# FEDERATED LEARNING BASED DYNAMIC QUALITY ROUTING IN WIRELESS SENSOR NETWORK

*Mr. N.Karthick** [1], *Dr. R.Sharmila* [2]

**ABSTRACT**

Wireless Sensor Networks (WSNs) are vulnerable to malicious nodes, limiting their applications. This work proposes a node classification approach for network security in wireless sensor networks using federated deep learning. The method leverages network deployment information and anomaly detection techniques to initially classify potential malicious nodes. Feature extraction from network data, encompassing activity, communication, traffic patterns, and resource consumption, provides a comprehensive understanding of node behavior. This detailed view, combined with federated learning, allows for collaborative model training across sensor nodes while preserving data privacy. The trained model identifies the patterns in the extracted features to classify legitimate and malicious nodes, continuously updating its knowledge with new data. Compared to traditional deep learning approaches, this method demonstrates superior performance in malicious node detection through a combination of federated learning and LEACH protocols. Network performance metrics like throughput, energy consumption, and delay are evaluated before and after implementing the proposed system. The trained model effectively identifies malicious nodes in unseen data, enabling the registration of legitimate nodes for further network operations.

**Keywords:** Block chain, Deep learning, Low Energy Adaptive Clustering Hierarchy (LEACH) protocol, Malicious node detection, Malicious node identification, Wireless Sensor Network.

Department of Computer Science[1],
Selvamm Arts and Science College,
Namakkal, Tamilnadu. nkmcapgp@gmail.com[1]
Department of Computer Applications[2]
Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India
sharmi.saravanan0521@ gmail.com[2]
* Corresponding Author

## I. INTRODUCTION

Wireless Sensor Network is a network comprising sensors distributed in various locations, collaborating to monitor and gather data on physical or environmental conditions. These sensors can measure parameters such as temperature, sound, pollution levels, humidity, and wind. Subsequently, the collected data is transmitted to a central location for analysis. WSNs are used in a variety of settings to collect data, but because of their open nature and resource limitations they become vulnerable to security risks. Some essential aspects of WSN Security includes threat under standing, security objectives, security mechanisms and balancing security and resource constraints. Wireless sensor network faces some threat understanding issues such as Denial-of-Service (DoS), data tampering, eavesdropping, and node capture. To handle these threats, WSN security focuses on four main objectives: integrity, confidentiality, availability and authentication. Access control, cryptography, key management, intrusion detection systems, and secure routing protocols are vital security mechanisms used in WSNs as layered security approach. Because of their limited processing power and battery life, traditional cryptographic algorithms used in wired networks may not be appropriate. Therefore, light weight cryptographic techniques are considered. These mechanisms are designed carefully to maintain strong security without hindering how sensor networks operate. Securing WSNs requires a layered approach that addresses various threats and vulnerabilities. By understanding the security goals and implementing appropriate mechanisms, WSNs can be made more resilient to attacks.

WSN relies on special instructions called routing protocol that navigates data packets from their source sensor node to a destination sensor node. These protocols are essential for efficient and reliable communication among sensor nodes while optimizing network resources such as band width, energy, and latency. Routing protocols in WSNs are designed to address the unique characteristics and

challenges of these networks, such as limited energy supply, dynamic network topology, and communication constraints. WSN routing protocols can be broadly classified into three categories based on their approach to routing:

Flat Routing Protocols: All sensor nodes in the network operate equally, making routing decisions based on metrics such as distance, energy level, orhop count.

Hierarchical Routing Protocols: Sensor nodes are divided into clusters or hierarchies, with designated cluster heads who are responsible for coordination and communication within the cluster.

Location-Based Routing Protocols: These routing protocols use the geographical location of sensor nodes to make routing decisions which are particularly useful where the physical location of nodes is important and known.

The ideal choice depends on several factors like network size, density, mobility, and application requirements. The ultimate goal is to optimize network performance interms of energy efficiency, latency, and reliability while ensuring that data reaches its destination in a timely and efficient manner.

WSNs can vary greatly, ranging from a few hundred in small-scale installations to tens of thousands in large-scale networks. Small-scale WSNs are commonly used in controlled environments like building monitoring or home automation systems which are fairly easy to manage due to their size. Medium-scale WSNs, consisting of several hundred to a few thousand nodes, are used in applications like environmental monitoring and industrial automation, requiring more sophisticated routing protocols and management techniques. Applications like smart cities, disaster response, and animal tracking use large-scale WSNs, which have tens of thousands of nodes spreads over a large area. These networks pose significant challenges in terms of scalability, energy efficiency, and network management.

Scalability in WSNs refers to the network's capacity to accommodate an increasing number of nodes while maintaining performance and efficiency which depends on various factors. Routing protocols are essential, in large-scale WSNs, location-based and hierarchical routing protocols are often used to improve scalability. For energy efficiency, protocols and algorithms are used to reduce energy consumption and prolong network lifetime. Performance optimization, resource management, and node health monitoring depends on efficient network administration. Large-scale WSNs face a unique challenge: balancing network performance with control message over head. Efficient and flexible physical node placement is ensured by using scalable deployment techniques such as mobile node deployment, grid deployment, and random deployment. To ensure efficient and reliable operation in various applications and environments, these aspects must be carefully taken in to account while building and implementing a scalable WSN.

Detecting malicious nodes in WSNs is vital for ensuring network security. This involves analyzing node behavior, detecting anomalies in energy consumption and data patterns, and employing various techniques like Intrusion Detection Systems (IDS), node energy consumption pattern and trust-based mechanisms. Collaborative detection, where nodes share information and machine learning algorithms trained on past data further enhance threat detection. By combining these techniques, WSNs can effectively detect and neutralize threats, ensuring the network's integrity and smooth operation.

