© Volume 2, Issue 11, November 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

Trust-Aware Byzantine-Resilient Federated Learning with Adaptive Differential Privacy for Secure Big Data Analytics

Dr. A Mohamed Azharudheen[1], Mrs.D. Lalitha[2], Mr. A. Akilan[3]
[1]*Assistant Professor and HEAD, Department of Computer Science and IT, Srinivasan College of Arts & Science, Perambalur-621 212, Tamil Nadu, India*
[2,3]*Assistant Professor, Department of Computer Science and IT, Srinivasan College of Arts & Science, Perambalur-621 212, Tamil Nadu, India*
https://doi.org/10.64643/jatir.141309

***Abstract*—The rapid growth of distributed big data environments across healthcare, financial services, Internet of Things (IoT), and cloud computing has increased the demand for secure and privacy-preserving machine learning mechanisms. Federated Learning (FL) enables collaborative model training across distributed data sources without transferring raw data to a centralized server. However, conventional federated learning approaches remain vulnerable to privacy leakage through model updates, membership inference attacks, gradient inversion, and malicious client participation. Furthermore, existing privacy-preserving mechanisms often introduce computational overhead and reduce analytical utility, while conventional aggregation strategies may fail to mitigate Byzantine and model poisoning attacks. To address these limitations, this research proposes a Trust-Aware Byzantine-Resilient Federated Learning with Adaptive Differential Privacy (TABR-FL-DP) framework for secure big data analytics. The proposed architecture integrates adaptive deep feature extraction, client-level differential privacy, trust-based client evaluation, Byzantine-resilient model aggregation, and fault-tolerant federated optimization. Initially, distributed clients perform local data preprocessing and feature extraction without exposing sensitive information. Gradient clipping and calibrated Gaussian noise are subsequently applied to protect individual client contributions under a formally accounted differential privacy mechanism. A trust evaluation engine assesses client reliability based on update consistency, historical behavior, and abnormal model-update characteristics. The proposed robust aggregation mechanism combines trust-weighted contributions with Byzantine-resilient outlier mitigation to reduce the influence of malicious participants. Additionally, an adaptive privacy-budget optimization mechanism balances privacy protection, analytical utility, communication overhead, and model convergence. The proposed framework will be evaluated using benchmark cybersecurity datasets, including CIC-IDS2017 and UNSW-NB15, against conventional centralized learning, FedAvg, and established privacy-**

preserving and Byzantine-resilient federated learning approaches. Performance will be assessed through classification accuracy, precision, recall, F1-score, differential privacy guarantees, attack success rate, communication overhead, latency, and fault tolerance. The research aims to establish a reliable and scalable privacy-preserving federated learning framework for secure and high-availability big data analytics.

***Index Terms*—Federated Learning, Differential Privacy, Byzantine-Resilient Aggregation; Trust-Aware Learning; Big Data Security; Model Poisoning Detection; Privacy-Preserving Machine Learning; Adaptive Privacy Budget; Deep Learning; Intrusion Detection.**

I. INTRODUCTION

The continuous development of big data technologies has transformed the way organizations collect, process, store, and analyze large-scale information. Applications in healthcare, banking, financial transactions, smart cities, industrial automation, and Internet of Things environments generate enormous volumes of heterogeneous data. These applications require intelligent data analytics to support real-time decision-making, predictive modelling, anomaly detection, and automated security management. However, the increasing distribution of sensitive information across multiple organizations and computing environments introduces significant challenges related to confidentiality, integrity, privacy, and availability.

Traditional centralized machine learning approaches require data to be transferred to a central processing environment for model training. Although this approach simplifies data management and model development, it creates privacy risks associated with unauthorized access, centralized data breaches, and sensitive information exposure. Encryption, anonymization, and access-control mechanisms provide important security protections, but they may introduce computational overhead and limitations in maintaining analytical utility.

Federated Learning has emerged as an alternative distributed machine learning paradigm that enables multiple clients to collaboratively train a global model without directly sharing their raw datasets. Each participating client performs local model training and transmits model updates to a coordinating server. The server aggregates these updates to improve the global model. This decentralized learning approach reduces direct raw-data movement and supports collaborative analytics across organizations.

Despite these advantages, federated learning does not automatically guarantee privacy or security. Model updates may reveal sensitive information through gradient inversion and membership inference attacks. Moreover, malicious participants may deliberately submit manipulated gradients to compromise the global learning process. Such attacks, commonly associated with Byzantine behavior and model poisoning, can reduce model accuracy, disrupt convergence, or introduce targeted misclassification.

Differential Privacy provides a mathematical framework for limiting the information revealed about individual data contributions. By applying gradient clipping and calibrated random noise,

differential privacy can reduce the possibility of identifying individual training records or clients. However, excessive noise may reduce model accuracy, while insufficient noise may provide inadequate privacy protection. Therefore, adaptive privacy-budget management is necessary to balance privacy requirements and analytical performance.

Similarly, Byzantine-resilient aggregation techniques aim to identify and mitigate malicious model updates during distributed training. Methods such as coordinate-wise median, trimmed mean, and Krum provide robustness against certain adversarial behaviors. Nevertheless, these approaches may face limitations when client data distributions are heterogeneous, malicious clients behave adaptively, or privacy-preserving mechanisms obscure individual model updates.

To address these challenges, this research proposes a Trust-Aware Byzantine-Resilient Federated Learning with Adaptive Differential Privacy (TABR-FL-DP) framework. The proposed framework integrates client-level privacy protection, trust-aware client evaluation, robust model aggregation, and adaptive privacy–utility optimization within a unified distributed learning architecture.

The proposed approach extends the previously published Federated Hybrid Deep Privacy Architecture (FHDPA) by introducing formal differential privacy accounting, explicit Byzantine threat mitigation, and trust-aware federated aggregation. Rather than relying primarily on entropy-calibrated anonymization, the new framework establishes measurable privacy guarantees and evaluates security through adversarial experiments.

1.1 Research objectives

The main objectives are:

1. To develop a secure federated learning architecture for distributed big data environments.
2. To introduce an adaptive differential privacy mechanism with formal privacy accounting.
3. To design a trust evaluation mechanism for identifying unreliable and malicious federated clients.
4. To develop Byzantine-resilient aggregation for reducing the impact of model poisoning attacks.
5. To formulate a multi-objective optimization mechanism balancing privacy, utility, trust, and computational efficiency.
6. To evaluate the proposed framework using benchmark datasets and established federated learning techniques.
7. To improve the reliability and availability of distributed privacy-preserving analytics.

