Mesopotamian journal of Cybersecurity
Vol.5,No.3, **pp**. 1324–1339
DOI: https://doi.org/10.58496/MJCS/2025/068; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Enhancing Advanced Persistent Threat Detection with Federated Learning and Neural Networks for Secure Cloud Computer Environment

Baydaa Flayyih Hasan [1,*], Wafaa Ayoub Kassara [1], Bushra Raad Zahi [1]

[1] *Technical College of Management , Baghdad, Middle Technical University, Baghdad, Iraq.*

ARTICLE INFO

Article History

Received 23 Apr 2025
Revised 20 Sep 2025
Accepted 30 Oct 2025
Published 14 Dec 2025

Keywords

Advanced Persistent Threats (APTs), federated learning (FL), Exploratory Data Analysis (EDA), Gated Recurrent Unit (GRU) model, Convolution neural networks (CNN) model

ABSTRACT

Rapid internet expansion and global cloud storage use have heightened the risk of stealthy, persistent, multi-stage Advanced Persistent Threats (APTs). Distributed and resource-limited cloud environments make identifying these stealthy and dynamic threats difficult for traditional Intrusion Detection Systems (IDSs). This paper present FedNN-APT, a Federated Learning (FL) and hybrid Neural Network (NN) APT detection system to overcome these difficulties. High detection accuracy and distributed, privacy-preserving training over several cloud devices are achieved by this approach. FedNN-APT integrates Gated Recurrent Units (GRUs) and Convolutional Neural Networks (CNNs) to learn temporal and spatial APT behavior features effectively. The framework trains local models on partitioned datasets using GRU-CNN, 1D-CNN, and GRU-Recurrent Neural Network (RNN) models, then selects the optimal model for federated aggregation. The final global model is collaboratively built while preserving data confidentiality. The system is evaluated using an APT Malware dataset consisting of 11,107 samples. Experimental findings reveal that the hybrid GRU-CNN model outperforms other models with an average accuracy of 0.9977, precision of 0.9989, recall of 1.00, and F1-score of 0.9988. The federated model has 0.99global accuracy across four clients. A comparative evaluation with current APT detection systems underscores the superiority of FedNN-APT, especially regarding detection accuracy and flexibility in resource-limited environments. Finally, in this paper introduced FedNN-APT, an advanced approach that combines federated learning and neural networks for the detection of APT attacks while enhancing data privacy via a cloud environment and the findings indicate that incorporating deep learning models into a federated learning framework offers a promising direction for future study in safe and scalable threat detection in cloud systems.

1. INTRODUCTION

The rapid advancement of cloud computing technologies has made cloud environments necessary for organizations and companies to manage and store data. Cloud Computing facilitates ubiquitous, accessible, on-demand network access to a shared pool of customizable computing resources (e.g., servers, networks, storage, services, and applications) that can be swiftly supplied and released with minimum administrative effort or service provider involvement [1]. Nonetheless, many cloud computing adopters worry about security concerns [2].

APTs represent one of the most perilous and intricate forms of assaults aimed at computing systems. APTs are recognized for their prolonged stealth, ability to steal critical data, and power to disrupt systems. They mostly focus on government entities, essential infrastructure, and commercial enterprises [3]. The enhanced skills of attackers and their proficiency in executing multi-stage, complex assaults on target systems are seen in the significant levels of concealment, variety, and complexity characteristic of APT attacks. APT assaults possess a greater attack area in cloud computing networks due to their distributed and scalable characteristics [4].

APT adversaries exploit cloud computing platforms' dynamic design and shared resource characteristics to execute covert and prolonged infiltrations. Additionally, storing and retrieving substantial volumes of sensitive data in cloud computing systems motivates APT attackers to execute intricate and targeted assault operations. Detection and defense are more challenging in these assaults due to their multi-phase nature, encompassing initial penetration, lateral movement, data exfiltration, and ultimate escape [5].

[*] *Corresponding author. Email: Baydaa@mtu.edu.iq*

In contrast with traditional computing environments, where attack methods tend to be more fixed and apparent, moreover, APT assaults can affect multiple companies according to shared resources and dependence, threatening not only the data security of individual companies but also potentially resulting in extensive data breaches and operational disruptions [6]. Consequently, the investigation and development of distinctive dynamic defensive strategies to effectively resist APT attacks has emerged as a significant priority for current cybersecurity. Nevertheless, traditional defensive measures encounter several obstacles in addressing APT threats.

A prevalent traditional technique for identifying assaults involves employing specialized Penetration Testing (PT). Despite the ongoing advancement of technologies, current penetration testing methodologies are becoming increasingly nonstandard, complex, and resource-demanding.

Also, Traditional feature-based or rule-based detection methods struggle to address small changes in extensive data due to the large amount and complexity associated with cloud computing systems [7]. Current APT detection methods have inherent limitations, as IDSs rely on signatures or behavioral patterns that APTs intentionally avoid. While advanced APT detection systems may enhance prevention, sophisticated APTs can bypass these defenses. Additionally, these systems often produce high false-positive rates, a lack of generalization, and an inability to handle decentralized data or maintain privacy [6].

Due to the limitations of the techniques employed to address this issue, APT assaults concern security businesses. Thanks to the Artificial Immune System (AIS), Deep Learning (DL), and Machine Learning (ML) algorithms, the use of artificial intelligence (AI) in APT intrusion detection is becoming increasingly popular among researchers. Security experts say these techniques might address APT and other cybersecurity issues [8].

Recently, Researchers are undertaking FL investigations concerning APT identification via cloud computing devices as a potential remedy to these challenges. It is a machine learning approach that enables algorithm training without necessitating data transmission among several decentralized devices that retain local data samples. FL is particularly advantageous for cloud computing environments and can facilitate the development of a global model without necessitating substantial processing resources or transmitting potentially sensitive data [4].

FL-based methodologies protect against such attacks by effectively identifying anomalies and complying with cloud device limitations. The academic community is rigorously exploring innovative FL-based strategies, which will likely influence cloud cybersecurity in the future [9]. In light of the studies above, researchers propose a federated learning framework that integrates multiple neural network architectures—GRU-CNN, 1D-CNN, and GRU-RNN—to improve APT detection in resource-limited cloud environments. The suggested approach, known as FedNN-APT, utilizes temporal and spatial feature extraction advantages while ensuring data privacy via distributed model training. The data are pre-processed, an essential step in improving forecast accuracy relative to raw data, differentiating our methodology from previous approaches.

The proposed system's main contribution is integrating NN with FL utilizing advanced cutting-edge NN architectures, such as hybrid GRU-CNN, 1D-CNN, and GRU-RNN models. Specifically, the proposed FedNN-APT system:

1. Presents novel hybrid GRU-CNN architecture throughout a federated learning framework, facilitating temporal and spatial data extraction for accurate APT detection.
2. The study presents an approach comprising several pre-processing processes, such as exploratory data analysis and noise elimination. This essential element enhances predictive accuracy compared to the use of raw data. When combined, these procedures achieve a high degree of precision in evaluating the system's state.
3. It guarantees data privacy preserving by training models locally on clients' devices, preventing the sharing of raw data and protecting sensitive information in cloud computing environments.
4. Assesses computational efficiency by examining training duration and memory usage, validating the model's appropriateness for implementation in resource-limited cloud environments.

The rest of this paper is organized as follows: Section 2 presents the relevant work on FL methods and APT detection. Section 3 illustrates the principle of an APT attack. Section 4 proposes the FedNN-APT model, Section 5 offers the experimental findings, while Section 6 presents the limitations of this paper, and Section 7 presents the conclusions and future work plans.

2. RELATED WORKS

Recent research has concentrated on detecting APTs using various methodologies. These technologies aim to address the shortcomings of traditional IDSs, which often face difficulties associated with complex attack patterns, scalability, and data privacy issues. This section addresses Federated Learning-Based APT Detection Schemes and Neural Network-Based APT Detection Schemes.

In 2021, X. Cheng et al. [9] proposed the differential privacy-enhanced FL for APT prediction in Internet of Things (IoTs) devices, leveraging 5G edge frameworks to mitigate IoT computational and communication challenges. Although proficient in dynamic reaction, it requires extensive training data and complex synchronization.

