Federated Learning-Based Ensemble CNN for Network Intrusion Detection in Healthcare Systems

Dr. T. Revathi [1], Mr. R. Prabhu [2], Mr. K. Vijesh Pethuram [3]
Department of Information Technology, Mepco Schlenk Engineering College

***Abstract*— The rising complexity of cyber threats in Industry 5.0-driven healthcare environments necessitates robust, privacy-preserving intrusion detection mechanisms. In this paper, we propose a Federated Learning-Based Ensemble Convolutional Neural Network (FLE-CNN) for Network Intrusion Detection in smart healthcare systems. Unlike traditional centralized models, our approach leverages federated learning to preserve data privacy by enabling collaborative model training across multiple decentralized healthcare systems without exposing sensitive patient data. The FLE-CNN integrates multiple models like CNN, SVM, DT and KNN into an ensemble, enhancing the detection accuracy of diverse intrusion types while mitigating overfitting. We utilize the NSL-KDD dataset to assess the model's performance. Experimental results demonstrate that the proposed FLE-CNN framework outperforms existing intrusion detection schemes in terms of accuracy, recall, precision, and F1-score, while also addressing key security concerns related to data sharing and system scalability. This work presents a significant advancement toward secure, intelligent, and privacy-preserving intrusion detection in next-generation healthcare infrastructures.**

***Index Terms*— Federated Learning, Network Intrusion Detection, Smart Healthcare, Convolutional Neural Network, Data Privacy.**

I. Introduction

The evolution toward Industry 5.0 has revolutionized healthcare delivery through personalized and intelligent services powered by the Internet of Medical Things (IoMT), Artificial Intelligence, and real-time data analytics. However, the increased connectivity of medical systems has exposed them to numerous cybersecurity vulnerabilities. Network Intrusion Detection Systems play a crucial role in safeguarding these systems, particularly when sensitive patient data is constantly transmitted across networks. Traditional centralized NIDS approaches face challenges such as data privacy risks, communication overhead, and vulnerability to single points of failure.

To overcome these limitations, federated learning has emerged as a decentralized paradigm that facilitates collaborative model training without compromising data privacy. In this work, we propose an intrusion detection scheme that leverages an ensemble of convolutional neural network (CNN), Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Decision Tree (DT) trained within a federated learning framework. This novel approach ensures robust and generalized intrusion detection performance across heterogeneous and distributed healthcare systems while addressing the stringent privacy regulations prevalent in the medical domain.

Additionally, our model incorporates four key modules—Capture Engine, Cloud Server, Proxy Server, and Hospital Server—that work together to identify and respond to intrusions in real-time. The proxy server acts as a decoy to trap attackers and collect malicious patterns, while the capture engine continuously sniffs traffic for anomalies. Leveraging periodic federated training and secure model aggregation, FLE-CNN ensures faster predictions, reduced bandwidth use, and superior anomaly detection accuracy. This paper outlines the design, implementation, and evaluation of the proposed system using the NSL-KDD dataset, validated by performance metrics and formal security analysis using the Scyther tool. The results demonstrate the system's ability to deliver high accuracy while preserving privacy, making it an effective solution for safeguarding Industry 5.0 healthcare infrastructures.

A. Research Motivation

The increasing adoption of smart healthcare technologies in Industry 5.0 environments has introduced new challenges in ensuring the security and privacy of sensitive medical data. Wearable sensors, remote diagnostics, and AI-driven patient monitoring systems have made healthcare delivery more efficient and personalized. However, these interconnected systems also present a significantly expanded attack surface for cybercriminals. Traditional intrusion detection systems (IDS) that rely on centralized data collection and analysis struggle to keep up with modern threat landscapes. These systems often require large volumes of patient data to be transmitted to a central location for model training, raising serious privacy concerns and increasing bandwidth consumption. Moreover, centralized systems are prone to single points of failure and may not effectively adapt to evolving attack vectors across diverse hospital networks.

Recent advances in federated learning offer a promising solution by enabling collaborative model training across distributed nodes without sharing raw data. When combined with ensemble-based deep learning techniques, particularly Convolutional Neural Network (CNN), this approach enhances

detection accuracy, adaptability, and robustness. This leads the motivation for this research.

B. Research Contributions:

This work proposes a novel privacy-preserving intrusion detection framework for Industry 5.0 healthcare environments by integrating Federated Learning with Ensemble and Convolutional Neural Network (CNNs). Unlike traditional centralized models, our approach decentralizes the training process, enabling individual hospital nodes to collaboratively train an intrusion detection system (IDS) without sharing sensitive patient data. This ensures enhanced privacy compliance while also improving scalability and system robustness. The proposed FLE-CNN architecture incorporates an ensemble of CNNs, each trained with different configurations across nodes. The ensemble design improves detection accuracy, reduces overfitting, and enhances the generalizability of the system across diverse attack patterns. To support real-time intrusion detection, we introduce a modular architecture comprising a Capture Engine Module, a Cloud FL Server, a Proxy Server for decoy-based malware collection, and a Hospital Server acting as the primary data node. These modules work in unison to detect anomalies efficiently and provide immediate alerts. Also, we include a proxy server that acts as a honeypot to trap attackers. This proxy mimics real hospital servers and collects malware samples, which are then used to fine-tune the global detection model via federated learning. This approach not only enhances the system's learning capability but also improves the anomaly detection rate in future iterations. The framework is evaluated using the NSL-KDD dataset, where it demonstrates high accuracy, precision, recall, and F1-score in detecting network intrusions. Additionally, to validate the security of the proposed design, we employ the Scyther tool for formal security analysis. This verification confirms the system's resilience against multiple attack vectors such as man-in-the-middle attacks, DDOS, DOS, and port scans.

