Outlier Detection in Incentivized Fault Tolerant Blockchain based Federated Machine Learning

D Dharani[1*], K AnithaKumari[2]

[1]D Dharani is with PSG College of Technology, Coimbatore - 641004, India as an Assistant Professor [Sr.Gr.] in Dept. of Information Technology (Corresponding Author, Phone: +91-9952890280, Email: ddi.it@psgtech.ac.in)

[2]K Anitha Kumari is with PSG College of Technology, Coimbatore - 641004, India, as Associate Professor in Department of IT (Phone: +91-9842525820, Email: kak.it@psgtech.ac.in)

Abstract: Federated Machine Learning (FML) offers an exciting pathway for collaborative model training, enabling numerous users to contribute without disclosing their data. Yet, maintaining the security and privacy of both data and model within distributed settings continues to pose significant challenges. Identifying outliers plays a vital role in pinpointing abnormal behaviors that have the potential to compromise the integrity of the model. The paper introduces a system that harnesses autoencoders in conjunction with anomaly scoring techniques and thresholding mechanisms to detect anomalies within the dataset prior to model training. In the context of FML, where the model is trained across network, vulnerabilities arise as model parameters are exposed to evasion attempts. These attempts aim to undermine model integrity by manipulating the aggregation process. A protocol termed incentivized Probabilistic Byzantine Fault Tolerance (iPBFT) is developed to ensure the integrity of the model in a distributed environment. The framework offers a holistic solution to enhance security and integrity in distributed machine learning environment without compromising the system performance. Therefore, it serves as a crucial advancement in enhancing the overall performance and effectiveness of analytical endeavors, facilitating reliable decision-making in edge computing systems.

Keywords – Federated Machine Learning, Outlier detection, Autoencoder, Incentivized Practical Byzantine Fault tolerant Blockchain network, Data and Model security.

1. INTRODUCTION

In the field of collaborative model training and distributed data analysis, federated machine learning has become a potent paradigm. The goal is to train machine learning models via a decentralized network of edge devices, leaving private and sensitive data on such devices while advancing the construction of a global model[1][2]. Applications for this strategy may be found in a number of industries, such as healthcare, banking, and the Internet of Things (IoT), where security, privacy, and regulatory

compliance are critical issues. Federated machine learning uses a number of edge devices, or nodes, to train a global machine learning model jointly without exchanging raw data.

This privacy-preserving strategy is especially important in situations where security, regulatory compliance, and data privacy are top priorities. Traditional federated algorithms, which have been employed in distributed systems to compile data or carry out computations over a network of linked nodes, are the roots of federated machine learning. However, a new set of opportunities for research and development has emerged due to the integration of machine learning algorithms and the unique problems related to training models in a federated system. The accuracy and convergence of the global model in federated machine learning might be at risk if abnormalities in the local updates of participating nodes are not identified and mitigated.

Dealing with outliers or aberrant behavior among the participating nodes is a major difficulty in federated machine learning. This is known as outlier detection. Data points or local model updates that substantially vary from the anticipated trends are called outliers. Because these outliers have the potential to negatively impact the global model's convergence and jeopardize its correctness, it is imperative to identify and deal with them. Because outliers may cause problems for traditional machine learning models, it's critical to identify and deal with them in the federated learning process. To discover and lessen these abnormalities, a variety of outlier detection strategies have been created and modified for use in federated machine learning, including statistical methods, clustering-based approaches, and machine learning models.

The auto encoder is a powerful outlier detection mechanism that gets trained on normal, non-outlier data. Its ability to reconstruct the input data helps identify the outlier. The trained auto encoder takes the new input data and reconstructs it. The reconstruction error which is the difference between input data and reconstructed data is computed. The data points having reconstruction error higher than the threshold is considered as an outlier. The autoencoder mechanism is utilized to detect and remove outliers from the training data originating from various client nodes within the distributed network[3][4][5].

In Hyperledger based blockchain network, Practical Byzantine Fault Tolerance (PBFT) is an eminent consensus mechanism that verifies agreement among nodes in distributed FML network. It plays a critical role in maintaining the integrity of the transactions in the network. A client in the

network initiates the transaction by broadcasting the transaction request to all other peers in the network. All the peer nodes, upon receiving the transaction request, broadcasts prepare message to ensure their readiness for committing the transaction. PBFT requires prepare message from at least [2/3 N] is the total number of nodes in the network. In the context of PBFT, as long as an adequate number of honest nodes exist to reach consensus, the network can securely progress despite the incidence of malicious or faulty nodes. This mechanism is impersonated in the federated machine learning environment inorder to identify the model tampering. The iPBFT in proposed system offers a tamper-proof and secure ledger to document the specifics of the training procedure, transaction history, and consensus amongst involved nodes. This mechanism involves detecting nodes with incentives below a threshold, labeling them as faulty nodes, and excluding them from ongoing communication rounds and training processes, preserving the authenticity of the global model. Incentives are allotted to nodes that provide correct commit message. In federated machine learning environments, trust and accountability issues are addressed by this additional security and transparency layer [6]. The primary goals of the proposed work are outlined as follows:

- **Stabilized distributed model training:** Outlier data introduces instability and bias, leading to convergence issues and flawed model. Excluding them from training process, helps smoothen the overall process.

- **Fault tolerance:** Without adequate fault tolerance mechanisms, model drift becomes a risk within distributed networks, potentially causing the global model to deviate from the true underlying data distribution. iPBFT, an incentive based Probabilistic Byzantine Fault Tolerance helps achieve the proper fault tolerance to the system. It in turn improves the model performance [7].

2. RELATED WORKS

Y. Mirsky et al. introduce an innovative approach to online network intrusion detection through the use of an ensemble of autoencoders. Their study tackles key challenges faced by network intrusion detection systems (NIDS), such as real-time processing, detecting previously unseen attacks, and minimizing false positives. The paper offers valuable insights into improving both the efficacy and efficiency of NIDS in dynamic network environments [8].

X. Yuan et al. explore the landscape of adversarial attacks and defenses in deep learning. They advocate for the use of outlier detection techniques to obtain cleaner datasets, thereby improving the

reliability and robustness of deep learning models against adversarial attacks. The paper highlights the critical role of data preprocessing in mitigating vulnerabilities within deep learning systems [9].

P. Garcia Teodoro et al. delve into detecting network intrusions by identifying deviations from normal behavior. They explore the use of genetic algorithms to optimize intrusion detection systems, particularly for feature selection and model tuning, and employ neural networks for pattern recognition in anomaly detection. The paper also highlights challenges such as false positives, adaptive system learning curves, and network traffic variability. Additionally, it addresses the limitations of anomaly-based detection, emphasizing its dependence on known data and the difficulty of establishing reliable baselines [10].

Jin et al. propose a distributed anomaly detection scheme for the Industrial Internet of Things (IIoT) that integrates blockchain and federated learning. This research addresses the challenges of detecting anomalies in IIoT environments while ensuring security and efficiency. By combining blockchain's decentralization with the collaborative nature of federated learning, the authors enhance the reliability of anomaly detection. Their findings contribute to improving security in smart manufacturing and IIoT applications [11].

Yang et al. provide an overview of federated machine learning, discussing its concepts and applications. The authors highlight the benefits of federated learning, such as enhanced data privacy and reduced communication costs. They also explore various real-world applications across fields like healthcare, finance, and smart devices. This work significantly contributes to the understanding of the potential impact of federated machine learning on future technologies [12].

Adnan et al. explore the intersection of federated learning and differential privacy in medical image analysis. The authors investigate how federated learning can enhance data privacy while enabling collaborative analysis of medical images across institutions. They present methodologies that ensure sensitive patient data remains protected during the learning process. This research contributes significantly to advancing privacy-preserving techniques in healthcare, demonstrating the potential for federated learning to improve medical image analysis while safeguarding patient confidentiality [13].

Li et al. explore abnormal client behavior detection within the framework of federated learning. This research addresses the challenges associated with identifying malicious or faulty clients that can

disrupt the learning process in federated settings. By developing innovative techniques for detecting such behaviors, the authors significantly contribute to enhancing the robustness and security of federated learning systems. Their findings provide valuable insights into maintaining model integrity and ensuring reliable collaborative learning among distributed clients [14].

Cui et al. investigate security and privacy enhancements in federated learning for anomaly detection within Internet of Things (IoT). The authors propose a framework that integrates federated learning with robust security measures to effectively detect anomalies while preserving data privacy. Their research addresses critical challenges associated with IoT environments, including the risks of data breaches and unauthorized access. This study significantly contributes to improving security and privacy in IoT applications, highlighting the effectiveness of federated learning in maintaining data confidentiality during anomaly detection [15].

CFE, CITP presents "Blockchain Basics: A Non-Technical Introduction in 25 Steps". This article serves as a comprehensive guide aimed at demystifying blockchain technology for readers without a technical background. By breaking down complex concepts into easily digestible steps, the author effectively highlights the fundamental principles and applications of blockchain. This non-technical introduction is particularly valuable for professionals seeking to understand the implications of blockchain technology in various sectors, making it an accessible resource for anyone interested in exploring the transformative potential of blockchain [16].

Sadeghi et al. conduct a cryptanalysis of the full-round SFN block cipher, a lightweight block cipher designed for Internet of Things (IoT) systems, in their 2023 article titled "Cryptanalysis of Full-Round SFN Block Cipher: A Lightweight Block Cipher Targeting IoT Systems," published in *Scientia Iranica*. The authors explore vulnerabilities within the cipher, providing insights into its security strengths and weaknesses [17].

Behniafar et al. introduce the Anomaly Detection Fog (ADF), a federated approach tailored for IoT. The authors present a framework that leverages federated learning to enhance anomaly detection capabilities within IoT environments. By enabling decentralized learning from multiple devices, ADF aims to improve the accuracy and efficiency of anomaly detection while preserving data privacy. This research significantly contributes to advancing security measures in IoT systems, addressing the challenges of detecting anomalies in distributed and resource-constrained environments [18].

Rezaeian et al. present a density-based unsupervised learning approach for evaluating generator coherency in complex domains. The authors develop a novel method that utilizes density-based learning techniques to assess the coherency of generators, enhancing the understanding of dynamic interactions in power systems. Their research addresses the challenges of evaluating generator performance in complex environments, offering a robust solution for improving reliability and efficiency in energy systems. This study significantly contributes to the field of power system analysis and provides valuable insights for optimizing generator operations [19].

Dalila Ressi et al. explore the integration of Machine Learning algorithms and AI into blockchain technology, emphasizing recent advancements and potential applications. Notable opportunities include the development of decentralized AI models and enhanced decision-making processes. The study concludes by addressing the challenges and outlining future directions for AI-enhanced blockchain systems [20].

Raed Abdel-Sater et al. provide a thorough literature review of recent advancements in federated learning and its applications across diverse domains. The system identifies key challenges, including data privacy, communication efficiency, and model accuracy, while introducing the novel Federated LLM algorithm. Additionally, it tackles the challenges of communication overhead by optimizing the transmission of model updates to enhance communication efficiency [21].

Sater et al. explores the use of IoT sensors in smart buildings to enhance energy efficiency and anomaly detection in multivariate temporal data. It introduces a federated learning model based on a stacked long short-term memory (LSTM) architecture, demonstrating more than twice the training convergence speed of centralized LSTM. Overall, the approach effectively reduces training costs while maintaining prediction accuracy [22].

3. MATERIALS AND METHODS

In this system, autoencoder model is designed to identify anomalies within the datasets of individual client nodes. The autoencoder computes anomaly scores for images, enabling the system to detect anomalies by establishing a threshold. Data points with anomaly scores surpassing the threshold are deemed anomalous, while those below it is considered benign. Subsequently, the benign dataset is utilized as input for training local models at the end nodes. To identify outliers during model

transmission in FML, iPBFT Blockchain network is employed. Various system testing with threat model are conducted to assess the performance of the model. The entire working of the proposed system is depicted in Figure 1.