In 2022, H. T. Thi et al. [10] presented an FL technique for identifying cyber threats in networks with “Software Defined Networking” (SDN) enabled. Threat information from partners is used to identify and respond to APT attacks in a proactive manner. Nonetheless, the lack of validation through real-world implementation constitutes a restriction.
In 2023, H. K. Alkhpor and F. M. Alserhani [11], developed an intelligent detection system using federated learning to identify APTs. They trained for machine learning models under the framework of centralized learning. They implemented CNN-based federated learning (CNN-FL) across multiple clients to ensure privacy and integrity. The proposed model implementation using the “UNSW-NB15 intrusion detection dataset” reached about 0.9018. The model fails to integrate temporal dependencies in network traffic, constraining its capacity to detect intricate APT attacks.
In 2023, H. Zhu et al. [12], introduced distinctive methodology, "Global Vision Federated Learning" (GV-FL), uses FL to accurately and effectively identify APT in IoT devices with limited resources. They test the suggested GV-FL framework on a practical IoTs device, the Raspberry Pi. The methodology was not assessed for classification accuracy or its robustness against temporal fluctuations. Also, numerous studies have utilized neural network methodologies to identify APT attacks.
In 2021, F. J. Abdullayeva [13] proposed a deep autoencoder neural network for classifying APT attacks, using deep learning to identify features automatically. This autoencoder learns these features first from training data, then from a SoftMax regression layer, achieving 0.9832accuracy on the "APT Malware dataset". The model has not undergone evaluation in a cloud or distributed setting, nor has it been combined with FL privacy protection methodologies.
In 2021,C. Do Xuan and M. H. Dao [14] introduced a new methodology for detecting APT assaults utilizing a Deep Neural Network (DNN) that integrates "multilayer perceptron" (MLP), CNN, and "long short-term memory" (LSTM). It collects IP attributes from network traffic and classifies them to detect APT assaults, attaining an accuracy range of 0.93 to 0.98 utilizing an NSL-KDD dataset. This approach enhances APT detection capabilities but cannot adapt to contemporary cloud infrastructures.
In 2021, M. Mamun and K. Shi [15] suggested deep task APT, a deep-learning model using a task-tree approach to detect insider APT. Combining LSTM and CNN analyzes user logs to predict malicious activities through behavioral anomaly detection. This model implementation utilizes the "OpTC dataset," the proposed model achieves an accuracy score of about 98.3%. It exclusively addresses insider attacks and does not consider the use of detecting multi-stage APT in cloud environments.
In 2023, G. Xiang et al. [16] introduced a new model for extracting APT attack events from web texts. It defines APT event types and schemas, constructs a Chinese dataset for APT event extraction, and proposes a “BERT-BiGRU-CRF-based” model. The proposed “BERT-BiGRU-CRF” shows promising results with a precision of 0.7013, a recall of 0.8011, and a score of 0.7479. This study is confined to textual analysis without examining network traffic; furthermore, the performance is restricted to an accuracy of 0.7479 which is inadequate for essential applications.
In 2023, W. Ren [17] suggested APT detection based on the graph CNN model (GCNN). The aim is to detect known attacks by leveraging vulnerabilities and attack contexts, using a knowledge graph processed with GCNN to enhance AT detection. The GCNN model implementation on the newly collected dataset achieved an accuracy score of about 0.959. The model exclusively identifies known assaults and is incapable of detecting other threats.
In 2024, C. Do Xuan and N. H. Cuong [18] proposed a new approach combining deep learning and attention networks for APT attack detection. CNN-LSTM networks analyze the preprocessed network traffic data. The classification the data using an attention network instead of direct classification. The CNN-LSTM-ATTENTION implementation on the network traffic dataset achieved an accuracy score of about 0.992. The framework is centralized data, which conflicts with the privacy requirements associated with cloud computing.
In 2024, H. Li et al. [19] presented an approach that included constructing a fusion provenance graph, transforming causal relationships into feature vectors, using a temporal network model, and implementing a prototype system for real-time detection of APT attacks, achieving an accuracy of 0.963.The approach encounters obstacles related to data redundancy, feature extraction, and the intricacies of high-dimensional data amalgamation.
In 2025, J. Lee et al. [20] suggested a High Discrimination APT Intrusion Detection System (HDAPT-IDS), which comprises a Cyber Clustering Module (CCM) and a Clustering Analysis Module (CAM). CCM performs an initial classification of traffic packets and employs the random forest method to forecast the primary class. In contrast, CAM chooses the relevant DNN based on CCM's prediction findings to ascertain the sub-class of traffic packets as the outcome. Attained an accuracy of 0.8143, however, it also encounters difficulties in feature extraction and the risk of misclassification. In 2025, J. Chen et al. [21] presented a grouped multi-agent deep reinforcement learning defense (GMADRLD) technique, which delineates a simulation environment modelling an attack-defense game including 100 attackers and 100 defenders over five servers inside a cloud computing framework. It divides all defenders into collaborative groups, enabling members within each group to collectively optimize the defense plan by exchanging knowledge and experiences. This setting is deficient in communication mechanisms and static defender grouping.

Despite the prominent previous studies mentioned above, security and privacy are a major concern in cyberspace, primarily due to the inherent vulnerabilities of cloud systems. Moreover, although several models have attained considerable accuracy, they frequently depend on centralized data processing or inadequately address temporal dependencies. Conversely, FedNN-APT addresses these constraints by facilitating decentralized learning, guaranteeing robust generalization, and preserving user data privacy. Consequently, the proposed technique addresses a significant research gap by offering a highly precise, privacy-preserving, and realistically implementable methodology for identifying APTs in cloud computing. Table I compares the proposed FedNN-APT system with related works utilizing FL and neural networks via cloud attack detection. The results show that the FedNN-APT outperformed previous approaches, illustrating its high accuracy and efficiency in APT detecting attacks.

TABLE I COMPARES THE FEDNN-APT SYSTEM AND SOME RELATED WORKS.

Study	Year	Methodology	Dataset/ Platform	Accuracy	Recall	F1-Score
[13]	2021	DL	APT Malware dataset	0.9832	0.83	0.82
[14]	2021	MLP,CNN,LSTM	NSL-KDD	0.93-0.98	0.873-0.951	0.877-0.944
[15]	2021	LSTM,CNN	OPTC	0.983	0.7587-0.882	No available
[11]	2023	CNN, FL	UNSW-NB15	0.9018	No available	No available
[16]	2023	BERT-BiGRU-CRF layers	APT web dataset	0.7479	0.8011	0.7479
[17]	2023	CNN	Custom Dataset	0.959	No available	0.956
[18]	2024	CNN, LSTM, Attention	Network Traffic	0.992	No available	No available
[19]	2024	Provenance Graph	Private Cloud	0.963	No available	No available
[20]	2025	ML and Clustering IDS	NSL-KDD & Edge-IIoTset Dataset	0.8143	0.89-0.90	0.87
[21]	2025	Deep Reinforcement Learning	Simulated Environment	No available	No available	No available
Proposed FedNN-APT	Proposed now	APT malware dataset	proposed hybrid GRU-CNN	0.99	1.00	0.9988

Table I above demonstrates that the suggested FedNN-APT system has enhanced detection accuracy (0.99) relative to comparable approaches. In contrast to several related studies that either do not incorporate federated learning or inadequately safeguard data privacy, FedNN-APT successfully integrates a hybrid GRU-CNN architecture with federated learning, achieving superior accuracy and privacy-preserving efficacy in a resource-constrained computing environment. This integration enables the efficient detection of complex APTs inside distributed and resource-limited cloud environments. Finally, the results of the FedNN-APT system can be applied to real-world cloud computing environments to enhance security against APT attacks effectively. The system can be integrated into cloud-based Security Information and Event Management (SIEM) systems to facilitate real-time threat detection and response. Privacy-preserving across Multiple Cloud Tenants or Data Centers, Federated learning enables joint training of detection models without sharing sensitive data, thus preserving data privacy. Furthermore, the system's lightweight model architecture enables deployment across diverse cloud infrastructures with varying computing capabilities. These features provide FedNN-APT with robustness and practical implement ability in real-world contexts to enhance the entire cloud security framework.

3. OVERVIEW APT THREAT APT ATTACKS

Initially aimed at military institutions, it began with the Moonlight Maze operation in 1996, targeting U.S. military and government networks. These assaults have since expanded to sectors like governmental, industrial education, banking, aviation, energy supply, chemistry, communications, medical, and consulting [14]. APT exploits vulnerabilities in various devices, including mobile devices and PCs, allowing hackers to remotely control compromised systems and steal sensitive data from organizations and government institutions [9]. APT attacks operate by taking advantage of the weaknesses present in various device types. APT attacks target mobile devices and personal computers because of their features. Using sophisticated attack tactics, hackers can remotely manage compromised technology and steal sensitive data from businesses and governmental institutions. APT often uses spear-phishing emails with malicious links or attachments to infect systems. They target an organization’s technology to gather sensitive information, bypassing security through social engineering and C&C communications. The acronyms used in this phrase aim to clarify APT [14].

- Advanced: This indicates that attackers are well-trained, well-funded, and well-organized, and they make full use of network penetration techniques.
- Persistent: This denotes the ongoing nature of these assaults. In this instance, the attackers create a persistent online presence and make a genuine effort to hack the system. The APT1 group's four years and ten months of assault are the longest-lasting APT attacks to date.

- Threat refers to the potential for a company's classified material containing strategic knowledge to leak. APT attacks often inflict significant harm on a target since their primary objective is to steal sensitive information. Federate learning for APT detection.

