ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2026.0577

UNCERTAINTY-AWARE FEDERATED LEARNING FOR TRAFFIC PREDICTION IN 5G NETWORK SLICING ENVIRONMENTS

Vishal Kumar and Vikas Maheshkar
Department of Information Technology, Netaji Subhas University of Technology, India

Abstract

The advent of 5th generation (5G) wireless networks and Internet of Things (IoT) has led to the development of the need for privacy-aware and precise traffic predictions in network slicing scenarios. Centralized machine learning models usually encounter various obstacles due to data privacy, high communication costs, and scalability concerns. In order to mitigate such problems, this paper presents a new method of Secure Uncertainty-Aware Federated Learning (Secure UA-FL), which utilizes Bayesian LSTM models in combination with Monte Carlo Dropout (MCD) technique to predict traffic uncertainty in distributed edge nodes. The model is based on the uncertainty-aware aggregation, which adjusts client weights adaptively during the training process, and uses Krum-based Byzantine defense approach for improved performance and security. The proposed model is tested on synthesized 5G traffic datasets corresponding to eMBB, URLLC, and mMTC 5G network slices. The Secure UA-FL framework provides a scalable, secure, and privacy-aware solution for traffic prediction in dynamic 5G network slicing scenarios.

Keywords:

Federated Learning, 5G Network Slicing, Traffic Prediction, Bayesian LSTM, Byzantine Defense

1. INTRODUCTION

The 5G wireless networks have dramatically changed the way we communicate today with their high-speed connectivity, the ability to communicate with ultra-low latency, and massive deployments of the Internet of Things (IoT). Network slicing is one of the key technologies of 5G, where a single physical network will be divided into multiple virtual networks with different Quality of Service (QoS) requirements [7]. Network slices can be categorized into three broad types: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communication (URLLC) and massive Machine-Type Communication (mMTC) with different traffic characteristics and service requirements. Highly dynamic resource allocation and intelligent network management require accurate traffic prediction mechanisms that are able to cope with rapidly changing network conditions [2]. Most of the current traffic prediction methods are based on centralized machine learning models that gather traffic information from the distributed edge nodes and then train the model in a central server. Centralized models have high prediction accuracy but have several drawbacks such as privacy leakage, excessive communication overhead, and scalability problems in large-scale 5G-IoT networks [3].

To facilitate collaborative model training while maintaining privacy of the user data, a distributed learning paradigm called Federated Learning (FL) has been proposed in the recent years [1, 6]. In FL, each edge node trains the machine learning models with local data and only sends model parameters to a central server. This greatly facilitates privacy in preservation and a lower cost of communication. But, practical 5G traffic prediction system encounters great difficulties in conventional FL. The distribution of traffic in slices and geographic regions is very skewed and non-IID. Second, in standard Federated Averaging (FedAvg), all clients are given the same weights, irrespective of their prediction quality. Third, FL systems are still susceptible to Byzantine attacks in which the malicious clients try to manipulate the global model by sending the corrupted model updates [11] [12].

Recently, some research has been conducted on FL based traffic prediction frameworks for 5G network slicing [1] – [6]. To address this, Dutta et al. [1] proposed a federated learning approach to address the prediction-based load distribution problem in 5G slicing environments. Rakkiannan and his colleagues [4] combined FL and Software Defined Networking (SDN) and Network Function Virtualisation (NFV) to automate network slicing at the edge. Likewise, Phyu et al. [6] studied federated traffic forecasting in mobile network slices. While these methods provide an improvement in preserving privacy, they are only partial solutions for uncertainty estimation and secure aggregation.

This paper presents an Uncertainty-Aware Federated Learning (UA-FL) framework to address the above challenges for traffic prediction in 5G network slicing applications. The proposed model combines Bayesian LSTM networks with Monte Carlo Dropout for uncertainty estimation of prediction at edge nodes. The proposed strategy dynamically adapts the contribution of the clients according to their levels of confidence to enhance learning under non-IID traffic condition. In addition, a Secure UA-FL variant integrates Krum-based Byzantine defense to provide robustness against malicious client updates.

The major contributions of this paper are stated below:

- A Bayesian LSTM model with Monte Carlo Dropout is developed for uncertainty-aware traffic prediction in 5G network slicing environments.
- An adaptive uncertainty-aware federated aggregation mechanism is proposed to boost learning performance under heterogeneous traffic distributions.
- A Byzantine-robust secure aggregation framework based on the Krum algorithm is integrated into the federated learning pipeline.
- Extensive experimental evaluation is conducted across eMBB, URLLC, and mMTC traffic slices to validate prediction accuracy, security robustness, and communication efficiency.

2. RELATED WORK

With the growing complexity and diversity of services in 5G wireless communication systems, traffic prediction has emerged as a key research topic. Traditionally time-series traffic forecasting has been performed using statistical models such as

AutoRegressive Integrated Moving Average (ARIMA) model. But these techniques cannot handle the complicated non-linear traffic behavior of modern 5G-IoT network [2].

The models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and CNN-LSTM models have been found to perform better in traffic prediction problems. Kulkarni et al. [2] gave a comprehensive survey on the machine learning based approaches to 5G network slicing traffic prediction and reported the RMSE value ranging from 0.04 to 0.06 for LSTM based approaches. Mihai et al. [3] proposed an adaptive network slicing framework for real-time traffic classification and have come up with a prediction-based slice management to ensure efficient resource allocation.

While the centralized deep learning models offer high prediction accuracy, there is a need for continuous flow of traffic data from distributed edge devices to centralized cloud servers. This results in significant communication overhead and privacy concerns, especially in 5G networks that enable IoT applications.

FL is a potential solution to distributed learning in wireless networks for privacy protection. Dutta et al. [1] proposed an FL scheme for load balancing in 5G network slicing environment based on prediction. Their method minimized communication overhead and yet achieved an acceptable prediction accuracy. However, the framework was not working well in the case of heterogeneous data.

