adaptability in diverse federated learning setups.

B. Performance Comparison with Centralized Model

This section evaluates an FL-based IDS model for detecting cyber threats in CPS while preserving data privacy. We compare its performance with a centralized ideal model that has access to all data. The comparison is shown in Fig. 16 presents a comparative analysis of detection accuracy between the centralized approach and proposed FedSecureIDS across five categories of cyberattacks, i.e., DoS, weaponization, exploitation, tampering, and reconnaissance. Both methods achieve high accuracy rates, exceeding 0.85 in all categories. However, the proposed method consistently outperforms the centralized approach across all attack vectors. This consistent superior performance of the proposed methodology underscores its effectiveness in accurately identifying and mitigating a range of cyber threats, highlighting its potential for enhanced application in cybersecurity contexts. These results are presented in Table II, which includes a detailed comparison of accuracy, F1-score, precision, and recall for both methodologies across the five attack categories.

Fig. 17 compares the performance of a centralized learning model and multiple FedSecureIDS configurations. The area under the curve (AUC) values indicate the effectiveness of each model in distinguishing between classes. The centralized

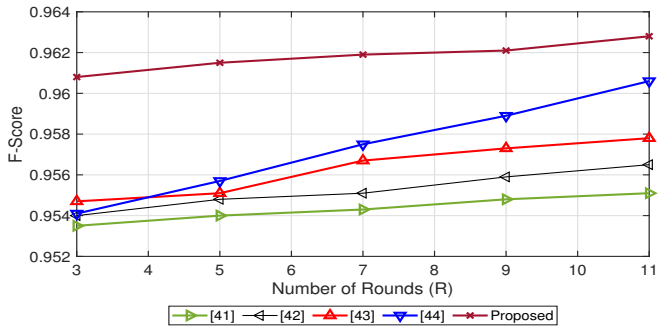

Fig. 14. Performance comparison of F1-score vs. number of rounds when edge devices = 15.

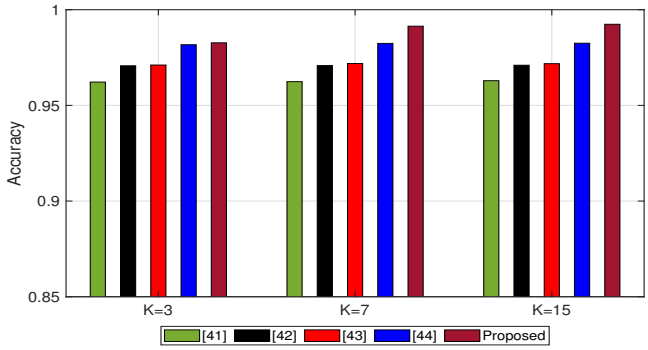

Fig. 15. Performance comparison of the proposed model with state of art studies.

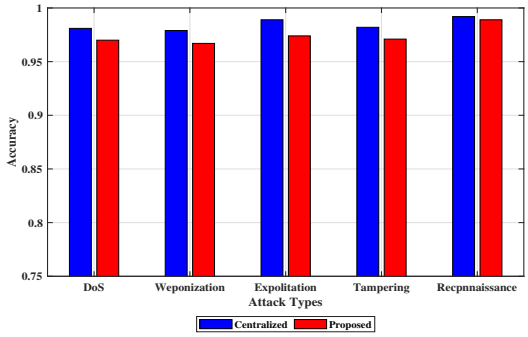

Fig. 16. Performance comparison of centralized and proposed model under different attack types.