1.2 Research questions

- RQ1: How effectively can adaptive differential privacy protect federated model updates against inference attacks?
- RQ2: Can trust-aware client evaluation improve resistance to malicious federated participants?
- RQ3: How does Byzantine-resilient aggregation influence global model convergence and classification performance?

- RQ4: Can adaptive privacy-budget allocation maintain analytical utility while satisfying formal privacy constraints?
- RQ5: How effectively can the proposed framework maintain availability under client failures and communication disruptions?

II. RESEARCH GAP

Although federated learning, differential privacy, and robust aggregation have been studied independently, integrating these mechanisms into an adaptive and trust-aware architecture presents several research challenges.

Your earlier FHDPA paper already addressed federated feature abstraction, entropy-calibrated anonymization, utility-constrained reconstruction, distributed threat intelligence and high-availability data delivery. The new work should therefore concentrate on the following additional research gaps.

No.	Existing research limitation	Identified research gap	Proposed solution
1	Federated learning may expose information through model updates.	Formal client-level privacy protection is required.	Adaptive DP with privacy accounting.
2	Conventional FedAvg is vulnerable to malicious updates.	Reliable malicious-client mitigation is required.	Byzantine-resilient aggregation.
3	Existing robust aggregation may treat all clients similarly.	Client reliability needs to be evaluated.	Trust-aware client scoring.
4	Static privacy settings may reduce model utility.	Privacy protection needs controlled adaptation.	Adaptive privacy-budget allocation.
5	Privacy mechanisms can increase computation and communication costs.	Resource-aware federated optimization is required.	Communication-efficient local training.
6	Client disconnections affect distributed training.	Fault tolerance needs explicit evaluation.	Client-failure-aware aggregation.
7	Privacy and security evaluations are often conducted separately.	Integrated adversarial evaluation is needed.	Joint privacy, poisoning and availability testing.

Table 1: Research Gap between Existing model and proposed model

2.1 Specific research gap

The central research gap is the absence of a unified, experimentally validated mechanism in the proposed research design that jointly addresses:

- Formal client-level differential privacy.
- Trust-aware client reliability assessment.
- Byzantine-resistant model aggregation.

- Adaptive privacy–utility optimization.
- Fault-tolerant federated model training.

The novelty claim must be validated against the latest related publications before submission. It should not be presented as proof that no previous work has ever combined these techniques.

III. LITERATURE REVIEW

Federated learning, differential privacy, and Byzantine-resilient optimization form the theoretical foundation of the proposed TABR-FL-DP framework. This section reviews important contributions related to distributed learning, privacy protection, malicious client detection, and secure model aggregation.

3.1 Federated Learning for Distributed Data Analytics

McMahan et al. (2017) introduced Federated Averaging (FedAvg), a communication-efficient learning algorithm that enables multiple distributed clients to collaboratively train a global model without sharing their raw datasets. The authors demonstrated the effectiveness of iterative model averaging under distributed and unbalanced data conditions. However, FedAvg does not independently provide formal privacy protection or robust defence against malicious participants. Federated learning has subsequently become an important approach for collaborative machine learning in privacy-sensitive environments. Nevertheless, the protection of model updates and the reliability of participating clients remain important research challenges.

3.2 Differential Privacy in Deep Learning

Abadi et al. (2016) introduced techniques for training deep neural networks with differential privacy. Their approach combines gradient clipping and calibrated noise addition to limit the information revealed through model training. The Moments Accountant technique provides a method for tracking cumulative privacy loss. This research provides an important foundation for privacy-preserving neural network training.

Mironov (2017) introduced Rényi Differential Privacy (RDP), which provides a mathematical framework for analysing privacy loss under repeated and composed mechanisms. RDP is particularly useful for evaluating privacy guarantees across multiple federated communication rounds.

Azharudheen and Vijayalakshmi [9] presented a study entitled “Improvement of Data Analysis and Protection Using Novel Privacy-Preserving Methods for Big Data Application” in The Scientific Temper, Volume 15, Issue 2, pages 2181–2189 (2024). The study focuses on improving data analysis and strengthening data protection in Big Data environments through privacy-preserving methods. The research addresses the importance of maintaining data confidentiality while enabling the effective analysis and utilization of large-scale datasets.

The proposed TABR-FL-DP framework extends this direction by considering adaptive privacy-budget management within a trust-aware federated learning environment.

3.3 Byzantine-Resilient Federated Learning

Blanchard et al. (2017) proposed Krum, a Byzantine-resilient gradient aggregation algorithm designed to tolerate arbitrary and potentially malicious worker behaviour. The approach selects a gradient update based on its proximity to other submitted updates rather than simply averaging all contributions.

Yin et al. (2018) investigated Byzantine-robust distributed learning and introduced coordinate-wise median and trimmed-mean aggregation strategies. Their work provides theoretical analysis of distributed optimization under Byzantine failures.

These approaches demonstrate the importance of robust aggregation in distributed learning. However, the proposed research will investigate how trust information and privacy-protected updates can be incorporated into robust aggregation under heterogeneous client distributions.

3.4 Model Poisoning and Backdoor Attacks

Bagdasaryan et al. (2020) demonstrated that federated learning can be vulnerable to model replacement and backdoor attacks. Their research showed that malicious participants can manipulate model updates to introduce attacker-controlled behavior into the global model.

This establishes the need for security mechanisms capable of identifying suspicious updates and reducing malicious influence during aggregation.

3.5 Intrusion Detection Datasets for Federated Security

Sharafaldin et al. (2018) introduced CIC-IDS2017, a network intrusion detection dataset containing benign traffic and multiple categories of contemporary attacks. The dataset includes labelled network flows suitable for machine learning-based intrusion detection.

For the proposed research, CIC-IDS2017 will provide a suitable experimental foundation for evaluating distributed intrusion detection, client heterogeneity, and adversarial model updates.

3.6 Summary of the Literature

Author and year	Research contribution	Relevance	Remaining challenge
McMahan et al. (2017)	Federated Averaging	Distributed learning	Privacy and malicious clients
Abadi et al. (2016)	Differentially private deep learning	Gradient privacy	Utility degradation
Mironov (2017)	Rényi Differential Privacy	Privacy accounting	Adaptive federated deployment
Blanchard et al. (2017)	Krum aggregation	Byzantine resistance	Heterogeneous client behaviour
Yin et al. (2018)	Median and trimmed mean	Robust optimization	Integration with privacy and trust

Bagdasaryan et al. (2020)	Federated backdoor attacks	Threat modelling	Robust mitigation
Sharafaldin et al. (2018)	CIC-IDS2017 dataset	Experimental evaluation	Dataset partitioning for federated experiments

Table 2: Summary of Literature Review

Literature review observation: The reviewed publications establish the individual foundations of federated learning, differential privacy and Byzantine robustness. The proposed research investigates their integration with trust-aware client evaluation and adaptive privacy management.