Phyu et al. [6] introduced a federated learning scheme for mobile traffic forecasting in network slices with BiLSTM models. They proved that federated learning is feasible to preserve data privacy and reduce dependence on a centralized data source. Similarly, Rakkiannan et al. [4] merged federated learning and SDN and NFV to facilitate automatic network slicing in the edge node.

To support 6G mobile networks, Ming et al. [5] proposed a new FL approach which combines deep reinforcement learning for prediction-based network slice mobility management. They have improved resource use and mobility prediction efficiency based on their approach.

Although these developments have been made, the majority of current FL-based traffic prediction systems still use the traditional FedAvg aggregation method, assuming that all client updates are equally important and do not consider uncertainty in client predictions.

A key challenge in federated learning systems is security, where the malicious clients may include corrupted updates in the global model. Ma et al. [11] introduced Byzantine-robust asynchronous federated learning for cellular traffic prediction and showed better robustness with Krum aggregation algorithm.

To achieve privacy protection, Pan et al. [12] put forward a deep reinforcement learning-based FL framework in vehicular networks, which is Byzantine robust. Their approach ensures better security in model poisoning attacks while keeping communication efficient.

In [13] the authors investigated how local differential privacy (LDP) can be used to achieve privacy-preserving federated learning in Internet of Vehicles (IoV) traffic prediction systems. They demonstrated in their paper how prediction accuracy and privacy guarantees are traded.

To provide a dynamic 5G network slice provisioning, Kholidy presented a Federated Network Slicing Orchestrator (FNSO) based on a combination of Hexagonal Fuzzy TOPSIS and Secure Federated Learning (SFL). The framework takes into account the security, cost and performance KPIs for slice deployment and enables to improve the slice acceptance ratio without exposing the private information of local service provisioning [14].

Ayepah-Mensah et al. proposed an adaptive Digital Twin Multi-Agent Federated Learning (DT-MAFL) in 5G supported network slicing for IoT networks. Our proposed method is based on Graph Attention Networks (GAT) for traffic demand prediction and federated reinforcement learning for privacy-preserving resource allocation, which not only enhances prediction accuracy but also minimizes communication overhead [15].

Bedda et al. proposed an asynchronous federated learning-based intelligent network slicing approach for 5G networks. They update the model parameters asynchronously to minimize the network overhead and latency, and preserve the privacy of users. The experimental results showed that it achieved a better prediction accuracy and reduced communication costs than traditional FedAvg and centralized learning approaches [16].

Brik and Ksentini suggested a federated learning-based approach to predict service-oriented Key Performance Indicators (KPIs) in 5G network slices. The framework ensures data remains within each slice and that updates to the models are consolidated in the central location, which maintains privacy while providing accurate predictions of the performance metrics of the service, like the response time of the Mobility Management Entity (MME) [17].

In this regard, Li et al. introduced a federated orchestration system for joint bandwidth and computational resource allocation in 5G-IoT networks. The proposed Federated-Orchestrator manages distributed resources without depending on exchanging local information among base stations. They optimise services asynchronously, which significantly decreases service response time, preserves scalability and privacy [18].

Wijethilaka and Liyanage suggested a Federated Learning based Security Orchestrator (FLeSO) to improve the security in Network Slicing environments. The framework allows for a coordinated security management within slices without compromising data privacy. Experimental evaluation on a real world slicing testbed showed better threat detection and security performance than traditional security mechanisms [19].

Moreira et al. proposed a federated learning-based security-native network slicing architecture. The framework combines the ML-Agents with the Security Agents to implement distributed learning-based DDoS and intrusion detection, leveraging on non-intrusive telemetry data. The high attack detection accuracy and low slice isolation and data privacy loss on the large-scale testbeds were confirmed by experimental results [20].

Additionally, Dangi et al. [10] gave a thorough survey of the security mechanisms offered for 5G network slicing environments using machine learning. The study highlighted the need of using secure and smart resource management techniques for future wireless networks.

Table.1. Comparison of Existing FL-Based Traffic Prediction Methods

Reference	Technique	Security Mechanism	Major Limitation
Dutta et al. [1]	Federated LSTM for prediction-based load distribution in 5G slicing	No security mechanism	Equal client weighting and weak performance under non-IID traffic
Rakkiannan et al. [4]	FL with SDN/NFV for automated edge network slicing	Basic FL privacy	Does not address malicious client attacks or uncertainty estimation
Ming et al. [5]	Federated Deep Reinforcement Learning for 6G slice mobility prediction	Partial secure mobility management	Focuses on mobility management rather than traffic prediction reliability
Phyu et al. [6]	Federated BiLSTM for mobile traffic forecasting	No Byzantine defense	FedAvg aggregation ignores varying prediction quality across clients
Ma et al. [11]	Byzantine-Robust Asynchronous Federated Learning	Krum Byzantine defense	Does not incorporate uncertainty-aware aggregation
Pan et al. [12]	Privacy-Preserving Byzantine-Robust FL with DRL	Byzantine defense + Differential Privacy	Designed mainly for vehicular networks instead of 5G slicing
Khudair et al. [13]	FL with Local Differential Privacy for IoV traffic prediction	Local Differential Privacy	Accuracy degradation due to excessive privacy noise
Kholidy et al. [14]	Secure Federated Network Slicing Orchestrator (FNSO) using HF-TOPSIS	Secure FL aggregation	Focuses on slice orchestration and resource provisioning rather than traffic prediction accuracy
Ayepah-Mensah et al. [15]	Digital Twin Multi-Agent Federated Learning (DT-MAFL) for slice demand forecasting	FL privacy preservation	Lacks Byzantine attack resilience and uncertainty-aware aggregation
Bedda et al. [16]	Asynchronous Federated Learning for network slice prediction	Privacy-preserving asynchronous FL	Does not address malicious clients or prediction uncertainty
Brik and Ksentini [17]	FL-based service-oriented KPI prediction for network slices	Data privacy through FL	Limited to KPI prediction and does not consider robust aggregation against attacks
Wijethilaka and Liyanage [19]	Federated Learning enabled Security Orchestrator (FLeSO)	Coordinated FL-based security management	Focuses on security orchestration rather than traffic forecasting performance
Moreira et al. [20]	Federated Learning-based intelligent security architecture for network slicing	FL-based intrusion and DDoS detection	Primarily designed for attack detection, not traffic prediction or uncertainty modeling
Proposed Secure UA-FL Framework	Bayesian LSTM with Uncertainty-Aware Secure FL	Krum Byzantine defense + Secure aggregation	Additional computation overhead due to MC Dropout inference