3.1 Autoencoder with Statistical Thresholding:

Autoencoders are widely used neural network architecture for unsupervised anomaly detection. A basic autoencoder consists of three main components: an encoder block (comprising one or more layers of neurons), a bottleneck layer (typically containing fewer neurons than the encoder), and a decoder block (sharing similar characteristics with the encoder). The primary goal of an autoencoder is to minimize the reconstruction loss or binary cross entropy loss when input data is passed through the network. This objective enables the bottleneck layer of the autoencoder to capture the most essential and representative features in a lower-dimensional space. Anomaly scores are computed in two ways: One is using the reconstruction loss that generates anomaly scores, another is based on Mahalanobis Distance calculated using binary cross entropy loss and Ledoit-Wolf covariance estimation on the encoded features and anomalies are identified using weighted average of statistical threshold calculations based on the Median Absolute Deviation (MAD), Z score and standard deviation. Data with an anomaly score greater than the threshold is considered as anomalous data, while data with an anomaly score below the threshold is categorized as benign data [23][24]. Mahalanobis Distance is a measure of the distance between a point and a distribution, taking into account the correlation between variables as shown in Equation 1.

$$M_{distance} = \sqrt{(x-\mu)^T . \Sigma . (x-\mu)} \quad (1)$$

where,

$M\_distance$ is the Mahalanobis Distance

x is the vector of encoded features of a data point

μ is the mean vector

Σ is the inverse covariance matrix

When an autoencoder is trained on normal instances (e.g., normal images of digits), the encoded features capture the typical patterns and variations present in the normal data. Anomaly detection using Mahalanobis Distance in the context of autoencoders involves assessing how well a new instance fits the learned distribution of normal instances in the feature space. Mahalanobis Distance can be calculated

using the learned covariance matrix from the normal instances in the feature space. Once Mahalanobis Distance is computed for a given instance, it serves as an anomaly score. A high Mahalanobis Distance suggests that the instance is far from the distribution of normal instances and may be considered an anomaly. Binary Cross Entropy is a loss function used in machine learning and deep learning to measure the difference between predicted binary outcomes and actual binary labels as shown in Equation 2. It quantifies the dissimilarity between probability distributions, aiding model training by penalizing inaccurate predictions. It is designed to measure the dissimilarity between the predicted probability distribution and the true binary labels of a dataset.

$$B_{loss} = -\frac{1}{\mathrm{N}} \sum_{\mathrm{i,j=1}}^{\mathrm{N,M}} y_{ji} \log\left(P_{ij}\right) \tag{2}$$

where,

B_{loss} is Binary Cross Entropy Loss

N is number of rows

M is number of classes

P_{ij} is probability of a particular class

y_{ji} is binary value of input image

The discrepancy between the initial input and the reproduced output within the autoencoder is termed the reconstruction loss as shown in Equation 3. In an autoencoder-driven anomaly detection framework trained solely on regular data, the objective is to replicate any input as faithfully as feasible to the familiar patterns learned from normal instances.

$$Reconstruction\,loss = \frac{1}{N} \sum_{i=1}^{N} \left| y_i - y'_i \right| \tag{3}$$

where,

N is the number of samples in the dataset

y_i is the target value of sample

y'_i is the predicted value of sample

After training the auto encoder, the anomaly score is calculated upon the data points. The threshold value is calculated using statistical approaches like Z-score, Standard Deviation, and Median Absolute Deviation (MAD), which quantify the deviation of data points from central tendencies. Data points exceeding the threshold is classified as outliers. Those data points are removed from the dataset and then the modified dataset is allowed for entering into the distributed network for further training process.

The Z-score, also known as the standard score, quantifies how many standard deviations a data point is from the mean. It is calculated as shown in Equation 4.

$$Z = \frac{x - \mu}{\sigma} \tag{4}$$

where,

x is the datapoint value

μ is the mean

σ is the standard deviation

Standard Deviation measures the extent of dispersion in a dataset as shown in Equation 5.

$$\sigma = \sqrt{\frac{(x - \mu)^2}{n}} \tag{5}$$

where,

x is the datapoint value

μ is the mean

n is the size of dataset

Mean Absolute Deviation is a robust measure of dispersion, less sensitive to outliers than standard deviation and is measured as shown in the Equation 6.

$$MAD = median(|\ x - median(x)|) \tag{6}$$

where,

x_i is the individual datapoint value

n is the total number of data points in the dataset

μ is the mean of the dataset

In the context of the proposed methodology, the weighted average for the threshold mechanism is a method used to combine the three thresholding techniques, which are Standard Deviation (SD), Median Absolute Deviation (MAD), and Z-score. Each of these techniques calculates a threshold independently based on different statistical properties of the data distribution. The weighted average mechanism involves assigning weights to each of these individual thresholds based on their respective importance or reliability in capturing outliers. Once the weights are assigned, the individual thresholds

are combined using a weighted average formula, where each threshold is multiplied by its corresponding weight and then summed together. The resulting weighted average threshold provides a comprehensive measure that takes into account the strengths of each individual thresholding technique, aiming to provide a more robust and reliable criterion for identifying outliers in the data as shown in Equation 7.

$$T = \frac{w_{1.}\left|Z_i\right| + w_2.\left(\left|x_i - \mu\right|\right)}{MAD} \qquad (7)$$

where,

T is Threshold

w_1 and w_2 are Coefficients or weights applied to different terms in the equation

$|Z_i|$ is value associated with Z score

$|x_i - \mu|$ is Standard deviation

MAD is Median Absolute Deviation

Following training, a threshold is determined using a threshold mechanism. Each image in the input array is then evaluated based on its reconstruction error loss compared to the threshold. If the error exceeds the threshold, indicating an outlier, the image is removed from the dataset. Finally, the remaining images, classified as 'good dataset' are sent to the nodes for further processing. This method efficiently identifies anomalies by combining reconstruction error and Mahalanobis distance, facilitating robust anomaly detection in various applications while ensuring the integrity of the dataset.

3.2 Incentivized Practical Byzantine Fault Tolerant (iPBFT) Blockchain network:

Practical Byzantine Fault Tolerance (PBFT) is a consensus algorithm vital for ensuring the reliability and fault tolerance of distributed systems, especially in the face of malicious or Byzantine nodes. To establish a tamper-resistant block in a blockchain utilizing weighted transactions, a proposer, and the requirement for approval from at least '2f+1' nodes, where 'f' represents the maximum tolerated Byzantine or potentially malicious nodes. Consensus mechanisms like PBFT ensure the acceptance of the block proposal by the required number of nodes. Throughout the approval process, nodes validate the transactions and verify signatures to ensure accuracy. If the proposal gains approval, it is added to the blockchain, ensuring tamper resistance due to the unlikely collusion of '2f' or more nodes to approve a malicious block. The blockchain continues with the selection of a new

proposer for the next block, maintaining the cycle. This robust process guarantees the integrity and continuity of the blockchain in the presence of potential adversities [23][24] [25].

The system incentivizes correct behavior among participating nodes by rewarding them with incentives for each correct commit message. It also employs iPBFT to detect and handle Byzantine faults by identifying faulty nodes during the consensus process. If the number of faulty nodes exceeds a predefined threshold in a communication round, system applies fault tolerance measures to maintain system integrity [26]. This includes removing the remaining faulty nodes and adjusting the protocol to continue with the consensus process. In scenarios where a significant portion of nodes is identified as faulty, the system proposes a mechanism to skip the communication round and revert to the output of the previous round to ensure system stability and continuity as shown in Figure 2.

The proposed fault tolerance mechanism is shown in the following Algorithm.

Algorithm: Incentivized Practical Byzantine Fault Tolerance (iPBFT)

Input: Nodes in PBFT with x faulty nodes
Output: Faulty nodes removed from network

Initialize totalNodes →0
CR →Communication Round
$Nodes_{removed}$ →Faulty nodes removed from current CR
update totalNodes with participating peers in network;
for i in **1 to n** CR do:
  if exists ($Nodes_{removed}$) from i-2th CR:
    add $Nodes_{removed}$ in ith CR;
    totalNodes = add($Nodes_{removed}$);
  for j in **1 to totalNodes** do:
    incentiveCalculation(j);
    x = findFaultyNodes(j);
  //based on the number of commits
    f=dynamicThresholdCalculation();
    if x gt f
      if x-f % gte 60%
        x-f nodes not removed
        Cancel the CR updates of ith CR
        Consider model weights of (i-1)th CR
      else
        Remove: (x-f) nodes in ith CR
  end

end

In the context of Federated Machine Learning, where local models collaborate with a central global model, a fault-tolerant weight transaction mechanism is proposed utilizing Practical Byzantine Fault Tolerance to safeguard against tampering of weights during transactions [27]. In proposed approach, participating nodes are incentivized with a reward of 1 for each correct commit message, fostering adherence to the protocol. To address Byzantine faults, it identifies faulty nodes in each communication round, ensuring system integrity. If the count of faulty nodes surpasses 2f+1, where f represents the maximum number of allowable faults, the system removes the remaining x nodes from the total of 2f+1+x faulty nodes. However, if x exceeds 40% of the total node count, system skips the round and revert to the output of the previous round for stability, discarding the output of the current round. Nodes removed in the current round are reintroduced in the subsequent round (i+2) to sustain fault tolerance. This comprehensive approach mitigates the impact of Byzantine faults on weight transactions between local and global models in FML systems, ensuring robustness and reliability.

To ensure the integrity of transactions, a threshold calculation mechanism is implemented. Specifically, the incentive threshold is determined as the sum of a base threshold, a node count factor, and a transaction count factor. The node count factor is computed by multiplying the node count by a predetermined coefficient, while the transaction count factor is derived from the transaction count multiplied by a similar coefficient. With a maximum threshold set at Number of transactions in each round and a base threshold is established, alongside coefficients of 0.1 for both node and transaction factors, this approach dynamically adjusts the incentive threshold to incentivize correct behavior among participating nodes while maintaining system stability and reliability. The formula for calculating dynamic incentive threshold as shown is Equation 8.

$$Threshold = BaseThreshold + NodeCountFactor + TransactionCountFactor \quad (8)$$

Where, node count factor is calculated by multiplying node count with node factor co-efficient. The transaction count factor is calculated by multiplying transaction count with transaction count factor.

In the proposed work, outlier detection and iPBFT components are introduced into a federated machine learning framework. Each participating node can use the auto-encoder to detect outliers in their local data before sharing updates with other nodes. Outliers can be handled differently or reported to a central

coordinator for further analysis. iPBFT consensus mechanism ensures that nodes in the federation agree on the updates to the shared model. This can enhance the security, transparency and trustworthiness of the federated learning process.

4. RESULTS AND DISCUSSION

4.1 Dataset:

The system is tested using MNIST and Fashion MNIST dataset. MNIST contains handwritten digits, Fashion MNIST features clothing items. These datasets are valuable for training and evaluating image classification algorithms due to their diverse contents and standardized formats [28] [29].

4.2 Autoencoder with statistical Thresholding mechanism:

A histogram illustrating the distribution of anomaly scores calculated using weighted average of Standard Deviation (SD), Median Absolute Deviation (MAD), and Z-score values are shown in Figure 3. The data points above the threshold are removed from dataset and not allowed for the training process. Each bar in the graph corresponds to a range of anomaly score values, and the height of each bar indicates the frequency of occurrence of these values [30].

4.3 Blocks added to Blockchain:

Figure 4 depicts the percentage of blocks added within a blockchain network operating with a fixed count of 10 nodes offers valuable insights into the network's performance and reliability of nodes. The x-axis of the chart represents the number of nodes, which is consistent at 10, while the y-axis illustrates the percentage of successfully added blocks. At this specific node count, one would anticipate an initially high percentage of block additions due to the network's manageable size, simplifying the consensus process and expediting quorum achievement. Consistency in the chart at 10 nodes underscores the network's reliability and predictability. It signifies the network's scalability and ability to efficiently process transactions, which is vital for potential expansion. However, the chart also highlights that as the number of nodes increases beyond a certain point, the network may experience diminishing returns, introducing complexities and potentially reducing efficiency. Additionally, the chart serves as a foundation for informed decisions when planning potential network expansion or upgrades.

4.4 Total commits per node over communication rounds:

Figure 5 offers crucial insights into the efficiency and reliability of the network's consensus process. In the initial rounds, a notable quantity of commits is expected, serving as an indicator of the iPBFT consensus algorithm's effective operation. The chart's paramount characteristic is the constancy of commit figures across all 100 rounds, signifying a network that is dependable and resilient, unfazed by fluctuations in transaction volume or network conditions. It is essential to keep a vigilant eye on the chart for any indications of diminishing returns, as a sudden decline in commit numbers during later rounds might imply that the network has reached its operational capacity or that increased complexity is affecting its performance. This data not only aids in evaluating the network's suitability for the given workload but also serves as a valuable reference for future decisions related to scalability and optimization, including infrastructure enhancements.

4.5 Incentive calculation and identification of Faulty node:

Nodes that provide correct commit messages on the transactions gets the incentives. Node that provides wrong commit messages might not receive any incentives. Since there are 100 communication rounds and there are 20 transactions in each communication round the maximum incentive received by a node at the end of 100^{th} communication round is 2000 as shown in Figure 6.

Nodes with incentives less than threshold value are considered as faulty nodes. PBFT can tolerate up to 'f' faulty nodes in a network of P nodes, where $P \leq \frac{n-1}{3}$. If the number of faulty nodes exceeds the number of faulty nodes allowed in iPBFT, the node will be removed in the next communication round. Figure 7 shows the number of nodes removed in each communication round.

Incentives are integer value calculated based on number of commit messages in every communication round. The generated incentives during the training process are shown in Figure 8.

4.6 Network metrics of transactions:

Network metrics of transactions were calculated using Prometheus and express-prom-bundle and the results is shown in Figure 9. Prometheus scrapes metrics from specified endpoints at defined intervals, making it well-suited for dynamic environments where services may frequently scale up or down. express-prom-bundle automatically tracks HTTP request metrics, including bandwidth, throughput, latency and scalability, providing insights into network performance and application behavior. Together, these tools provide a robust framework for monitoring network health and performance, enabling proactive management of network-related issues in applications.

In PBFT, bandwidth of nodes denotes the ability of each node to transmit and receive messages across the network. This capacity is pivotal in shaping the scalability and robustness of PBFT-based distributed systems. It is calculated as shown in Equation 9.

$$Bandwidth = R \times S \times N \quad (9)$$

where,

R is the number of messages sent per unit time, S is the message size, N is the number of participating nodes.