IV. PROPOSED METHODOLOGY

4.1 Overview of the Proposed TABR-FL-DP Framework

The proposed framework follows a distributed collaborative learning architecture in which participating clients train local models while preserving the confidentiality of their original datasets.

The framework consists of seven interconnected modules:

1. Distributed Data Collection and Preprocessing.
2. Adaptive Deep Feature Extraction.
3. Client-Level Differential Privacy.
4. Trust Evaluation and Risk Assessment.
5. Byzantine-Resilient Model Aggregation.
6. Adaptive Privacy–Utility Optimization.
7. Fault-Tolerant Global Model Delivery.

The methodology follows an iterative training process in which privacy protection, client evaluation and robust aggregation are performed during each communication round.

4.2 System Model

Let the distributed environment consist of N participating clients:

$C = \{C1, C2, \ldots\ldots, C_N\}$

Each client maintains its own local dataset:

$Di = \{(x_{ij}, y_{ij})\}^{ni}{}_{j=1}$

Where:

- Ci represents the participating client.
- Di represents the local dataset.
- xij represents the input feature vector.
- yij represents the corresponding class label.
- ni represents the number of local samples.

The global learning objective is:

Where

$$\min_{w} F(w) = \sum_{i=1}^{N} p_i F_i(w)$$

$$p_i = \frac{n_i}{\sum_{j=1}^{N} n_j}$$

Here, Fi (w) represents the local training loss and F (w) represents the global objective.

4.3 Module 1: Distributed Data Collection and Preprocessing

The first module collects data from multiple distributed clients without transferring raw records to the central coordinating server.

The preprocessing operations include:

- Missing-value handling.
- Duplicate-record removal.
- Data normalization.
- Categorical feature encoding.
- Class-label transformation.
- Feature selection.
- Local training and testing partitioning.

For numerical feature normalization:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

The normalized features are used for subsequent local model training.

Expected contribution: Improve consistency across participating clients while maintaining local data ownership.

4.4 Module 2: Adaptive Deep Feature Extraction

The second module extracts relevant features from local network traffic records using a lightweight deep learning architecture.

The proposed feature extraction process can combine:

- Deep Belief Network for hierarchical feature learning.
- Convolutional Neural Network for local feature extraction.
- Autoencoder for dimensionality reduction.

The hybrid feature representation is:

$Zi = f\theta\ (Xi)$

Where:

- Xi is the preprocessed input.
- fθ is the adaptive feature encoder.
- Zi is the extracted latent representation.

The feature encoder will be evaluated against conventional feature extraction techniques to determine whether it improves classification performance and computational efficiency.
Unlike the previous FHDPA model, this module is not the principal novelty. The main new contributions are privacy accounting, trust assessment and robust aggregation.

4.5 Module 3: Adaptive Client-Level Differential Privacy
This module protects model updates before they are used in global aggregation.
Step 1: Local gradient computation

$$g_i^t = \nabla F_i(w_t)$$

Step 2: Gradient clipping

$$\bar{g}_i^t = g_i^t \min\left(1, \frac{C}{\|g_i^t\|_2}\right)$$

Where C is the clipping threshold.
Step 3: Gaussian noise addition

$$\tilde{g}_i^t = \bar{g}_i^t + \mathcal{N}(0, \sigma_t^2 C^2 I)$$

Where σ_t is the noise multiplier for communication round t.
Step 4: Privacy-budget management
The privacy accountant tracks cumulative privacy loss across the training rounds.
The final guarantee is expressed as:
(ε, δ) - Differential Privacy
For the Gaussian mechanism, privacy accounting must consider the sensitivity, sampling probability, noise multiplier and number of composed rounds.
The adaptive noise multiplier is proposed as:

$$\sigma_t = \sigma_{min} + (\sigma_{max} - \sigma_{min}) r_t$$

Where r_t is a normalized privacy-risk indicator.
A higher risk may trigger stronger noise, subject to the remaining privacy budget and utility constraints.
Important: This adaptive rule is a proposed control policy, not itself a proof of differential privacy. The complete sequence of mechanisms must be analyzed by a valid accountant, and the privacy parameters must be calibrated before deployment.

4.6 Module 4: Trust Evaluation and Client Risk Assessment

The trust evaluation engine identifies unreliable or suspicious client behavior.

Four major characteristics are considered:

1. Model update consistency.
2. Historical client reliability.
3. Update similarity.
4. Abnormal behavior and security risk.

The proposed trust score is:

$$T_i^t = w_1 A_i^t + w_2 C_i^t + w_3 H_i^t + w_4 (1 - R_i^t)$$

Subject to:

$$\sum_{k=1}^{4} w_k = 1$$

Where:

- A_i^t: Agreement score.
- C_i^t: Consistency score.
- H_i^t: Historical reliability.
- R_i^t: Normalized security-risk score.

All components are normalized between zero and one.

The agreement score can be calculated using cosine similarity:

$$A_i^t = \frac{\tilde{g}_i^t \cdot G_{ref}^t}{\|\tilde{g}_i^t\|_2 \|G_{ref}^t\|_2}$$

Here, G^t_{ref} is a robust reference direction calculated from eligible client updates.

Historical consistency can be evaluated using:

$$C_i^t = \exp(-\kappa d_i^t)$$

Where d^t_i represents the normalized deviation from the client's historical update behavior.

The system can classify clients into three groups:

Trust score	Client classification	Action
0.75–1.00	Trusted	Eligible for normal aggregation
0.40–0.74	Suspicious	Additional validation
Below 0.40	High-risk	Temporarily excluded and investigated

Table 2: Client Classification based on Trust Score

4.7 Module 5: Byzantine-Resilient Model Aggregation

This module reduces the impact of malicious or abnormal model updates.

The proposed system uses a two-stage process.

Stage 1: Robust outlier mitigation

Apply coordinate-wise trimmed mean to the received updates.

For each parameter coordinate k:

$$G_{trim,k} = TrimmedMean(\tilde{g}_{1,k}, \ldots, \tilde{g}_{N,k})$$

The method removes a predefined number of extreme values from both ends of each coordinate before calculating the mean.