The Table.1 presents a comparative analysis of existing federated learning-based traffic prediction methods for 5G network slicing environments. The comparison highlights that most existing approaches lack uncertainty estimation and comprehensive secure aggregation, whereas the proposed Secure UA-FL framework integrates both uncertainty-aware aggregation and Byzantine-robust security mechanisms for improved prediction accuracy and reliability.

Although quite a few aspects of federated learning for 5G traffic prediction and network slicing have been resolved, there are still key issues that are not addressed. Past FedAvg based solutions do not address the uncertainty in the client prediction or do not consider the reliability differences of models for non-IID traffic distributions. Besides, most of the frameworks do not have a wide-ranging Byzantine-robust secure aggregation and are seldom tested with different slice types including eMBB, URLLC, and mMTC. To solve these problems, this paper introduces an Secure UA-FL framework, which combines the Bayesian LSTMs and secure aggregation based on the Krum protocol for accurate, robust and secure traffic prediction.

3. PROPOSED METHODOLOGY

The Secure Uncertainty-Aware Federated Learning (Secure UA-FL) methodology helps out with secure, scalable, and privacy-preserving traffic prediction in 5G network slicing systems. It uses Bayesian LSTMs and Monte Carlo Dropout for estimating uncertainty, along with a process that considers this uncertainty during aggregation. Also, it employs a Byzantine-robust secure aggregation mechanism.

This setup works through distributed edge clients and a central federated server. In this secure workflow, no edge client shares raw traffic data; they just help train a global prediction model together.

3.1 SYSTEM ARCHITECTURE

The proposed Secure UA-FL architecture aims to securely and scalably predict traffic in dynamic 5G network slicing environments while maintaining privacy. To achieve high prediction accuracy under heterogeneous traffic conditions, the architecture integrates distributed edge learning, Bayesian Long Short-Term Memory (LSTM) networks, and a Byzantine-robust, uncertainty-aware aggregation mechanism.

It is based on several edge client devices (5G base station, edge gateway, etc.) securely communicating with a centralized federated server. All clients are constantly collecting local traffic

data on different network slices, such as eMBB, URLLC and mMTC. The traffic data generated by these slices is inherently heterogeneous and non-IID, since they have very different QoS requirements.

For each client, a local Bayesian-LSTM model is trained from the client's own traffic logs to process this data. The Bayesian LSTM architecture is selected due to its capacity to capture temporal relationships among the sequential traffic patterns and uncertainty in the system's predictions. Our decentralised design means that raw data of traffic never leaves the edge device, unlike traditional centralised designs. Rather, only the learned model parameters and uncertainty statistics are passed around between client and servers in the rounds of federated communication. This greatly improves data privacy and reduces the amount of overhead in communication in large-scale 5G-IoT networks.

Monte Carlo (MC) Dropout is then turned on during inference after the local training phase. This enables the model to estimate predictive uncertainty by making several stochastic forward passes through the model. Of course, there will be clients with lower uncertainty values that will be more confident in their predictions, and therefore more reliable contributors.

Once the updates to these models and uncertainty measures are uploaded, the centralized federated server manages the collaborative learning process. To make the system secure, the server applies the Krum algorithm that computes similarity of each pair of messages to determine their similarity and eliminate Byzantine attacks as a result of the similarity. Finally, the server performs an uncertainty-aware weighted aggregation with higher influence being given to the clients with lower uncertainty.

Finally, the new global Bayesian LSTM model is sent back to all the clients around the world for the next communication round. The iterative federated learning process keeps running until convergence, which allows for traffic prediction for intelligent 5G network slicing management to be accurate, secure, and communication efficient.

The complete FL architecture is shown in Fig.1 and it includes the Uncertainty-Aware Aggregation (UAA) and the Secure Aggregation (Krum).

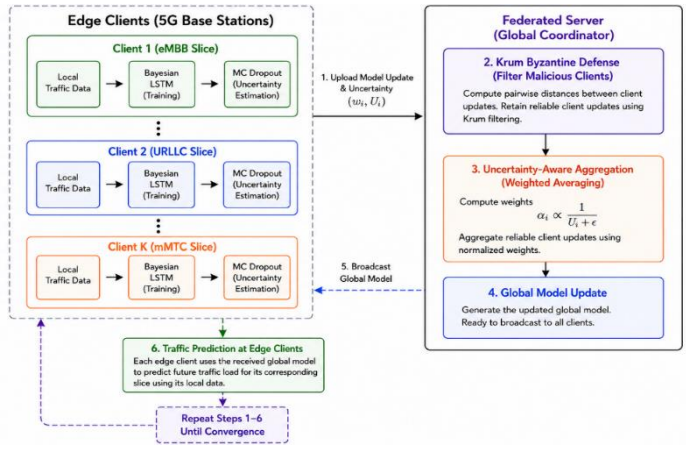

Fig.1. Proposed Secure Uncertainty-Aware Federated Learning Architecture

3.2 BAYESIAN LSTM WITH MONTE CARLO DROPOUT

The proposed framework employs the Bayesian Long Short-Term Memory (LSTM) networks at distributed edge clients to learn time-series dependency of network traffic data. The LSTM model can capture the long-range traffic characteristics of the dynamic 5G network, making it an ideal choice for time-series forecasting.