4. PROPOSED FEDNN-APT SYSTEM

Researchers propose framework of the FedNN-APT system for APT detection in a cloud computing environment. The proposed system comprises five stages: create a local dataset, preprocessing dataset, splitting dataset, Local training model and select best model, and finally, implementing the federated learning process. The following stages are explained in detail below, as shown in Fig.1:

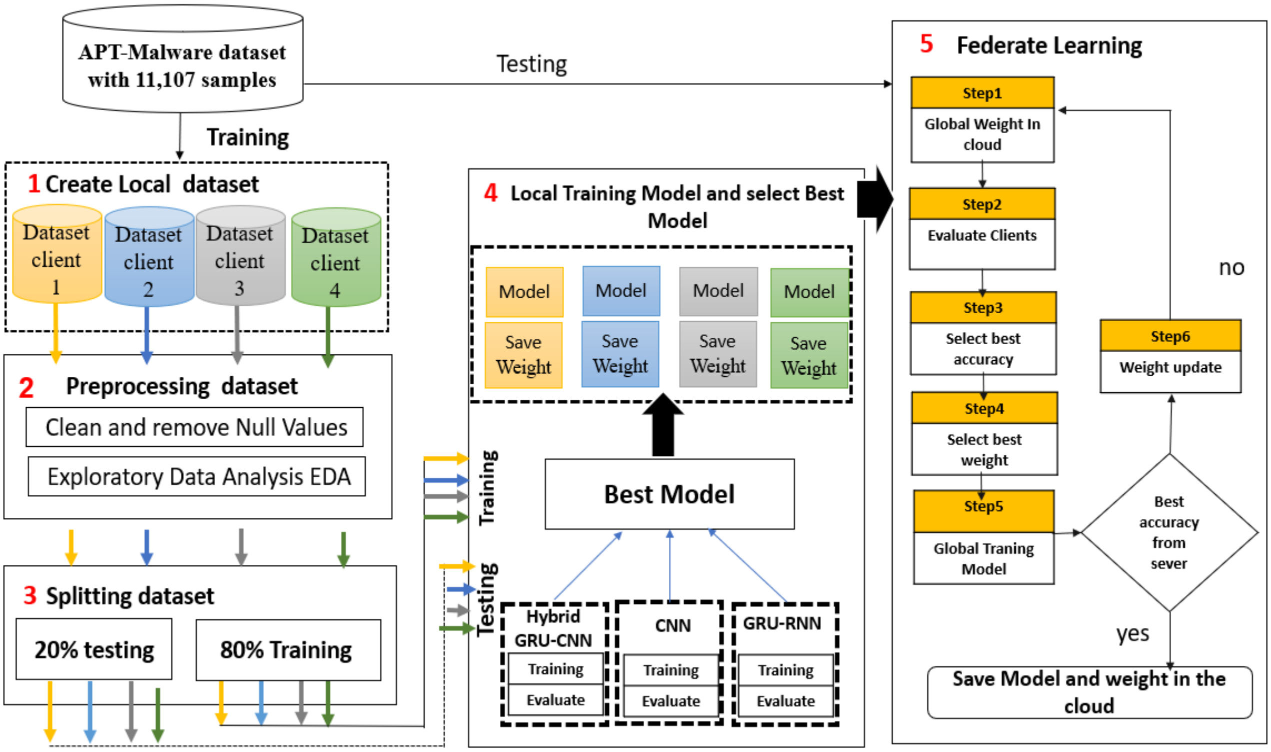


Fig. 1. Framework of the Proposed FedNN-APT System

4.1 Dataset and Local Distribution Preparation

The FedNN-APT system utilized an APT malware dataset, available at https://github.com/cyberresearch/APTmalware. The. The dataset includes 2086 APT malware samples and 9021 normal malware samples. It was initially divided into 80% training (8886 samples) and 20 % testing (2221 samples). The training dataset was then randomly partitioned into smaller groups to create four local datasets: Dataset client 1, Dataset client 2, Dataset client 3, and Dataset client 4. Each client is allocated a local dataset to emulate a distributed, privacy-preserving setting. This dataset's characteristics fall approximately into eight groups. The features include Obfuscated String Statistics (3 features), Packer (64 features), Packer (64 features), Imported API (3917 features), Buckets (98 features), Optional Header (30 features), MS-DOS Header (17 features), File Header (18 features), and Mutex (7 features).

4.2 Preprocessing dataset

The preprocessing stage involved preparing the four local datasets with a reliable and accurate training set. This stage included removing zero values and performing exploratory data analysis to remove outliers and enhance the efficiency and stability of the training process. Below are the preprocessing steps in detail:

4.2.1 Clean Dataset and Remove Null Values

The input dataset contains duplicate rows and null values, which the proposed system removes to ensure data integrity and enhance detection model accuracy. By eliminating these redundancies, the system prevents skewed results and ensures that the model is trained on accurate, reliable data.

4.2.2 Exploratory Data Analysis (EDA) Technique

EDA aims to understand the distribution of data and the correlations between characteristics [22]. In this instance, the goals are to identify the dataset's general structure, enumerate its key features, and visually depict the dataset. Thus, let's examine any relationship between attributes within a collection of distinct classes. Fig. 2 illustrates the behavioral correlations among a set of classes and one or more attributes.

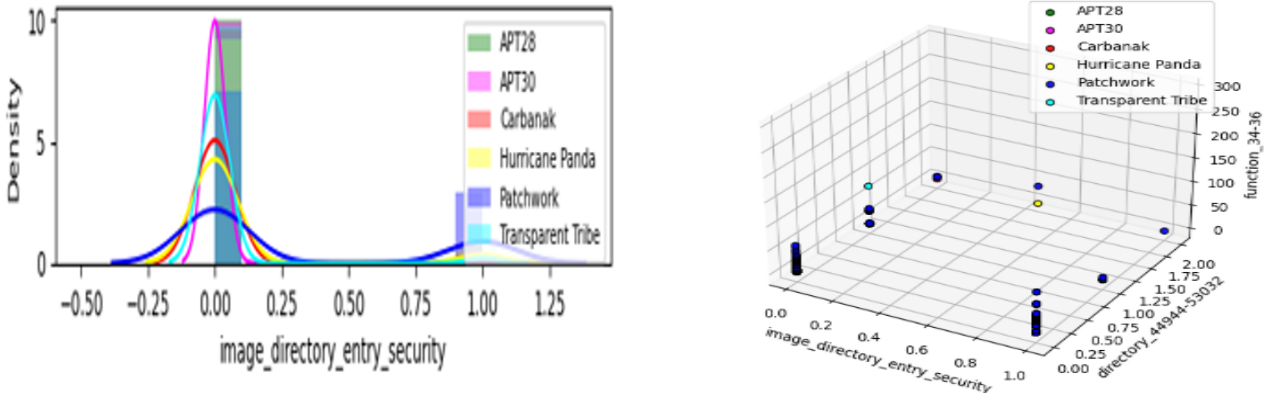
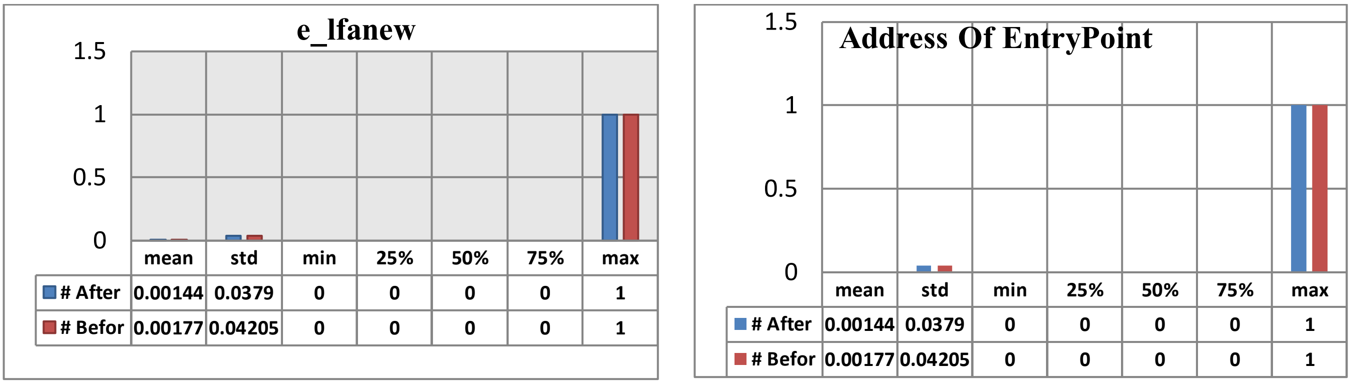


Fig. 2. An Example of the Relation between Attributes and Classes in the Dataset: 2D Histogram representation; 3D Histogram Representation

Outlier detection is essential for identifying inconsistent or distant data points in the FL framework. The FedNN-APT first uses EDA statistics (mean, standard deviation, min, max, and quarts) to detect and remove outliers. Fig.3 provides an example summary of this process with a blue line referring to data before outlier removal and an orange afterward. The horizontal axis represents EDA statistics, while the vertical axis displays their values.



	mean	std	min	25%	50%	75%	max
🟦 # After	0.00144	0.0379	0	0	0	0	1
🟥 # Befor	0.00177	0.04205	0	0	0	0	1

	mean	std	min	25%	50%	75%	max
🟦 # After	0.00144	0.0379	0	0	0	0	1
🟥 # Befor	0.00177	0.04205	0	0	0	0	1

Fig.3. An Example of Remove Outlier Values Using EDA Statistics: a) e-anew Feature; b) Address of Entry Point Feature

4.3 Splitting Dataset

Each local dataset is partitioned into 80% for training and 20% for testing to assess the model's performance locally before the federated learning stage. Table II presents comprehensive information on all local datasets.