Machine learning (ML) techniques are increasingly applied in WSNs to strengthen security measures, addressing key challenges like intrusion detection, secure routing, and data confidentiality. These approaches reform WSN security, offering innovative solutions to safe guard networks against threats. The development of Intrusion Detection Systems for WSNs relies heavily on ML techniques such as Support Vector Machines (SVM), Random Forest, and Neural Networks. These systems analyze network traffic and sensor data, identifying abnormal patterns of malicious activities. Moreover, ML techniques enhances routing protocol security by detecting and mitigating attacks, ensuring the integrity of routing messages and node behavior. They play a vital role in key management, predicting key distribution patterns and detecting unauthorized key access which fortifies communication security between sensor nodes. ML techniques ensure data confidentiality by employing encryptional gorithms, rendering sensor data unreadable to

unauthorized entities. Further more, ML helps anomaly detection in sensor data, flagging deviations that may signal security breaches or sensor malfunctions. By analyzing past data, ML model scanidentify patterns of normal behavior, enhancing security measures. ML also contributes to energy-efficient security solutions in WSNs, optimizing security mechanisms to minimize energy consumption without compromising security levels.

## II. LITERATURE REVIEW

A novel approach for enhancing Wireless Sensor Networks (WSNs). It leverages Bi-Concentric Hexagons and Mobile Sink technology to optimize cluster head selection, energy efficiency, and data aggregation. Additionally, Sec DL ensures high-level security through the One Time-PRESENT (OT-PRESENT) cryptography algorithm and enhances Quality of Service (QoS) using the Crossover based Fitted Deep Neural Network (Co-FitDNN) model. It enhances IoT - user security through data mining-based authentication, improving network performance as shown by ns-3.26 modeling [1]. The Secure Wireless Sensor Network Middleware (SWSNM), which addresses WSN security challenges using the generative adversarial network algorithm. SWSNM generates fake data and distinguishes it from real data, enhancing data accuracy and security. It improves energy efficiency, through put, and reduces end-to-end delay, promising significant enhancements in WSN performance and security [2]. A deployed model to isolate the DoS attacker nodes, leveraging quality metrics to optimize routing, rerouting, and data transmission, resulting insignificant improvements in network performance. Data collected by sensor nodes undergo analysis and processing in distributed systems before encryption and transmission to the base station. Security remains a para mount concern amidst vast data generation, leading to extensive research focusing on trade-offs between power consumption, delay, and latency [3]. A novel intrusion detection system (IDS) for WSNs utilizing a deep learning model in which the system optimally selects cluster heads (CH) based on energy, delay, and distance constraints using the self-improved sea lion optimization (SI-SLnO)model. A multidimensional two-tier hierarchical trust model evaluates the trust of CH and nodes, while deep learning-based intrusion detection employs optimized neural networks trained via the SI-SLnO algorithm. The approach demonstrates superiority over existing techniques through extensive evaluation [4]. The surveyed the related works and developed a deep learning-based intrusion detection systems trained on WSN-DS data set, detecting Black hole, Gray hole, Flooding, and Scheduling attacks. It was found that the traditional intrusion detection systems are less effective against evolving attacks like denial of service (DoS) [5].

A Deep Learning-based Defense Mechanism, for light weight DoS detection during Data Forwarding Phase (DFP). Extensive simulations demonstrate DLDM's effectiveness in accurately isolating adversaries, achieving high detection rates, throughput, and packet delivery ratio, while reducing energy consumption and false alarms [6]. deep neural network (DNN) for intrusion detection systems, leveraging cross-correlation to select optimal features and construct a DNN structure for intrusion detection. Experimental results demonstrate superior performance of the proposed DNN compared to conventional machine learning models like support vector machine, decision tree, and random forest, effectively identifying attacks in wireless sensor networks [7]. A proposed IDS framework using machine learning algorithm for WSNs, offering flexibility across various attack types and automating detection model creation from training data, there by reducing manual labour [8]. A novel routing scheme for WSNs combining block chain, meta-heuristic, and deep learning algorithms to counter diverse cyber-attacks. Block chain handles routing information dissemination, while Salp Swarm Optimization achieves optimal routing. Deep Convolutional Neural Network predicts routing variations and facilitates optimal decisions. Method improved efficiency, evaluated on latency, energy consumption, and through put metrics, and compared with existing methods like particle swarm optimization and reinforcement learning-based neural network for delay ratio [9]. A protocol emphasizing data availability along side confidentiality, integrity, and authentication utilizing blowfish encryption, EAX mode, and RSA algorithm, out performing existing protocols in Energy Efficiency, Network Lifetime, Average Delay, and Packet Delivery Ratio [10].

A block chain-based security architecture for data storage and access control, alongside a node synchronization energy optimization algorithm showingimproved network lifetime, latency, and throughput, making it reliable for resource-constrained WSNs [11]. The enhanced WSN security and reliability using block chain, implementing Q-tables for routing data collection and SHA256 encryption for data security. Assessing resilience against Sybil attacks, Proof of Work (PoW) outperforms Proof of Authority (PoA) in maintaining block chain integrity, despite higher computational costs. This highlights the efficacy of different consensus algorithms in securing block chain-integrated WSNs against attacks [12]. The integrated block chain into IoT-WSN architectures to boost security, privacy, and scalability, leveraging decentralized control and cryptographic features. While acknowledging challenges like increased computational overhead and interoperability issues [13]. An integrated block chain techniques with wireless sensor nodes to enhance cyber security in smart grid systems, addressing vulnerabilities high lighted by recent cyber-attacks. Leveraging a Proof of Authority(PoA) Ethereum Block chain frame work, the study evaluates performance across various SCAD A network topologies, aiming to bolster data integrity, transmission reliability, and trustworthiness in smart grid operations [14]. A lightweight yet highly secure node validation method by integrating Block chain with WSN, addressing security concerns without excessive resource usage. Through Block chain-assisted Node Validation (Block Node) and Valid Cluster Formation (VCF) techniques, it ensures high-level security while maintaining energy efficiency. Experimental results from NS-3.25 simulations demonstrate superior performance in security, encryption, delay, energy consumption, delivery ratio, and throughput [15].