The network architecture of the Bayesian LSTM is:

- Two stacked LSTM layers with 128 hidden units each
- Dropout layers with dropout probability of 0.3
- Fully connected dense output layer
- Monte Carlo Dropout enabled during inference

The model takes as input historical traffic measurements and is used to forecast traffic loads for various network slices. Prediction uncertainty is estimated at the inference time by using Monte Carlo (MC) Dropout, where MC-Dropout refers to multiple stochastic forward passes of the Bayesian LSTM model. The predictive mean is calculated as:

$$\mu(x) = \frac{1}{M}\sum_{m=1}^{M} f(x; w_m) \tag{1}$$

where, M denotes the number of Monte Carlo forward passes and $f(x; w_m)$ represents the model prediction during m^{th} stochastic forward pass.

The predictive uncertainty variance is estimated as:

$$\sigma^2(x) = \frac{1}{M}\sum_{m=1}^{M}\left(f(x; w_m) - \mu(x)\right)^2 \tag{2}$$

where, $\mu(x)$ represents the predictive mean and $\sigma^2(x)$ represents predictive uncertainty.

In the proposed framework, M=20 Monte Carlo forward passes are performed during inference. For client i, the predictive uncertainty is denoted by:

$$U_i = \sigma_i^2(x) \tag{3}$$

Clients with higher uncertainty values are considered less reliable during federated aggregation.

The uncertainty thresholds are defined as follows:

- Low uncertainty: U_i<0.01
- Moderate uncertainty: 0.01≤U_i<0.05
- High uncertainty: U_i≥0.05

The mechanism of uncertainty estimation makes it possible to adapt the weights for client updates, thus enhancing the robustness of learning in the presence of non-IID client traffic.

The architecture of the Bayesian LSTM with Monte Carlo Dropout applied in each edge client is shown in Fig.2. Stacked LSTM layers are fed by historical traffic sequences for prediction. The uncertainty estimation is done by Monte Carlo Dropout and then passed to the uncertainty-aware federated aggregation.

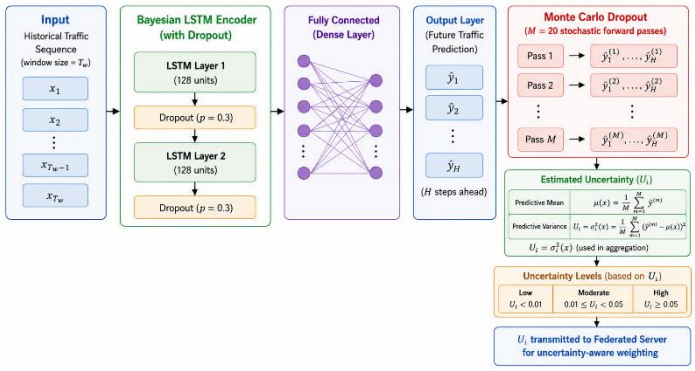

Fig.2. Bayesian LSTM with Monte Carlo Dropout Architecture

3.3 UNCERTAINTY-AWARE AGGREGATION

All the clients training in the classic Federated Averaging (FedAvg) are given the same weight, no matter how correct their predictions are or how trustworthy their local data has been. But in practice, slices and geographical regions have very different traffic distributions leading to non-IID client data sets. The proposed framework overcomes this limitation by adopting an uncertainty-aware aggregation mechanism that dynamically adjusts its aggregation weights based on the prediction confidence. In the global model updating, lower uncertain clients are given bigger weights in the aggregation. The aggregation weight for client *i* is computed as:

$$\alpha_i = \frac{1}{U_i + \grave{o}} \tag{4}$$

where, U_i denotes prediction uncertainty of client *i* and ϵ is a small positive constant used for numerical stability

The weights are normalized as:

$$\tilde{\alpha}_i = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j} \tag{5}$$

where *k* denotes the number of participating clients.

The global model is then updated using weighted aggregation:

$$w^{(t+1)} = \sum_{i=1}^{K} \tilde{\alpha}_i \, w_i^{(t+1)} \tag{6}$$

where, $w_i^{(t+1)}$ represents the local model parameters of client *i* and $w^{(t+1)}$ represents the updated global model. This is an uncertainty-based aggregation that stabilizes federated learning by mitigating the impacts of noisy / unreliable clients. As a result, the framework has more accurate and robust prediction in non-IID traffic distributions.

3.4 BYZANTINE-ROBUST AGGREGATION

Federated learning systems are vulnerable to Byzantine attacks, in which malicious clients intentionally upload corrupted model updates to affect the overall performance of the model. To improve security and robustness, the proposed design integrates the Krum Byzantine defense mechanism before uncertainty-aware aggregation.

The Krum algorithm computes pairwise Euclidean distances between client model updates. The distance between two client updates is defined as:

$$d_{ij} = \left\| w_i^{(t+1)} - w_j^{(t+1)} \right\|_2 \tag{7}$$

where d_{ij} denotes the Euclidean distance between client updates $w_i^{(t+1)}$ and $w_j^{(t+1)}$.

The Krum score for client *i* is defined as:

$$\text{Score}(i) = \sum_{j \in N_i} d_{ij} \tag{8}$$

where, N_i denotes the set of nearest neighbouring client updates and d_{ij} represents Euclidean distance between client updates

A client with a large Krum score will be considered a malicious or Byzantine client and will not be included in the aggregation process. The proposed framework employs a two-stage aggregation process:

3.4.1 Stage 1: Byzantine Filtering:

Krum Byzantine filtering removes malicious client updates.

$$R_t = \text{Krum}\left(\{w_i^{(t+1)}\}\right) \tag{9}$$

where R_t denotes the set of reliable clients retained after Byzantine filtering.

3.4.2 Stage 2: Uncertainty-Aware Aggregation:

Reliable client updates are aggregated using uncertainty-aware weights:

$$\alpha_i = \frac{\frac{1}{U_i + \grave{o}}}{\sum_{j \in R_t} \frac{1}{U_j + \grave{o}}} \tag{10}$$

The global model is updated as:

$$w^{(t+1)} = \sum_{i \in R_t} \alpha_i \, w_i^{(t+1)} \tag{11}$$

This secure aggregation method is integrated to make a system much more robust against model poisoning attacks without sacrificing much prediction accuracy.