TABLE II DETAILS OF FOUR LOCAL DATASETS

Dataset name	Total samples	80% Training	20% Testing
Client 1 dataset	2221	1778	444
Client 2 dataset	2221	1778	444
Client 3 dataset	2221	1777	444
Client 4 dataset	2223	1778	445
Total Samples	8886	7106	1777

4.4 Local training model and select best model

This section presents the framework for training the local model. The proposed system trains each local dataset using three detection models: GRU-RNN, 1D-CNN, and the proposed Hybrid GRU-CNN. Each model is then evaluated using testing data and corresponding evaluated metrics**,** as you shown in Fig.4:

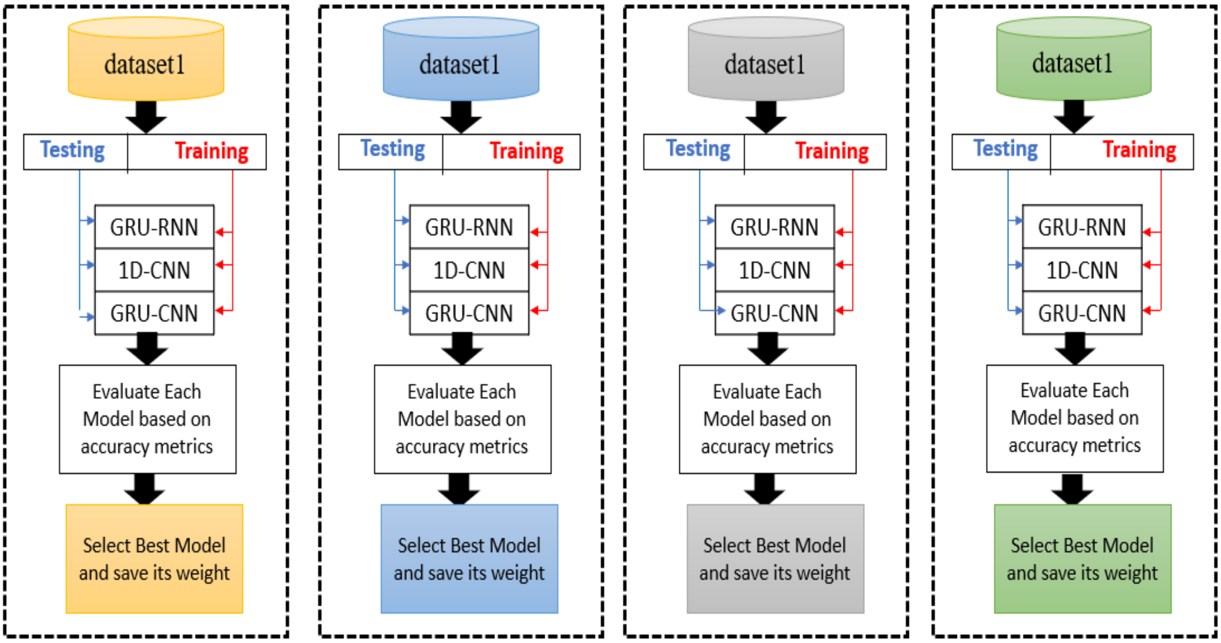

dataset1
Testing
Training
GRU-RNN
1D-CNN
GRU-CNN
Evaluate Each Model based on accuracy metrics
Select Best Model and save its weight
dataset1
Testing
Training
GRU-RNN
1D-CNN
GRU-CNN
Evaluate Each Model based on accuracy metrics
Select Best Model and save its weight
dataset1
Testing
Training
GRU-RNN
1D-CNN
GRU-CNN
Evaluate Each Model based on accuracy metrics
Select Best Model and save its weight
dataset1
Testing
Training
GRU-RNN
1D-CNN
GRU-CNN
Evaluate Each Model based on accuracy metrics
Select Best Model and save its weight

Fig .4. Framework of the Proposed Training Local Models

4.4.1 Gated Recurrent Unit (GRU) Model

An enhanced version of the LSTM network is the GRU network. It combines neuron and hidden states, optimizes the three LSTM gate architectures, and unifies input and forgets gates into a single update gate [18]. It may successfully lessen the "gradient disappearance" of RNN and lower the LSTM network unit's parameter count. GRU reduces the model's training time. In formulae (1) through (5), the mathematical formulas are explained [11].

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \qquad (1)$$

$$u_t = \sigma(W_u \cdot [h_{t-1}, x_t]) \qquad (2)$$

$$\tilde{h}_t = \emptyset(W_{\tilde{h}} \cdot [r_t \times h_{t-1}, x_t]) \qquad (3)$$

$$h_t = (I - u_t) \times h_{t-1} + u_t \times \tilde{h}_t \qquad (4)$$

$$y_t = \sigma(W_o \cdot h_t) \qquad (5)$$

Where the input vector, state memory variable at the previous moment, state memory variable at the current moment, reset gate state, update gate state, state of the current candidate set, and output vector at the current moment are represented by the variables r_t,u_t,$\tilde{h}_t$,h_t, x_t, and y_t , respectively; The weight matrices for the relevant inputs of the network activation functions are W_r, W_u, and W_o; [] indicates vector connection; · is the matrix dot product; × is the matrix cross product; σ is the sigmoid activation function; φ is the tan h activation function. I stand for identity matrix. The following is the mathematical definition of σ and φ [9]:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad (6)$$

$$\emptyset(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (7)$$

The update and reset gates are the main components of the GRU network. Following sigmoid nonlinear transformation, which ascertains the degree to which the state variable was brought into the present state at the previous time, the update gate receives the splicing matrix of the input parameter $x_t$and the state memory variable h_{t-1}at that moment. The reset gate regulates how much data goes to the candidate that was previously set. I $-$ $u_t$times h_{t-1}, which stores the information from the previous time, and $u_t$times $\tilde{h}_t$, which records the information from the present time, are added to provide the current time output [21].

4.4.2 Convolution neural networks (CNN) Model

A "convolutional", "batch normalization" (BN), "activation function", "pooling", "fully connected", and classification layers make up a CNN design in general. The feature maps are extracted using the convolutional layer. For instance, the feature map may be expressed as follows, assuming the input size is Q×P×C, where C represents the input channel number [14].

$$z^l = \sum_{c=1}^{c} \sum_{\rho=0}^{\Gamma-1} \sum_{\nu=0}^{\gamma-1} k(q+\rho, p+\nu, c) \times W^l(\rho, \nu) + b^l \tag{8}$$

Where $W^l(.)$ represents the convolutional kernel size,$b^l$is the l_{th} bias, and k is the input size, with q and p denoting row and Column indices of the input size , respectively. Additionally, q=$[1, L_q]$, where $L_q = \left[\frac{Q-\Gamma}{stride} + 1\right]$,and p= $[1, l_p]$ where $L_P = \left[\frac{P-\gamma}{stride} + 1\right]$ .The feature map generated by the “convolution layer” is normalization by “batch normalization “ layer ,as reference [21]:

$$\acute{z}^L = \frac{z^L - \mu_b}{\sqrt{\sigma_b^2 + \delta}} \tag{9}$$

$$g^L = \gamma \acute{z}^L + \beta \tag{10}$$

Where μ_b and σ_b^2 are average and variance of the mini-batch size, respectively. δ is a numerical stability coefficient, which prevents the denominator from becoming zero. γ is a scale factor and β is a shift factor over the mini-batch size. The rectified linear unit (ReLU) activation function decides whether the output of the BN is forwarded to the next layer, and it is expressed as [17]:

$$f\ ReLU(\acute{g}_k^l) = \max\,(o, g^l) \tag{11}$$

Where the SoftMax activation function is used after the fully connected layer, which is given as:

$$f\ softMax(g_k^l) = \frac{e^{g_k^l}}{\sum_x e_x^{g^l}} \tag{12}$$

Where g_k^l is the l_{th} feature maps k element and g_x^l is pre-activation output [20].