A block chain-based authentication protocol for WSNs to address security vulnerabilities such as ID spoofing. Leveraging block chain's features, including cryptographic security and immutability, the protocol enhances data authentication in WSNs. The system model incorporates sensor nodes, cluster nodes, base stations, and private block chain networks [16]. A secure attack localization and detection in IoT-WSNs, utilizing block chain-based encryption and trust evaluation. Federated machine learning is enhanced for data security by identifying and classifying malicious nodes [17]. A block chain-based scheme for registration, data sharing, mutual authentication, and non repudiation in IoT-WSNs utilizing a consortium block chain for identity storage. Coordinators execute smart contracts, aiding sensor nodes in authentication and data processes. Ambient data storage employed an AI-based Inter Planetary File System (IPFS), while the Stellar consensus protocol enhances transaction throughput [18]. A block chain-based architecture to address cyber security issues in IoT devices, offering distributed data storage, immutability, and enhanced security. This architecture enables traceability and efficiency, ensured high performance and scalability for massive data storage, accommodating diverse WSN communication protocols seamlessly [19]. A Double Lined Hash Blocks (DLHB) block chain system enhances healthcare security via Advanced Encryption Standard (AES) encryption, Prime Padding Rivest Cipher Key Generation (PP- RCKG), and Shuffle Structure Chain Link (SSCL), ensuring data integrity and confidentiality. Utilizing Master Node Aggregation (MNA) and master node authentication, it decentralizes communication, improving security verification and validation. WSNs are essential for various applications, but they are vulnerable to attacks from malicious nodes. These attacks can compromise the integrity, confidentiality, and availability of data, limiting the effectiveness and reliability of WSNs [20]. The objective of this research is to enhance the security and performance of WSNs using Deep Learning (DL) and privacy-preserving techniques. The focus is on addressing evolving attack patterns, ensuring real-time adaptation, leveraging diverse network data, and preserving privacy. Main objective includes:

- Enhancing malicious node detection in WSNs using federated deep learning
- Evaluating the efficiency of federated learning with LEACH for malicious node classification in WSN security

This paper is organized into five sections. The introduction outlines the essentials of wireless sensor network security, including routing protocols, network size scalability, malicious node detection processes, and the application of machine learning for accurate node classification models. The literature review examines recent research and existing studies on WSN Security, focusing on deep learning approaches and the utilization of block chain technology to provide context for the proposed methodology. The methodology section details the federated deep learning model for node classification, explaining the various stages involved, such as feature extraction, federated learning, and anomaly detection. The results and analysis section presents the experimental outcomes achieved by the federated learning model on various benchmark data-sets and assesses its performance using relevant metrics. Finally, the conclusion section summarizes the key findings of the research, evaluates the effectiveness of the proposed federated learning model for node classification in wireless sensor networks, and identifies potential areas for future exploration.

## III. METHODOLOGY

Proposed system is designed for anomaly detection in a network of sensor nodes. Sensor data is collected and pre-processed by the nodes themselves. Features are then extracted and used to train a deep learning model locally on each node for real-time adaptation. To improve the model and protect privacy, federated deep learning allows nodes to collaboratively train the model without sharing raw data. The base station aggregates encrypted data while maintaining confidentiality and uses trust scores stored on a block chain to make routing decisions. Finally, the base station monitors the network performance and stores data securely. Figure 1 depicts the overall architecture of the proposed method.

### 3.1. Sensor Nodes

Sensor nodes act as the data collectors in this proposed system, gathering raw measurements like temperature or pressure from their surroundings. This data can be noisy or inconsistent, so preprocessing techniques like filtering and scaling clean and prepare it for analysis. In feature extraction process the key characteristics of both the sensor nodes and data are extracted for anomaly detection. Each sensor node integrates the process of deep learning model that continuously learns from this processed data with these features, adapting to real-time changes. While local training offers this real-time benefit, sharing knowledge across nodes can further improve the model's performance. Federated learning tackles this challenge by enabling collaborative training without compromising privacy.

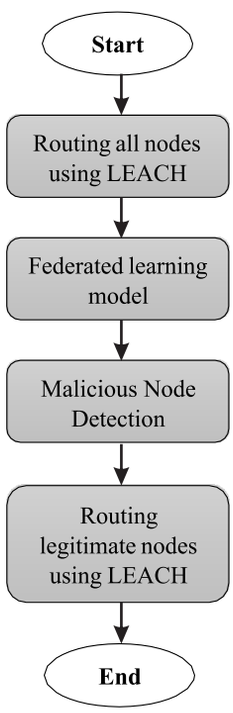


Figure 1: Architecture of the proposed work

### 3.2. Feature Extraction

In wireless sensor network, the malicious nodes can be detected by extracting various features from the network data by considering traffic, behavioral, communicational and resource- based features.

#### 3.2.1. Traffic features

In WSN, traffic features help us detect potential security threats. Differences in the usual packet size of a specific data type indicate it's an attempt to inject malicious code or

manipulate data by breaking communication or attacking vulnerabilities. Similarly, each node in a WSN has a common rate of data or packet transmission. Unusual changes in this rate like sudden increase or decrease signify that the node is malfunctioning or dropping packets to interrupt communication. Furthermore, Sensor nodes usually follow established routing protocols to send data towards the base station. Sudden alterations in the direction of packet flow are a sign that either routing protocols are disrupted or communications are being spied on. Lastly, Unusual changes in a node's energy consumption compared to its typical usage patterns could suggest that the node is performing malicious activities that require more energy.

#### 3.2.2. Communicational features

Variations between a node's described list and the reports of surrounding nodes might be an attempt to manipulate its neighbor list to isolate other nodes, disrupt routing protocols, or launchtargeted attacks. A severe issue is indicated when there's a consistent lack of acknowledgments (ACK's) from a specific node because when a node sends data, it expects an acknowledgment signal from the receiver confirming successful reception. A sudden increase in control messages from a specific node is an attempt to disrupt routing protocols by sending excessive route requests, sending fake signals to confuse other nodes, or launch a denial-of-service attack.

#### 3.2.3. Behavioral features

A node that constantly neglects to participate in network activities like routing data, responding to queries, and maintaining network health is considered suspicious. Their lack of participation leads to communication disruption. Another behavioral feature to consider is Clustering Behavior, particularly in cluster-based WSNs. Nodes that don't follow already established clustering protocols or display abnormal behavior within a cluster. Then, the lack of clustering behavior could be malicious.