3.5 TRAINING ALGORITHM

The overall federated training procedure of the proposed Secure UA-FL framework is summarized in Algorithm 1.

Algorithm 1. Secure Uncertainty-Aware Secure Federated Learning (Secure UA-FL) for 5G Traffic Prediction

Input: Global model parameters $w^{(0)}$, Number of clients *K* and communication rounds *T*.

Output: Optimized global traffic prediction model $w^{(T)}$

1. **Initialize** the global Bayesian LSTM model: $w^{(0)}$
2. **For each communication round** t=1,2,…,T:

   Server-side

   a. Select a subset of participating clients: $S_t \subseteq \{1,2,\ldots,K\}$

   b. Broadcast the current global model: $w^{(t)} \rightarrow S_t$

3. **For each selected client** $i \in S_t$:

a. Train the Bayesian LSTM using local traffic dataset D_i

$$w_i^{(t+1)} = \text{LocalTrain}\left(w^{(t)}, D_i\right)$$

b. Perform Monte Carlo Dropout inference using M stochastic forward passes:

$$\hat{y}_i^{(m)} = f(x_i; w_i^{(t+1)}, \text{Dropout}), \quad m = 1, \ldots, M$$

c. Compute predictive mean

$$\mu_i = \frac{1}{M}\sum_{m=1}^{M} \hat{y}_i^{(m)}$$

d. Compute predictive uncertainty

$$U_i = \frac{1}{M}\sum_{m=1}^{M}\left(\hat{y}_i^{(m)} - \mu_i\right)^2$$

e. Upload local model update and uncertainty: $\{w_i^{(t+1)}, U_i\}$.

4. **Federated Server Processing:**

a. Compute pairwise Euclidean distances between client updates

$$d_{ij} = \left\| w_i^{(t+1)} - w_j^{(t+1)} \right\|_2$$

b. Calculate Krum score for each client

$$\text{Score}(i) = \sum_{j \in N_i} d_{ij}$$

where N_i denotes the set of nearest neighboring updates.

c. Apply Krum Byzantine filtering and retain reliable clients

$$R_t = \text{Krum}\left(\{w_i^{(t+1)}\}\right)$$

d. Assign uncertainty-aware aggregation weights

$$\alpha_i = \frac{\frac{1}{U_i + \epsilon}}{\sum_{j \in R_t} \frac{1}{U_j + \epsilon}}$$

where ϵ is a small constant for numerical stability.

$$w^{(t+1)} = \sum_{i \in R_t} \alpha_i \, w_i^{(t+1)}$$

e. Aggregate reliable client models

5. **Update** global model: $w^{(t)} \leftarrow w^{(t+1)}$
6. **Broadcast** updated global model to all participating clients.
7. **Repeat** Steps 2–6 until convergence or $t=T$.

Return $w^{(T)}$

The optimized uncertainty-aware secure federated learning model for 5G traffic prediction.

4. EXPERIMENTAL SETUP

This section describes the dataset generation process, federated learning configuration setup, model parameters, Byzantine attack simulation scenarios and evaluation metrics used to validate both the UA-FL and Secure UA-FL frameworks for traffic prediction in 5G network slicing environments.

4.1 DATASET GENERATION

Due to the limited availability of publicly accessible 5G operator traffic datasets, synthetic traffic traces were generated to emulate realistic network slicing environments. To closely mimic practical 5G-IoT environments with diverse traffic, the dataset was generated following realistic patterns documented in existing literature [2, 3, 7] and 3GPP standards.

Three major network slicing categories were considered in this study:

- Enhanced Mobile Broadband (eMBB)
- Ultra-Reliable Low-Latency Communication (URLLC)
- Massive Machine-Type Communication (mMTC)

The eMBB slice generated high-bandwidth burst traffic representing applications such as video streaming and augmented reality services. The URLLC slice produced stable low-latency traffic patterns suitable for mission-critical applications including industrial automation and autonomous systems. The mMTC slice simulated periodic low-rate traffic generated by large-scale IoT devices and sensor networks. Traffic sequences were generated using temporal variations, Gaussian noise injection, and slice-specific utilization trends to imitate real-world network fluctuations. Historical traffic measurements were converted into sequential input windows for time-series forecasting. Each traffic sample contained multiple timesteps representing previous network load conditions used for predicting future traffic demand.

Table.2. Generated Dataset Statistics

Parameter	Value
Geographic Cells	50
Observation Period	30 Days
Time Resolution	10 Minutes
Timesteps per Cell	4,320
Network Slice Types	eMBB, URLLC, mMTC
Total Traffic Records	216,000
Total Time-Series Samples	~2.6 Million
Data Distribution	Non-IID across clients

The Table.2 summarizes the characteristics of the generated dataset used for training and evaluation. The large-scale multi-slice dataset captures diverse traffic behaviors across geographically distributed cells and provides sufficient variability for evaluating federated learning under realistic non-IID conditions.

Table.3. Traffic Characteristics of Different 5G Network Slices

Slice Type	Application Examples	Data Rate	Traffic Pattern	Peak Hours	Variance (CV)
eMBB	Video Streaming, AR/VR	50–150 Mbps	Bursty	18:00–23:00	High (CV = 0.70)
URLLC	Autonomous Systems, Industry 4.0	10–30 Mbps	Stable	08:00–17:00	Low (CV = 0.15)

mMTC	IoT Sensors, Smart Devices	1–5 Mbps	Periodic	Every 6 Hours	Moderate (CV = 0.35)

The Table.3 summarizes the traffic characteristics used to generate realistic network slicing scenarios for eMBB, URLLC, and mMTC services. The generated traffic samples were partitioned across 20 federated clients using a non-IID distribution strategy, where each client received different proportions of eMBB, URLLC and mMTC traffic. This setup emulates realistic heterogeneous edge environments commonly encountered in practical 5G deployments.