C. Organization of the paper

The paper's remaining sections are arranged as follows. The specifics of other comparable intrusion detection systems that are currently in use and relevant to smart healthcare are included in Section II. Section III lists the system design for the suggested FLE-CNN. Section IV provides the process of FLE-CNN's specifics. Additionally, Section V provides the FLE-CNN's practical implementation. Section VI then presents the crucial security analysis of FLE-CNN and also showcases the formal security verification of the proposed Federated leaning based CNN Ensemble Intrusion detection system.

II. Related Works

The integration of federated learning (FL) into intrusion detection systems (IDS) has garnered significant attention, particularly in the context of healthcare and IoT environments. A comprehensive survey by Belenguer et al. [1] categorizes FL-based IDS approaches, highlighting their potential in preserving data privacy while maintaining detection efficacy. Similarly, Agrawal et al. [2] discuss the challenges and future directions of FL in IDS, emphasizing the need for robust aggregation strategies and addressing issues related to data heterogeneity.

In the realm of healthcare, FIDChain [3] presents a blockchain-enabled federated learning (FL) framework specifically designed for IoT-based healthcare applications. The architecture combines the strengths of blockchain technology and federated learning to address critical challenges such as data privacy, traceability, and model integrity in distributed healthcare environments. By integrating blockchain, FIDChain ensures that all model updates and transactions are recorded in a tamper-proof ledger, thereby enhancing trust and accountability among participating nodes. The framework utilizes lightweight artificial neural networks (ANNs) to accommodate the limited computational resources of IoT healthcare devices, making it practical for real-world deployment in edge-based medical systems.

Another study by Almaghthawi et al. [4] introduces a blockchain-based federated learning intrusion detection system (FL-IDS) designed to enhance cybersecurity in distributed networks while preserving data privacy. In their approach, machine learning models are collaboratively trained across multiple nodes without the need to share raw data, addressing the critical privacy concerns associated with sensitive domains such as healthcare and industrial IoT. The framework employs blockchain technology to provide a decentralized, immutable ledger for tracking model updates and verifying the integrity of the contributions from each participating node.

For IoT networks, Nguyen and Beuran [5] propose FedMSE, a semi-supervised FL approach combining Shrink Autoencoder and Centroid one-class classifier to enhance anomaly detection in decentralized settings. Additionally, Gourceyraud et al. [6] explore an unsupervised FL-based IDS, introducing a federated K-means++ initialization technique to improve clustering performance without compromising data privacy.

In the context of the Internet of Healthcare Things (IoHT), a study by Almarashdeh et al. [7] proposes a privacy-preserving FL framework integrating differential privacy mechanisms to secure sensitive health data during collaborative model training. Their approach emphasizes minimizing the risk of data leakage during transmission by ensuring that only noise-perturbed model updates are shared among participating healthcare nodes. This not only safeguards the privacy of patients' medical records but also complies with regulatory standards like HIPAA and GDPR. The framework incorporates a lightweight deep learning model suitable for resource-constrained IoHT devices, enabling efficient real-time anomaly detection without compromising on performance. Additionally, the authors demonstrate how their system maintains high detection