#### 3.2.4. Resource-based features

If a node's remaining energy level is significantly lower than its neighboring nodes, it could be compromised, which may be exploited by attackers to launch malicious activities until the node's battery runs out. If their memory usage suddenly increases, it could mean there's malicious software on the node. Monitoring memory usage, unusual spikes can be marked for further investigation.

By analyzing a combination of these features and employing techniques like federated deep learning algorithms, WSNs can be made more robust against malicious nodes. The specific features chosen will depend on the network's configuration, security requirements, and the type of data being collected.

### 3.3. Federated Learning

The process of node classification in network security begins with an initial list that incorporates both legitimate nodes $(L = \{l_1,l_2,....,l_n\})$ and malicious nodes $(M = \{m_1,m_2,....,m_n\})$ The malicious nodes can be classified through network deployment information and federated learning techniques used in anomaly detection. In the initial step, various features $(F = \{f_1,f_2,....,f_n\})$ are extracted from the network data $(X)$ to provide a comprehensive understanding of node behavior and its characteristics. These features encompass activity-based $(Ai)$, communication-based $(Ci)$, and traffic-based patterns $(Ti)$ and resource consumption metrics $(RCi)$ After analyzing the data from all nodes including both legitimate nodes and malicious nodes, a detailed view of network activity is obtained, allowing for a sophisticated understanding beyond the initial malicious node classification. Figure 2 depicts the entire architecture of the proposed federated learning system.

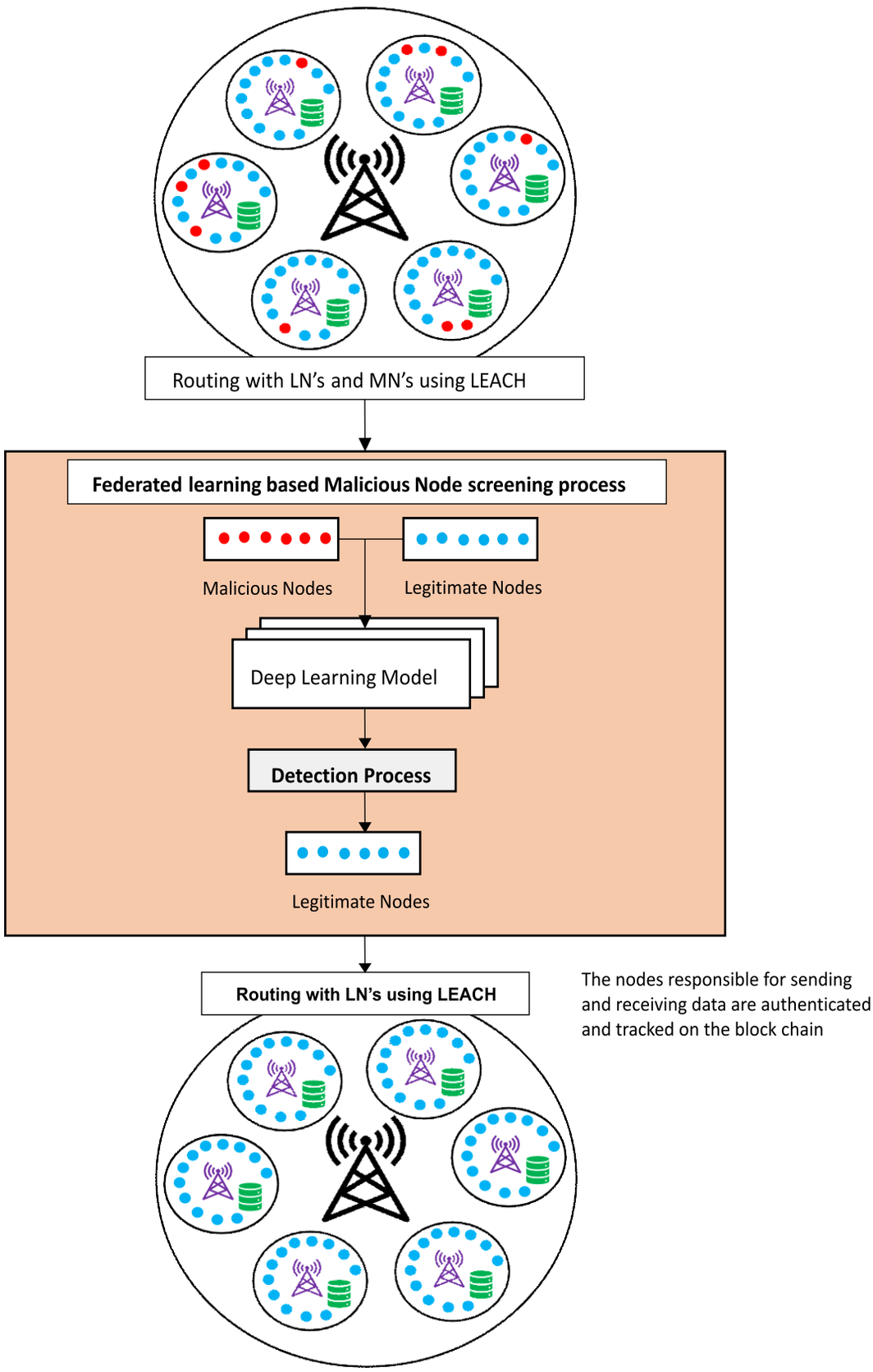


Figure 2 : Architecture of the proposed federated learning system

The proposed federated learning model is helpful in the node classification process that ensures data privacy. Initially, a cluster head distributes a base model to all sensor nodes. A global model with initial weights ($\theta_0$) is distributed to participating nodes. All sensor nodes ($S_i$) then train a local copy of this model ($f(X_i)$) using their private network data ($X_i$), which remains private on the sensor. Crucially, the raw data is never shared. Instead, each node creates an update ($\Delta\theta_i$) that captures the differences between the local data and the global model. These updates, essentially refined versions of the model based on the node's specific information, are then sent back to the cluster head.

$$\Delta\theta_i \approx \nabla\theta(f_\theta(X_i), t_i) \qquad (1)$$

Where, $L$ be the loss function and $t_i$ be the target. The cluster head aggregates these updates from all nodes ($\Sigma\Delta\theta_i$), effectively combining their learnings without exposing the raw data. Then, this combined knowledge is to improve the global model ($\theta$), which is distributed to the nodes again.