4.2 EXPERIMENTAL CONFIGURATION

The federated learning environment comprised 20 distributed edge clients (located at a geographically separated 5G base stations). The simulated traffic data was split into clients in an non-IID manner, to mimic realistic heterogeneous traffic conditions. In each communication round, 30% of the clients participated in local model training and sharing of update. The collaborative learning process was conducted over 50 rounds of global models to achieve stable convergence of global models. Centralized federated server coordinated the selection, secure aggregation, updating of global model and distribution of the updated model to participating clients. We used a Bayesian Long Short-Term Memory (LSTM) network with two stacked LSTM layers, each having 128 hidden units, as the proposed traffic prediction model. To support Monte Carlo Dropout based uncertainty estimation during inference, dropout layers with a dropout probability of 0.3 were added to the model. The model was implemented using Python and TensorFlow. It was decided to use the Adam optimizer due to its rapid convergence and stability in sequential learning tasks. The learning rate was set to 0.001, and the number of samples in a batch to 32.

Table.4. Model Hyperparameter Configuration

Parameter	Value
Operating System	Windows 11 Home
CPU	Intel Core i5-13420H
GPU	NVIDIA RTX 2050
RAM	16 GB DDR5
Total Clients	20
Client Participation Rate	30%
Data Distribution	Non-IID
Byzantine Defense	Krum
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Hidden Units	128
Dropout Rate	0.3
Monte Carlo Passes	20
Communication Rounds	50

The Table.4 summarizes the hardware platform, federated learning settings, and model hyperparameters used throughout the experimental evaluation.

4.3 BYZANTINE ATTACK SIMULATION

Three typical Byzantine attack scenarios were simulated to measure the security strength of the proposed framework in federated learning.

- Random Noise Attack: Malicious clients uploaded random noises on top of the model updates to break the global model convergence.
- Label-Flipping Attack: The training labels were modified, intentionally, before training the local model, leading to misleading model updates.
- Adaptive Byzantine Attack: Malicious clients send carefully designed updates that resemble the clients' behavior and try to harm the global model performance.

These attack models are the different degrees of sophistication of such an attack and they give a full assessment of the defense mechanism Krum.

4.4 EVALUATION METRICS

The performance of the proposed UA-FL framework was measured by means of prediction accuracy, communication efficiency, and security related metrics. The performance of the prediction was evaluated in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Better forecasting accuracies are reflected by lower RMSE and MAE values. RMSE is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \quad \text{(xx)}$$

In the same way, MAE is calculated as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \quad \text{(xx)}$$

Byzantine client identification effectiveness was assessed by the use of Detection Rate (DR) and False Positive Rate (FPR) to measure security performance. Detection Rate is the percentage of clients identified as malicious by the defense mechanism, while False Positive Rate is the percentage of clients mistakenly identified as malicious. To assess the efficiency of the communication between clients and the federated server, the total information exchanged between them during the training procedure was calculated. Furthermore, convergence behavior was examined in comparison to communication rounds to assess training stability and optimization efficiency.

5. RESULTS AND DISCUSSION

This section presents the experimental evaluation and analysis of the proposed UA-FL and Secure UA-FL frameworks for traffic prediction in 5G network slicing environments. The framework is evaluated in terms of prediction accuracy, convergence behavior, security robustness, and communication efficiency.

5.1 PERFORMANCE COMPARISON

The performance of the proposed UA-FL and Secure UA-FL frameworks was evaluated against conventional centralized learning and standard Federated Averaging (FedAvg) approaches for traffic prediction in 5G network slicing environments. The

comparison was conducted using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics across heterogeneous eMBB, URLLC, and mMTC traffic datasets.

Table.5. Overall Performance Comparison

Method	RMSE	MAE	Training Type	Security
LSTM	0.0506	0.0273	Centralized	None
GRU	0.0540	0.0275	Centralized	None
CNN-LSTM	0.0512	0.0250	Centralized	None
FedAvg	0.0493	0.0253	Federated	None
UA-FL	**0.0492**	**0.0249**	Federated	None
Secure UA-FL	**0.0493**	**0.0247**	Federated	**Krum**

The Table.5 clearly indicates that federated learning can achieve prediction accuracy comparable to centralized learning approaches while preserving data privacy. The proposed Uncertainty-Aware Aggregation method improves the quality of model updates by assigning adaptive weights to client contributions. Furthermore, the Secure UA-FL version maintains similar prediction accuracy while incorporating Byzantine defense.

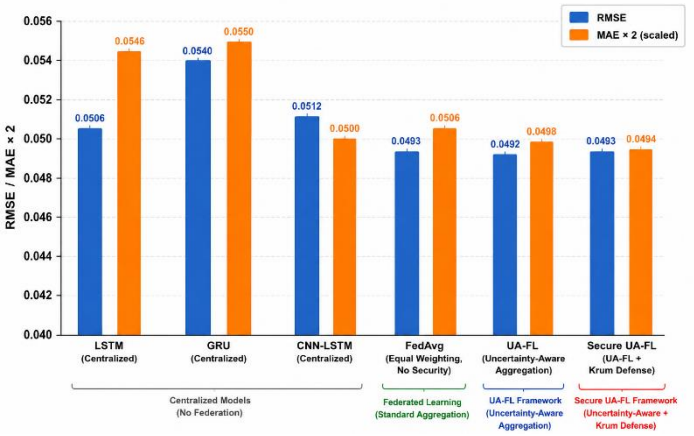

Fig.3. Performance Comparison Across All Methods

The Fig.3 illustrates the comparative performance of all evaluated methods in terms of RMSE and MAE. The Fig.demonstrates that the proposed UA-FL framework achieves consistently competitive performance across all evaluated metrics.