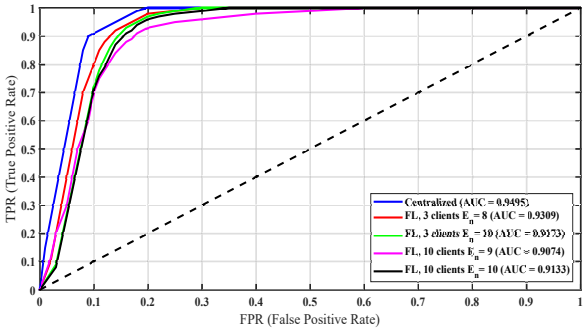

Fig. 17. ROC curves for centralized and FedSecureIDS model.

model achieves the highest AUC of 0.9429, while various FL setups, including 3 clients with $E_n = 8$ and $E_n = 10$, and 10 clients with $E_n = 9$ and $E_n = 10$, exhibit slightly lower but comparable AUC values, demonstrating the feasibility of FL in distributed environments. The centralized model achieves the highest accuracy with an AUC of 0.9429 due to its access to a complete, unified dataset. This comprehensive data exposure allows for robust feature learning. Conversely, the FedSecureIDS model achieves an accuracy of 0.93, is trained on subsets of data across clients, and faces challenges such as non-IID data and limited data per client, slightly reducing their AUC values. Despite this, FL models maintain strong performance while enhancing data privacy, making them suitable for sensitive applications in ICPS. FedSecureIDS demonstrates high performance in detecting known cyber threats, its efficacy against emerging, sophisticated attacks may vary. Additionally, environments with constrained computational resources might face challenges in implementing the model efficiently.

1) Key management analysis: Table IV shows the comparative analysis between the asymmetric key management protocols and our proposed symmetric key exchange protocol. Comparison results show that symmetric key exchange protocols have advantages over asymmetric key management protocols in terms of key size, key generation time, computation speed, communication overhead, and security threats.

2) Symmetric and asymmetric cryptography performance analysis: We compared the performance of asymmetric and symmetric key cryptographic systems. Results show that our proposed symmetric key cryptographic system has advantages over asymmetric cryptographic systems in terms of communication and computation overheads [53].

3) Operational efficiency and performance analysis: When we compared the performance of our proposed secure symmetric key cryptographic system with other systems in terms of computation and communication costs, our proposed secure system showed better performance in terms of communication and computation costs. We implemented our proposed system on an industrial NFV-Based IPv6 network and measured the operational efficiency [45].

VIII. Conclusion and Future Work

ICPS in Industry 4.0 is highly vulnerable to cyber threats due to device interconnectivity, making intrusion detection, and data privacy a significant challenge. This paper proposed a FedSecureIDS model where multiple smart devices collaboratively construct an IDS using CNN, LSTM, and MLP architectures. Rigorous testing on a real-world CPS dataset demonstrated the method's superior performance, achieving 98.68% accuracy, 98.78% precision, 98.64% recall, and a 99.05% F1-score. A lightweight symmetric encryption technique was integrated to secure communications between smart devices and servers, ensuring data privacy. The method's resilience to well-known cyber attacks was validated through formal analysis using tools such as AVISPA and BAN logic,

TABLE IV
COMPARISON OF ASYMMETRIC AND SYMMETRIC KEY MANAGEMENT PROTOCOLS.

Features	**[32]–[38]**	**Proposed**
Key type	Asymmetric	Symmetric
Key property	Static	Dynamic
Key size (bits)	3072	128
Key generation time complexity	Polynomial	Constant
Communication overhead	Very high	Very low
Computation speed	Very slow	Very fast
Security issues	Low	Very Low
Encryption / decryption	Slow	Fast

offering strong assurance of effectiveness. Furthermore, informal security analysis confirmed its ability to mitigate cyber threats, establishing the framework as a reliable option for securing ICPS environments.

Several challenges in FL based IDS persist, including vulnerability to data poisoning attacks, which necessitate robust countermeasures. Enhancing model training efficiency and managing communication overhead through client clustering is crucial. Additionally, it is important to reduce latency and computational overhead for real-world deployment and to use different testing environments for model validation. Optimizing latency and computational overhead for real-world deployment alongside diverse testing environments is essential for model validation. Data inference attacks also pose risks, demanding stronger privacy-preserving techniques. Addressing these challenges will improve the security and efficiency of FL-based IDS in ICPS.

ACKNOWLEDGMENTS

The authors extend their appreciation to Taif University, Saudi Arabia, for supporting this work through project number (TU-DSPP-2024-41).

AVAILABILITY OF DATA AND MATERIALS

Data and materials are available and can be provided on request.