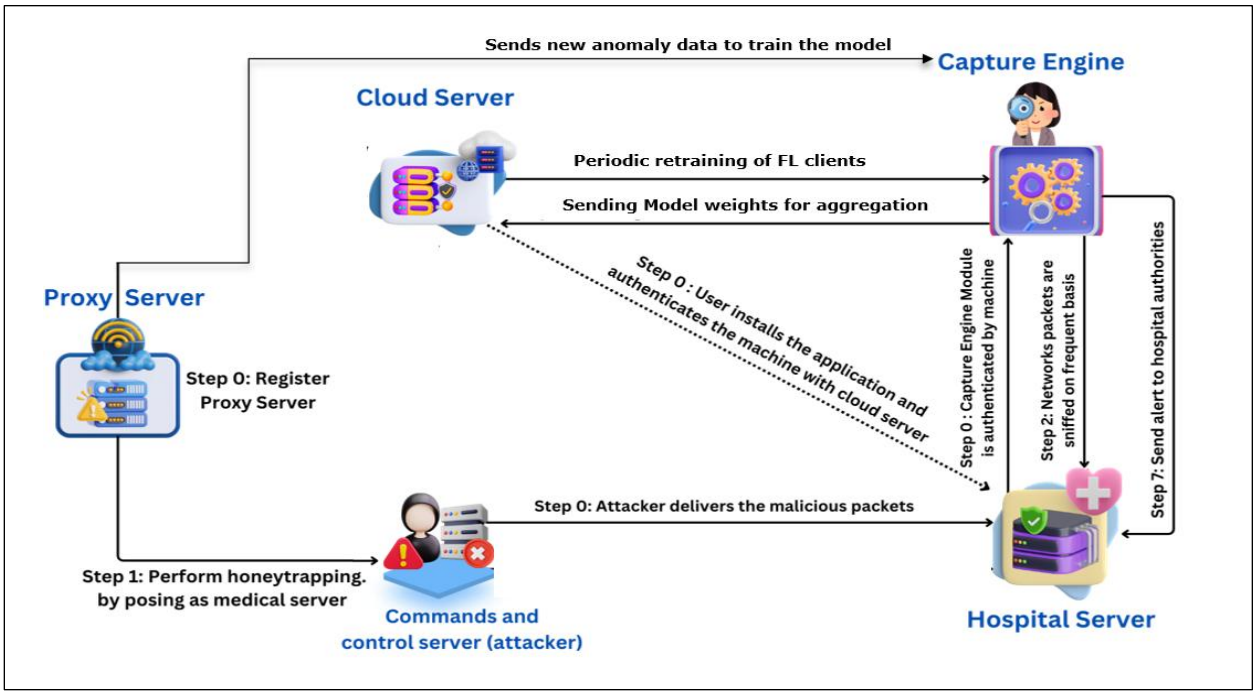

Fig. 1 Process flow of Proposed FLE-CNN IDS

accuracy while resisting membership inference and model inversion attacks, highlighting its practical applicability in real-world healthcare settings. This work serves as a foundational reference for integrating advanced privacy-preserving mechanisms into federated intrusion detection systems tailored for smart healthcare environments.

Lazzarini et al. [8] evaluated the use of federated learning for intrusion detection in IoT environments, employing shallow artificial neural networks and various aggregation algorithms. Their study demonstrated that federated learning approaches could achieve comparable performance to centralized models while preserving data privacy. The research also highlighted the impact of different aggregation methods on model efficacy.

These studies collectively underscore the efficacy of FL in enhancing IDS capabilities across various domains, addressing challenges related to data privacy, system scalability, and detection accuracy.

III. System Design

The Federated Learning-Based Ensemble Convolutional Neural Network (FLE-CNN) system shown in Fig.1 for intrusion detection in healthcare environments is structured around four main modules—Capture Engine Module, Cloud Server, Proxy Server, and Hospital Server—which together form a decentralized, privacy-aware, and intelligent intrusion detection framework. The Hospital Server Module acts as the central node within the hospital's secure network. It authenticates communication with the cloud server, orchestrates local training of the ensemble which consists of SVM, KNN, DT and a CNN model, and handles data from IoT devices and internal systems. It also integrates the capture engine and facilitates real-time anomaly detection. The hospital server accumulates insights from its own traffic patterns, including those from ESP32 sensors, and prepares model updates for federated training. The Cloud Server coordinates the federated learning process. It collects encrypted model weights (not raw data) from multiple hospital servers and applies the Federated Averaging to generate global weights. These weights are then distributed back to each client (hospital node) for the next round of local training. The use of federated learning ensures that patient-sensitive data remains localized, preserving privacy and complying with healthcare data protection regulations.

A key enhancement to this system is the Proxy Server Module, which acts as a decoy designed to interact with potential attackers. By mimicking a legitimate medical server, it attracts malicious actors and captures attempted intrusions and malware payloads. These interactions are not only logged but also transformed into labelled anomalous data points that are used to enrich the global model's knowledge base during future federated training cycles. At the edge of the network, the Capture Engine Module is deployed within the hospital's local infrastructure. It is responsible for continuously sniffing network traffic, including packets generated by connected healthcare IoT devices such as ESP32 boards with attached pulse sensors. These devices monitor patient vitals in real-time and communicate with the hospital server over Wi-Fi. The capture engine intercepts packets from such devices, extracts relevant features e.g., protocol type,

source/destination bytes, flag bits, and forwards them to the local ensemble CNN model for classification. The model then predicts whether each packet is benign or anomalous. Crucially, this process occurs without sending raw data outside the local network, aligning with the privacy goals of federated learning. Together, these modules form a cyclical and collaborative learning framework. Periodic retraining occurs using the newly captured anomalies and benign traffic, including data originating from medical IoT devices like the ESP32. This continuous improvement loop enhances the detection model's accuracy and adaptability while maintaining low latency, reduced bandwidth usage, and high data privacy.

IV. The Proposed Intrusion Detection Process

The proposed Federated Learning-based Ensemble CNN system in Fig.1 is designed to detect intrusions in a decentralized smart healthcare environment. The architecture involves interaction between the Cloud Server, Capture Engine, Hospital Server, Proxy Server, and the potential attacker's command and control server. Each step of the process illustrated in the diagram corresponds to critical events in both normal operation and during a malicious attack attempt.

Initially, when the Capture Engine Module is installed on a local machine, the user must authenticate the device with the Cloud Server. This step ensures that only verified machines participate in the federated learning process. The authentication process may include device verification and secure registration to prevent rogue devices from joining the FL network. Simultaneously, a Proxy Server is registered within the system. This proxy mimics a legitimate medical server and is strategically deployed to act as a honeypot. It is used to lure attackers and gather malicious traffic without affecting actual hospital infrastructure. This module plays a crucial role in proactively detecting new or zero-day attacks. The Proxy Server actively pretends to be a vulnerable or legitimate hospital device. It interacts with attackers or malware sources by exposing dummy services or interfaces. When an attacker attempts to exploit this fake node, their activities are recorded in detail—including commands, payloads, and attempted breaches. The Capture Engine begins frequent and continuous sniffing of network packets within the hospital LAN/Wi-Fi. It captures data from internal systems as well as from connected medical IoT devices like ESP32 with pulse sensors. These packets are passed through a local ensemble model, which classifies them as either benign or anomalous.