5.2 CONVERGENCE ANALYSIS

The convergence behavior of the UA-FL framework was analyzed over multiple communication rounds to evaluate training stability and optimization efficiency. Experimental observations showed that the proposed UA-FL framework converged faster and exhibited more stable learning behavior compared to conventional FedAvg-based training. The uncertainty-aware aggregation mechanism significantly reduced oscillations during global model updates by assigning lower aggregation weights to highly uncertain client models. In addition, the Krum-based Byzantine filtering stage prevented malicious updates from negatively affecting convergence performance. As the communication rounds increased, the global model gradually achieved lower RMSE values and stable convergence, demonstrating the effectiveness of combining uncertainty estimation with secure federated aggregation.

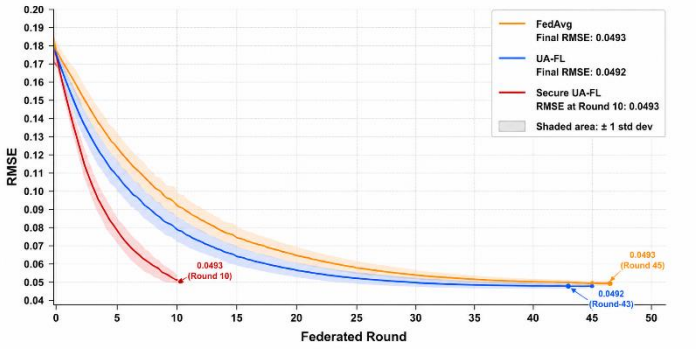

Fig.4. Convergence Comparison - FedAvg vs. UA-FL vs. Secure UA-FL

The Fig.4 demonstrates that the uncertainty-aware framework converges faster than conventional FedAvg. Secure UA-FL achieved a stable RMSE of 0.0493 within 10 communication rounds, after which additional training provided negligible improvement due to Krum-based filtering of noisy updates. Consequently, training was terminated early to reduce communication overhead. The results indicate that uncertainty-aware aggregation is particularly beneficial during early training stages when client update quality varies significantly. Although the RMSE improvement is modest, the proposed framework simultaneously provides uncertainty-aware aggregation, Byzantine robustness, and communication efficiency without compromising prediction accuracy.

5.3 SECURITY ANALYSIS

To evaluate the robustness of our framework against malicious participants, three Byzantine attacks have been simulated: Random Noise, Label-Flipping, and Adaptive Byzantine attacks. Byzantine clients intentionally send corrupted model updates to disrupt federated learning performance. The detection performance was assessed by Detection Rate (DR), False Positive Rate (FPR) and the prediction RMSE.

Table.6. Byzantine Attack Detection Performance

Attack Type	Detection Rate (%)	FPR (%)	RMSE
Random Noise	100.00	0.00	0.1501
Label-Flipping	94.12	2.08	0.1520
Adaptive Byzantine	88.24	3.12	0.1540

The Table.6 shows the effectiveness of the Krum defence mechanism with various attack strategies. Random Noise attacks were tested with 0 false positives and 100% detection. The framework achieved high detection rates >88% for both Label-Flipping and Adaptive Byzantine attacks. Although prediction error increased slightly under stronger attacks, the framework maintained low false positive rates (<4%).

5.4 DIFFERENTIAL PRIVACY EXPLORATORY ANALYSIS

The influence of differential privacy noise on the results of traffic prediction was explored in an exploratory study.

Table.7. Differential Privacy Analysis

Privacy Budget (ε)	Utility Observation
0.5	High privacy, reduced convergence
1.0	Strong privacy with moderate instability
2.0	Stable training begins
5.0	Near-baseline performance
∞	Best predictive accuracy

Trade-off between privacy and utility is noted in Table 7. When privacy budgets are strong, a lot of noise comes into the sequential traffic pattern, which results in unstable convergence and poorer predicted accuracy. The moderate level of privacy ($\varepsilon \geq 5$) yields near baseline performance, suggesting that there is a practical privacy level that balances privacy preservation with forecasting performance.

5.5 COMMUNICATION EFFICIENCY

An important factor to consider in the implementation of large-scale federated learning is communication overhead. Federated learning exchanges model parameters, not raw traffic data, significantly decreasing the amount of communication.

Table.8. Communication Overhead Comparison

Approach	Data per Client	Total per Round	Total Training	Reduction
Centralized	125 MB raw data	N/A (single location)	2.5 GB	Baseline
FedAvg	4 MB (model up/down)	24 MB	1.2 GB	52%
UA-FL	4 MB + 1 KB (uncertainty)	24 MB	1.2 GB	52%
Secure UA-FL	4 MB + 1 KB	24 MB (10 rounds)	240 MB	**90%**

As shown in Table.8, federated learning achieves much less communication overhead than centralized training. The uncertainty values sent across UA-FL incur hardly any extra cost in terms of communication. The secure variant is also lightweight, further reducing the need for communication as a result of the faster convergence.

5.6 DISCUSSION

The results show that the proposed Secure UA-FL framework is indeed effective in achieving a tradeoff between prediction accuracy, robustness and security for 5G network slice traffic predictions. In contrast to traditional federated traffic prediction methods using FedAvg aggregation [1] and [6], the proposed method incorporates uncertainty-aware aggregation to mitigate the impact of unreliable client updates. Moreover, the proposed Byzantine-robust federated learning schemes do not account for prediction uncertainties during aggregation, such as those proposed in Ma et al. [11] and Pan et al. [12].

The proposed Secure UA-FL framework can solve both problems: Bayesian uncertainty estimation and Krum-based filtering. Previous studies could not be directly compared as the experiments were performed on a synthetically generated 5G traffic dataset that has different traffic characteristics and evaluation settings. The findings show, however, that incorporating uncertainty awareness and secure federated aggregation can enhance the reliability and resilience of distributed 5G network slicing environments.

6. CONCLUSION

This paper introduced a novel Secure Uncertainty-Aware Federated Learning (Secure UA-FL) framework designed for traffic prediction in 5G network slicing applications. This framework incorporates Bayesian LSTM-based uncertainty quantification along with uncertainty-aware aggregation and Krum Byzantine attack detection methods. Results revealed good prediction accuracy in terms of RMSE = 0.0493 and MAE = 0.0247, successful detection of malicious behavior and reduction of communication load, when compared to centralized architectures. Additionally, the proposed framework showed resilience towards heterogeneous, non-IID distribution in terms of eMBB, URLLC, and mMTC network slices.