REFERENCES

[1] A. K. Sutrala *et al.*, "Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2316–2330, Mar. 2021.

[2] M. S. Obaidat, I. Traore, and I. Woungang, Eds., Biometric-Based Physical and Cybersecurity Systems. Cham: Springer International Publishing, 2019, pp. 1–10.

[3] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Cyber-physical-social systems: A state-of-the-art survey, challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 389–425, 2019.

[4] H. Alkahtani and T. H. Aldhyani, "Developing cybersecurity systems based on machine learning and deep learning algorithms for protecting food security systems: Industrial control systems," *Electron.*, vol. 11, no. 11, p. 1717, 2022.

[5] M. A. Alohali *et al.*, "Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment," *Cogn. Neurodynamics*, vol. 16, no. 5, pp. 1045–1057, 2022.

[6] P. Voigt and A. Von dem Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, 1st ed., Cham: Springer International Publishing, 2017, vol. 10, no. 3152676, pp. 10–5555.

[7] Y. Guo *et al.*, "Efficient and flexible management for industrial Internet of things: A federated learning approach," *Comput. Netw.*, vol. 192, p. 108122, 2021.

[8] Q. Gou, L. Yan, Y. Liu, and Y. Li, "Construction and strategies in IoT security system," in *Proc. IEEE GreenCom-iThings-CPSCom* 2013.

[9] A. Ashok, A. Hahn, and M. Govindarasu, "A cyber-physical security testbed for smart grid: System architecture and studies," in *Proc. CSIIRW*, 2011.

[10] A. Tabassum, A. Erbad, W. Lebda, A. Mohamed, and M. Guizani, "FedGAN-IDS: Privacy-preserving ids using gan and federated learning," *Comput. Commun.*, vol. 192, pp. 299–310, 2022.

[11] T. D. Nguyen, P. Rieger, M. Miettinen, and A. R. Sadeghi, "Poisoning attacks on federated learning-based iot intrusion detection system," in *Proc. DISS*, 2020.

[12] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. AISTATS*, 2020.

[13] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. ICML*, 2019.

[14] C. Xie, K. Huang, P. Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proc. ICLR*, 2020.

[15] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. ACM SIGSAC*, 2015.

[16] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC*, 2017.

[17] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," 2018, *arXiv:1805.04049.*

[18] N. M. García, "Multi-agent system for anomaly detection in industry 4.0 using machine learning techniques," *Advances Distrib. Comput. Artificial Intell. J.*, vol. 8, no. 4, pp. 33–40, 2019.

[19] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber-physical systems using recurrent neural networks," in *Proc. IEEE HASE*, 2017.

[20] A. Khazraei, H. Kebriaei, and F. R. Salmasi, "Replay attack detection in a multi-agent system using stability analysis and loss-effective watermarking," in *Proc. IEEE ACC*, 2017.

[21] S. Boddupalli and S. Ray, "REDEM: Real-time detection and mitigation of communication attacks in connected autonomous vehicle applications," in *Proc. IFIPIoT*, 2019.

[22] S. Boddupalli, A. S. Rao, and S. Ray, "Resilient cooperative adaptive cruise control for autonomous vehicles using machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15655–15672, Sep. 2022.

[23] Y. Huang *et al.*, "Real-time detection of false data injection in smart grid networks: An adaptive cusum method and analysis," *IEEE Syst. J.*, vol. 10, no. 2, pp. 532–543, Jun. 2014.

[24] A. Varshovi, M. Rostamipour, and B. Sadeghiyan, "A fuzzy intrusion detection system based on categorization of attacks," in *Proc. IEEE IKT*, 2014.

[25] A. Mishra, B. B. Gupta, and R. C. Joshi, "A comparative study of distributed denial of service attacks, intrusion tolerance and mitigation techniques," in *Proc. IEEE EISIC*, 2011.

[26] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Comput. Netw.*, vol. 204, p. 108693, Dec. 2022.

[27] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu, "FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4059–4068, 2021.

[28] T. Li, Z. Huang, P. Li, Z. Liu, and C. Jia, "Outsourced privacy-preserving classification service over encrypted data," *J. Netw. Comput. Appl.*, vol. 106, pp. 100–110, 2018.