Throughput of nodes refers to the rate at which transactions or messages can be processed and finalized by each individual node within the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. Factors influencing the throughput of nodes in PBFT include the processing power of each node, network latency, message size, and the number of transactions being processed concurrently. Increasing the throughput of nodes can enhance the overall transaction processing capacity and scalability of the PBFT-based distributed system. It is calculated as shown in the Equation 10.

$$Throughput = R / N \quad (10)$$

where,

R is transactions per unit time, N is number of participating nodes.

Scaling transactions in the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm refers to its capacity to manage an increasing number of transactions while maintaining efficient performance and responsiveness. In the realm of distributed systems and blockchain networks, scalability is a crucial factor, as it determines how effectively the system can accommodate a growing number of users and transactions without a decrease in performance or an increase in transaction confirmation times. Scalability is calculated as shown in the Equation 11.

$$Scalability = \left(Throughput\left(N\right)\right) / \left(Throughput\left(M\right)\right) \quad (11)$$

where,

Throughput(N) is Transaction throughput with N nodes, Throughput(M) is Transaction throughput with M nodes.

The transaction latency in PBFT represents the duration it takes for a transaction to be proposed, validated by the consensus protocol, and ultimately added to the distributed ledger. This latency is affected by several factors, including network latency (the time required for messages to transmit between nodes), processing time at each node, and the number of replicas necessary to reach a

consensus. A lower latency implies quicker transaction processing, which is beneficial for applications that require real-time or high-throughput performance. Latency is calculated as shown in Equation 12.

$$Latency = PT + NT + CT + ET \quad (12)$$

where,

PT is proposal time, NT is network propagation time, CT is consensus time, ET is execution time.

4.7 Federated Machine Learning – Local Model Loss and Accuracy:

In Federated Machine Learning, local model loss quantifies the error incurred by individual nodes during training on their respective datasets. This metric assesses the model's fit to local data, crucial for evaluating each participant's model effectiveness. Typically minimized using optimization algorithms like stochastic gradient descent, the loss function measures discrepancies between model predictions and actual target values. Calculated periodically, it provides ongoing insights into model performance, guiding adjustments and optimizations throughout the training process in the federated learning framework that is depicted in Figure 10.

Local model accuracy in Federated Machine Learning as shown in Figure 11, evaluates the alignment between a participant’s machine learning model predictions and their local dataset’s actual data. Each participant trains their model independently on their confidential dataset, assessing accuracy by comparing predictions to ground truth values. Expressed as a percentage, this metric is crucial for understanding model performance within specific data environments. It provides insights into how effectively each participant’s model is leveraging local data, contributing to the overall effectiveness of the federated learning process.

4.8 Federated Machine Learning – Global Model Loss and Accuracy:

The Figure 12 shows the relationship between global accuracy and communication rounds in Federated Machine Learning that can provide valuable insights into the performance of the federated model as it progresses through rounds of collaborative training. This is because the federated model starts with a less accurate initialization and limited information from each node. As communication rounds progress, the global accuracy tends to improve. This is because the model benefits from aggregating insights and updates from multiple nodes, leading to better generalization.

In Federated Machine Learning, global loss as shown in Figure 13, represents the total error spanned across all participant nodes during the training phase of the global model on their distinct local datasets. On the other hand, local model loss is a measure of how effectively individual participant models perform on their specific datasets. Global loss, in particular, offers a holistic perspective of the model's performance, evaluating how accurately the global model corresponds with the collective data from all nodes in the federated network. This metric plays a critical role in guiding efforts aimed at improving the global model's accuracy and adaptability across diverse data sources, thereby enhancing the overall performance of the FML system [31][32].

The overall system is evaluated using various metrics as shown in Equations 13,14, 15 and 16. It makes it possible to assess the model's overall performance metrics on the whole as federated system, as shown in Figure 14.

$$Accuracy = \left(TP + TN\right) / \left(TP + TN + FP + FN\right) \quad (13)$$

$$Precision = TP / \left(TP + FP\right) \quad (14)$$

$$Recall = TP / \left(TP + FN\right) \quad (15)$$

$$F1Score = \left(2 \times \left(precision\right) \times \left(recall\right)\right) / \left(precision + recall\right) \quad (16)$$

where,

TP is the true positives, FP is the false positives, TN is the true negatives, FN is the false negatives.

4.9 Computational overhead of proposed framework:

To evaluate the computational overhead of the proposed framework, it is crucial to analyze the functionality of the various integrated components within the system. This includes examining the interactions between outlier detection, the fault-tolerant blockchain mechanism, and federated machine learning.

4.9.1 Computational overhead of Outlier Detection:

In the Federated Machine Learning framework, the process of detecting outliers—such as malicious nodes or poor-quality models—is computationally intensive [33], requiring a time complexity denoted as '*TIMEoutlier*' which is computed as follows.

Let,

P be the number of participants

Q be the data batches submitted by each node

The usage of statistical methods for detecting anomalies in the input data batch can asymptotically require O(P*Q) time.

$$TIMEoutlier = O(P * Q) \quad (17)$$

4.9.2 Computational overhead of Fault tolerant blockchain mechanism:

The incentive mechanism assigns rewards to participating nodes based on their contributions to the model and penalizes nodes detected as faulty nodes to maintain the integrity of the network. This involves computing contributions and validating rewards through smart contracts on the blockchain. Let, P be the number of participants, then the computational overhead of incentive distribution is calculated as shown in Formula 18.

$$TIMEincentive = O(P) \quad (18)$$

Each model update is recorded on the blockchain to ensure the trustworthiness of the updates. Let's denote the time required for processing these transactions as '*TIMEtransactions'*. For the entire blockchain network comprising P nodes, the total time complexity for processing model transaction updates is calculated as shown in Formula 19.

$$TIMEmodel = O(P * TIMEtrasactions) \quad (19)$$

Hence, the total computational overhead of the incentivized fault tolerant blockchain is computed using the Formula 20.

$$\text{TIMEblockchain} = O(TIMEincentive + TIMEmodel) \quad (20)$$

4.9.3 Computational overhead Federated Learning Model Aggregation:

Federated learning involves aggregating model updates from participating nodes. Let Q be the size of the model, and P be the number of participating nodes. The aggregation operation computes a weighted average of individual models. The time complexity of model aggregation is shown in Formula 21.

$$\text{TIMEfederation}(\text{P},\text{Q}) = O(P * Q) \quad (21)$$

4.9.4 Total computation overhead:

Combining the overheads from each component helps analyze the overall computational complexity of the system which is shown in Formula 22.

$$TIMEoverall = O(TIMEoutlier + \text{TIMEblockchain} + \text{TIMEfederation}) \quad (22)$$

Where, the TIMEoutlier and TIMEfederation scale linearly with P*Q, meaning as the number of participants increases or model size grows, the overhead increases linearly. The scalability of TIMEblockchain is influenced by the iPBFT consensus mechanism, which exhibits quadratic growth in relation to the number of consensus nodes (k).

The system uses novel iPBFT protocol which is a specific implementation of BFT designed for practical use cases. The iPBFT is designed to have lower computational complexity comparatively. As the number of nodes increases in the network, the overhead becomes more pronounced due to the quadratic message complexity of iPBFT. It achieves improved scalability compared to traditional BFT by reducing its communication complexity, as traditional BFT often exhibits cubic complexity. Many BFT algorithms become impractical in large networks due to their high communication overhead, resulting in increased response times and decreased throughput [34]. iPBFT is designed for practical applications and performs better in permissioned networks with a limited number of nodes. Proposed novelty of calculating incentives in PBFT does not impact much in the scaling as incentivization requires only O(P) time where P is number of participants. Hence, the novel iPBFT still works in quadratic time complexity. However, its quadratic scaling can still pose challenges as the network size increases significantly.

4.10 Threat model for analyzing scalability of proposed framework:

When designing a scalable system, it is crucial to account for how scaling affects security and the emergence of new threats as the system expands. A threat model for scalability helps identify, analyze, and mitigate potential risks that arise with increasing system size, complexity, and resource demands. In the proposed system, various threat model analyses were conducted, and the resulting findings are discussed in detail.

4.10.1 Threat model for Autoencoder based outlier detection:

In addition to evaluating model performance metrics and assessing the impact of threats such as adversarial attacks. By analyzing system behavior, valuable insights can be gained into the effectiveness of outlier detection algorithms in identifying anomalies and mitigating their impact on model training. This analysis helps assess how well the algorithms maintain model reliability in the presence of anomalous data. Moreover, observing system response to adversarial attacks provides valuable information on the robustness of the system and its ability to withstand malicious manipulation. Incorporating the Fast Gradient Sign Method (FGSM) for inducing adversarial attacks can have a notable impact on the accuracy of the autoencoder model. FGSM operates by perturbing

the input data in alignment with the gradient of the loss function concerning the input, with the objective of maximizing the loss and inducing misclassification. The introduction of such adversarial perturbations often leads to a decrease in accuracy observed within the autoencoder. This decline in accuracy underscores the susceptibility of the model to adversarial manipulation and highlights the importance of robustness and resilience in defending against such attacks. Evaluating the impact of FGSM-induced attacks on the autoencoder's accuracy provides valuable insights into the model's vulnerability to adversarial perturbations and underscores the need for enhanced defense mechanisms to safeguard against such threats. The classification report on figure 15 provides a comprehensive overview of the model's performance by presenting the counts of true positives, false positives, true negatives, and false negatives. In the threat model the accuracy seems to have reduced due to the adversarial attack.

4.10.2 Threat model for incentivized fault tolerant Blockchain network:

Figure 16 illustrates a scenario where nodes are intentionally shut down during different communication rounds in iPBFT network. It shows the communication rounds and the corresponding number of available nodes in each round within the network.

Figure 17 illustrates shutting down nodes during various communication rounds in an iPBFT network leads to noticeable drops in overall accuracy. This deliberate disruption of nodes at different stages of communication leads to inconsistencies and fluctuations in the accuracy levels observed across the network.

Figure 18 illustrates that Shutting down a specific node during entire communication rounds makes its accuracy zero. This analysis highlights the dynamic nature of accuracy in response to node disruptions, emphasizing the need for robust fault tolerance mechanisms and resilient consensus protocols in iPBFT networks.

Figure 19 illustrates when tampered data is transmitted from one port to another using Postman API, it introduces a significant challenge to the integrity and accuracy of the sender port within the iPBFT network. This leads to fluctuations in the accuracy of the tampered node in the accuracy graph.

As the system scales, the attack surface expands, increasing the risk of malicious node behavior and communication vulnerabilities. Ensuring secure data transmission and maintaining consensus

efficiency under growth are critical challenges. Resource exhaustion and potential synchronization issues could also arise, particularly in distributed environments. The proposed work demonstrates strong performance across various scenarios, effectively addressing the challenges associated with outlier detection in incentivized fault-tolerant blockchain-based federated machine learning [35].

4.11 Potential Trade-offs in System Performance:

The integration of outlier detection mechanisms may increase the overall training time for the federated learning model. This is due to the additional computations required to identify and handle outliers, which can lead to longer convergence times, especially in large datasets. Implementing proposed algorithms may necessitate additional computational resources, such as memory and processing power. This can strain the infrastructure, particularly in environments with limited resources or when scaling to a larger number of nodes. Increased resource consumption can lead to higher operational costs, which is particularly relevant in blockchain environments where resource allocation may impact transaction fees and overall system performance. Despite the inherent trade-offs, the system offers numerous advantages that justify its implementation. By carefully managing the trade-offs associated with resource consumption, training time, and operational costs, the advantages of the proposed framework can significantly outweigh the drawbacks, making it a valuable solution in the context of incentivized fault-tolerant blockchain-based federated machine learning.

5. CONCLUSION

The system uses autoencoder as an outlier detection mechanism which helps to filter malicious data before the data is allowed to get trained in the distributed network. The global model drift due to the malicious data injection is completely avoided because of this mechanism. Hence the data security is enhanced inside Federated Machine Learning environment. Practical Byzantine Fault Tolerance algorithms are essential for ensuring the reliability and integrity of distributed systems, particularly in scenarios involving malicious or faulty nodes. Incentive based PBFT mechanism (iPBFT) enables systems to maintain functionality and consensus on transactions, even in the presence of Byzantine faults, thereby enhancing system resilience and reducing susceptibility to disruptions or attacks. With strong guarantees of safety and liveness, iPBFT ensures that transactions are processed correctly and the system continues to progress despite potential faults. This fault tolerance mechanism enhances system liveness, stability, and reliability, thus mitigating the impact of node failures or network partitions and increasing system availability.

Future research will focus on improving the computational efficiency of both the autoencoder and the iPBFT protocol. Suggested avenues for exploration include investigating various architectures, such as transfer learning and Generative Adversarial Networks (GANs), for outlier detection instead of solely relying on autoencoders. To improve the computational efficiency of the iPBFT protocol, one potential approach is to analyze communication patterns and explore strategies that minimize message complexity and reduce the number of communication rounds required. However, it is important to note that there will be trade-offs associated with these optimizations.