REFERENCES

[1] N. Dutta, S.P. Patole, R. Mahadeva and G. Ghinea, "Federated Learning Framework for Prediction based Load Distribution in 5G Network Slicing", *Proceedings of International Conference on Contemporary Computing*, Vol. 12, pp. 421-426, 2024.

[2] D. Kulkarni, M. Venkatesan and A.V. Kulkarni, "Traffic Prediction with Network Slicing in 5G: A Survey", *Proceedings of International Conference on Artificial Intelligence and Smart Energy*, Vol. 7, pp. 1360-1365, 2023.

[3] R. Mihai, L.M. Tufeanu, L. Fricosu, A. Vulpe, R. Craciunescu and M. Vochin, "Adaptive 5G Network Slicing with Real-Time Traffic Classification and Experimental Validation", *IEEE Access*, Vol. 13, pp. 216906-216915, 2025.

[4] T. Rakkiannan, G. Ekambaram, N. Palanisamy, R.R. Ramasamy, S. Muthusamy, A.K. Loganathan, H. Panchal, K. Thangaraj and A. Ravindaran, "An Automated Network Slicing at Edge with Software Defined Networking and Network Function Virtualization: A Federated Learning Approach", *Wireless Personal Communications*, Vol. 131, pp. 639-658, 2023.

[5] Z. Ming, H. Yu and T. Taleb, "Federated Deep Reinforcement Learning for Prediction-based Network Slice Mobility in 6G Mobile Networks", *IEEE Transactions on Mobile Computing*, Vol. 23, No. 12, pp. 11937-11953, 2024.

[6] H.P. Phyu, D. Naboulsi and R. Stanica, "Mobile Traffic Forecasting for Network Slices: A Federated-Learning Approach", *Proceedings of International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 6, pp. 745-751, 2022.

[7] X. Foukas, A. Elmokashfi, G. Patounas and M.K. Marina, "Network Slicing in 5G: Survey and Challenges", *IEEE Communications Magazine*, Vol. 55, No. 5, pp. 94-100, 2017.

[8] W. Rafique, J.R. Barai, A.O. Fapojuwo and D. Krishnamurthy, "A Survey on Beyond 5G Network Slicing

for Smart Cities Applications", *IEEE Communications Surveys and Tutorials*, Vol. 27, No. 1, pp. 595-628, 2025.

[9] M. Andalibi, M.R. Hashemi and Z. Zali, "Enhancing Quality of Service in Internet of Things Networks using 5G-Inspired Resource Management Approaches", *Proceedings of International Conference on Internet of Things and Applications*, Vol. 34, pp. 1-11, 2025.

[10] R. Dangi, A. Jadhav, G. Choudhary, N. Dragoni, M.K. Mishra and P. Lalwani, "ML-based 5G Network Slicing Security: A Comprehensive Survey", *Future Internet*, Vol. 14, No. 4, pp. 1-7, 2022.

[11] H. Ma, K. Yang and Y. Jiao, "Cellular Traffic Prediction via Byzantine-Robust Asynchronous Federated Learning", *IEEE Transactions on Network Science and Engineering*, Vol. 12, No. 4, pp. 2402-2414, 2025.

[12] Y. Pan, Z. Su, Y. Wang, J. Zhou and M. Mahmoud, "Privacy-Preserving Byzantine-Robust Federated Learning via Deep Reinforcement Learning in Vehicular Networks", *IEEE Transactions on Vehicular Technology*, Vol. 74, No. 6, pp. 9461-9475, 2025.

[13] M.J. Khudair, F.S. Mubarek and S.A. Aliesawi, "Privacy-Preserving Federated Learning with Local Differential Privacy for Traffic Prediction in IoV Systems", *International Journal of Applied Mathematics*, Vol. 38, No. 6, pp. 1170-1188, 2025.

[14] H.A. Kholidy, "Dynamic Network Slicing Orchestration in Open 5G Networks using Multi-Criteria Decision Making and Secure Federated Learning Techniques", *Cluster Computing*, Vol. 28, No. 237, pp. 1-14, 2025.

[15] D. Ayepah-Mensah, G. Sun, Y. Pang and W. Jiang, "Adaptive Digital Twin and Communication-Efficient Federated Learning Network Slicing for 5G-Enabled Internet of Things", *Proceedings of International Congress on Teletraffic*, Vol. 12, pp. 124-131, 2024.

[16] K. Bedda, Z.M. Fadlullah and M.M. Fouda, "Efficient Wireless Network Slicing in 5G Networks: An Asynchronous Federated Learning Approach", *Proceedings of International Conference on Internet of Things and Intelligence Systems*, Vol. 22, pp. 285-289, 2022.

[17] B. Brik and A. Ksentini, "On Predicting Service-Oriented Network Slices Performances in 5G: A Federated Learning Approach", *Proceedings of International Conference on Local Computer Networks*, Vol. 7, pp. 164-171, 2020.

[18] Y. Li, A. Huang, Y. Xiao, X. Ge, S. Sun and H.C. Chao, "Federated Orchestration for Network Slicing of Bandwidth and Computational Resource", *IEEE Access*, Vol. 2, pp. 1-30, 2020.

[19] M. Abouaomar, H.A. Kholidy and A. Erradi, "A Federated Learning Approach for Improving Security in Network Slicing", *Proceedings of International Conference on Global Communications*, Vol. 21, pp. 1-7, 2022.

[20] R. Moreira, R.S. Villaca, M.R.N. Ribeiro, J.S.B. Martins, J.H. Correa, T.C. Carvalho and F.O. Silva, "An Intelligent Native Network Slicing Security Architecture Empowered by Federated Learning", *Future Generation Computer Systems*, Vol. 164, pp. 71-89, 2025.