4.4.3 Hybrid GRU-CNN Model

The third classification model for APT attack detection is hybrid GRU-CNN, as illustrated in Figure 5. It starts with three 1D-Convoluation layers (512,256 filters) to extract features, followed by ReLU activation for non-linearity. Two 2D-max-pooling layers downsample the feature maps, and a 512-memory-unit GRU Lauer captures complex sequential patterns. Batch normalization improves stability, while a flattened layer reshapes outputs for dense layers. A dropout layer with a 0.5 rate is added for regularization. The model concludes with three thick layers, the last providing APT classification results, as you shown in Fig.5:

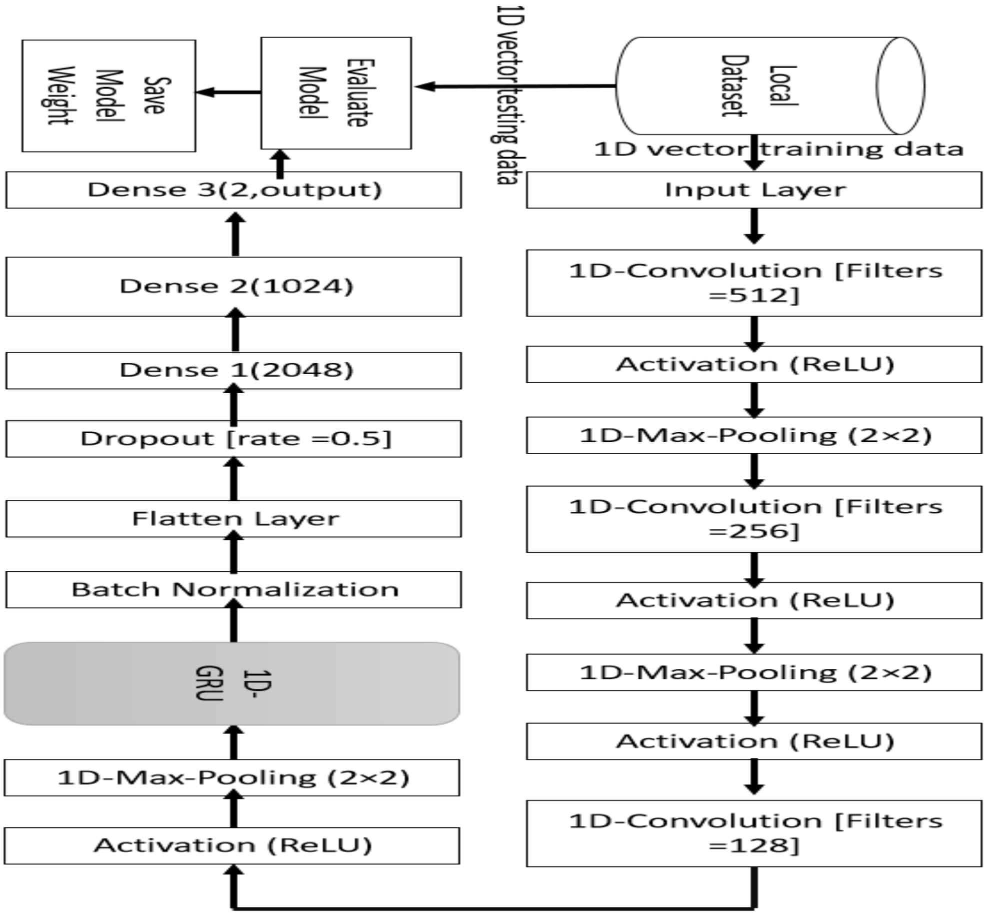


Fig.5. Structure of the Proposed Hybrid GRU-CNN Model

4.5 Federated Learning

In deep learning, FL is known as a distributed technique in which numerous clients work together to develop global models. Instead of exchanging personal local data, client local models are used to protect participant-contributed privacy. However, the number of clients and the level of their contributions have a significant impact on the effectiveness of federated learning [12]. In order, to serve in predetermined rounds, individuals improved their local models. In federated learning, a large number of clients participate in a distributed training process for the global model. Clients only train the model locally and exchange the model parameters for updates to the global model, allowing for data privacy, computational efficiency, and a wider detection range. The framework of FL in the FedNN-APT system is presented in Fig.6 In this work, the FL aims to develop a global model with optimal weights while preserving the privacy of client data. The central server aggregates the received updates from all clients to produce a new global model. This is typically done by using six steps:

- Global Weight Initialization in the Cloud: weights to be assigned to clients are initialized in the global model by the server.
- Client Evaluation: Using its data, each client assesses the model locally and modifies the weights as necessary.
- Best Accuracy from Clients is selected: to determine which client's models are the most accurate, the server evaluates their accuracy.
- Best Weights from Clients are selected: The server then chooses the weights from the previous phase that are most accurate.
- Best Global Training Model Selection: Lastly, the server uses the total weights and accuracy of all clients to determine which global model is the best.
- Check accuracy from the server; if it reaches to best accuracy, then create the final global model with optimal weight else, update weight using Equation (13) [10] :

$$M_g^{t-1} = M_g^t + \alpha \frac{1}{k} \sum\nolimits_1^k w_t^k \qquad (13)$$

In federated learning, if k clients have the same objective of collaboratively training a model, the global model M_g is distributed to the clients at each iteration and each client trains the model using local data. Each client transmits the model parameters back to the central computer after the local training is finished, and the global model is created by combining the model parameters from all of the clients. Where α a weight for the update process is, w_t^k is the model parameters, and M_g^t is the global model.

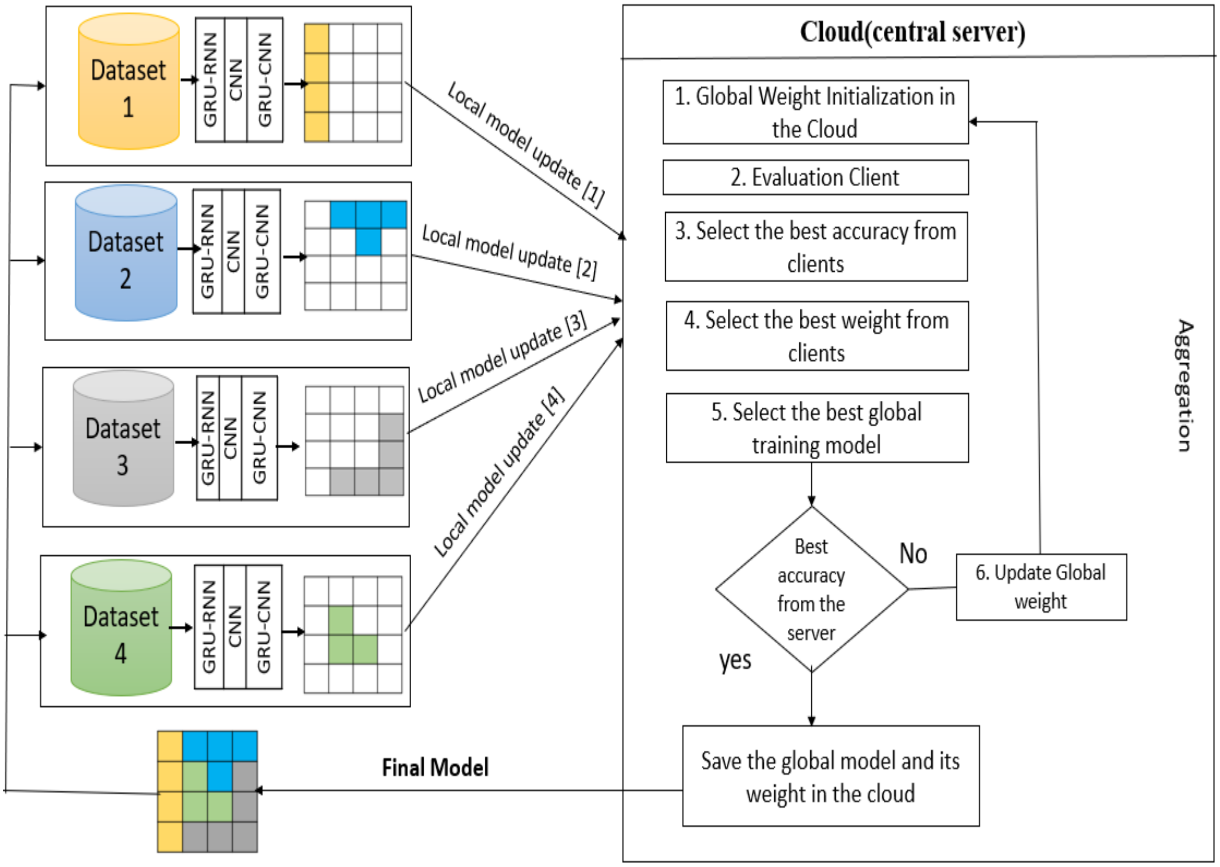

Dataset 1
GRU-RNN
CNN
GRU-CNN
Dataset 2
Dataset 3
Dataset 4
Local model update [1]
Local model update [2]
Local model update [3]
Local model update [4]
Cloud(central server)
1. Global Weight Initialization in the Cloud
2. Evaluation Client
3. Select the best accuracy from clients
4. Select the best weight from clients
5. Select the best global training model
Best accuracy from the server
No
yes
6. Update Global weight
Aggregation
Save the global model and its weight in the cloud
Final Model

Fig. 6. Framework of Federated Learning in FedNN-APT System

5. EXPERIMENTAL AND RESULTS

The suggested FedNN-APT system aims to mitigate the constraints of the cloud computing environment by utilizing federated learning techniques. The proposed approach minimizes the necessity of transmitting substantial volumes of sensitive data to the server, reducing network expenses and preserving privacy. Moreover, the most highly assessed model optimizes performance and efficiency. Each model is trained on lightweight cloud machines equipped with a hardware processor including a 20.0 GB RAM capacity and a "Core i7-8550U CPU" operating at "1.80 GHz" and "2.00 GHz". To assess the viability of implementing the proposed FedNN-APT system in cloud computing environments, investigated its computational performance comprehensively, concentrating on two primary metrics: training time per epoch and memory size.