At this point, an attacker unknowingly delivers malicious packets to the Proxy Server, thinking it is a real hospital machine. These packets are logged, analysed, and labelled as anomalous data, which is crucial for enhancing the detection model. These samples will later be incorporated into the FL training loop to improve model robustness. If the local model identifies a packet as anomalous (e.g., a DoS attempt, scanning, or malware delivery), an alert is triggered and sent to hospital authorities. This real-time alerting mechanism helps IT security teams respond quickly to prevent data breaches, system downtime, or patient safety risks. In parallel, each hospital node periodically sends its model weights to the Cloud Server. The cloud server aggregates these updates using Federated Averaging and sends back the improved global model. Importantly, no raw patient or network data is shared—only encrypted model weights—thus preserving privacy and complying with medical data regulations. Newly captured attack data from the Proxy Server and Hospital Server (including malicious packets delivered by attackers or suspicious traffic from IoT devices) is periodically used to retrain local models. This updated data helps the models evolve and stay resilient against new and evolving threats

TABLE I
NOTATIONS USED IN EIDS-HS

Notations	Abbreviations
FLE-CNN	Federated learning-based Ensemble CNN scheme to detect intrusions in the industry 5.0-driven healthcare system
DL	Deep learning
DoS	Denial-of-service attack
IDS	Intrusion detection systems
CNN	Convolutional neural network
SVM	Support Vector Machine
KNN	*k*-nearest neighbours
TP	True positive
FP	False positive
TN	True negative
FN	False negative

The details of the notations used in the proposed EIDS-HS are given in Table I.

A. Data Acquisition

The proposed FLE-CNN intrusion detection system was evaluated using the NSL-KDD dataset [9], a refined and improved version of the widely used KDD CUP 99 dataset. This dataset includes a wide range of attack types such as denial-of-service (DoS), probing and various attacks. It contains records of approximately five million network connections, each characterized by 41 features. These features encompass various aspects of network behavior, such as protocol type, service, duration, and source/destination statistics, making the dataset well-suited for intrusion detection research. In the context of this study, a binary classification approach was adopted. The goal was to classify each record as either a normal connection or an intrusion. The dataset includes a total of 79,179 normal instances and 81,161 intrusion instances, providing a balanced and realistic testing ground for evaluating the performance of FLE-CNN.

B. Preprocessing:

To prepare the dataset for model training and evaluation, several preprocessing steps were undertaken. Initially, all duplicate records were removed to ensure quality of the dataset.

Algorithm 1: Federated Local Ensemble Model Training (FL Clients)

Input: Local dataset Di at each hospital
Parameters: k for KNN, number of communication rounds R, Real-time packet stream P
Output: Model weights: $wSVM, wKNN, wDT, wCNN$ *and Prediction*: $yFinal$
Training Phase:
1. For each hospital client Hi ∈ H:
2. Load the local dataset Di
3. Remove duplicate records
4. Split Di into feature matrix X and target variable y
5. Perform train-test split (70/30)
6. Apply standard scaling to numeric features
7. Apply one-hot encoding to categorical features in X
8. Apply label encoding to the target variable y
9. Apply PCA to reduce dimensionality
10. Train models: SVM, KNN (with k), DT, and CNN
11. Eval model accuracies: $aSVM, aKNN, aDT, aCNN$
12. Compute normalized ensemble weights:

$$w_j = \frac{a_j}{(aSVM + aKNN + aDT + aCNN)},$$

$$for\ j \in \{SVM, KNN, DT, CNN\}$$

13. Send local weights wj to the cloud server
14. **End For**
Real-Time Prediction Phase:
15. **For each incoming packet p ∈ P:**
16. Apply preprocessing steps (scale, encode, PCA) to the real-time packet data.
17. Run p through each trained model and get the predictions: $ySVM, yKNN, yDT, yCNN$
18. Compute the ensemble prediction using current weights:

$$yFinal = w1 \times p1 + w2 \times p2 + w3 \times p3.$$

19. If $yFinal = Anomaly$, then raise alert to Hospital Authorities.
20. End For

Algorithm 2: Federated Ensemble Aggregation and Global Update (Cloud Server)

Input: Weights from all Hospital clients
Output: Global ensemble weights $wSVM, wKNN, wDT, wCNN$
1. For each communication round r = 1 to R:
2. Receive local weights of all models from all hospital clients Hi ∈ H.
2. Take average of each of the model weights sent by all Hospital Clients which are the global weights.
3. Send the calculated global weights of each model to all the Hospital clients.
4. Clients update their local ensemble model with the global weights.
4. **End For**

Following this, standard scaling was applied to all numerical features to normalize the data and bring the features onto a comparable scale. This step is essential for improving the convergence and performance of neural network models such as FLE-CNN. Next, one-hot encoding was used to transform categorical features into a suitable numerical representation. Since many machine learning models, including convolutional neural networks, require numerical input, this transformation enabled categorical attributes such as protocol types and service types to be included in the model training. Also, label encoding was done to the target variable to prepare it for ML tasks.