Stage 2: Trust-weighted aggregation

The trust weights are:

$$q_i^t = \frac{T_i^t}{\sum_{j \in S_t} T_j^t}$$

Where St represents the eligible client set.

The proposed aggregation rule is:

$$G_t = \sum_{i \in S_t} q_i^t \tilde{g}_i^t$$

However, this simple weighted mean alone is not Byzantine-robust. The implementation must apply the selected outlier mitigation to the individual updates or use a formally justified robust estimator before combining trust weights.

The global model is updated as:

$$W_{t+1} = W_t - \eta G_t$$

Where η represents the learning rate.

Security consideration: The aggregation design must define its adversary assumptions, including the maximum malicious-client fraction, collusion and non-IID data. Trust scores based only on update similarity may unfairly penalize legitimate clients with different data distributions. Therefore, trust assessment should also use historical evidence and independent validation where available.

4.8 Module 6: Adaptive Privacy–Utility–Trust Optimization

The sixth module manages the trade-off between privacy protection, model accuracy, trustworthiness and system performance.

The proposed multi-objective optimization function is:

Max $J = \lambda_1 P + \lambda_2 U + \lambda_3 T - \lambda_4 D - \lambda_5 L$

Subject to:

$\varepsilon <= \varepsilon max$

Accuracy >= Amin

Latency <= Lmax

Where:

- P: Privacy protection.
- U: Analytical utility.
- T: Client trustworthiness.
- D: Data or model distortion.
- L: Processing and communication latency.
- λ_1 …….λ_5 : Optimization coefficients.

Privacy should be represented by formal privacy guarantees and attack-resistance measurements rather than entropy alone.

The optimization coefficients can be tuned using grid search or Bayesian optimization on a validation dataset.

4.9 Module 7: Fault-Tolerant Global Model Delivery

The final module ensures that model training and delivery can continue despite client failures.

It incorporates:

- Client timeout handling.
- Partial participation.
- Delayed update handling.
- Model checkpointing.
- Global model recovery.
- Communication-efficient update transmission.

The availability ratio is calculated as:

Availability = $T_{Operational} / T_{total}$ x 100

The framework will be tested under different client dropout rates to evaluate its resilience.

4.10 Complete Proposed Architecture

The following diagram represents the proposed system-level architecture. The proposed framework integrates Federated Learning, Adaptive Differential Privacy and Trust-Weighted Robust Aggregation to ensure secure and privacy-preserving Big Data processing. It involves local data preprocessing, model training, gradient clipping, privacy protection and secure model aggregation. Trust evaluation and Byzantine outlier mitigation help prevent malicious updates, while adaptive optimization improves model performance. Finally, the updated global model is securely delivered to participating clients, ensuring privacy, reliability and fault tolerance.

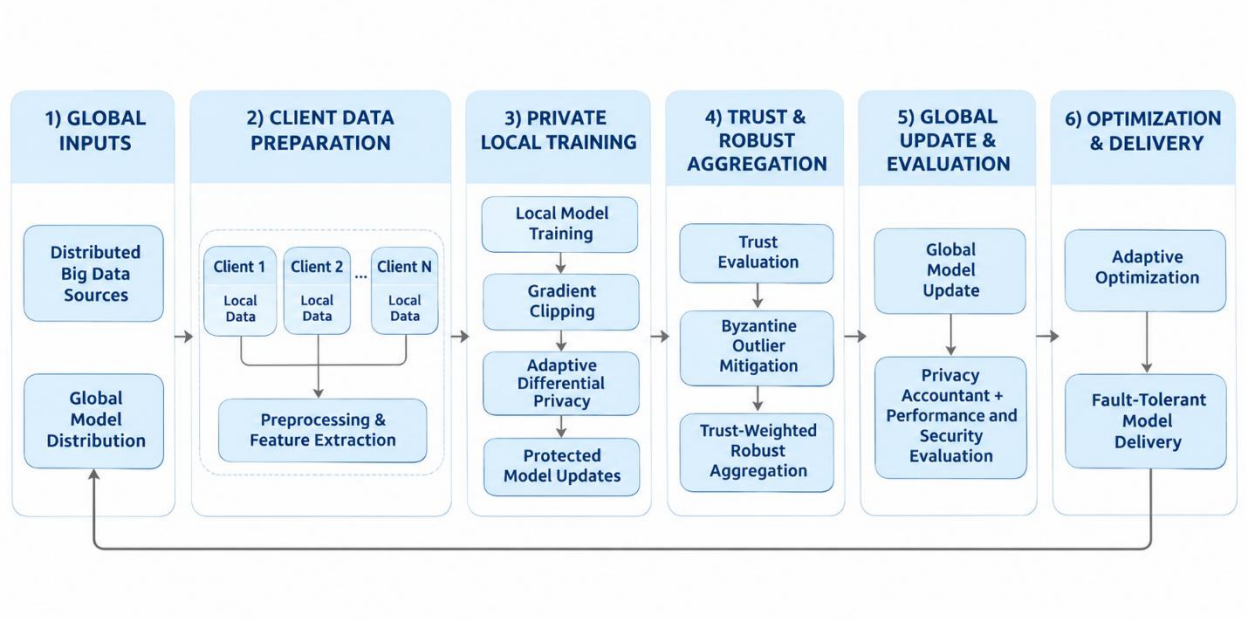


Figure 1. Proposed TABR-FL-DP system architecture.

This figure can be redrawn as a publication-quality vector diagram. It represents the seven principal modules and the iterative feedback process.

V. PROPOSED ALGORITHM

Algorithm 1: Trust-Aware Byzantine-Resilient Federated Learning with Adaptive Differential Privacy

Algorithm name: TABR-FL-DP Algorithm

Input:

- Distributed client datasets $D_1, D_2, \ldots, D_N$.
- Initial global model W_0.
- Number of federated clients N.
- Number of communication rounds R.
- Privacy budget (ε, δ).
- Gradient clipping threshold C.
- Client trust threshold τ.
- Maximum tolerated malicious-client fraction f.

Output: Optimized privacy-preserving global model W*.