- Training time per Epoch

The training duration for each of the three local detection models (GRU-RNN, 1D-CNN, and Hybrid GRU-CNN) was assessed across all four customer datasets, yielding average time intervals (in seconds) of 10.4, 9.6, and 11.1, respectively. Although the Hybrid GRU-CNN model achieved a longer training time, its superior accuracy justifies this slight computational discrepancy. The training time remained within acceptable standards, facilitating real-time performance.

- Memory size

The memory usage of the three local detection models (GRU-RNN, 1D-CNN, and Hybrid GRU-CNN) during training was assessed, yielding values of 1020, 850, and 1145 MB, respectively. The memory usage of the Hybrid GRU-CNN model is slightly elevated because of its hybrid design; nonetheless, it remains appropriate for cloud nodes and edge servers, choosing GRU over LSTM decreased memory needs by around 20%, rendering it more lightweight and scalable for federated learning contexts.

These findings validate that the system is implementable in practical cloud settings with constrained computational resources.

The proposed FedNN-APT system evaluated three local training models and federated learning based on standard evaluation measurements. The four indicators listed below will serve as references for the assessment indicators [18]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + TN} \quad (16)$$

$$f1\ score = \frac{2 \times (precision \times recall)}{precision + recall} \quad (17)$$

True positive (TP) refers to APT instances accurately detected as APT, while True Negative (TN) describes regular instances correctly recognized as standard. In contrast, False Positive (FP) occurs when normal instances are wrongly classified as APT, and False Negative (FN) happens when APT instances are mistakenly classified as normal.

Table III summarizes the accuracy and loss values throughout the training and validation process for three detection local models, evaluated across all four local datasets over 20 epochs. It demonstrates that the hybrid GRU-CNN model outperforms other detection models in terms of both accuracy and loss. Specifically, the hybrid GRU-CNN achieves the highest precision and lowest loss values across the training and validation process, where the actual value of accuracy is 1 positive, 0 negative so the closer the value is to one, the better, and the optimal value of loss is the closer it gets to zero.

TABLE III SUMMARY RESULTS OF THE THREE DETECTION MODELS VIA TRAINING ANF VALIDATION PROCESS

Local Dataset	**Models**	**Training**		**Validation**	
		Accuracy	**loss**	**accuracy**	**Loss**
Dataset client1	GRU-RNN	0.95094	0.75237	0.94934	0.77696
	1D-CNN	0.99930	0.00687	0.99969	0.00479
	GRU-CNN	**0.99936**	**0.00935**	**0.99970**	**0.00457**
Dataset client2	GRU-RNN	0.97051	0.41993	0.7200	0.39978
	1D-CNN	0.99942	0.08892	0.99431	0.09074
	GRU-CNN	**1.00000**	**0.00000**	**1.00000**	**0.00000**
Dataset client3	GRU-RNN	0.95001	0.76669	0.95626	0.76290
	1D-CNN	0.99970	0.00462	0.99983	0.00264
	GRU-CNN	**0.99988**	**0.00363**	**1.00000**	**0.00000**
Dataset client4	GRU-RNN	0.99400	0.08890	0.99942	0.09070
	1D-CNN	0.99998	0.00105	0.99999	0.00021
	GRU-CNN	**1.00000**	**0.00000**	**1.00000**	**0.00000**

Table III demonstrates that the hybrid GRU-CNN model consistently exhibited superior training and validation metrics, signifying its capacity to capture spatial and temporal features that the other models cannot. Moreover, the model exhibited stability and accuracy across all four client datasets, demonstrating its capacity for generalization and robustness to data fluctuation.

Table IIII summarizes the testing results for three detection models, highlighting their excellent performance in the confusion matrix, accuracy, precision, recall, and f1 score. Among them, the hybrid GRU-CNN model outperforms both the GRU and CNN models, with an average accuracy of about 0.9977, an average precision of about 0.9989, an average recall of about 1.00, and an average f1 score of about 0.9988. These results underscore that the hybrid model effectively leverages the strengths of both individual models, resulting in superior accuracy and reliability for APT detection.

TABLE IIII SUMMARY OF THE CONFUSION MATRIX VIA TESTING PROCESS

Mod	Local dataset	TP	TN	FP	FN	Accuracy	Precision	Recall	F1
GRU-RNN	1	435	1	4	4	0.9820	0.9919	0.9919	0.9919
	2	439	0	1	4	0.9901	0.9977	0.9901	0.9949
	3	433	4	2	6	0.9881	0.9966	0.9856	0.9910
	4	437	2	3	3	0.9854	0.9932	0.9932	0.9932
		Average				**0.9864**	**0.9949**	**0.9902**	**0.9928**
1D-CNN	1	440	1	2	1	0.9944	0.9946	0.9977	0.9961
	2	440	1	1	2	0.9930	0.9977	0.9955	0.9965
	3	441	0	2	1	0.9944	0.9954	0.9977	0.9964
	4	441	0	2	1	0.9944	0.9954	0.9977	0.9964
		Average				**0.993**	**0.9964**	**0.996**	**0.9963**
Hybrid GRU-CNN	1	443	0	1	0	0.9977	0.9977	1.00	0.9988
	2	442	0	0	1	0.9977	1.00	0.9977	0.9988
	3	443	0	1	0	0.9977	0.9977	1.00	0.9988
	4	444	0	0	1	0.9977	1.00	0.9977	0.9988
		Average				**0.9977**	**0.9989**	**1.00**	**0.9988**

Table IIII indicates that the hybrid GRU-CNN model effectively reduces false positives and negatives. Identifying APT is crucial, since the oversight of a single attack can result in significant system vulnerabilities.

The 1D-CNN model performed well (average accuracy: 0.993, F1 score: 0.9963); however, its recall rate (0.996) was somewhat inferior to that of the hybrid model. This indicates that 1D-CNNs may overlook some assault patterns, particularly those reliant on temporal factors, despite their considerable accuracy. The GRU-RNN model performed the least effectively, particularly with the recall rate (0.9902) and accuracy (0.9864). This is likely due to its inability to capture spatial dimensions in network behaviors.

The balanced hybrid GRU-CNN effectively captures both temporal via GRU and spatial characteristics through CNN in APT attacks, a system recall rate of 1.00 indicates its capability to identify all instances of APT across various clients without omission. This serves as a critical metric for security applications, as undetected assaults can yield severe consequences.

The following figure explains depicts the average confusion matrices for the three assessed models (1D-CNN, GRU-RNN, and Hybrid GRU-CNN) derived from the testing process. The visual representation aligns with the numerical findings in the above table, indicating that the Hybrid GRU-CNN attains the lowest misclassification rate and the superior performance metrics, as shown in Fig.7:

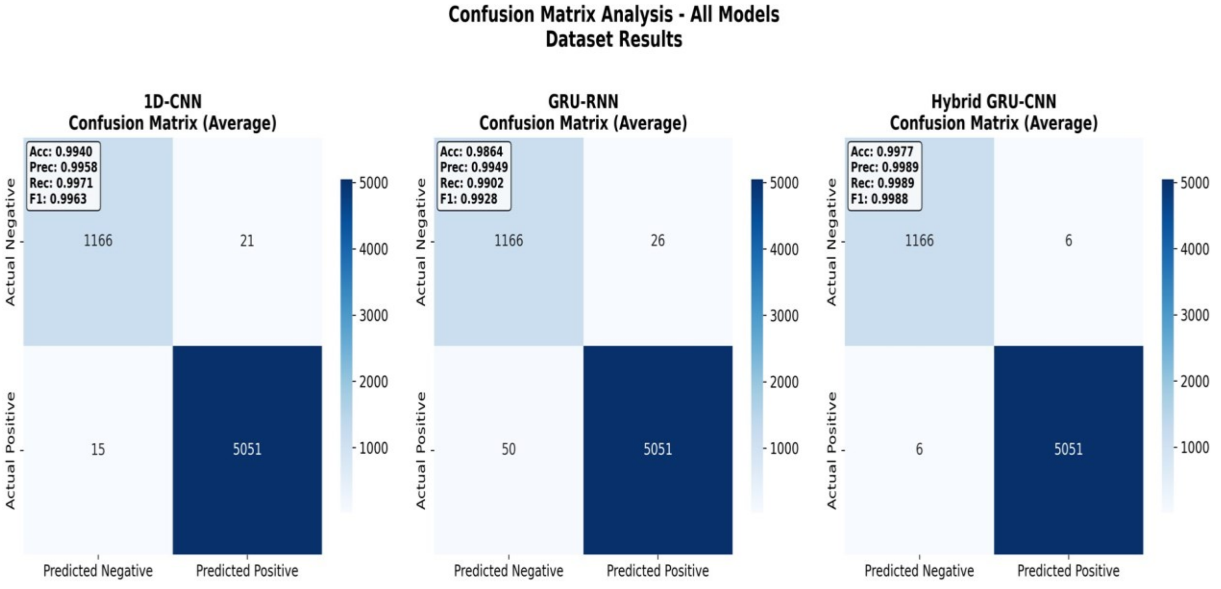
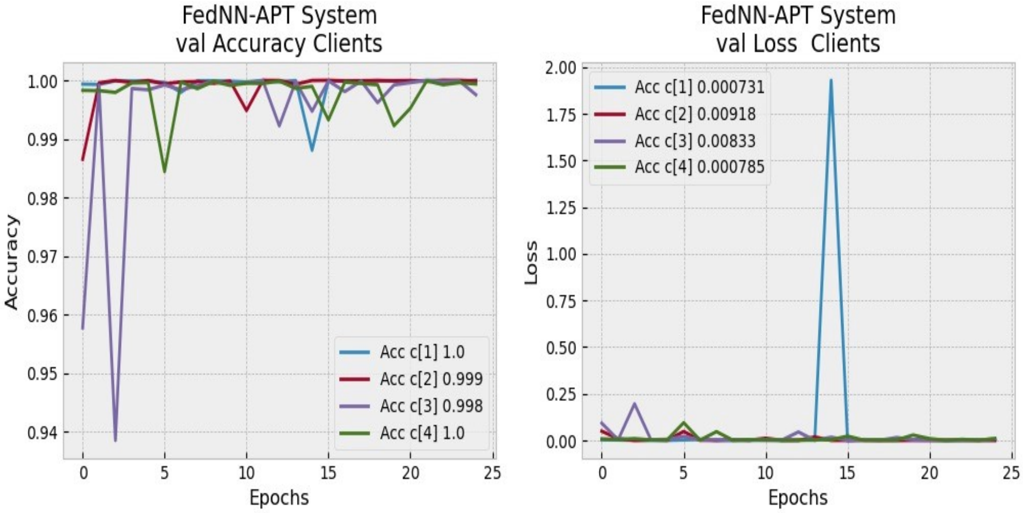