To reduce the dimensionality of the dataset and improve computational efficiency, Principal Component Analysis (PCA) was employed. PCA helped retain the most informative components while discarding less significant features, thus preserving maximum variance in a lower-dimensional space. This step was particularly beneficial in enhancing the generalization ability of the FLE-CNN model. The entire data preparation pipeline was structured and implemented as described in Algorithm 1. This included the division of the dataset into features (X) and labels (y), followed by a 70/30 split for training and testing. The training data was used to fit scalers and PCA transformers, which were then applied to the testing data to ensure consistency. Afterward, the preprocessed training set was used to train the FLE-CNN model, and the performance was evaluated on the testing set. Where necessary, hyperparameter tuning was conducted to optimize the model.

C. Federated Learning Framework:

In the proposed Federated Learning Ensemble CNN framework, each hospital functions as an independent client with its own local dataset. Instead of sharing sensitive patient data, each client performs local training using various machine learning models - Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree (DT), and Convolutional Neural Network (CNN). The data undergoes preprocessing steps including duplication removal, feature scaling, encoding, and dimensionality reduction using Principal Component Analysis. Once trained, the performance of each model is evaluated locally, and their accuracies are used to compute normalized ensemble weights that reflect their individual contribution to the final prediction.

These local model weights are then securely transmitted to a central cloud server for aggregation, where global ensemble weights are constructed based on the collected client updates. This decentralized approach ensures data privacy while enabling collaborative model improvement across distributed medical institutions. Additionally, this methodology enhances

the robustness of models by capturing diverse weights from multiple sources, which may not be present in a single dataset. As a result, the federated ensemble learning framework contributes to a more generalized and accurate intrusion detection system, particularly suitable for sensitive domains like healthcare where data confidentiality is paramount.

TABLE II
DETAILS OF SIMULATION PARAMETERS

Parameter	Value
Operating system	Microsoft Windows(R) 11 Home
Platform used	Visual Studio Code, Jupyter Notebook
Processor	8 × Intel(R) i5 CPU @ 2.20GHz
GPU	4 GB Intel(R) Xeon Graphics
Random access memory (RAM)	8 GB
Programming language	Python 3.8
Used libraries	Scikit-learn, Scapy, Numpy
Machine learning models	Support Vector Machine (SVM), K-NearestNeighbours(KNN), Decision Tree, Convolutional Neural Network (CNN)

V. PRACTICAL IMPLEMENTATION

This section describes the practical execution setup of the proposed federated intrusion detection framework, outlining the hardware specifications and software tools employed. The implementation was carried out on a machine running Microsoft Windows 11 Home, equipped with an Intel(R) i5 processor featuring 8 logical cores, operating at 2.20 GHz, and supported by 8 GB of RAM. The graphical processing was handled by an integrated 4 GB Intel Xeon Graphics unit.

Development and simulation were performed using Visual Studio Code and Jupyter Notebook environments. The programming was done in Python 3.8, utilizing prominent libraries such as Scikit-learn for machine learning, Scapy for packet manipulation, and NumPy for numerical computations. Detailed simulation parameters are presented in Table II.

A range of machine learning algorithms was deployed on the local client datasets, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), and Convolutional Neural Network (CNN), to train individual models at each participating client.

Evaluation Metrics:

To assess the performance of the proposed system, four primary evaluation metrics were considered: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP indicates instances where malicious activities were correctly detected as threats, while TN denotes cases where benign traffic was accurately recognized as safe. Conversely, FP accounts for situations where normal behaviour was wrongly flagged as malicious, and FN represents instances where actual threats were incorrectly classified as normal.

*Accuracy***:** Measures the proportion of correctly predicted instances among all observations.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Recall (Sensitivity): Indicates the proportion of actual positives that were identified correctly.

$$Recall = \frac{TP}{TP + FN}$$

Precision: Reflects the proportion of positive predictions that were indeed correct.

$$Precision = \frac{TP}{TP + FP}$$

F1-Score: Represents the harmonic mean of precision and recall, giving a balanced measure of accuracy in imbalanced datasets.

$$F1 - Score = \frac{2\ x\ (Precision\ x\ Recall)}{Precision + Recall}$$

These metrics were used to comprehensively evaluate the detection capabilities of the proposed system and ensure its effectiveness in identifying cyber threats with high precision and reliability.

TABLE III
COMPARISONS OF DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	95.33%	96.86%	94.35%	94.10%
SVM	92.00%	90.96%	88.62%	91.77%
KNN	94.78%	92.17%	94.24%	95.00%
EIDS-HS	95.12%	93.11%	95.27%	94.18%
FLE-CNN (Proposed)	**97.67%**	**95.33%**	**94.27%**	**95.52**

Table III illustrates the comparative performance metrics including accuracy, precision, recall, and F1-score for various machine learning models and the proposed framework. These evaluation criteria are essential for understanding each model's effectiveness in correctly identifying malicious and benign instances in the dataset.

From the Table III, it is evident that FLE-CNN outperforms all other models across all metrics. It achieved the highest accuracy of 97.67%, demonstrating superior capability in correctly classifying data instances. Additionally, it recorded a precision of 95.33%, recall of 94.27%, and a leading F1-score of 95.52%, indicating a balanced performance in terms of both sensitivity and specificity. In comparison, the EIDS-HS [10] framework, which integrates multiple models in an ensemble, showed 95.12% accuracy and a F1-score of 94.18%, making it the second-best performer. FLE-CNN also notably surpassed the traditional classifiers such as Decision Tree, SVM, and KNN in most metrics.