[29] W. Li, W. Meng, and M. H. Au, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments," *J. Netw. Comput. Appl.*, vol. 161, p. 102631, 2020.

[30] Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "Multi-task network anomaly detection using federated learning," in *Proc. SoICT*, 2019.

[31] D. Costa, D. Garrido, and D. Silva, "Efficient secure communication for distributed multi-agent systems," in *Proc. ICAART*, 2021.

[32] B. Li *et al.*, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2020.

[33] Q. Kong *et al.*, "Privacy-preserving aggregation for federated learning-based navigation in vehicular fog," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8453–8463, Dec. 2021.

[34] Z. Shi, Z. Yang, A. Hassan, F. Li, and X. Ding, "A privacy-preserving federated learning scheme using homomorphic encryption and secret sharing," *Telecommunication Syst.*, vol. 82, no. 3, pp. 419–433, Mar. 2023.
[35] W. Du *et al.*, "An efficient and robust privacy-preserving framework for cross-device federated learning," *Complex Intell. Syst.*, pp. 1–15, 2023.
[36] P. Verma, J. G. Breslin, and D. O'Shea, "FLDID: Federated learning enabled deep intrusion detection in smart manufacturing industries," *Sensors*, vol. 22, no. 22, p. 8974, Nov. 2022.
[37] Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2017.
[38] C. Fang *et al.*, "Privacy-preserving and communication efficient federated learning in Internet of things," *Comput. Security*, vol. 103, p. 102199, Dec. 2021.
[39] E. Barker and Q. Dang, NIST Special Publication 800-57 Part 1, Revision 5: Recommendation for Key Management: Part 1—General, National Institute of Standards and Technology, 2020, p. 58.
[40] Z. Ashraf, A. Sohail, and M. Yousaf, "Robust and lightweight symmetric key exchange algorithm for next-generation IoE," *Internet Things*, vol. 22, p. 100703, Jan. 2023.
[41] W. Schneble and G. Thamilarasu, "Attack detection using federated learning in medical cyber-physical systems," in *Proc. ICCCN*, 2019.
[42] T. D. Nguyen *et al.*, "DÏoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE ICDCS*, 2019.
[43] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul. 2020.
[44] V. Mothukuri *et al.*, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Apr. 2022.
[45] Z. Ashraf, A. Sohail, and M. Iqbal, "Design and implementation of lightweight certificateless secure communication scheme on industrial NFV-based IPv6 virtual networks," *Electron.*, vol. 13, no. 13, p. 2649, Jul. 2024.
[46] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.
[47] S. G. O. P. A. L. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015, *arXiv:1503.06462*.
[48] Z. Ling and Z. H. J. Hao, "An intrusion detection system based on normalized mutual information antibodies feature selection and adaptive quantum artificial immune system," *International J. Semantic Web Inf. Syst.*, vol. 18, no. 1, pp. 1–25, Jan. 2022.
[49] Z. Ling and Z. H. J. Hao, "Intrusion detection using normalized mutual information feature selection and parallel quantum genetic algorithm," *International J. Semantic Web Inf. Syst.*, vol. 18, no. 1, pp. 1–24, Jan. 2022.
[50] L. Zhang, S. Jiang, X. Shen, B. B. Gupta, and Z. Tian, "PWG-IDS: An intrusion detection model for solving class imbalance in IIoT networks using generative adversarial networks," 2021, *arXiv:2110.03445*.
[51] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[52] "AVISPA Code and Simulation Results," GitHub Repository, [Online]. Available: https://github.com/zashraf-sudo/researchpaper-5-code. [Accessed: Feb. 10, 2024].
[53] Z. Ashraf, A. Sohail, and M. Yousaf, "Lightweight and authentic symmetric session key cryptosystem for client-server mobile communication," *J.Supercomputing*, vol. 79, no. 14, pp. 16181–16205, Apr. 2023.