Fig. 7. The Confusion Matrices for FedNN-APT System.

Since the hybrid GRU-CNN performs best in training and testing local models, the FedNN-APT system selects it and incorporates its weights to apply a federated learning approach. This approach aggregates insights from local models to create a robust global final model. The following figure explains the accuracy and loss metrics for client-level validation of the federated learning model based on a hybrid GRU-CNN with four clients. Client-level validation assesses the model performance on each client's unique data distribution. The model achieved high accuracy scores of 1.000, 0.999, 0.998, and 1.000 with lower loss scores of 0.0007, 0.0091, 0.0083, and 0.0007, as shown in Fig.8:

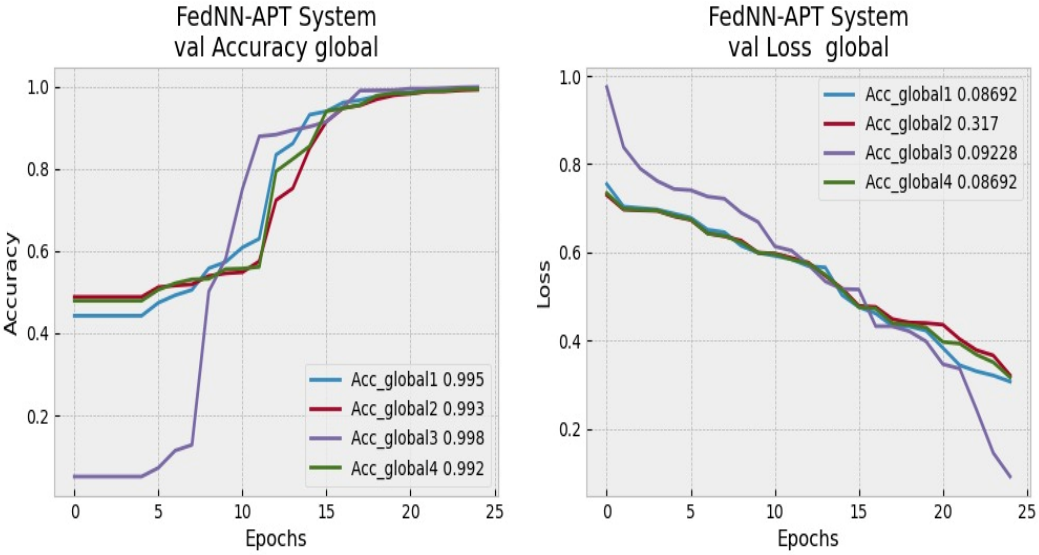

FedNN-APT System val Accuracy Clients

FedNN-APT System val Loss Clients

Fig.8. Results of FL at Client-Level Validation.

This result indicate the model's superiority in accuracy and loss validation performance across all local datasets, showing that each local agent of the hybrid GRU-CNN model can effectively learn data patterns specific to its environment with few mistakes.

Validating a model performance on a centralized or aggregated dataset is referred to as global validation. The following figure explains the global-level validation results, the FL achieving excellent validation accuracy for four clients with scores of 0.995,0.993,0.998, and 0.992, and lowest loss scores of 0.0869,0.0317,0.09228, and 0.08692, as shown in Fig.9:


FedNN-APT System val Accuracy global

FedNN-APT System val Loss global

Fig.9. Results of FL at Global-Level Validation.

These results indicate that the performance is robust and generalizable in various contexts. Additionally, Federated training reduces the need to transmit data to a central server, conserving network bandwidth and mitigating privacy concerns. Only the model weights are sent, reducing the expense of data transmission, which is crucial for real-time applications in cloud computing with limited resources.

The following figure explains shows client-level testing results based on FL and four datasets. The FL shows excellent results with values 1.00, 0.999, 0.999, and 0.999 and reduced loss scores of 0.00005, 0.0097, 0.00428, and 0.00616, as shown in Fig.10:

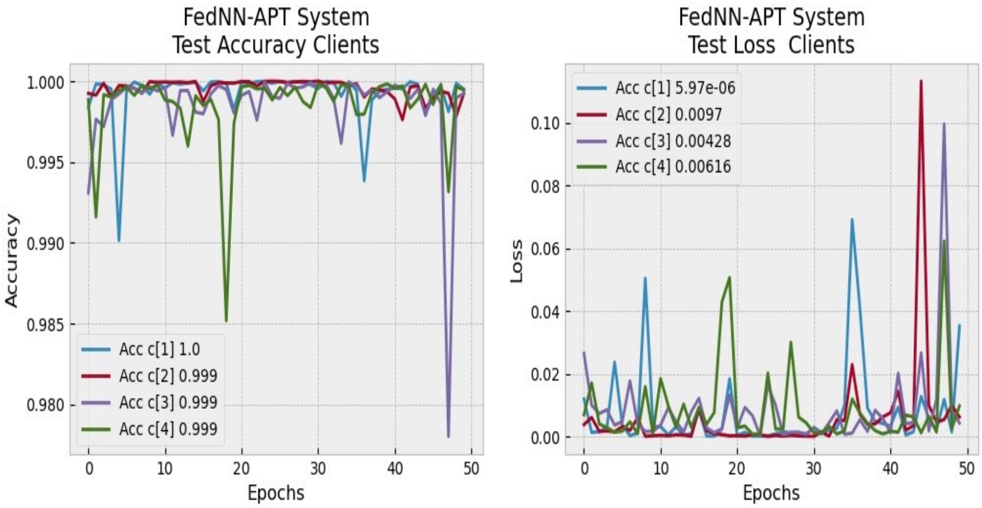


Fig.10. Results of FL at Client Level Testing.

Fig.10 above indicates that the model underwent testing across all clients and demonstrated high accuracy.This degree of accuracy indicates that the model can detect assaults in any local context, even when the data is distributed among clients. Conversely, the substantial decline in loss values indicates that the model is stable and proficient in accurately representing the dataset. These findings indicate that the model not only assimilates knowledge from the training data but also maintains robust performance throughout the test process.

The following figure illustrates worldwide federated deep learning testing results, using data collected from all clients. Excellent testing accuracy ratings (0.998, 0.984, 0.999, and 0.98) suggest powerful and accurate APT attack detection. The model achieved lower loss scores (0.1021, 0.1798, 0.1070, and 0.1021), as shown in Fig.11:

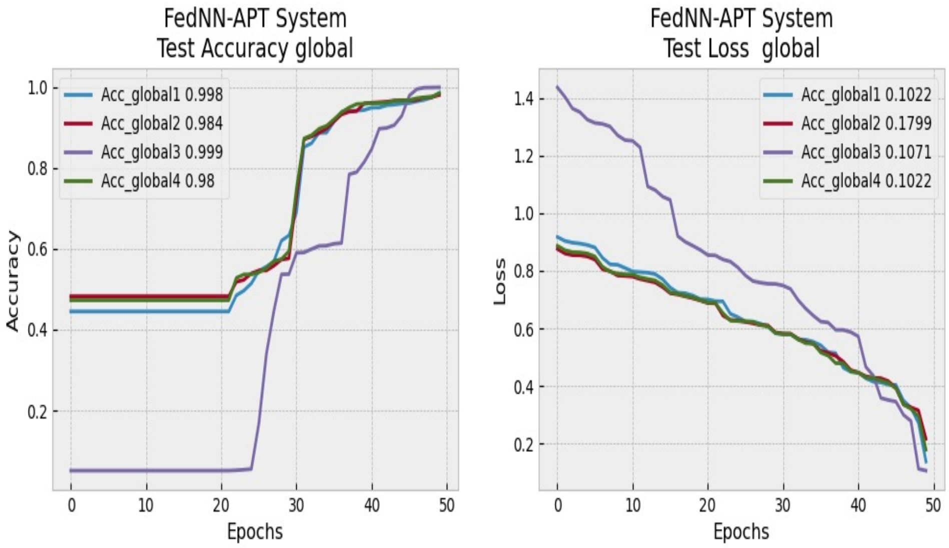


Fig.11. Results of FL at Global Level Testing.