REFERENCES

1. Li, T., Sahu, A. K., Talwalkar, A., et al., "Federated learning: challenges, methods, and future directions." *IEEE Signal Processing Magazine*, **37**.3 (2020): 50-60. https://doi.org/10.1109/msp.2020.2975749
2. Choudhury, O., Gkoulalas-Divanis, A., Salonidis, T., et al., "Differential privacy-enabled federated learning for sensitive health data." *arXiv preprint* arXiv:1910.02578 (2019). arXiv:1910.02578.
3. Yang, B., Cui, L., et al., "Security and privacy-enhanced federated learning for anomaly detection." *IEEE Transactions on Industrial Informatics* 18.5 (2021): 3492-3500. https://doi.org/10.1109/TII.2021.3107783
4. Contreras-Cruz, M. A., Correa-Tome, F. E., Lopez-Padilla, R., et al., "Generative adversarial networks for anomaly detection in aerial images." *Computers and Electrical Engineering* 106 (2023): 108470. https://doi.org/10.1016/j.compeleceng.2022.108470
5. Cao, X., Sun, G., Yu, H., et al., "PerFED-GAN: personalized federated learning via generative adversarial networks." *IEEE Internet of Things Journal* 10.5 (2022):3749-3762. https://doi.org/10.1109/jiot.2022.3172114
6. Ali, M., Karimipour, H., Tariq, M., "Integration of blockchain and federated learning for internet of things: recent advances and future challenges." *Computers & Security* 108 (2021): 102355. https://doi.org/10.1016/j.cose.2021.102355
7. Yang, X., Li, T., "A Blockchain-based federated learning framework for secure aggregation and fair incentives." *Connection Science* 36.1 (2024): 2316018. https://doi.org/10.1080/09540091.2024.2316018
8. Mirsky, Y., Doitshman, T., Elovici, Y., et al., "Kitsune: an ensemble of autoencoders for online network intrusion detection." *arXiv preprint arXiv:1802.09089* (2018). https://doi.org/10.14722/ndss.2018.23204

9. Yuan, X., He, P., Zhu, Q., et al., "Adversarial examples: attacks and defenses for deep learning." *IEEE transactions on neural networks and learning systems* 30.9 (2019): 2805-2824. https://doi.org/10.1109/tnnls.2018.2886017
10. Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., et al., "Anomaly-based network intrusion detection: techniques, systems and challenges." *computers & security* 28.1-2 (2009): 18-28. https://doi.org/10.1016/j.cose.2008.08.003
11. Jin, X., Ma, C., Luo, S., et al., "Distributed IIoT anomaly detection scheme based on blockchain and federated learning." Journal of Communications and Networks 26.2 (2024): 252-262. https://doi.org/10.23919/jcn.2024.000016
12. Yang, Q., Liu, Y., Chen, T., et al., "Federated machine learning: concept and applications." *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019): 1-19. https://doi.org/10.1145/3298981
13. Adnan, M., Kalra, S., Cresswell, J. C., et al., "Federated learning and differential privacy for medical image analysis." *Scientific reports* 12.1 (2022): 1953. https://doi.org/10.1038/s41598-022-05539-7
14. Li, S., Cheng, Y., Liu, Y., Wang, W., et al., "Abnormal client behavior detection in federated learning." *arXiv preprint arXiv:1910.09933* (2019). https://doi.org/10.1063/pt.5.028530
15. Cui, L., Qu, Y., Xie, G., et al., "Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures." *IEEE Transactions on Industrial Informatics* 18.5 (2021): 3492-3500. https://doi.org/10.1109/tii.2021.3107783
16. CFE, CITP. "Blockchain basics: a non-technical introduction in 25 steps." *The CPA Journal* 93.3/4 (2023): 13-13. https://doi.org/10.1007/978-1-4842-2604-9
17. Sadeghi, S., Mahmoudzadeh Niknam, M., Bagheri, N., et al., "Cryptanalysis of full-round sfn block cipher a lightweight block cipher, targeting iot systems." Scientia Iranica (2023). https://doi.org/10.24200/sci.2023.59581.6319
18. Behniafar, M., Mahjur, A., Nowroozi, A., "Anomaly detection fog (ADF): a federated approach for internet of things." Scientia Iranica 30.2 (2023): 465-476. https://doi.org/10.24200/sci.2022.57774.5410
19. Rezaeian Koochi, M. H., Hemmatpour, M. H., "Density-based unsupervised learning approach for generators coherency evaluation in complex domain." Scientia Iranica (2024). https://doi.org/10.24200/sci.2024.62573.7915
20. Ressi, D., Romanello, R., Piazza, C., et al., "AI-enhanced blockchain technology: a review of advancements and opportunities." *Journal of Network and Computer Applications* (2024): 103858. https://doi.org/10.1016/j.jnca.2024.103858
21. Abdel-Sater, R., Ben Hamza, A., "A federated large language model for long-term time series forecasting." *arXiv preprint arXiv:2407.20503* (2024). https://arxiv.org/pdf/2407.20503

22. Bahga, A., Madisetti, V., “Blockchain applications: a hands-on approach” Vpt, 2017. https://doi.org/10.4236/jsea.2016.910036
23. Cao, X., Sun, G., Yu, H., et al., "PerFED-GAN: personalized federated learning via generative adversarial networks." *IEEE Internet of Things Journal* 10.5 (2022): 3749-3762. https://doi.org/10.1109/jiot.2022.3172114
24. Liu, S., et al., "{XFT}: practical fault tolerance beyond crashes." *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016. https://doi.org/10.1109/msp.2010.134
25. Dautov, R., Husom, E. J., "Raft protocol for fault tolerance and self-recovery in federated learning." Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. 2024. https://doi.org/10.1145/3643915.3644093
26. Bach, L. M., Mihaljevic, B., Zagar, M., "Comparative analysis of blockchain consensus algorithms." *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 2018. https://doi.org/10.23919/mipro.2018.8400278
27. Dataset1: https://yann.lecun.com/exdb/mnist/
28. Dataset 2: https://www.kaggle.com/datasets/zalando-research/fashionmnist/
29. Zhang, L., Luo, Y., Bai, Y., et al., "Federated learning for non-iid data via unified feature learning and optimization objective alignment." *Proceedings of the IEEE/CVF international conference on computer vision*. 2021. https://doi.org/10.1109/iccv48922.2021.00438
30. Mortezaie, Z., Hassanpour, H., Beghdadi, A., "Re-identification in video surveillance systems considering appearance changes." Scientia Iranica 31.2 (2024): 103-117. https://doi.org/10.24200/sci.2023.57617.5331
31. Ahrabi Tabriz, M., Rafiei Atani, T., Ashtiani, M., et al., "Visual creativity through concept combination using quantum cognitive models." Scientia Iranica 31.2 (2024): 118-135. https://doi.org/10.24200/sci.2023.61494.7340
32. Khosravani Pour, L., Farrokhi, A., "Language recognition by convolutional neural networks." Scientia Iranica 30.1 (2023): 116-123. https://doi.org/10.24200/sci.2022.59110.6064
33. Abdel Sater, R., Ben Hamza, A., "A federated learning approach to anomaly detection in smart buildings." *ACM Transactions on Internet of Things* 2.4 (2021): 1-23. https://doi.org/10.1145/3467981
34. He, F., Feng, W., Zhang, Y., et al., "Improved fault-tolerant consensus based on the pbft algorithm." *IEEE Access* 10 (2022): 30274-30283. https://doi.org/10.3390/electronics12092049
35. Dewo, K. T., Yasin, V., Budiman, T., et al., "IT infrastructure dashboard monitoring application development using grafana and promotheus, a case study at astra polytechnic school." *2023 International Conference of Computer Science and Information Technology (ICOSNIKOM)*. IEEE, 2023. 10.1109/ICoSNIKOM60230.2023.10364485

APPENDICES

Figure 1: System Architecture

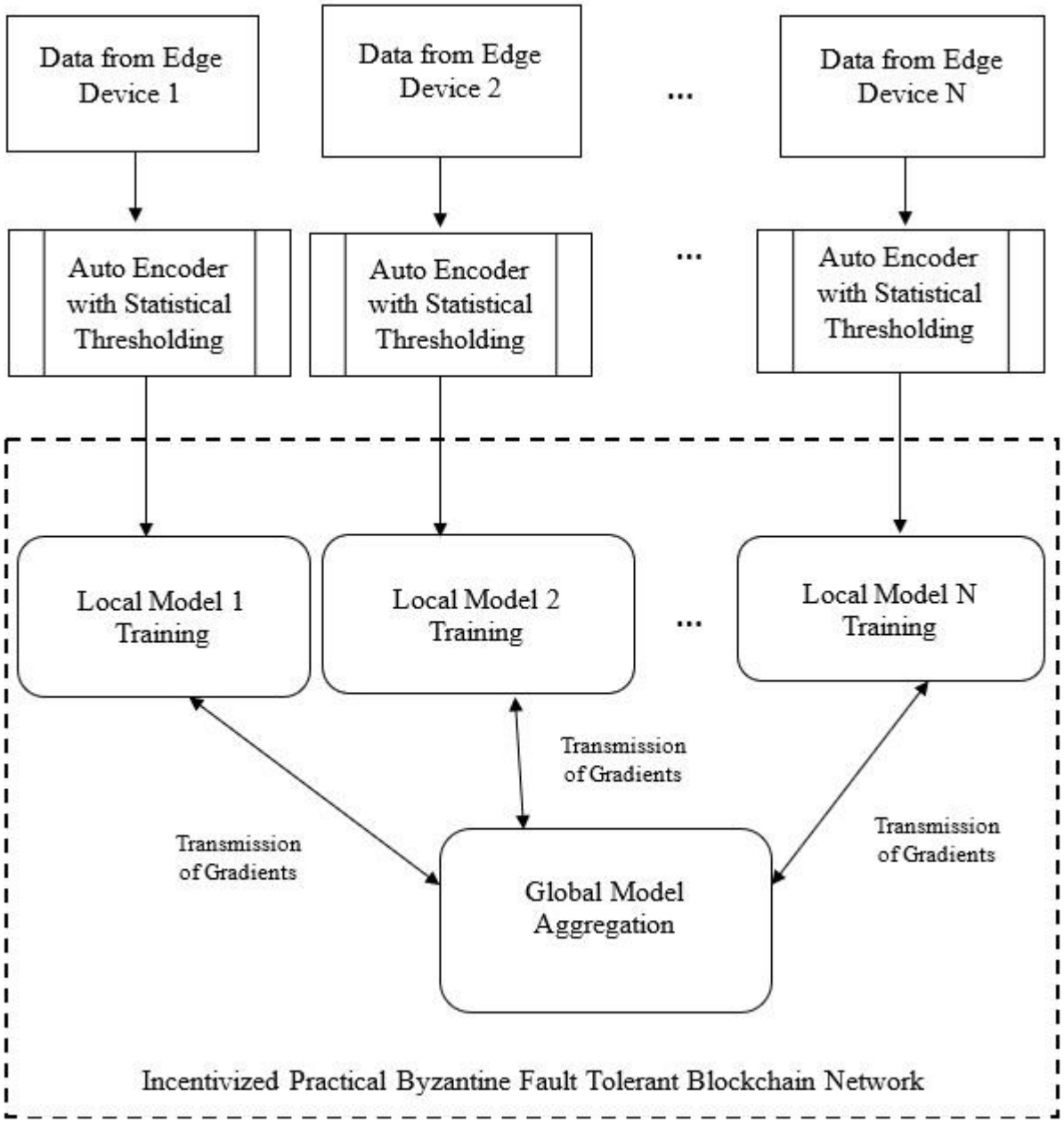


Figure 2: Working of incentivized Practical Byzantine Fault tolerant Blockchain Network