Imtiaz Ali Soomro is pursuing a Ph.D. in Electrical and Computer Engineering at Sir Syed CASE Institute of Technology, Islamabad, Pakistan. He obtained his B.E. in Telecommunications from Hamdard University, Islamabad, Pakistan 2010. He earned his M.S. in Electrical Engineering with a specialization in Telecom and Networking from COMSATS University, Islamabad, Pakistan, in 2012. His research interests are focused on the application of federated learning for IoT, wireless networks, and cybersecurity, particularly on privacy-preserving technologies and secure communication in distributed systems. He is also an IEEE Member, actively contributing to the research community through his innovative work in machine learning, IoT, and cybersecurity.

Hamood ur Rehman Khan obtained his B.S. degree in Electronics Engineering from Ghulam Ishaq Khan Institute of Technology in 2000, M.S. degree from the University of Michigan in 2005 and Ph.D. degree from King Fahd University of Petroleum and Minerals in 2019, both in Electrical Engineering. He is currently a Senior Member Technical Staff at the Center for Advanced Research and Engineering (CARE). He is currently an Adjunct Professor with the Computer Science Department, Sir Syed-CASE-Institute of Technology. At CARE he has been leading projects ranging from IoT Platform-as-a-Service systems, cybersecurity products, and advanced VLSI based AI platforms for large language model (LLM) inference. During the period 2000–2003 he was with Avaz Networks Inc, California as a Senior VLSI Design Engineer working on high-density voice over IP (VoIP) system-on-chips for gateway media switches. His primary research interests are information theory and signal processing for networked systems like WSNs and IoT.

Syed Jawad Hussain is an Associate Professor and Chairperson at the Sir Syed Case Institute of Technology, Islamabad, Pakistan. He holds a Ph.D. in Computer Science from Massey University, New Zealand, focusing on developing high-definition video quality experience models. His research interests include multimedia communication networks, machine learning, quality of service (QoS), quality of experience (QoE), data and network security, and statistical modeling. Dr. Hussain has extensive experience in academia and industry, having held various leadership roles, including Head of Department positions at institutions in Pakistan and abroad. He has worked on numerous research and consultancy projects, focusing on machine learning, data security, and multimedia communications. Dr. Hussain has published extensively in prestigious journals and conferences, contributing significantly to the field of computer science.

Zeeshan Ashraf, SMIEEE received the MSCS degree in Computer Networking and Security Services from UMT, Lahore, Pakistan, and the Ph.D. degree in Computer Science from IQRA University, Islamabad Campus, Islamabad, Pakistan. He has more than 15 years of experience in the IT field. He is CCNP (routing and switching) certified by CISCO. He is the Author of several books published by international publishers. His several research articles have been published in well-reputed international journals, having a good impact factor in Q1 and Q2. Currently, he is working as an Associate Professor in the Department of Computing and Information Technology, at IISAT, Gujranwala, Pakistan. His main research interests include next-generation virtualized internet architecture, IPv6 routing, performance modeling, IoT, optimization techniques, and security services in different networks.

PLACE PHOTO HERE

Mrim M. Alnfiai is currently an Associate Professor in Information Technology with Taif University, Saudi Arabia. Her research interests include assistive technology, human–computer interaction, accessibility, usable security, AI, and machine learning. She has published several papers at ISI journals and assistive technology, HCI, and accessibility conferences, including ASSETS, ANT, FNC, CIST, JAIHC, and ICCA. Her current research interest includes designing accessible tools for visually impaired people, including people with no or low vision. She has conducted several studies and experiences to understand visually impaired abilities and behaviors and design accessible systems that help them interact easily with technology. She has also published several articles on accessibility and authentication mechanisms for visually impaired users. She has also published articles related to using NFC technology and machine learning to enhance the healthcare systems.

PLACE PHOTO HERE

Nouf Nawar Alotaibi is an Assistant Professor in the Faculty of Education at Najran University. She teaches courses in gifted education and talent development, creativity, and the history and systems of psychology with educational applications. She is studying learning designs that combine advanced content, complex thinking, conceptual understanding, and intellectual character development. Currently, she is working on complex thinking, specifically focused on developing students' abilities to think both critically and creatively within domain-specific traditions.