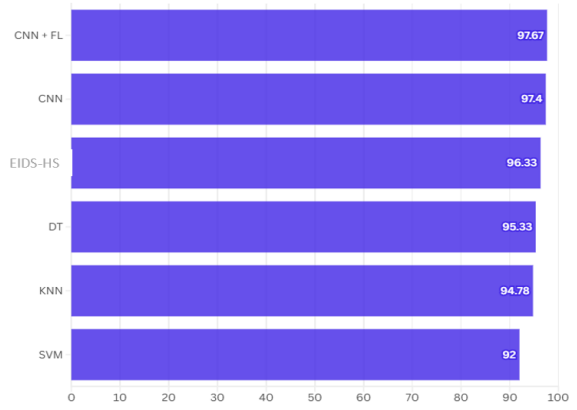

Fig. 2. Accuracy values of different methods.

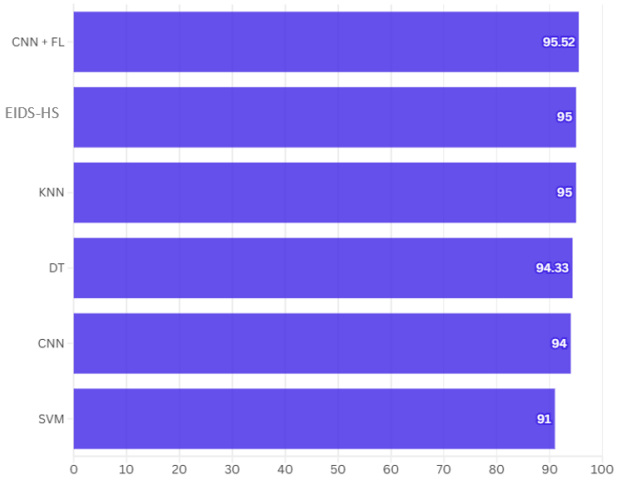

Fig. 3. F1-Score values of different methods.

VI. Security Verification

The security features compared across various schemes in Table IV are denoted as ΥF_1 to ΥF_8. Specifically, ΥF_1 refers to the accuracy of the scheme; ΥF_2 indicates whether the scheme provides secure session key establishment; ΥF_3 refers to the support for standard mutual authentication among different entities; ΥF_4 denotes whether secure data exchange is ensured; ΥF_5 addresses protection against data leakage attacks; ΥF_6 signifies the ability to maintain the integrity of exchanged data; ΥF_7 reflects the availability of machine learning or deep learning-based intrusion detection mechanisms; and ΥF_8 represents the availability of formal security verification using the Scyther tool; and ΥF_9 denotes the availability of FL-based Framework "YES" indicates the scheme supports or ensures that specific feature, while "NO" means the feature is either unsupported or insecure in the given scheme. Our FLE-CNN has a computational complexity of O(n).

TABLE IV
COMPARISONS OF DIFFERENT METHODOLOGIES

Features	Sherin et al. [28]	Esmaeili et al. [29]	Zou et al. [30]	Ketepalli et al. [31]	EIDS-HS [10]	FLE-CNN (Proposed)
$\Upsilon F1$	81.67%	82.80%	85.95%	94.74%	95.12%	**97.67%**
$\Upsilon F2$	NO	NO	NO	NO	YES	**YES**
$\Upsilon F3$	NO	NO	NO	NO	YES	**YES**
$\Upsilon F4$	NO	NO	NO	NO	YES	**YES**
$\Upsilon F5$	NO	NO	NO	NO	YES	**YES**
$\Upsilon F6$	NO	NO	NO	NO	YES	**YES**
$\Upsilon F7$	YES	YES	YES	YES	YES	**YES**
$\Upsilon F8$	NO	NO	NO	NO	YES	**YES**
$\Upsilon F9$	NO	NO	NO	NO	NO	**YES**

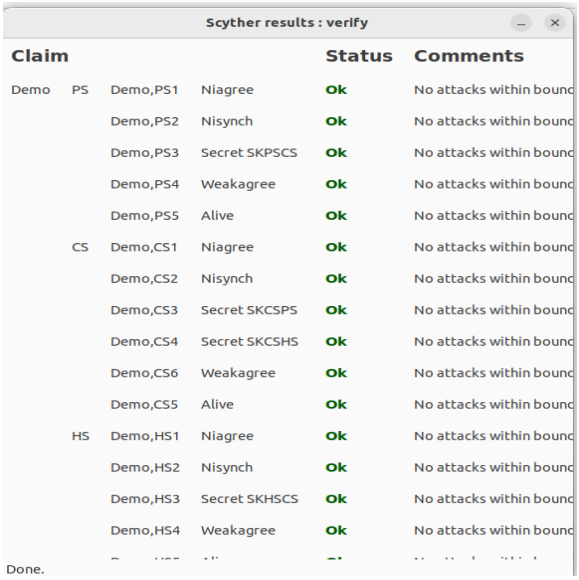

Fig. 4. Outcome of formal security verification.