$$\theta_{t+1} = \theta_t + \sum \Delta\theta_i \qquad (2)$$

Where, t be the current training iteration. This iterative process of local training, update sharing, and global improvement continues, allowing the model to continuously learn and refine its ability to classify legitimate and malicious nodes while keeping all sensor data confidential. Following this, the federated deep learning model is trained using the extracted features. Federated learning facilitates collaborative model training across multiple nodes while preserving data privacy and scalability. The model learns to identify underlying patterns and correlations in the extracted features that classifies the legitimate nodes and malicious nodes, continuously updating its knowledge based on new data. The training data for the federated deep learning model consists of labeled data points, where each data point represents the features extracted from a specific node and is labeled as either "legitimate" or "malicious." By integrating initial classification, feature extraction, and federated deep learning, this comprehensive approach enables the development of a robust and adaptive system for effective node classification, enhancing network security and resilience against malicious activities. Once trained the model have the ability to identify the malicious node from the unseen data. Finally, the trained model can be used classifying the anomaly nodes in the wireless sensor network. Following this, list of legitimate nodes can be registered in the base station for further process.

### 3.4. Anomaly Detection

The federated deep learning model can classify the five classes: legitimate node (LN) and four malicious node types (MN-1, MN-2, MN-3, MN-4). The network consists of three convolutional layers for feature extraction, followed by activation layers to introduce non-linearity. Two pooling layers are included for dimensionality reduction. Finally, fully-connected layers perform high-level reasoning and classification, with the final layer having an output size of 5 and using a Softmax function to assign probabilities to each class. The node is then classified based on the class with the highest probability.

The anomaly detection of the proposed model is described in Figure 3. Initially, a network is established with X number of sensor nodes ($S\{s_1, s_2, ...., s_x\}$), including cluster heads (CHs) and a base station (BS). Sensor nodes collect data from their environment and transmit it to the cluster head for further processing. These nodes are organized into clusters ($C\{c_1, c_2, ...., c_x\}$), where k is the number of clusters. Routing within the network is facilitated by the LEACH protocol.

$$T(s_i) = \begin{cases} \frac{p}{1 - p * (r \bmod \frac{1}{p})} & if\, s_i \in CH(t-1) \\ p & otherwise \end{cases} \qquad (3)$$

Where, $p$ be the desired percentage of cluster heads, $r$ be the current round number and ($t$-1) is the set of cluster heads in previous round ($t$-1). The cluster heads ($CH_i \in C$) aggregate and process the collected data before forwarding it to the base station.

**Algorithm – Anomaly_Detection()**

**Step 1: Initialization:**

- Deploy sensor nodes in the environment.
- Set desired cluster head percentage (p).
- Initialize round number (r = 0).

**Step 2: Cluster Head Selection:**

- For each sensor node $s_i$
  - Calculate threshold for becoming a cluster head in current round:

$$T(s_i) = \begin{cases} p/(1 - p * (r \bmod \frac{1}{p})) & if s_i \in CH(t-1) \\ p & otherwise \end{cases}$$

  - Generate a random number between 0 and 1 If the random number is less than $T(s_i)$, node $s_i$ becomes a cluster head for the current round.
- Update the set of cluster heads $C(t)$.
- Increment round number ($r = r+1$).

**Step 3: Data Collection and Aggregation:**

- Sensor nodes transmit their data to their assigned cluster head.
- Cluster heads aggregate and pre-process the collected data.

**Step 4: Federated Learning:**

- Cluster heads perform local training on the federated deep learning model using their aggregated data.
- Local models are uploaded to the base station securely.
- The base station aggregates the local models to update the global model.
- The updated global model is distributed back to the cluster heads.

**Step 5: Anomaly Detection:**

- Cluster heads use the updated global model to classify the received data from sensor nodes.
- Classification outputs:
  - Legitimate node ($LN$)
  - Malicious node type ($MN$ - $i$)
- If a sensor node is classified as $MN$ - $i$ ($i = 1, 2, 3, 4$), it is flagged as anomalous.

In wireless sensor networks, malicious nodes are identified using federated deep learning models. Malicious node type such as backdoors, DoS, fuzzers, and shell code are classified using the benchmark UNSW-NB15 dataset. Malicious node type such as smurf, neptune, backdos, and spy were classified using the benchmark KDDCUP99 dataset. Comparative analysis reveals that the combination of federated learning with the LEACH protocols outperforms the deep learning approach in malicious node detection. Network performance is evaluated before and after implementing malicious node detection, focusing on metrics such as throughput, energy consumption, and delay.

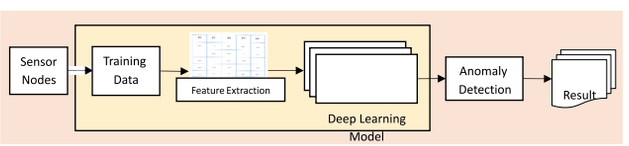


Figure 3 : Anomaly Detection of the proposed model

### 3.5. Calculating Transmission Time and Energy

In our proposed model, the uplink channel power gain from the x-th client to the access point is represented as:

$$h(s) = h_0 \rho_x(s)(d_0/d_x)^v \qquad (4)$$

Where, $h_0$ is the path loss constant, $d_x$ signifies the distance from the x-th client to the access point, $d_0$ represents the reference distance, $p_x$(t) indicates the power gain from the small- scale fading channel between the x-th client and the access point during the s-th communication round. $(d_0/d_x)^v$ represents the large-scale path loss with ν being the path loss factor which is determined by the distance. The below equation 5 describes the time taken for the x-th client to transmit data to the base station in the s-th round.

$$\varphi^{up}{}_x(s) = \frac{a_x\, i_x(s)}{Blog_2\left(\frac{1+T_{(s)}h_x(s)}{BP_0}\right)} \quad (5)$$

B represents system bandwidth, ($s$) represents the transmission power of the x-th node during s-th communication round, $P_0$ represents noise power spectral density and $\alpha$x represents the number of bits transmitted by x-th node. In addition, Equation 6 describes the energy consumed by the x-th node in the s-th communication round.

$$z^{up}{}_x(s) = \frac{T_x\,(s)a_x\, i_x(s)}{Blog_2\left(\frac{1+T_{(s)}h_x(s)}{BP_0}\right)} \quad (6)$$

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

### 4.1. Dataset Description

The UNSW-NB15 dataset [21] contains 49 features extracted using Argus and Bro-IDS tools. The dataset primarily consists of two record categories: normal and attack traffic. Attack records are further classified into nine distinct families based on the nature of the attacks. That includes analysis, backdoors, DoS, exploits, fuzzers, generic, reconnaissance, shellcode and worms.