Algorithm TABR_FL_DP:

1. Initialize global model W0
2. Initialize client trust scores Ti = 1.0
3. Initialize privacy accountant
4. Set communication round t = 1

5. FOR each communication round t:
    Select participating clients St
    FOR each client i in St:
        Load local dataset Di
        Perform data preprocessing
        Extract features using adaptive encoder
        Train local model using current global model Wt
        Calculate local gradient gi
        Clip gradient using threshold C
        Apply calibrated Gaussian noise
        Generate protected update g_tilde_i
        Evaluate local update consistency
        Submit protected update and permitted
        trust-evaluation evidence
    END FOR
    Evaluate eligible client reliability
    Update trust scores
    Detect suspicious model updates
    Apply Byzantine-resilient outlier mitigation
    Perform trust-aware robust aggregation
    Update global model Wt+1
    Update cumulative privacy accounting
    Check privacy budget and utility constraints
    IF privacy budget is exhausted:
        Stop private training
    END IF
    Evaluate validation accuracy and convergence
    Save global model checkpoint
    Handle client failures and delayed updates
    t = t + 1
6. END FOR
7. Return optimized global model W*
END Algorithm

5.1 Algorithmic explanation

Step	Operation	Expected outcome
1	Global model initialization	Common learning model
2	Client selection	Distributed participation
3	Local preprocessing	Clean training data
4	Feature extraction	Relevant representations

5	Local model training	Local model updates
6	Gradient clipping	Bounded sensitivity
7	Gaussian noise addition	Privacy protection
8	Trust evaluation	Client reliability
9	Outlier mitigation	Malicious update reduction
10	Robust aggregation	Secure global optimization
11	Privacy accounting	Formal privacy tracking
12	Model update	Improved global model
13	Fault management	Training continuity

Table 3: TABR_FL_DP Algorithm Expected Outcomes

Implementation clarification: The privacy accountant must operate on the actual private mechanism and sampling process. The algorithm should stop or change to a separately accounted training mechanism when the privacy budget is exhausted; it must not continue private training without accounting for additional privacy loss.

5.2 Computational complexity

Let:

- N = Number of participating clients.
- d = Model update dimensionality.
- R = Number of communication rounds.
- E = Number of local training epochs.

The approximate complexity of local training is:

$O(R\ N\ E\ C_{local})$

Where C_{local} represents the per-epoch computational cost.

The communication complexity for transmitting one model update per client per round is approximately:

$O(R\ N\ d)$

The coordinate-wise trimmed-mean aggregation requires sorting update values independently across coordinates, with approximate complexity:

$O(Nd \log N)$

The actual cost will depend on model architecture, client participation, aggregation strategy, and implementation environment.

VI. SUITABLE RESEARCH DIAGRAMS

The sequence diagram illustrates the communication process between two clients, a Federated Server and a Privacy Accountant. Initially, the federated server distributes the global model to participating clients. Each client performs local model training, applies gradient clipping and adds differential privacy noise to protect sensitive information. The protected model updates are sent to

the federated server, which evaluates trust and robustness before aggregating valid updates. The Privacy Accountant monitors privacy loss and provides updated privacy guarantees. Finally, the server generates and distributes the updated global model to all participating clients, ensuring secure and privacy-preserving federated learning.

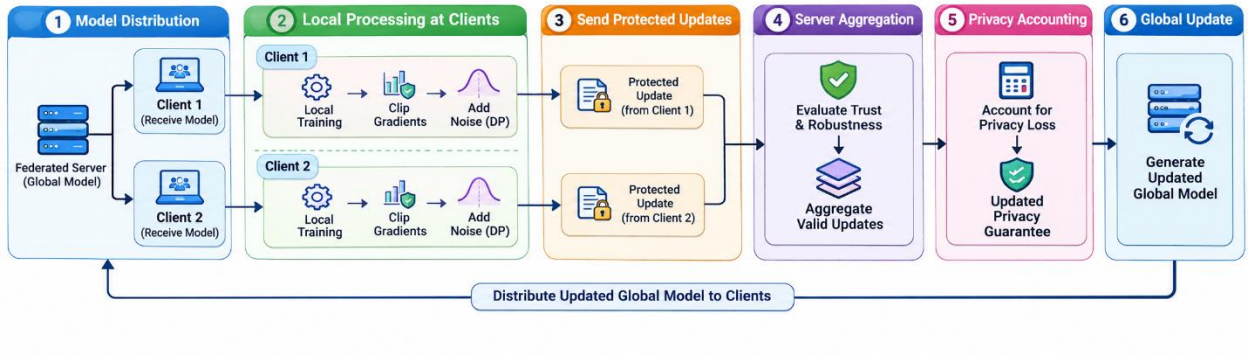


Figure 2. Federated model training and communication sequence

VII. EXPERIMENTAL DESIGN AND EVALUATION

7.1 Experimental environment

The proposed framework can be implemented using the following tools.

Component	Recommended Technology
Programming language	Python 3
Deep learning	PyTorch
Federated learning	Flower
Data processing	Pandas, NumPy
Machine learning	Scikit-learn
Privacy mechanism	Opacus or an appropriate DP accounting implementation
Visualization	Matplotlib
Development environment	Google Colab / Jupyter Notebook
Operating system	Windows 10/11 or Linux

Table 4: Recommended Technology for Privacy Enhancement

The combination of PyTorch and Flower is recommended because it supports distributed model-training experiments and customization of client-server workflows.

7.2 Dataset selection

1. CIC-IDS2017

Primary dataset

- Benign and malicious network traffic.
- Multiple attack categories.
- Suitable for intrusion detection and federated client partitioning.

2. UNSW-NB15

Secondary dataset

- Normal and attack network traffic.
- Suitable for evaluating generalization.
- Useful for additional cross-dataset experiments.

7.3 Experimental client configuration

Parameter	Experiment setting
Number of clients	10, 20, 30
Communication rounds	50, 100, 150
Local epochs	1, 3, 5
Malicious clients	0%, 10%, 20%, 30%
Client dropout	0%, 10%, 20%, 30%
Privacy budget ε	0.5, 1, 2, 4, 8
Delta δ	Set according to the dataset and privacy definition
Aggregation methods	FedAvg, Median, Trimmed Mean, Krum, proposed method

Table 5: Experimental Client Configuration Settings

These are suggested experimental settings, not completed experimental observations. The actual number of clients and feasible privacy budgets should be selected after considering dataset size, model dimensionality and computation resources.

Privacy performance

- Cumulative privacy budget.
- Membership inference attack success.
- Gradient inversion reconstruction quality.
- Privacy–utility trade-off.

Security performance

- Attack success rate.
- Malicious client detection rate.
- False alarm rate.
- Model poisoning resistance.
- Global model degradation under attacks.

System performance

- Communication overhead.
- Training time.
- Inference latency.
- Throughput.
- Client dropout tolerance.

- Model convergence rounds.