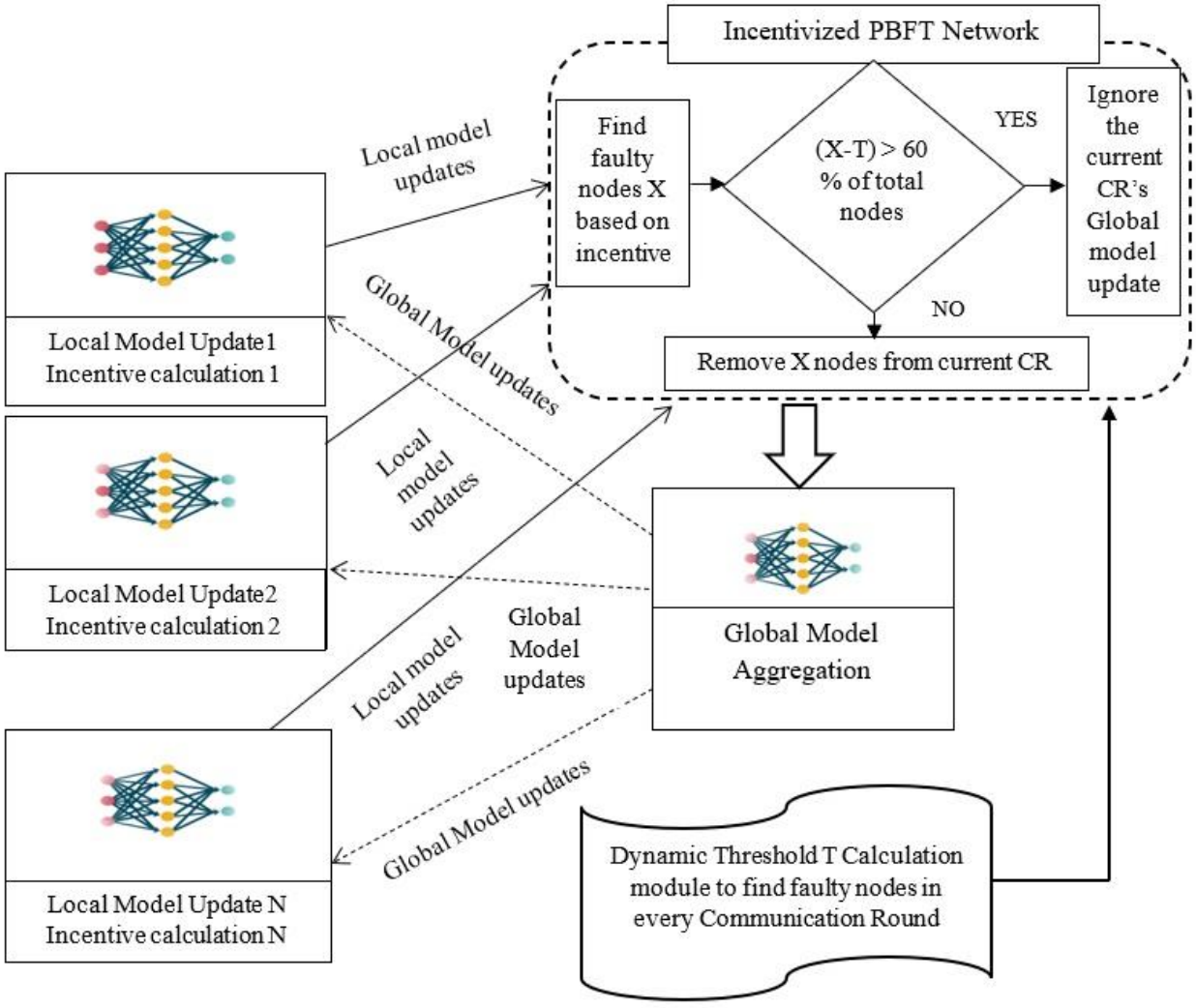


Figure 3: Anomaly scores and estimated threshold value

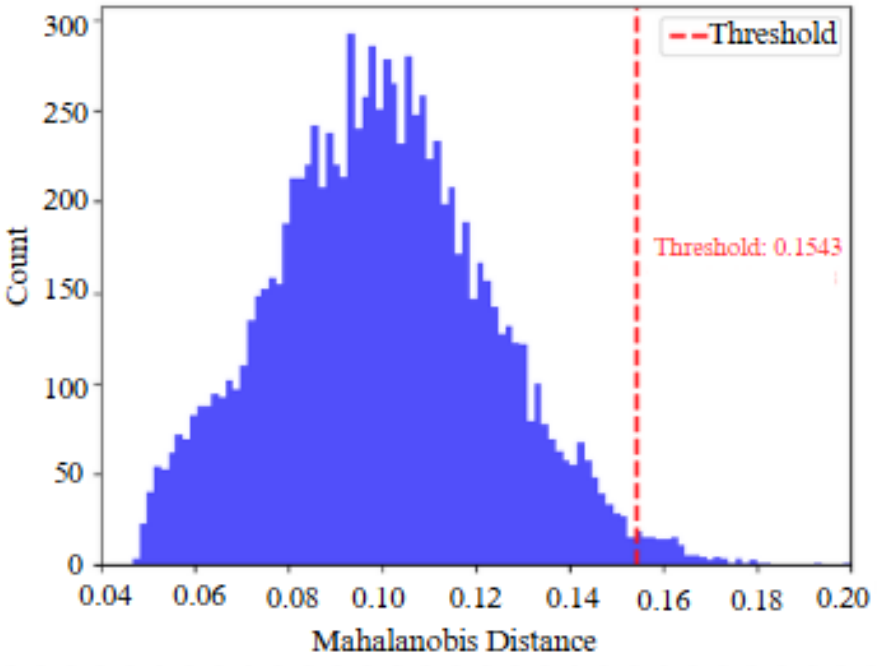
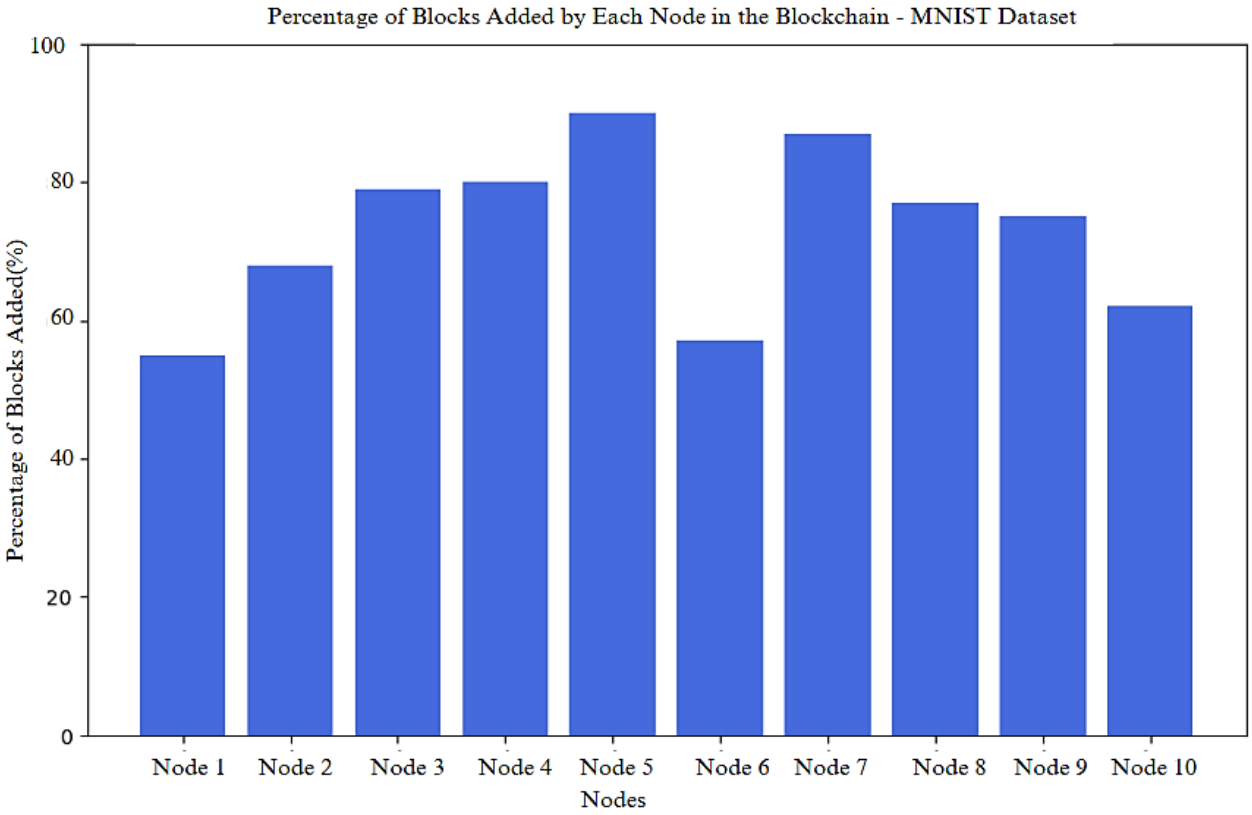