In KDDCUP99 dataset [22] the connections are classified as either normal or belonging to a specific attack category. There are four main attack types: DOS, R2L, U2R, and probing. Each connection record is relatively small, around 100 bytes. The training data includes normal class and 24 different attack types. They are back dos, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, and warezmaster. Connection records are dissected into three distinct feature sets for analysis: basic features capture fundamental details of individual TCP connections, content features leverage domain knowledge to analyze the content flowing within a connection, and traffic features are calculated based on network traffic patterns observed over a two-second window.

### 4.2. Results and Performance Analysis

Calculation of Training and Testing Loss, as well as Training and Testing Accuracy, for the UNSW-NB15 dataset is achieved by repeating epochs until optimal values are obtained. The performance of the federated learning has been measured and tabulated in Table 1.

Table 1 : Comparative Analysis of the Training and Testing loss, Training and Testing accuracy obtained from the UNSW-NB15 dataset

| Epochs | Training Loss | Testing Loss | Training accuracy | Testing accuracy |
|---|---|---|---|---|
| 1 | 0.198 | 0.102 | 0.905 | 0.901 |
| 2 | 0.133 | 0.119 | 0.947 | 0.927 |
| 3 | 0.127 | 0.118 | 0.953 | 0.905 |
| 4 | 0.174 | 0.131 | 0.971 | 0.963 |
| 5 | 0.152 | 0.107 | 0.942 | 0.932 |
| 6 | 0.121 | 0.164 | 0.922 | 0.907 |
| 7 | 0.199 | 0.102 | 0.918 | 0.965 |
| 8 | 0.117 | 0.143 | 0.934 | 0.929 |
| 9 | 0.181 | 0.105 | 0.965 | 0.915 |
| 10 | 0.176 | 0.132 | 0.956 | 0.942 |
| **Overall** | **0.1578** | **0.1223** | **0.9413** | **0.9286** |

The overall average of the training and testing accuracy are 0.9413 and 0.9286 respectively. The overall average of the training loss and testing loss identified during the epochs are 0.1578 and 0.1223 respectively. Figure 4 represents the results obtained from the epochs, included the representation of training and testing accuracy, allow us to make informed decisions about the dataset's performance. By studying these findings, areas for improvement and optimize the performance of UNSW-NB15 dataset can be identified.

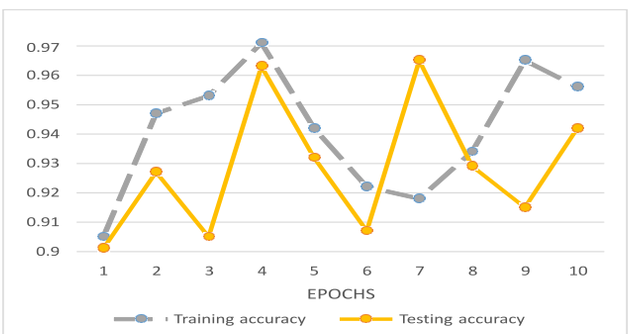


Figure 4 : Representation of the results of training accuracy, and testing accuracy obtained by the epochs from the UNSW-NB15 dataset

In Figure 5, the training and testing loss obtained for multiple epochs are seen. The average loss value obtained during these epochs is also calculated. By analyzing these results, analysts can gain valuable insights into the UNSW-NB15 dataset and identify areas where improvements can be made. Therefore, it is crucial to carefully review these findings to optimize the performance of UNSW-NB15 dataset.

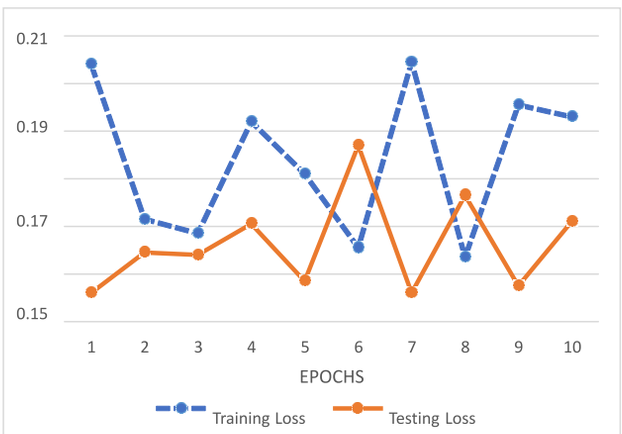


Figure 5 : Representation of the results of training loss, and testing loss obtained by the epochs from the UNSW-NB15 dataset

Figure 6 displays the sensitivity and specificity values for a UNSW-NB15 dataset by a federated deep learning classifier across different categories and general performance.

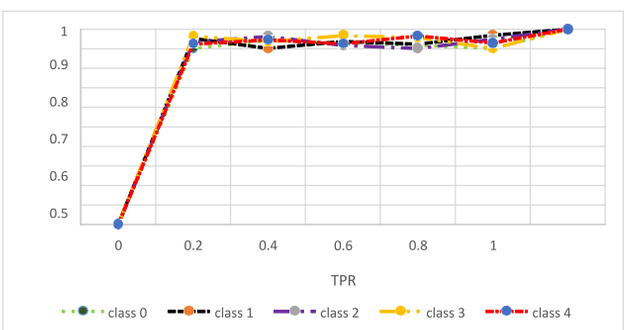


Figure 6 : Performance Analysis of federated deep learning classifier based on sensitivity and specificity for UNSW-NB15 dataset

Sensitivity represents the true positive rate (TPR) and specificity represents the true negative rate (TNR). The overall average values for sensitivity and specificity of class 0 (0.96, 0.9466), class 1 (0.9528, 0.945), class 2 (0.9512, 0.9506), class 3 (0.9506, 0.9564), and class 4 FPR(0.9572, 0.9602). Figure 7 displays the sensitivity and specificity values of a federated deep learning classifier for KDDCUP99 dataset and their overall performance.