VIII. PROPOSED TABLES FOR EXPERIMENTAL RESULTS

The following tables should be populated after running the implementation.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Centralized ML	TBD	TBD	TBD	TBD
FedAvg	TBD	TBD	TBD	TBD
FedAvg + DP	TBD	TBD	TBD	TBD
Median aggregation	TBD	TBD	TBD	TBD
Trimmed Mean	TBD	TBD	TBD	TBD
Proposed TABR-FL-DP	TBD	TBD	TBD	TBD

Table 6: Comparison of federated learning models

Model	ε	δ	Membership Inference Advantage	Reconstruction Quality
FedAvg	N/A	N/A	TBD	TBD
Fixed-DP FL	TBD	TBD	TBD	TBD
Adaptive-DP FL	TBD	TBD	TBD	TBD
Proposed TABR-FL-DP	TBD	TBD	TBD	TBD

Table 7: Privacy performance comparison

For meaningful comparisons, privacy guarantees must be reported under comparable adjacency definitions and accounting assumptions. The attack metrics should also be measured consistently.

Malicious clients (%)	FedAvg accuracy	Krum accuracy	Trimmed Mean accuracy	Proposed accuracy
0	TBD	TBD	TBD	TBD
10	TBD	TBD	TBD	TBD
20	TBD	TBD	TBD	TBD
30	TBD	TBD	TBD	TBD

Table 8: Byzantine attack evaluation

Client ID	Agreement	Consistency	History	Risk	Final trust
C1	TBD	TBD	TBD	TBD	TBD
C2	TBD	TBD	TBD	TBD	TBD
C3	TBD	TBD	TBD	TBD	TBD
C4	TBD	TBD	TBD	TBD	TBD
C5	TBD	TBD	TBD	TBD	TBD

Table 9: Client trust evaluation

Model	Training time (s)	Communication cost (MB)	Latency (ms)	Availability (%)
Centralized ML	TBD	TBD	TBD	TBD
FedAvg	TBD	TBD	TBD	TBD
FedAvg + DP	TBD	TBD	TBD	TBD
Byzantine FL	TBD	TBD	TBD	TBD
Proposed TABR-FL-DP	TBD	TBD	TBD	TBD

Table 10: System performance comparison

This table is particularly important for demonstrating which proposed components contribute to the final performance.

Experimental Configuration	DP	Trust	Byzantine Defence	Adaptive Optimization
Configuration A	No	No	No	No
Configuration B	Yes	No	No	No
Configuration C	Yes	Yes	No	No
Configuration D	Yes	Yes	Yes	No
Full proposed model	Yes	Yes	Yes	Yes