The Fig. 4 demonstrate the formal security verification of our intrusion detection system using the Scyther tool, a popular tool for the automatic verification of security protocols. The first screenshot shows the verification results, where all defined security claims for three roles Proxy Server-PS, Cloud Server-CS, and Hospital Server-HS have passed successfully. Each

claim, such as Niagree, Nisynch, Secret, Weakagree, and Alive, has a status of "OK", with comments indicating "No attacks within bounds", confirming that the protocol resists known attack patterns within the defined model boundaries. The Fig. 5 displays the protocol specification in Scyther's input language, outlining the interactions and cryptographic assumptions between the roles. The protocol defines secure exchanges and uses claims to verify critical security properties like authentication, secrecy, and agreement. The successful verification validates the robustness and correctness of the protocol used.

```
const SKPSCS:Function; const SKCSPS:Function;
const SKHSCS:Function; const SKCSHS:Function;
protocol Demo(PS,CS,HS)
{
  role PS
  {
    const TS1,TS2,DT1,DT2;
    send_1(PS,CS,{DT1}SKPSCS,TS1);
    recv_2(CS,PS,{DT2}SKCSPS,TS2);
    claim_PS1(PS,Niagree);
    claim_PS2(PS,Nisynch);
    claim_PS3(PS,Secret,(SKPSCS));
    claim_PS4(PS,Weakagree);
    claim_PS5(PS,Alive);
  }
  role CS
  {
    const TS1,TS2,DT1,DT2,TS3,TS4,DT3,DT4;
    recv_1(PS,CS,{DT1}SKPSCS,TS1);
    send_2(CS,PS,{DT2}SKCSPS,TS2);
    send_4(CS,HS,{DT4}SKCSHS,TS4);
    recv_3(HS,CS,{DT3}SKHSCS,TS3);
    claim_CS1(CS,Niagree);
    claim_CS2(CS,Nisynch);
    claim_CS3(CS,Secret,(SKCSPS));
    claim_CS4(CS,Secret,(SKCSHS));
    claim_CS4(CS,Weakagree);
    claim_CS5(CS,Alive);
  }
  role HS
  {
    const TS3,TS4,DT3,DT4;
    recv_4(CS,HS,{DT4}SKCSHS,TS4);
    send_3(HS,CS,{DT3}SKHSCS,TS3);
    claim_HS1(HS,Niagree);
    claim_HS2(HS,Nisynch);
    claim_HS3(HS,Secret,(SKHSCS));
    claim_HS4(HS,Weakagree);
    claim_HS5(HS,Alive);
  }
}
```

Fig. 5. SPDL snippet of the proposed FLE-CNN.

VII. Conclusion

Our proposed FLE-CNN, a Federated Learning-based Ensemble Convolutional Neural Network Intrusion Detection System for Healthcare Sector provides efficient and secure intrusion detection in distributed environments. The model demonstrated superior performance across multiple evaluation metrics, achieving an accuracy of **97.67%**, precision of **95.33%**, recall of **94.27%**, and F1-score of **95.52%**, outperforming traditional machine learning models such as Decision Tree, SVM, KNN, and even existing ensemble methods like EIDS-HS [10]. Additionally, the proposed system addressed key security and functionality requirements that were lacking in previous schemes, and formal verification using Scyther tool.

As part of future work, the proposed FLE-CNN framework can be further extended to support real-time intrusion detection across diverse IoT and edge computing environments. We also plan to enhance the scalability of the system by incorporating lightweight federated optimization algorithms. Moreover, integrating explainable AI (XAI) techniques could further improve transparency and trust in critical decision-making processes, especially in healthcare and industrial control systems.

References

[1] A. Belenguer, J. Navaridas, and J. A. Pascual, "A review of Federated Learning in Intrusion Detection Systems for IoT," *arXiv preprint arXiv:2204.12443*, 2022.

[2] S. Agrawal el., "Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions," *arXiv preprint arXiv:2106.09527*, 2021.

[3] A. M. Abdel-Basset et al., "FIDChain: Federated Intrusion Detection System for Blockchain-Enabled IoT Healthcare Applications," *Healthcare*, vol. 10, no. 6, p. 1110, 2022.

[4] A. Almaghthawi et al., "Federated-Learning Intrusion Detection System Based Blockchain Technology," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 20, no. 11, pp. 16–30, 2024.

[5] V. T. Nguyen and R. Beuran, "FedMSE: Federated learning for IoT network intrusion detection," *arXiv preprint arXiv:2410.14121*, 2024.

[6] M. Gourceyraud et al., "Federated Intrusion Detection System Based on Unsupervised Machine Learning," *arXiv preprint arXiv:2503.22065*, 2025.

[7] A. Almarashdeh et al., "Privacy-Preserving Federated Learning-Based Intrusion Detection System for IoHT Devices," *Electronics*, vol. 14, no. 1, p. 67, 2025.

[8] Riccardo Lazzarini et al., "Federated Learning for IoT Intrusion Detection", AI 2023, 4(3), 509-530; https://doi.org/10.3390/ai4030028

[9] "NSL-KDD dataset." Accessed: Apr. 2025. [Online]. Available: http://nsl.cs.unb.ca/nsl-kdd/

[10] Mohammad Wazid et. al., "An Ensemble-Based Machine Learning-Envisioned Intrusion Detection in Industry 5.0-Driven Healthcare Applications", IEEE Transactions on Consumer Electronics, Volume: 70, Issue: 1, February 2024, DOI: 10.1109/TCE.2023.3318850

[11] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," Front. Comput. Sci., vol. 14, pp. 241–258, Apr. 2020.

[12] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," IEEE Access, vol. 7, pp. 82512–82521, 2019.

[13] G. Thamilarasu, A. Odesile, and A. Hoang, "An intrusion detection system for Internet of Medical Things," IEEE Access, vol. 8, pp. 181560–181576, 2020.