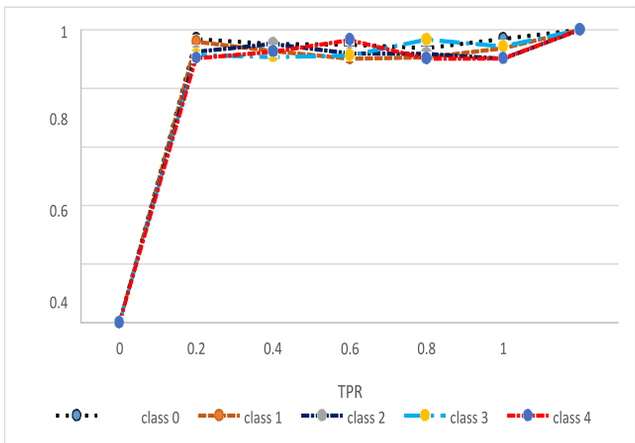


Figure 7 : Performance Analysis of federated deep learning classifier based on sensitivity and specificity for KDDCUP99 dataset

Based on the rate of TPR and TNR, The overall average values for sensitivity and specificity of class 0 (0.9426, 0.9366), class 1 (0.9348, 0.9256), class 2 (0.9336, 0.9366), class 3 (0.9346, 0.9484), and class 4 (0.9432, 0.9446). The calculation of the training and testing loss, as well as the training and testing accuracy, for the dataset KDDCUP99 is done by repeating epochs until optimal values are obtained. The performance of the federated learning has been measured and presented in Table 2 for further comparative analysis.

Table 2 : Comparative Analysis of the Training and Testing loss, Training and Testing accuracy obtained from the KDDCUP99 dataset

| Epochs | Training Loss | Testing Loss | Training accuracy | Testing accuracy |
|---|---|---|---|---|
| 1 | 0.111 | 0.128 | 0.924 | 0.934 |
| 2 | 0.149 | 0.132 | 0.943 | 0.912 |
| 3 | 0.127 | 0.116 | 0.953 | 0.931 |
| 4 | 0.182 | 0.154 | 0.975 | 0.926 |
| 5 | 0.136 | 0.128 | 0.913 | 0.934 |
| 6 | 0.175 | 0.192 | 0.935 | 0.951 |
| 7 | 0.151 | 0.167 | 0.961 | 0.913 |
| 8 | 0.125 | 0.103 | 0.927 | 0.952 |
| 9 | 0.147 | 0.119 | 0.945 | 0.934 |
| 10 | 0.162 | 0.132 | 0.978 | 0.922 |
| **Overall** | **0.1465** | **0.1371** | **0.9454** | **0.9309** |

The overall average of the training accuracy and testing accuracy are 0.9454 and 0.9309, respectively. The overall average of the training loss and testing loss identified during the epochs are 0.1465 and 0.1371, respectively. Figure 8 represents the results obtained from the epochs, included the representation of training and testing accuracy, allow us to make informed decisions about the dataset's performance. By studying these findings, areas for improvement and optimize the performance of KDDCUP99 dataset can be identified.


Figure 8 : Representation of the results of training accuracy, and testing accuracy obtained by the epochs from the KDDCUP99 dataset

In Figure 9, the training and testing loss obtained for multiple epochs are seen. The average loss value obtained during these epochs is also calculated. By analyzing these results, analysts can gain valuable insights into the UNSW-NB15 dataset and identify areas where improvements can be made. Therefore, it is crucial to carefully review these findings to optimize the performance of KDDCUP99 dataset. Figure 10 shows the delay that occurs in the network due to the presence of malicious nodes.


Figure 9 : Representation of the results of training loss, and testing loss obtained by the epochs from the KDDCUP99 dataset

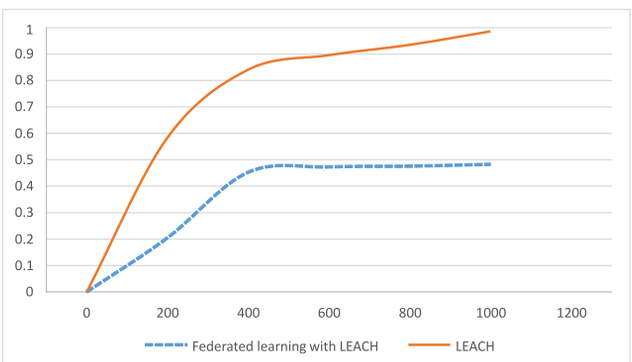


Figure 10 : Delay using LEACH and Federated Learning using LEACH

When using LEACH for routing, the detection of malicious nodes is not performed, which results in longer time for data transmission. Malicious nodes are detected and removed using Federated Learning using LEACH then routing is performed in the presence of legitimate nodes only. As a result, delay is minimized. From the figure 10, it is observed that the LEACH using Federated Learning is more accurate in detecting malicious nodes compared to LEACH

alone. Figure 11 shows the total throughput obtained from the network in terms of rounds using LEACH and Federated Learning using LEACH.

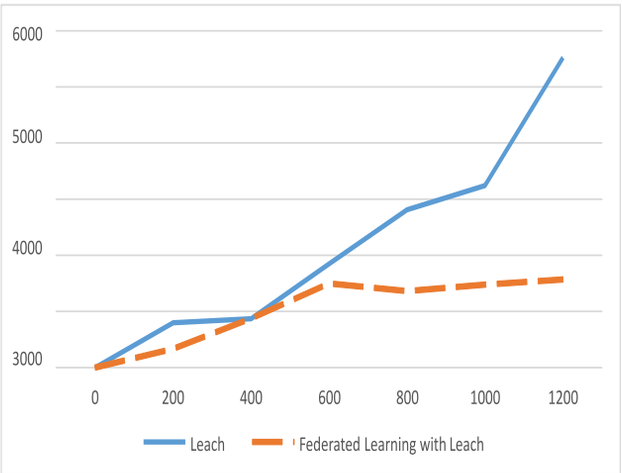


Figure 11 : Throughput using LEACH and Federated Learning using LEACH

In LEACH protocol, data packets are typically transmitted with the highest throughput when no malicious nodes are presented in the network. However, when federated learning isintegrated with LEACH for identifying and eliminating malicious nodes, routing occurs only among legitimate nodes, which can lead to a reduction in overall throughput.