Table 11: Ablation study

IX. SUITABLE GRAPHS FOR THE RESEARCH PAPER

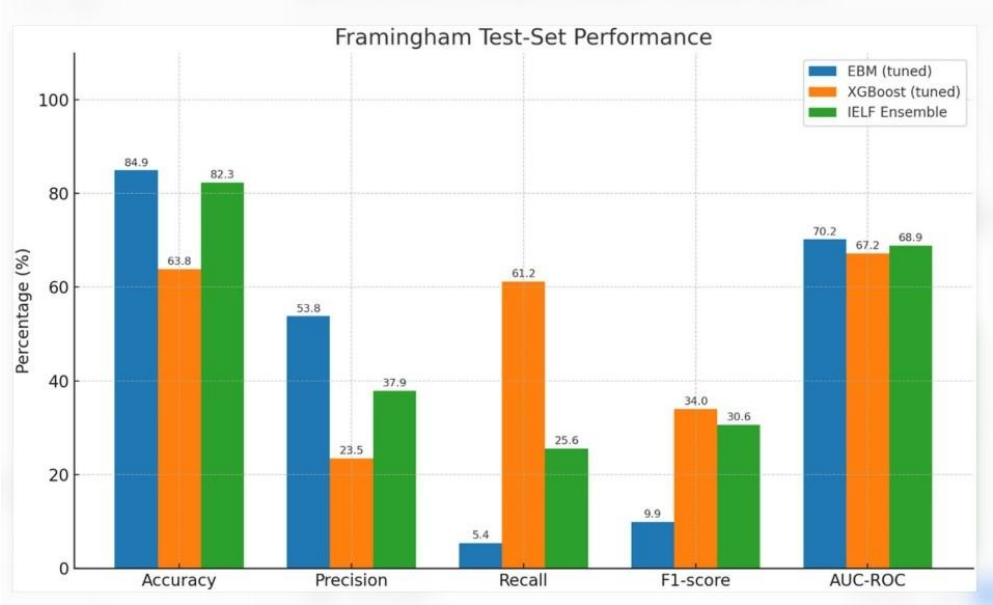


Figure 3. Accuracy comparison



Figure 4. Privacy budget versus model accuracy

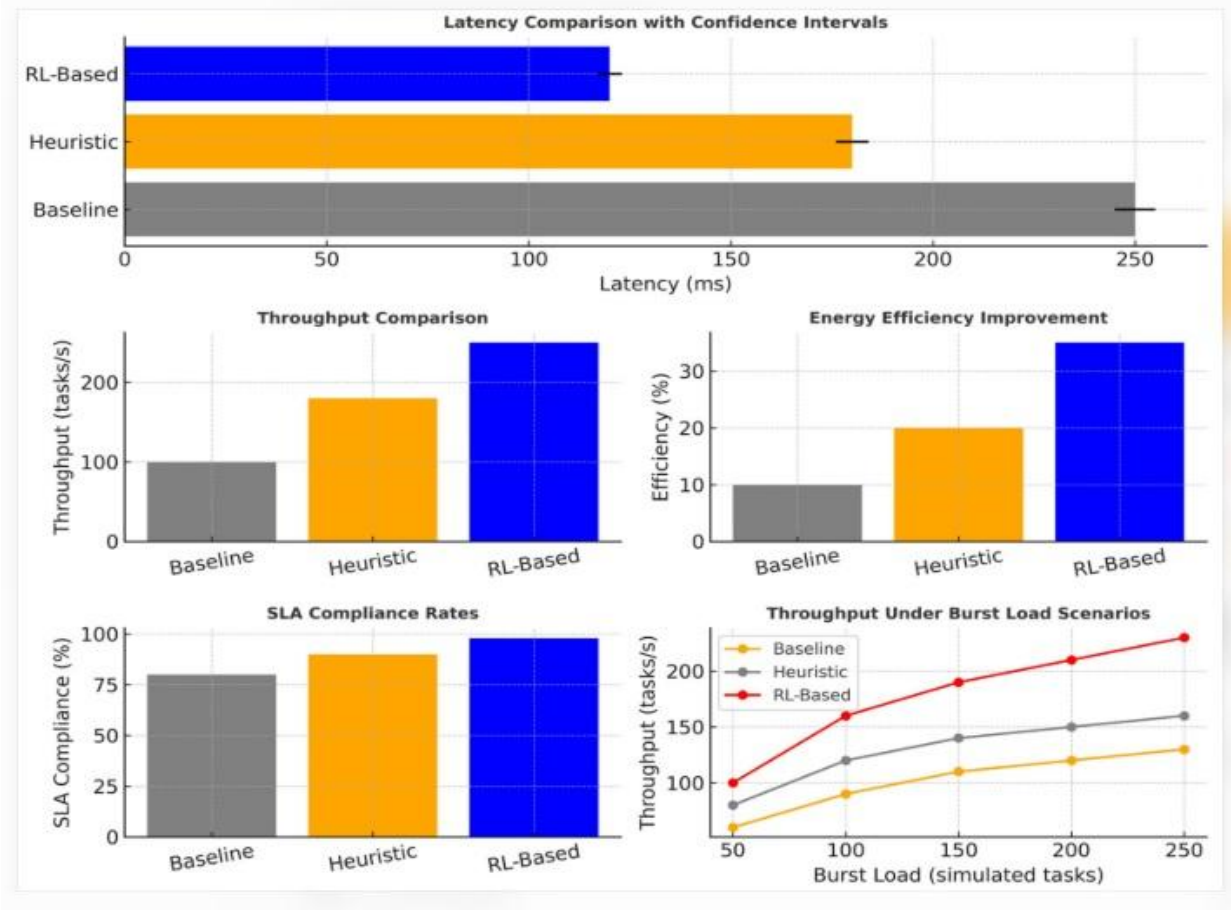


Figure 5. System Performance

X. CONCLUSION

This research proposes a Trust-Aware Byzantine-Resilient Federated Learning with Adaptive Differential Privacy (TABR-FL-DP) framework for secure and reliable big data analytics. The proposed architecture addresses important limitations associated with conventional federated learning, particularly privacy leakage through model updates, malicious client participation, model poisoning attacks, and reduced analytical utility caused by excessive privacy protection.

The framework integrates adaptive deep feature extraction, client-level differential privacy, trust-based client evaluation, Byzantine-resilient model aggregation, and privacy–utility optimization. Through gradient clipping and calibrated Gaussian noise addition, the proposed mechanism aims to provide mathematically accountable privacy protection while limiting the exposure of sensitive client information. The trust evaluation mechanism assesses participating clients using update consistency, historical reliability, and security-risk characteristics. Furthermore, the Byzantine-resilient aggregation strategy aims to reduce the influence of malicious model updates and improve global model reliability.

The proposed methodology extends the previously developed FHDPA architecture by introducing formal differential privacy accounting and explicit malicious-client mitigation. The experimental evaluation using CIC-IDS2017 and UNSW-NB15 datasets will investigate classification performance, privacy guarantees, attack resistance, communication efficiency, and fault tolerance. The research is expected to contribute a unified framework for secure distributed machine learning, particularly in privacy-sensitive applications such as healthcare analytics, financial security, industrial IoT, and cloud-based big data processing. Future extensions may investigate explainable artificial intelligence, secure multi-party computation, adaptive client selection, and blockchain-assisted federated model auditing.

XI. FUTURE ENHANCEMENT

The proposed framework can be further extended through the following research directions:

1. Explainable Federated Learning: Integrate SHAP or other explainable AI methods to interpret security decisions.
2. Blockchain-Assisted Model Auditing: Maintain transparent records of model updates and training activities.
3. Advanced Privacy Mechanisms: Investigate Rényi Differential Privacy and privacy amplification through subsampling.
4. Adaptive Client Selection: Select clients based on trustworthiness, computational capability and data contribution.
5. Lightweight Edge Intelligence: Optimize the framework for resource-constrained IoT devices.
6. Cross-Domain Federated Learning: Extend the framework to healthcare, banking and industrial environments.

7. Advanced Adversarial Defence: Investigate colluding attackers, adaptive poisoning and backdoor attacks.

REFERENCES

[1]. A. Mohamed Azharudheen, Kumudham A S. Kalaivani, and N. Sharin, "Federated Hybrid Deep Privacy Architecture for Adaptive and High-Availability Big Data Security," 2026. Previously published research paper supplied for this study.

[2]. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol. 54, pp. 1273–1282, 2017. [Online]. Available: PMLR publication.

[3]. M. Abadi, A. Chu, I. Goodfellow, B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318, 2016. [Online]. Available: Google Research publication.

[4]. I. Mironov, "Rényi Differential Privacy," in Proceedings of the 30th IEEE Computer Security Foundations Symposium (CSF), pp. 263–275, 2017. [Online]. Available: Google Research publication.

[5]. P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: NeurIPS publication.

[6]. D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates," in Proceedings of the 35th International Conference on Machine Learning, PMLR, vol. 80, pp. 5650–5659, 2018. [Online]. Available: PMLR publication.

[7]. E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How To Backdoor Federated Learning," in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR, vol. 108, pp. 2938–2948, 2020. [Online]. Available: PMLR publication.

[8]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018. Dataset: CIC-IDS2017 official portal.

[9]. A. Mohamed Azharudheen and V. Vijayalakshmi, "Improvement of Data Analysis and Protection Using Novel Privacy-Preserving Methods for Big Data Application," The Scientific Temper, vol. 15, no. 2, pp. 2181–2189, 2024.

[10]. A. Mohamed Azharudheen and V. Vijayalakshmi, "Privacy-Preserving Data Protection: A Novel Mechanism for Maximizing Availability Without Compromising Confidentiality," Journal of Information System Engineering and Management, vol. 10, issue 21, 2025.

[11]. A. Mohamed Azharudheen, A. Kumudham, and S. Kalaivani, "A Scalable Deep Learning-Optimized Data Security Architecture for High-Availability Big Data Environments," International Journal of Engineering Research & Technology (IJERT), vol. 14, issue 12, 2025, DOI: 10.17577/IJERTV14IS120127.

[12]. A. Mohamed Azharudheen and K. Kavibharathi, "A Hybrid Deep Learning-Driven Privacy Preservation Framework for High-Availability Big Data System," Technische Sicherheit, vol. 26, issue 1, pp. 1–16, January 2026.

[13]. C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, nos. 3–4, pp. 211–407, 2014. DOI: 10.1561/0400000042.