Figure 4: Blocks added in Blockchain



,

Figure 5: Total commits per node

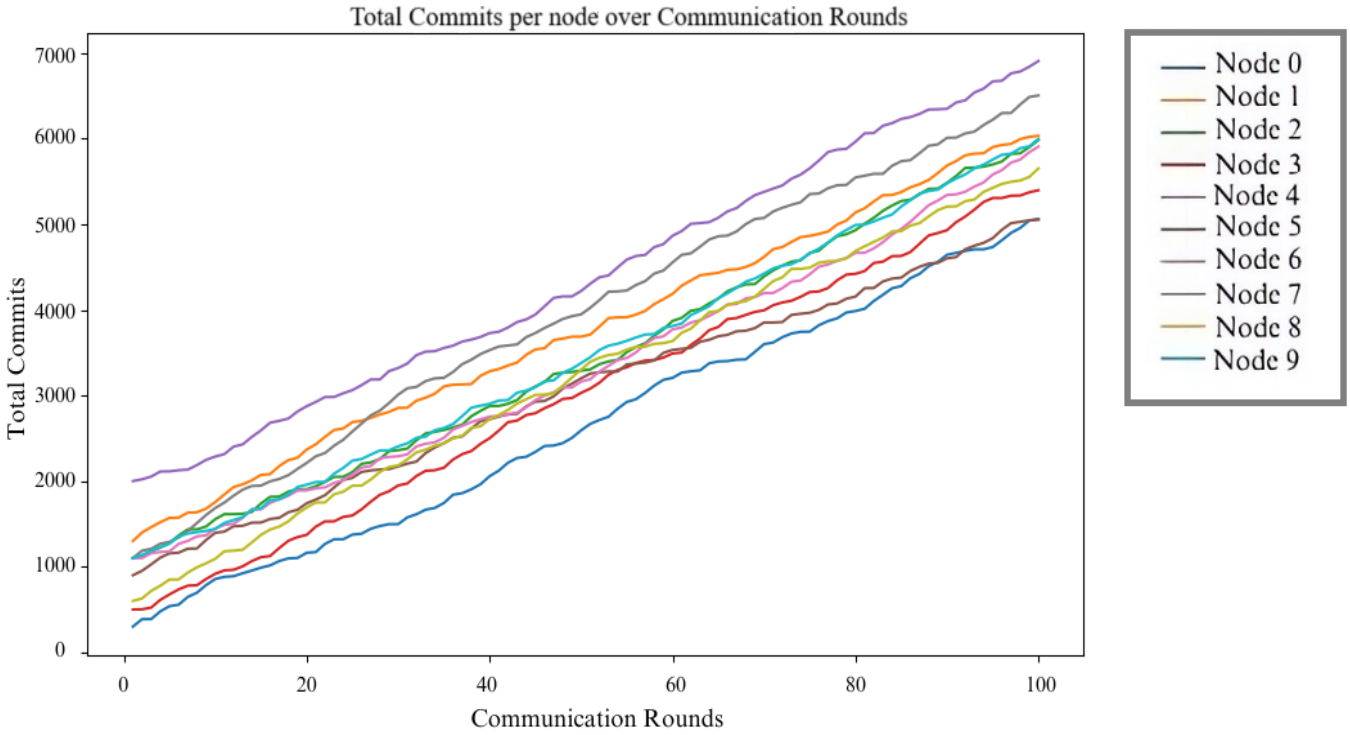


Figure 6: Incentives of nodes over communication rounds

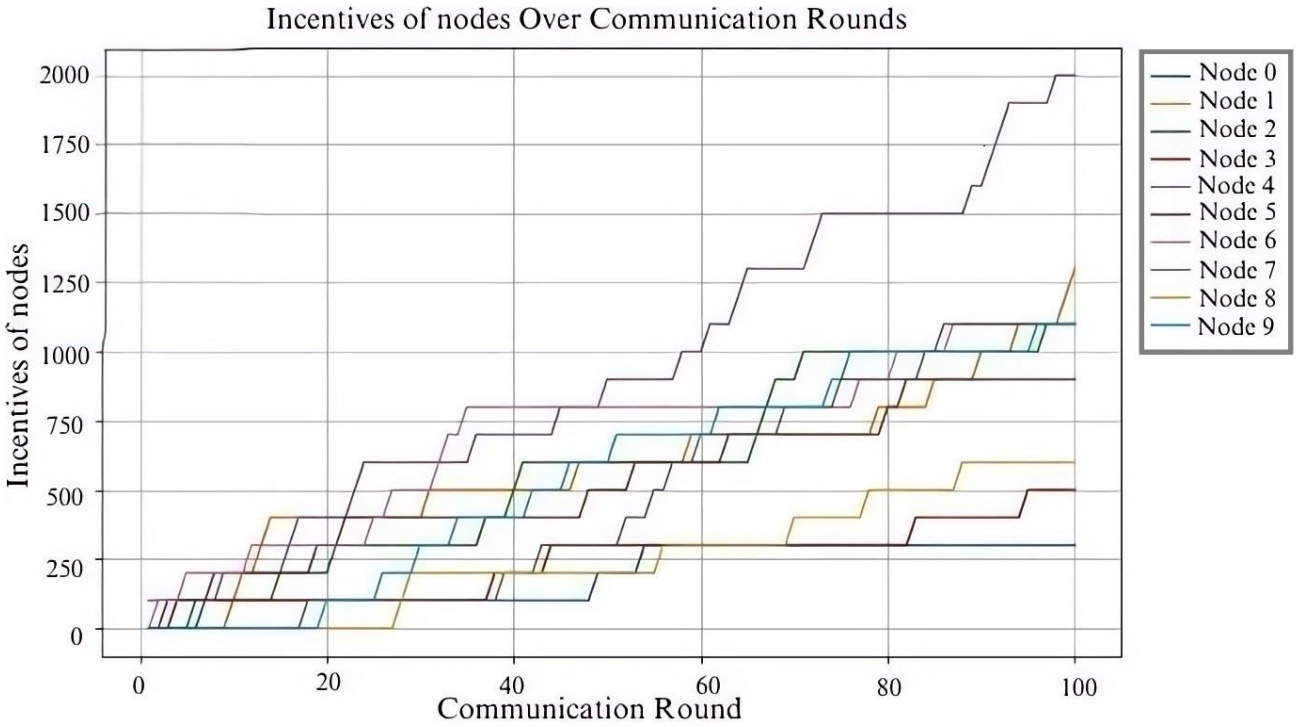


Figure 7: Faulty nodes over communication rounds

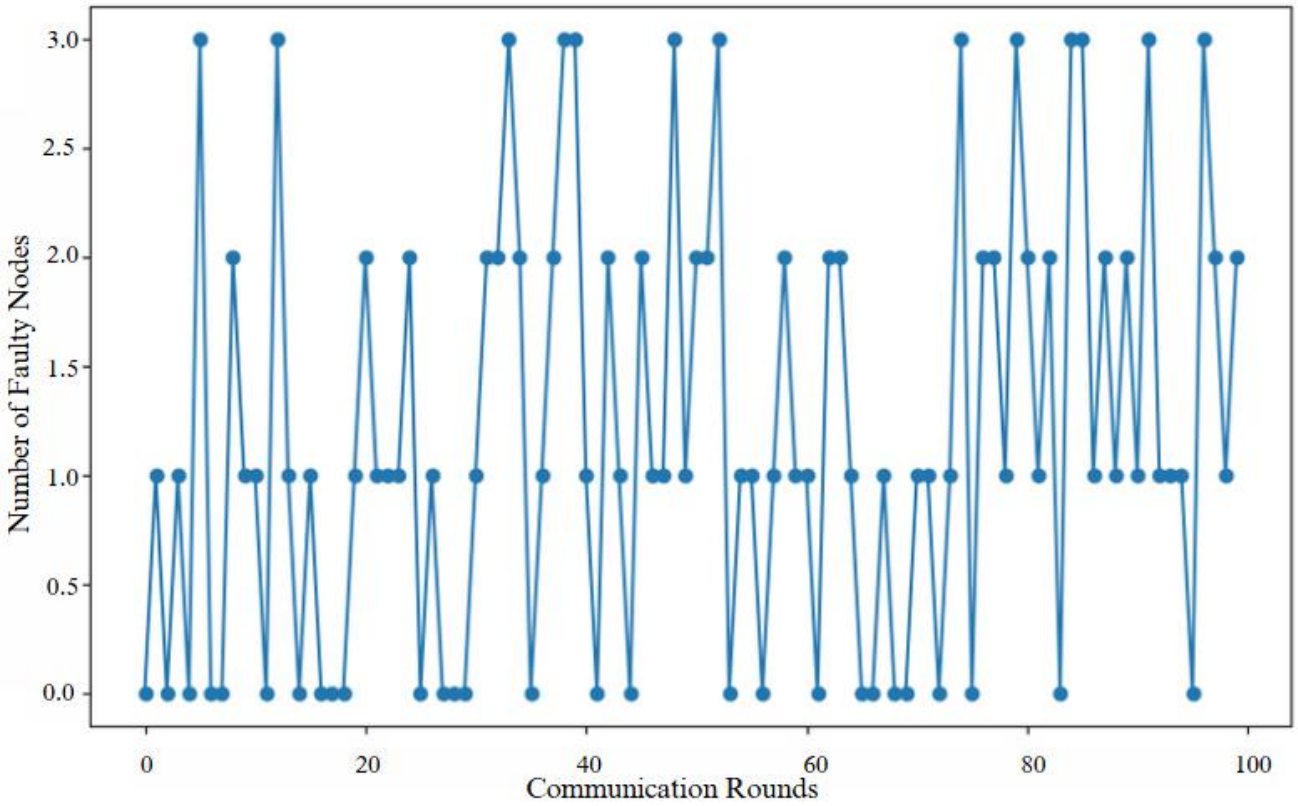


Figure 8: Incentives of nodes during training process

Node_0	Node_1	Node_2	Node_3	Node_4	Node_5	Node_6	Node_7	Node_8	Node_9
8	16	17	16	17	7	13	18	8	5
10	11	9	12	13	15	12	20	16	0
0	10	0	9	10	18	0	13	16	13
13	12	7	16	16	11	7	16	16	7
18	0	7	16	0	10	11	16	0	12
19	10	16	12	15	20	18	14	11	18
15	0	15	18	10	14	10	13	17	20
20	7	16	17	9	9	14	8	11	0
7	10	19	15	0	15	14	10	19	11
0	20	20	10	8	10	11	20	16	19
11	13	15	9	15	17	18	11	15	19
19	15	8	16	7	7	11	7	12	13
13	7	11	8	11	18	15	12	9	17
19	16	19	18	9	13	14	7	19	14
12	16	10	10	0	8	16	16	18	11
17	11	7	18	7	0	11	15	19	0
14	14	10	19	11	14	17	17	11	16
9	15	14	9	9	8	17	9	11	7
8	7	14	15	8	13	13	11	19	0
10	10	7	17	14	16	17	14	11	15
10	13	17	13	17	9	18	17	17	14
18	8	0	15	9	7	15	19	12	7
8	20	17	9	16	0	10	0	15	19
15	0	0	18	15	9	15	12	19	11
10	14	16	12	17	14	15	13	20	18
0	10	10	14	7	12	11	0	18	17
17	14	15	9	11	19	19	10	0	20
19	9	13	14	11	19	13	14	18	13
15	15	7	16	8	10	19	17	20	11
8	11	19	0	9	13	8	8	17	16
13	10	8	12	20	13	7	7	16	9

Figure 9: Network metrics of transactions

Bandwidth (KB/s)	Throughput (KB/s)	Latency (ms)	Scalability
5376.41	70.97	17	10
9710.28	29.04	9	10
7850.01	83.62	12	10
4276.17	61.61	3	10
9533.34	24.7	3	10
4275.23	29.26	10	10
6491.85	99.01	19	10
1539.77	92.65	2	10
3270.94	44.36	4	10
5307.42	10.23	1	10
6632.57	41.89	20	10
5583.21	19.23	6	10
1365.5	7.21	3	10
7502.83	74.07	19	10
3868.29	2.04	11	10
6675.02	22.06	14	10
7267.19	43.13	19	10
8568.31	25.68	19	10
4076.93	64.49	7	10
5004.12	11.92	5	10
4452.39	66.49	14	10
3052.04	55.91	18	10
9559.45	99.78	16	10
5038.74	20.2	18	10
1416.73	76.11	5	10
5953.38	14.03	6	10
3332.84	94.85	12	10
9328.12	44.13	20	10
3273.31	42.85	4	10
9986.56	75.67	13	10

Figure 10: Local model loss across communication rounds for MNIST and Fashion MNIST Datasets

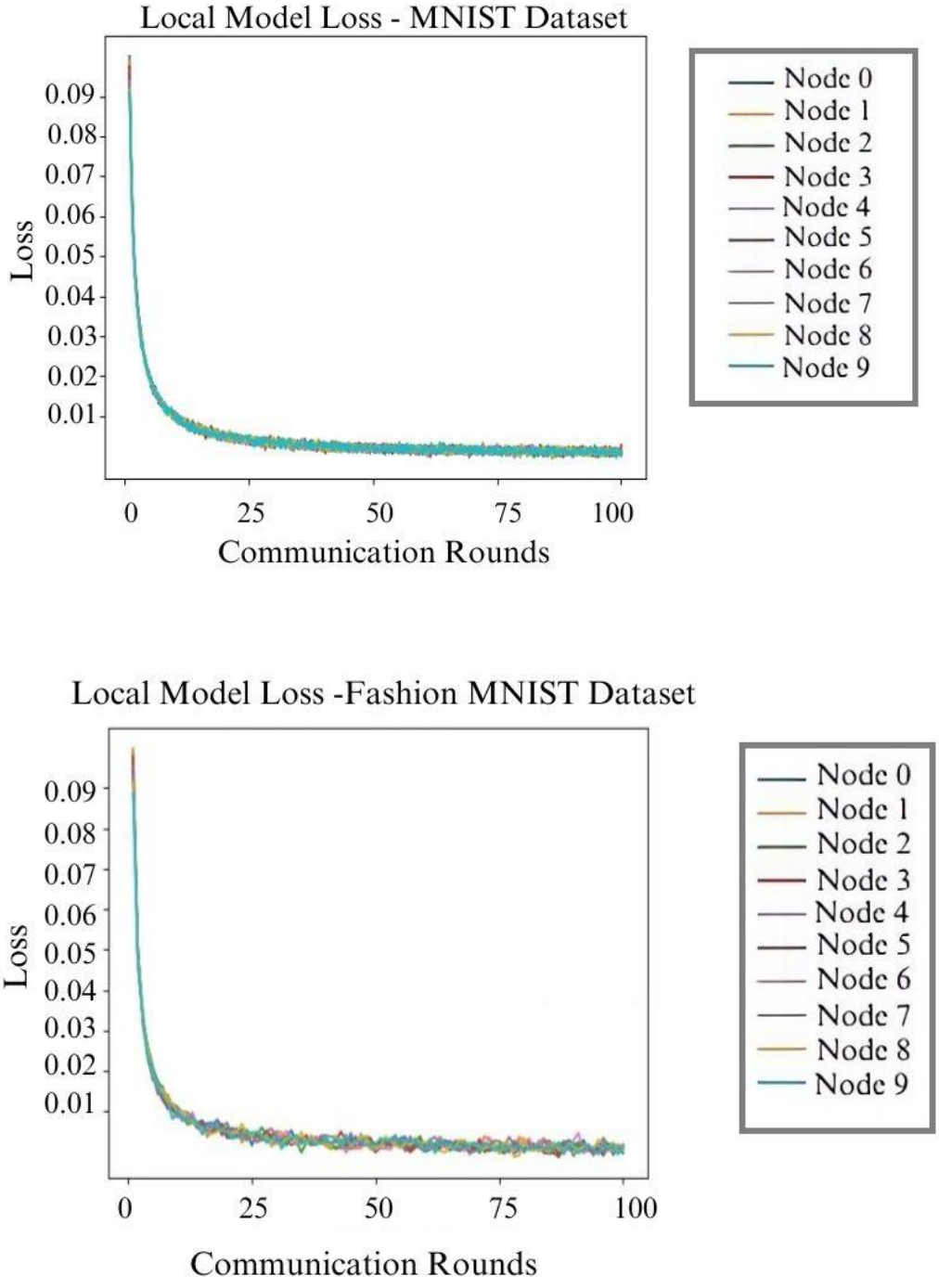


Figure 11: Local model Accuracy across communication rounds for MNIST and Fashion MNIST Datasets

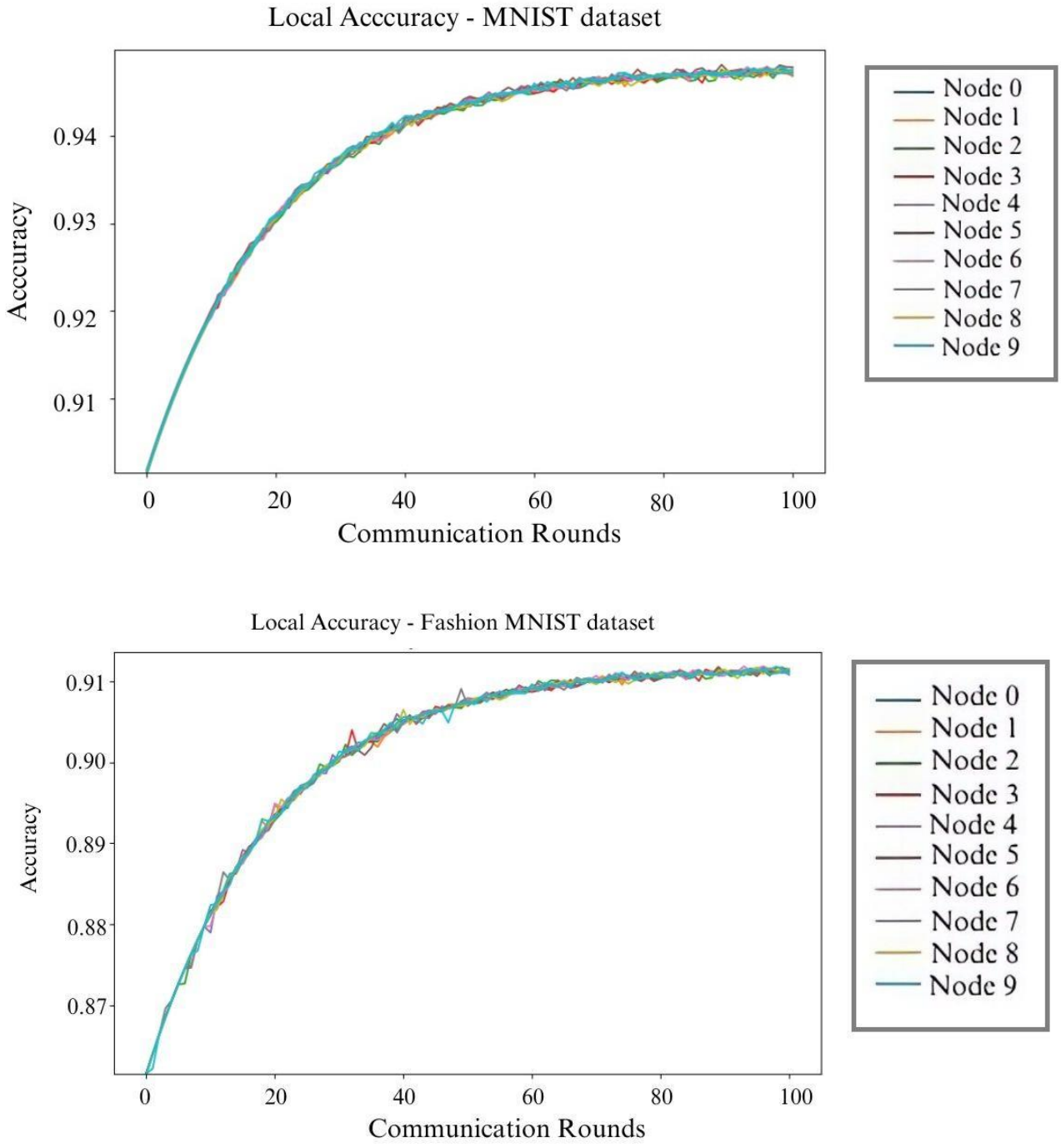


Figure 12: Global Accuracy across communication rounds for MNIST and Fashion MNIST Datasets