## V. CONCLUSION

In conclusion, this study presented a novel anomaly detection system for Wireless Sensor Networks (WSNs) that prioritizes real-time adaptability and data privacy. The system leverages the power of federated deep learning, enabling sensor nodes to collaboratively train models for efficient anomaly detection without compromising raw data privacy. Furthermore, the Base Station plays a crucial role in secure data aggregation, leveraging trust scores on a blockchain for optimized routing and ensuring secure data storage. This comprehensive approach offers a promising solution for enhancing the security and adaptability of WSNs in various applications.

## REFERENCES

[1] Sujanthi, S. and Nithya Kalyani, S., 2020. SecDL: QoS-aware secure deep learning approach for dynamic cluster-based routing in WSN assisted IoT. Wireless Personal Communications, 114(3), pp.2135-2169.

[2] Alshinina, R.A. and Elleithy, K.M., 2018. A highly accurate deep learning based approach for developing wireless sensor network middleware. IEEE Access, 6, pp.29885-29898.

[3] Saravana Kumar, N.M., Suryaprabha, E., Hariprasath, K. and Vijayakumar, V., 2023. Deep learning based hybrid security model in wireless sensor network. Wireless Personal Communications, 129(3), pp.1789-1805.

[4] Kagade, R.B. and Jayagopalan, S., 2022. Optimization assisted deep learning based intrusion detection system in wireless sensor network with two-tier trust evaluation. International Journal of Network Management, 32(4), p.e2196.

[5] Salmi, S. and Oughdir, L., 2023. Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network. Journal of Big Data, 10(1), p.17.

[6] Premkumar, M. and Sundararajan, T.V.P., 2020. DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. Microprocessors and Microsystems, 79, p.103278.

[7] Gowdhaman, V. and Dhanapal, R., 2022. An intrusion detection system for wireless sensor networks using deep neural network. Soft Computing, 26(23), pp.13059-13067.

[8] Yu, Z. and Tsai, J.J., 2008, June. A framework of machine learning based intrusion detection for wireless sensor networks. In 2008 IEEE International conference on sensor networks, ubiquitous, and trustworthy computing (sutc 2008) (pp. 272-279). IEEE.

[9] Revanesh, M. and Sridhar, V., 2021. A trusted distributed routing scheme for wireless sensor networks using blockchain and meta-heuristics-based deep learning technique. Transactions on Emerging Telecommunications Technologies, 32(9), p.e4259.

[10] Dener, M., 2022. SDA-RDOS: a new secure data aggregation protocol for wireless sensor networks in IoT resistant to DOS attacks. Electronics, 11(24), p.4194.

[11] Vulavala, T.V., Shaik, R., Shaik, K.Z., Pathan, M.K. and Satamraju, K.P., A Secure Internet of Things Model Using Blockchain with Integrated Power Optimization. Telecommunications and Radio Engineering.

[12] Singh, S. and Chander, S., 2024. Prevention of Sybil Attack on Block Chain to Ensure Security of Wireless Sensor Network. International Journal of Intelligent Systems and Applications in Engineering, 12(8s), pp.14-24.

[13] Gijare, V.V., Rajput, S.D., Ambala, S., Aware, M.S., Ashok, W.V., Gandhi, Y. and Memon, I., 2024. Designing a Decentralized IoT-WSN Architecture Using Blockchain Technology. In WSN and IoT (pp. 291-313). CRC Press.

[14] Almasabi, S., Shaf, A., Ali, T., Zafar, M., Irfan, M. and Alsuwian, T., 2024. Securing Smart Grid Data with Blockchain and Wireless Sensor Networks: A Collaborative Approach. IEEE Access.

[15] Gnanasundari, P. and Sheela Sobana Rani, K., 2024. Blockchain-assisted node validation approach for security provisioning in WSN environment. Journal of Intelligent & Fuzzy Systems, (Preprint), pp.1-15.

[16] Dener, M. and Orman, A., 2023. BBAP-WSN: a new blockchain-based authentication protocol for wireless sensor networks. Applied Sciences, 13(3), p.1526.

[17] Gebremariam, G.G., Panda, J. and Indu, S., 2023. Blockchain-based secure localization against malicious nodes in IoT-based wireless sensor networks using federated learning. Wireless communications and mobile computing, 2023.

[18] Khan, A.U., Javaid, N., Khan, M.A. and Ullah, I., 2023. A blockchain scheme for authentication, data sharing and nonrepudiation to secure internet of wireless sensor things. Cluster Computing, 26(2), pp.945-960.

[19] Maftei, A.A., Lavric, A., Petrariu, A.I. and Popa, V., 2023. Massive data storage solution for IoT devices using blockchain technologies. Sensors, 23(3), p.1570.

[20] Ramani, R., Prasad, D.R., Kumar, C.M.S., Karthikeyan, T., Choubey, S.B. and Rajasekar, S.S., 2024. Enhanced WSN Cloud Security Based on Double Linked Hash Blockchain Security using Prime Padding Rivest Cipher Key policy. International Journal of Intelligent Systems and Applications in Engineering, 12(11s), pp.144-153.

[21] Moustafa, N. and Slay, J., 2015, November. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

[22] Stolfo, J., Fan, W., Lee, W., Prodromidis, A. and Chan, P.K., 2000. Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection. Results from the JAM Project by Salvatore, pp.1-15.

[23] Deng, X., Li, J., Ma, C., Wei, K., Shi, L., Ding, M., Chen, W. and Poor, H.V., 2022. Blockchain assisted federated learning over wireless channels: Dynamic resource allocation and client scheduling. IEEE Transactions on Wireless Communications.