[14] M. M. Alani and A. I. Awad, "An intelligent two-layer intrusion detection system for the Internet of Things," IEEE Trans. Ind. Informat., vol. 19, no. 1, pp. 683–692, Jan. 2023.

[15] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, A. K. M. N. Islam, and M. Shorfuzzaman, "Permissioned blockchain and deep learning for secure and efficient data sharing in industrial healthcare systems," IEEE Trans. Ind. Informat., vol. 18, no. 11, pp. 8065–8073, Nov. 2022.

[16] Z. Chen, J. Duan, L. Kang, and G. Qiu, "Class-imbalanced deep learning via a class-balanced ensemble," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 10, pp. 5626–5640, Oct. 2022.

[17] V. Stephanie, I. Khalil, M. S. Rahman, and M. Atiquzzaman, "Privacypreserving ensemble infused enhanced deep neural network framework for edge cloud convergence," IEEE Internet Things J., vol. 10, no. 5, pp. 3763–3773, Mar. 2023.

[18] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," Appl. Sci., vol. 9, no. 20, p. 4396, Oct. 2019.

[19] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," IEEE Access, vol. 9, pp. 7550–7563, 2021.

[20] J. Liu, Z. Tian, R. Zheng, and L. Liu, "A distance-based method for building an encrypted malware traffic identification framework," IEEE Access, vol. 7, pp. 100014–100028, 2019.

[21] Y.-L. Wan, J.-C. Chang, R.-J. Chen, and S.-J. Wang, "Feature-selectionbased ransomware detection with machine learning of data analysis," in Proc. 3rd Int. Conf. Comput. Commun. Syst. (ICCCS), 2018, pp. 85–88.

[22] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

[23] P. Prasse, L. Machlica, T. Pevný, J. Havelka, and T. Scheffer, "Malware detection by analysing network traffic with neural networks," in *Proc. IEEE Security Privacy Workshops (S&P)*, 2017, pp. 205–210.

[24] M. Piskozub, R. Spolaor, and I. Martinovic, "MalAlert: Detecting malware in large-scale network traffic using statistical features," *SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 151–154, 2019.

[25] R. Abedin and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, pp. 1–22, 2022.

[26] R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy, and T. Reddy Gadekallu, "Malicious URL Detection using Logistic Regression," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Barcelona, Spain, 2021, pp. 1–6.

[27] R. M. Swarna Priya et al., "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Comput. Commun.*, vol. 160, pp. 139–149, Jul. 2020.

[28] V. J. Immanuel Jeo Sherin and N. Radhika, "Stacked ensemble-IDS using NSL-KDD dataset," *J. Pharmaceut. Negative Results*, vol. 13, no. 3, pp. 351–356, 2022.

[29] M. Esmaeili, S. H. Goki, B. H. K. Masjidi, M. Sameh, H. Gharagozlou, and A. S. Mohammed, "ML-DDoSnet: IoT intrusion detection based on denial-of-service attacks using machine learning methods and NSL-KDD," *Wireless Commun. Mobile Comput.*, vol. 2022, Aug. 2022, Art. no. 8481452.

[30] L. Zou, X. Luo, Y. Zhang, X. Yang, and X. Wang, "HC-DTTSVM: A network intrusion detection method based on decision tree twin support vector machine and hierarchical clustering," *IEEE Access*, vol. 11, pp. 21404–21416, 2023.

[31] G. Ketepalli and P. Bulla, "Feature extraction using LSTM autoencoder in network intrusion detection system," in *Proc. 7th Int. Conf. Commun. Electron. Syst. (ICCES)*, Coimbatore, India, 2022, pp. 744–749.

[32] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[33] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[34] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smartcard security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[35] "NSL-KDD dataset." Kaggle Accessed: Mar. 2025. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[36] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, Ottawa, ON, Canada, 2009, pp. 1–6.

[37] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Netw.*, vol. 25, no. 8, pp. 4737–4750, Nov. 2019.

[38] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKEIoD: Lightweight authenticated key exchange protocol for the Internet of Drone environment," *IEEE Access*, vol. 8, pp. 155645–155659, 2020.

[39] C. J. F. Cremers. "Scyther: Semantics and verification of security protocols." 2022. Accessed: Nov. 2022. [Online]. Available: https://pure.tue.nl/ws/files/2425555/200612074.pdf

[1] **Dr. T. Revathi** is currently serving as the Senior Professor and Head of the Department of Information Technology at Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India. She joined the institution in July 1986 and has over 38 years of teaching experience. She holds a Bachelor's degree in Electrical and Electronics Engineering from Madurai Kamaraj University (1986), an M.E. in Computer Science from Bharathiar University (1995), and earned her Ph.D. in Computer Networks from Manonmaniam Sundaranar University in 2008. Dr. Revathi's research interests include Big Data and Sensor Networks. Over the course of her academic career, she has guided 14 Ph.D. scholars to completion, with 1 currently pursuing under her supervision. She has an extensive publication record, including 61 journal papers and 44 conference papers, and has delivered 27 guest lectures at various academic and research forums. Additionally, she has contributed to the field through 5 published books, and has 3 patents granted, 4 patents published, and 1 patent filed.
She has successfully completed 3 funded projects and has held key academic positions.

[2] **Mr. R. Prabhu** is a student of Information Technology Department at Mepco Schlenk engineering college, Sivakasi.

[3] **Mr. K. Vijesh Pethuram** is also a student of Information Technology Department at Mepco Schlenk engineering college, Sivakasi