Fig.11 above illustrates that the model attained high test accuracy across four datasets. These outcomes demonstrate the Federated model's capacity to generalize and effectively identify APT attacks in various and diverse environments. Despite minor variations in accuracy among clients, it remained very high, illustrating the model's robustness and efficacy in APT detection, even with diverse data distributions between clients. The loss rates are satisfactory, signifying that the model is neither overfitting nor underfitting.

6. LIMITATIONS

Although the promising efficacy of the suggested FedNN-APT system certain limitations warrant elucidation:

- The system's performance was assessed solely on the APT Malware dataset, which, despite its widespread use, may not comprehensively reflect the intricate characteristics of APTs in modern cloud computing systems. This dataset may be deficient in dynamic threat behaviors, thereby constraining the system's generalizability across various cloud computing environments and attack scenarios.
- While the FedNN-APT system demonstrates excellent performance in experiments, it lacks an evaluation of its performance in a real-world cloud computing environment. Practical deployment situations often exhibit considerable variations in network latency, bandwidth fluctuations, and heterogeneous client device capabilities, affecting performance and model convergence.

7. CONCLUSIONS AND FUTURE WORKS

This study presented a novel intelligent system, FedNN-APT, which integrates Federated Learning with Neural Networks, including hybrid GRU-CNN, 1D-CNN, and GRU-RNN models, to attain accurate and effective APT detection in resource-constrained cloud devices while ensuring a distributed, privacy-preserving network.

Randomly created four local datasets from the APT Malware dataset for each client device to implement and evaluate the proposed system. The employed preprocessing for these local datasets, which involved several steps: cleansing data by removing null values, and applying Exploratory Data Analysis (EDA) techniques to eliminate outliers. The preprocessing significantly enhanced the efficiency and stability of the proposed system, with experimental results indicating that all neural network models performed adequately; however, the hybrid GRU-CNN exhibited superior performance relative to other models, achieving an average accuracy of 0.9977, precision of 0.9989, recall of 1.00, and F1 score of 0.9988 ,as shown in TABLE I, TABLE III, Fig.7.

The FL approach achieved an overall average accuracy of 0.99 across the four local datasets, confirming its suitability for privacy-preserving distributed cloud applications. Compared to existing detection methods, the proposed FedNN-APT system demonstrated superior attack detection performance while maintaining a distributed, privacy-preserving framework, as shown in TABLE I, Fgi.9, Fig.10, Fig.11.

In future research, suggestion an alternative dataset, such as APTracker-2024 or Linux-APT-Dataset-2024, to generalize and evaluate the proposed system in diverse threats scenarios, also want to deploy the suggested system in a real-time cloud computing environment.

Conflicts Of Interest

As researchers declare that there is no conflict of interest.

Funding

This research was conducted independently and individually and was not funded by any institution.

Acknowledgment

Finally, as a researcher, I extend my gratitude to the university for their valuable observations and information that presented the research in the best form.

References

[1] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," *J. Internet Serv. Appl.*, vol. 4, no. 1, pp. 1–13, 2013, doi: 10.1186/1869-0238-4-5.

[2] Z. T. M. Al-Ta'i and S. M. Sadoon, "Visual cryptography based on chaotic logistic map in multi-cloud," in *AIP Conf. Proc.*, vol. 3097, no. 1, 2024, doi: 10.1063/5.0209467.

[3] L. Ge, L. Wang, and L. Xu, "An APT trojans detection method for cloud computing based on memory analysis and FCM," in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, 2016, pp. 179–183, https://doi.org/10.1109/ICISCE.2016.48.

[4] G. Shenderovitz and N. Nissim, "Bon-APT: Detection, attribution, and explainability of APT malware using temporal segmentation of API calls," *Comput. Secur.*, vol. 142, p. 103862, 2024.

[5] J. Zhang, S. Liu, and Z. Liu, "Attribution classification method of APT malware based on multi-feature fusion," *PLoS One*, vol. 19, no. 6, p. e0304066, 2024, https://doi.org/10.1016/j.cose.2024.103862.

[6] C. Do Xuan, D. T. Huong, and D. Duong, "New approach for APT malware detection on the workstation based on process profile," *J. Intell. Fuzzy Syst.*, vol. 43, no. 4, pp. 4815–4834, 2022, https://doi.org/10.3233/JIFS-212880.

[7] Y. Wang, H. Liu, Z. Li, Z. Su, and J. Li, "Combating Advanced Persistent Threats: Challenges and Solutions," *IEEE Netw.*, vol. 38, no. 2, pp. 1–9, 2024, doi: 10.1109/MNET.2024.3389734.

[8] S. Li, Q. Zhang, X. Wu, W. Han, and Z. Tian, "Attribution classification method of APT malware in IoT using machine learning techniques," *Secur. Commun. Netw.,* vol. 2021, no. 1, p. 9396141, 2021, doi: 10.1155/2021/9396141.

[9] X. Cheng, Q. Luo, Y. Pan, Z. Li, J. Zhang, and B. Chen, "Predicting the APT for Cyber Situation Comprehension in 5G-Enabled IoT Scenarios Based on Differentially Private Federated Learning," *Secur. Commun. Netw.*, vol. 2021, no. 1, p. 8814068, 2021, https://doi.org/10.1155/2021/8814068.

[10] H. T. Thi, N. D. H. Son, P. T. Duy, and V.-H. Pham, "Federated learning-based cyber threat hunting for apt attack detection in SDN-enabled networks," in *2022 21st International Symposium on Communications and Information Technologies (ISCIT)*, 2022, pp. 1–6, https://doi.org/10.1109/ISCIT55906.2022.9931222.

[11] H. K. Alkhpor and F. M. Alserhani, "Collaborative federated learning-based model for alert correlation and attack scenario recognition," *Electronics*, vol. 12, no. 21, p. 4509, 2023, https://doi.org/10.3390/electronics12214509.

[12] H. Zhu, H. Wang, C.-T. Lam, L. Hu, B. K. Ng, and K. Fang, "Rapid APT detection in resource-constrained IoT devices using global vision federated learning (GV-FL)," in Proc. *Int. Conf. Neural Inf. Process. (ICONIP)*, 2023, pp. 568–581, https://doi.org/10.1007/978-981-99-8126-7_44.

[13] F. J. Abdullayeva, "Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm," *Array*, vol. 10, p. 100067, 2021, https://doi.org/10.1016/j.array.2021.100067.

[14] C. Do Xuan and M. H. Dao, "A novel approach for APT attack detection based on combined deep learning model," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13251–13264, 2021, https://doi.org/10.1007/s00521-021-05952-5.

[15] M. Mamun and K. Shi, "DeepTaskAPT: insider apt detection using task-tree based deep learning," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2021, pp. 693–700, https://doi.org/10.1109/TrustCom53373.2021.00102.

[16] G. Xiang, C. Shi, and Y. Zhang, "An APT event extraction method based on BERT-BiGRU-CRF for APT attack detection," *Electronics*, vol. 12, no. 15, p. 3349, 2023, https://doi.org/10.3390/electronics12153349.

[17] W. Ren *et al.*, "APT attack detection based on graph convolutional neural networks," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 184, 2023, https://doi.org/10.1007/s44196-023-00369-5.

[18] C. Do Xuan and N. H. Cuong, "A novel approach for APT attack detection based on feature intelligent extraction and representation learning," *PLoS One*, vol. 19, no. 6, p. e0305618, 2024, https://doi.org/10.1371/journal.pone.0305618.

[19] H. Li, C. Yang, B. Zha, L. Liu, Z. Zhang, and S. Zhong, "A Real-time APT Attack Detection Scheme Based on Fusion Provenance Graph in Private Clouds," *Proc. - 2024 Int. Conf. Netw. Netw. Appl. NaNA 2024*, pp. 490–495, 2024, doi: 10.1109/NaNA63151.2024.00087.

[20] J. S. Lee, Y. Y. Fan, C. H. Cheng, C. J. Chew, and C. W. Kuo, "ML-based intrusion detection system for precise APT cyber-clustering," *Comput. Secur.*, vol. 149, 2024, doi: 10.1016/j.cose.2024.104209.

[21] J. Chen, X. Lan, Q. Zhang, W. Ma, W. Fang, and J. He, "Defending Against APT Attacks in Cloud Computing Environments Using Grouped Multi-Agent Deep Reinforcement Learning," *IEEE Internet Things J.*, early access, 2025, doi: 10.1109/JIOT.2025.3542119.

[22] T. Milo and A. Somech, "Automating exploratory data analysis via machine learning: An overview," in *Proc. 2020 ACM SIGMOD Int. Conf. Manage. Data*, 2020, pp. 2617–2622, doi: 10.1145/3318464.3383126.