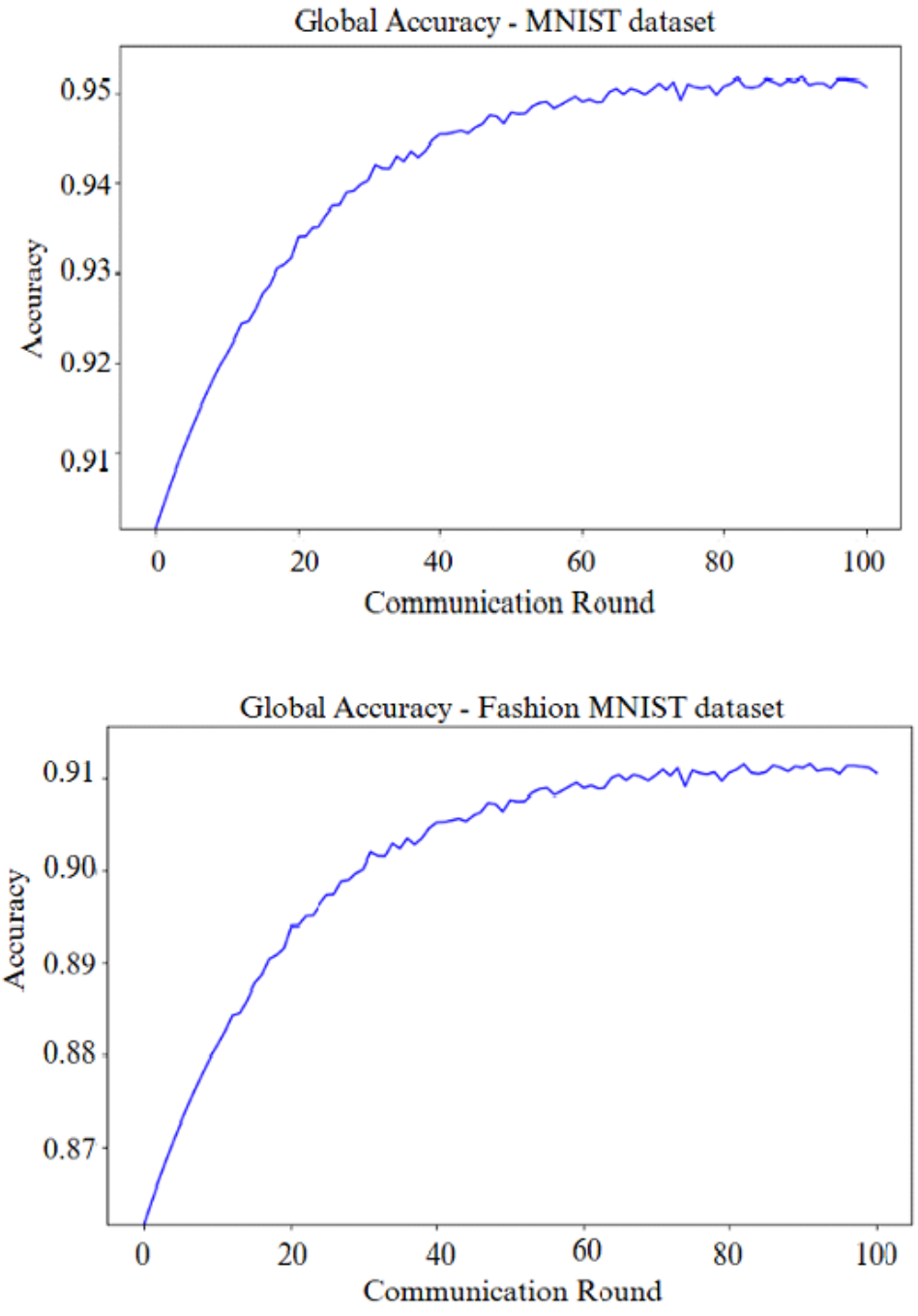


Figure 13: Global loss across communication rounds for MNIST and Fashion MNIST Datasets

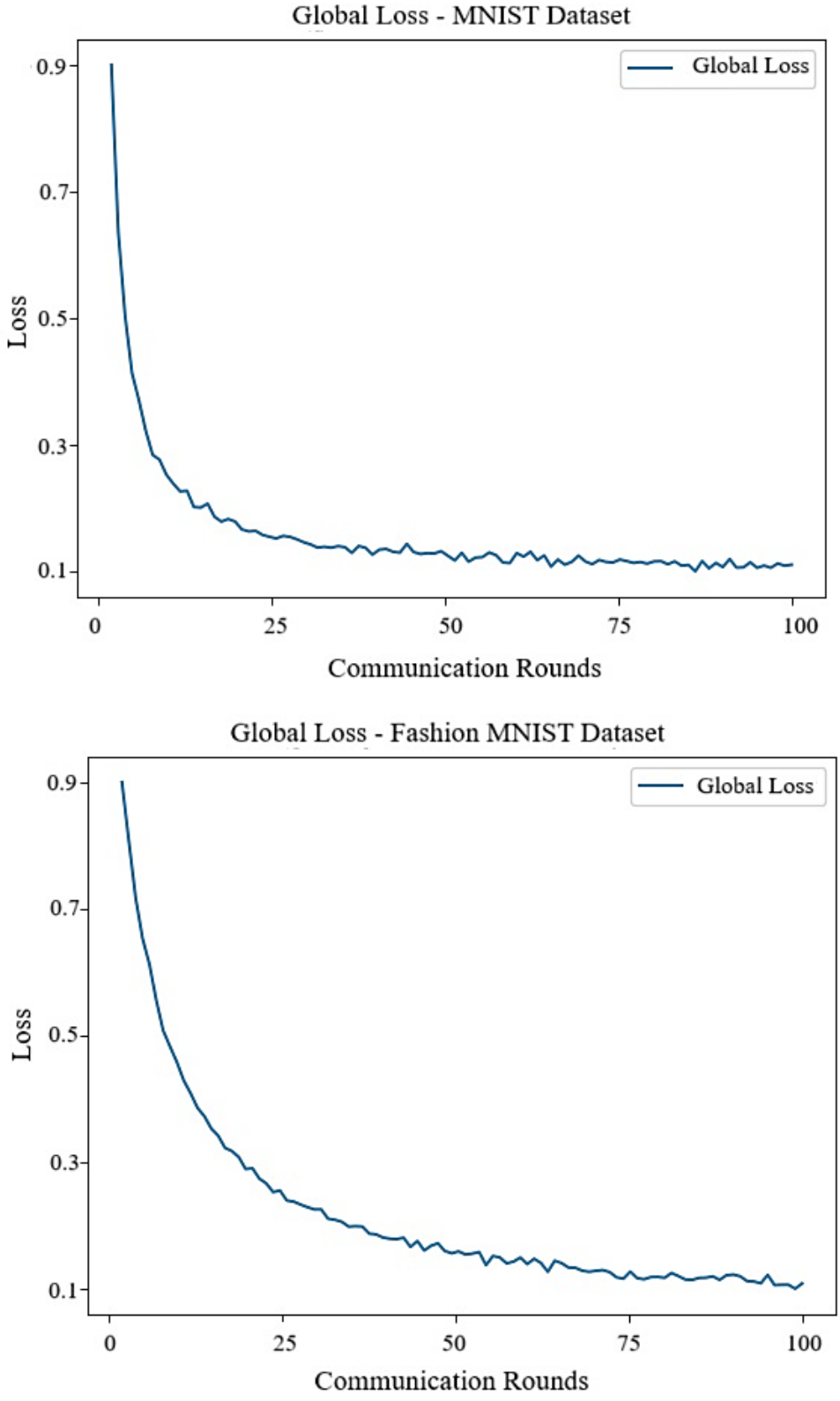


Figure 14: FML Global Model Evaluation with various Metrics

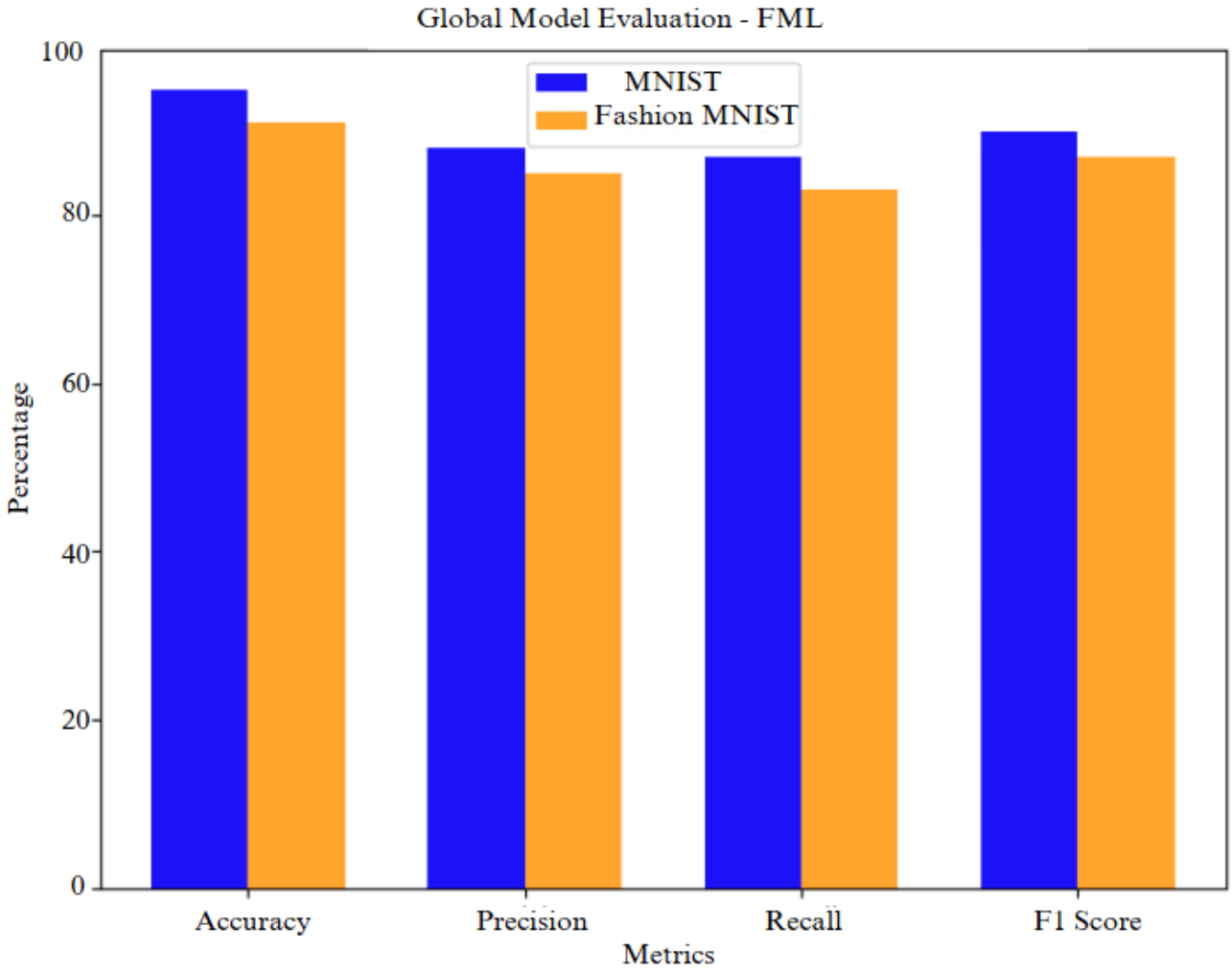


Figure 15: Classification report for threat model analysis of autoencoder

Classification Report:

	Precision	Recall	F1-score	Support
False	0.89	0.86	0.87	9000
True	0.61	0.63	0.67	1000
Accuracy			0.77	10000
Macro avg	0.64	0.63	0.64	10000
Weighted avg	0.80	0.77	0.78	10000

Figure 16: Threat model of node availability for each communication round

Comm_Round	Node Avalaibility
1	1, 1, 1, 0, 0, 1, 1, 1, 0, 1
2	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
3	0, 1, 1, 1, 1, 1, 1, 1, 1, 1
4	1, 0, 1, 1, 1, 1, 1, 1, 0, 1
5	1, 1, 1, 1, 1, 1, 0, 1, 1, 0
6	1, 1, 1, 1, 1, 1, 0, 1, 1, 1
7	1, 1, 1, 1, 0, 1, 1, 1, 1, 1
8	1, 0, 1, 1, 1, 1, 0, 1, 1, 1
9	1, 1, 1, 0, 0, 1, 1, 0, 1, 1
10	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
11	1, 0, 1, 1, 0, 1, 1, 1, 1, 1
12	1, 1, 1, 1, 0, 1, 1, 0, 0, 1
13	0, 1, 1, 0, 1, 1, 1, 1, 1, 0
14	0, 1, 0, 1, 1, 1,1, 1, 0, 1
15	1, 1, 1, 1, 1, 1, 1, 1, 1, 0
16	1, 0, 1, 1, 0, 1, 1, 1, 1, 1
17	1, 1, 1, 0, 1, 1, 1, 1, 0, 1
18	1, 1, 0, 1, 1, 1, 1, 1, 1, 0
19	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
20	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
21	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
22	1, 1, 1, 1, 1, 1, 1, 0, 1, 1
23	0, 1, 1, 0, 1, 1, 1, 1, 1, 1
24	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
25	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
26	1, 1, 1, 1, 1, 0, 0, 1, 1, 1
27	1, 1, 1, 1, 1, 1, 1, 1, 0, 1
28	0, 1, 1, 0, 1, 1, 1, 1, 1, 1
29	1, 1, 1, 1, 1, 1, 0, 1, 1, 1
30	1, 1, 1, 1, 1, 0, 0, 1, 1, 1
31	1, 0, 1, 0, 0, 1, 1, 1, 1, 1
32	0, 1, 1, 1, 1, 1, 1, 1, 1, 1
33	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
34	1, 1, 1, 0, 1, 1, 1, 1, 1, 0
35	1, 1, 1, 0, 1, 1, 1, 0, 1, 1

Figure 17: Threat model of Local model accuracy with different node shut downs

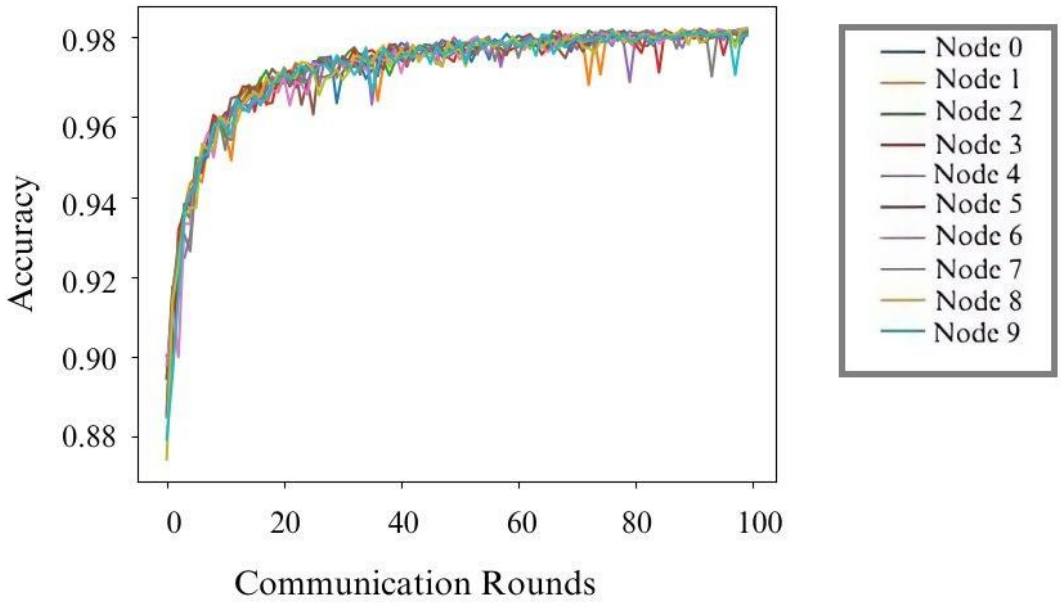


Figure 18: Threat model of local model accuracy for the individual node shutdown

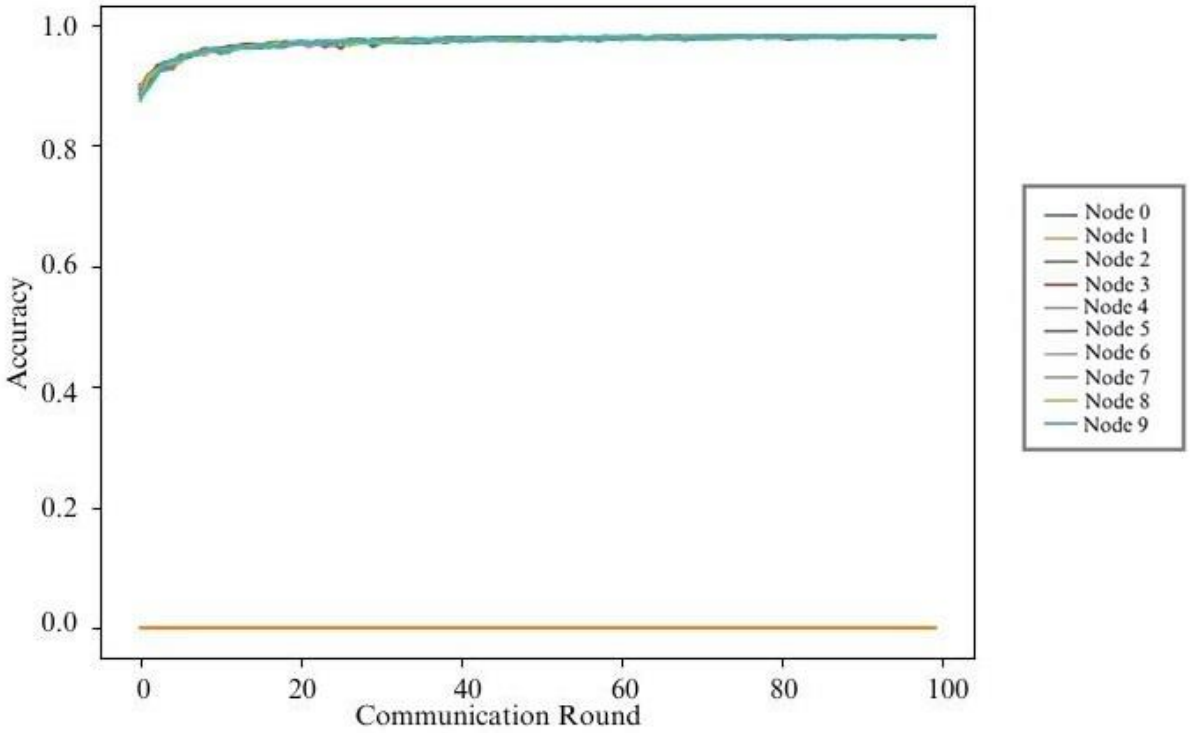
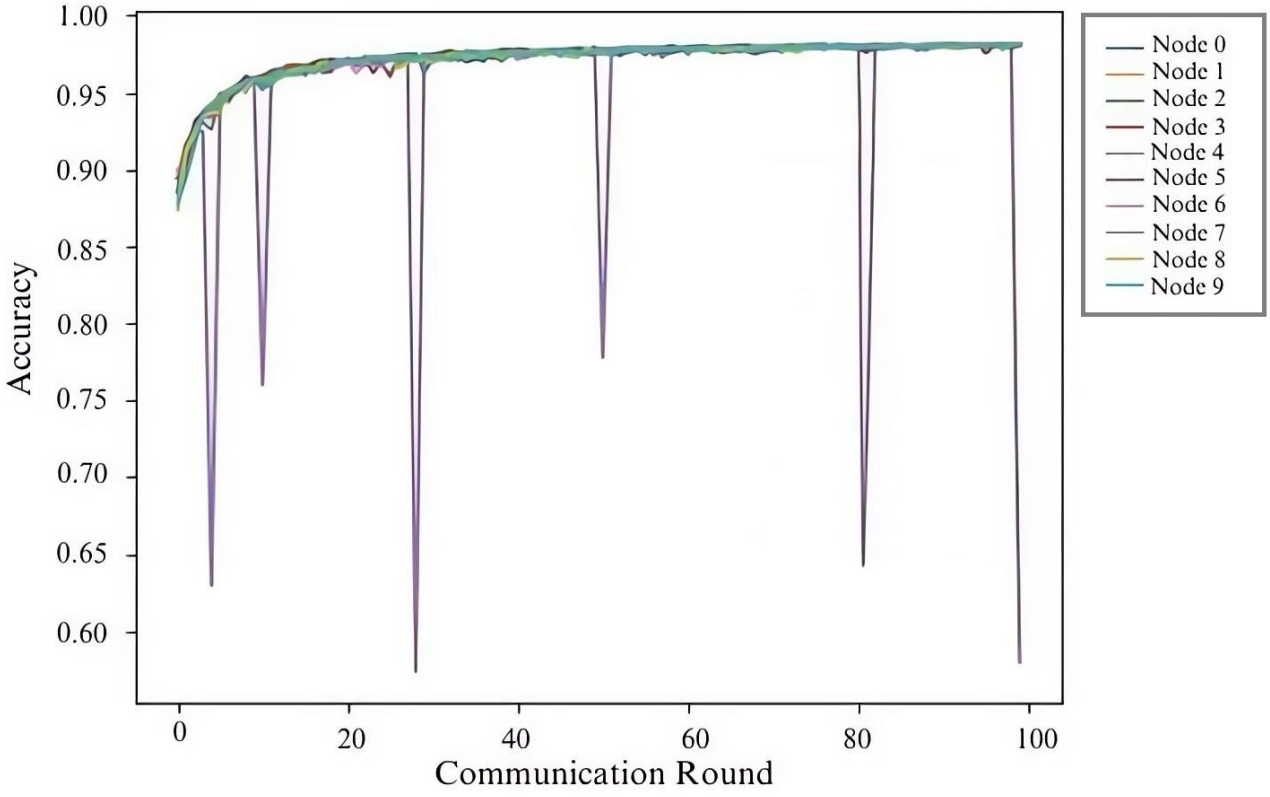


Figure 19: Threat model of local model accuracy for sending tampered data as input



AUTHORS BIOGRAPHY

Ms. D Dharani is working as an Assistant Professor [Senior Grade] in the Department of IT in PSG College of Technology, India. She received her M.Tech degree in Information Technology in the year 2016 and B.Tech degree in Information Technology in the year 2014 from Anna University, Chennai, India, and is currently pursuing PhD in Information and Communication Technology under Anna University, Chennai, India. She has published papers in International/National Journals/Conferences. She has published a Book entitled "Edge Computing: Fundamentals, Advances and Applications" by CRC Press, Taylor & Francis Group, LLC. She had completed PSG CARE Sponsored One year Certification Course on Data Science. She had successfully completed NPTEL and Coursera certification courses on Blockchain Platforms and Cybersecurity for IoT. Her research interest includes areas of blockchain technology, Edge computing security and Data analytics. You may contact her at ddi.it@psgtech.ac.in or dharani0609@gmail.com

Dr. K. Anitha Kumari [IEEE Senior Member] is an Associate Professor in the IT Department at PSG College of Technology, India, with 14+ years of experience. She has presented UGC-sponsored research on Quantum Cryptography in the USA and visited foreign universities, including CQT at the National University of Singapore. She holds a granted patent [Grant No. 409140, 2022] and five published patents. She passed the Patent Agent Examination (PAE) in 2024 and has published 130+ technical papers, three editorial books, and one authored book on Edge Computing. As an active reviewer for IEEE and Springer journals, she contributes to international conferences. She is an IPR Cell Member and BTech IT Coordinator, securing AICTE funding (~₹12L) for Edge Computing research and contributing to the ₹1.5Cr DST-FIST-funded Center for Cyber Security Research. A Gold Medalist from Anna and Avinashilingam Universities and Best Outgoing Student (2010) at PSG Tech, she has guided one Ph.D. scholar to completion and currently supervises seven. She also serves as a Ph.D. Thesis Evaluator, Oral Examination Panel Examiner, and Doctoral Committee Member. You may contact her at kak.it@psgtech.ac.in.