[14]. R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," arXiv:1712.07557, 2017. [Online]. Available: arXiv.

[15]. N. Agarwal, A. T. Suresh, F. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-Efficient and Differentially-Private Distributed SGD," in Advances in Neural Information Processing Systems, vol. 31, 2018.

[16]. B. Balle, G. Barthe, and M. Gaboardi, "Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences," in Advances in Neural Information Processing Systems, vol. 31, 2018.

[17]. I. Mironov, K. Talwar, and L. Zhang, "Rényi Differential Privacy of the Sampled Gaussian Mechanism," arXiv:1908.10530, 2019. [Online]. Available: arXiv.

[18]. N. Papernot et al., "Scalable Private Learning with PATE," in International Conference on Learning Representations, 2018. [Online]. Available: OpenReview.

[19]. Y. Aono, T. Hayashi, L. Wang, S. Moriai, and H. S. T. Sato, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1333–1345, 2018.

[20]. L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," in 2019 IEEE Symposium on Security and Privacy, pp. 691–706, 2019.

[21]. R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321, 2015. DOI: 10.1145/2810103.2813687.

[22]. L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients," in Advances in Neural Information Processing Systems, vol. 32, 2019.

[23]. J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting Gradients—How Easy Is It to Break Privacy in Federated Learning?," in Advances in Neural Information Processing Systems, vol. 33, pp. 16937–16947, 2020.

[24]. M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-Box Inference Attacks Against Centralized and Federated Learning," in 2019 IEEE Symposium on Security and Privacy, pp. 739–753, 2019. DOI: 10.1109/SP.2019.00065.

[25]. B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 603–618, 2017. DOI: 10.1145/3133956.3134012.

[26]. L. Bai, H. Hu, Q. Ye, H. Li, L. Wang, and J. Xu, "Membership Inference Attacks and Defenses in Federated Learning: A Survey," ACM Computing Surveys, vol. 57, no. 4, article 89, 2025. DOI: 10.1145/3704633. [Online]. Available: ACM publication.

[27]. J. Konečný, B. McMahan, and D. Ramage, "Federated Optimization: Distributed Optimization Beyond the Datacenter," arXiv:1511.03575, 2015. [Online]. Available: arXiv.

[28]. A. R. S. J. Rieke et al., "The Future of Digital Health with Federated Learning," npj Digital Medicine, vol. 3, article 119, 2020. DOI: 10.1038/s41746-020-00323-1.

[29]. M. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, Privacy-Preserving and Federated Machine Learning in Medical Imaging," Nature Machine Intelligence, vol. 2, pp. 305–311, 2020. DOI: 10.1038/s42256-020-0186-1.

[30]. T. Yang et al., "Applied Federated Learning: Improving Google Keyboard Query Suggestions," arXiv:1812.02903, 2018. [Online]. Available: arXiv.

[31]. P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[32]. D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates," in Proceedings of the 35th International Conference on Machine Learning, PMLR, vol. 80, pp. 5650–5659, 2018.

[33]. E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The Hidden Vulnerability of Distributed Learning in Byzantium," in Proceedings of the 35th International Conference on Machine Learning, PMLR, vol. 80, pp. 3521–3530, 2018.

[34]. C. Xie, O. Koyejo, and I. Gupta, "Zeno: Distributed Stochastic Gradient Descent with Suspicion-Based Fault-Tolerance," in Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference, pp. 197–208, 2019.

[35]. C. Xie, K. Huang, P. Chen, and B. Li, "DBA: Distributed Backdoor Attacks Against Federated Learning," in International Conference on Learning Representations, 2020.

[36]. E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How To Backdoor Federated Learning," in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR, vol. 108, pp. 2938–2948, 2020.

[37]. A. F. A. A. Baruch, M. Baruch, and Y. Goldberg, "A Little Is Enough: Circumventing Defenses for Distributed Learning," in Advances in Neural Information Processing Systems, vol. 32, 2019.

[38]. M. Fang, X. Cao, J. Jia, and N. Gong, "Local Model Poisoning Attacks to Byzantine-Robust Federated Learning," in 29th USENIX Security Symposium, pp. 1605–1622, 2020.

[39]. S. M. S. Karimireddy, L. He, and M. Jaggi, "Learning from History for Byzantine Robust Optimization," in Proceedings of the 38th International Conference on Machine Learning, PMLR, vol. 139, pp. 5311–5319, 2021.

[40]. X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-Robust Federated Learning via Trust Bootstrapping," in Network and Distributed System Security Symposium (NDSS), 2021. DOI: 10.14722/ndss.2021.24434.

[41]. C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The Limitations of Federated Learning in Sybil Settings," in 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID), pp. 301–316, 2020.

[42]. Y. Zhang, D. Zeng, J. Luo, Z. Xu, and I. King, "A Survey of Trustworthy Federated Learning with Perspectives on Security, Robustness and Privacy," in Companion Proceedings of the ACM Web Conference, pp. 1167–1176, 2023. DOI: 10.1145/3543873.3587681.

[43]. "A Survey of Trustworthy Federated Learning: Issues, Solutions, and Challenges," ACM Transactions on Intelligent Systems and Technology. DOI: 10.1145/3678181. [Online]. Available: ACM publication.

[44]. "A Systematic Literature Review of Robust Federated Learning: Issues, Solutions, and Future Research Directions," ACM Computing Surveys. DOI: 10.1145/3727643. [Online]. Available: ACM publication.

[45]. H. Liu et al., "Membership Inference Defense in Distributed Federated Learning Based on Gradient Differential Privacy and Trust Domain Division Mechanisms," Security and Communication Networks, vol. 2022, article 1615476, 2022. DOI: 10.1155/2022/1615476. [Online]. Available: Wiley publication.

[46]. N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in Military Communications and Information Systems Conference (MilCIS), pp. 1–6, 2015. DOI: 10.1109/MilCIS.2015.7348942.

[47]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), pp. 108–116, 2018. DOI: 10.5220/0006639801080116.

[48]. N. Moustafa, B. Turnbull, and K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4815–4830, 2019.

[49]. G. Andresini, A. Appice, N. D. Mauro, C. Loglisci, and D. Malerba, "Multi-Channel Deep Feature Learning for Intrusion Detection," IEEE Access, vol. 8, pp. 53346–53359, 2020.

[50]. N. Moustafa, "A New Distributed Architecture for Evaluating AI-Based Security Systems at the Edge: Network TON_IoT Datasets," Sustainable Cities and Society, vol. 72, article 102994, 2021.