DOI: 10.21917/ijsc.2025.0506

ENSEMBLE STRATEGY TO MITIGATE ADVERSARIAL ATTACK IN FEDERATED LEARNING

R. Anusuya[1], D. Karthika Renuka[2], Ashok Kumar[3], S.K. Prithika[4], S. Mridula[5], T. Subhaashini[6] and R. Tharsha[7]

[1,2,4,5,6,7]Department of Information Technology, PSG College of Technology, India

[3]Department of Electrical and Electronics Engineering, Thiagarajar College of Engineering, India

Abstract

Concerns about privacy are crucial in the data-driven healthcare industry of today. Federated Learning (FL) lowers the danger of data breaches by facilitating cooperative model training without exchanging raw patient data. Differential Privacy (DP), which introduces noise into model updates to protect patient data, improves FL's decentralized methodology. This is particularly useful for applications like early cardiovascular disease detection, allowing accurate models while maintaining privacy. Hospitals train models locally, sharing updates with a central server that refines a global model. Challenges include achieving model convergence and managing communication overhead. Ongoing research aims to optimize these processes, ensuring secure, privacy-preserving healthcare solutions.

Keywords:

Federated Learning (FL), Differential Privacy (DP), Data-driven Healthcare, Privacy-preserving Solutions, Early Cardiovascular Disease Detection, Model Convergence, Communication Overhead

1. INTRODUCTION

Federated Learning (FL), which enables organizations to train machine learning models cooperatively without exchanging sensitive patient data, presents a viable option for the healthcare industry in a time when data privacy is crucial. With this decentralized method, clinics and hospitals can share just model updates, like weights or gradients, with a central server, while keeping their data local. It preserves privacy while benefiting from diverse datasets, resulting in robust models that capture various patient characteristics and enhance predictive accuracy. However, FL poses risks of potential data leakage through model updates, especially from small datasets. Differential Privacy (DP), which obscures individual contributions by adding noise to model updates, is used to lessen this and stop adversaries from deducing specific patient data. By encrypting model updates as they are being transmitted, Secure Aggregation strengthens security even further and guarantees that the data is safe even in the event that the central server is compromised. The integration of FL, DP, and Secure Aggregation allows healthcare organizations to develop more accurate predictive models without compromising patient privacy. This approach facilitates personalized treatment plans by leveraging insights from multiple institutions, leading to better patient outcomes. It also accelerates medical research by pooling knowledge while maintaining data privacy. Challenges include ensuring necessary infrastructure, refining DP techniques to balance privacy with model performance, and navigating regulatory requirements. As technology evolves, innovations like blockchain could further enhance data security and transparency. FL has the potential to revolutionize healthcare analytics and promote safe and efficient improvements in patient care with continued cooperation between legislators, data scientists, and healthcare professionals.

2. LITERATURE SURVEY

Despite being intended to protect privacy by maintaining data decentralization, federated learning (FL) is susceptible to attacks that can extract private information from shared updates, such as membership inference and model inversion. It is not enough to rely on just one privacy-preserving technique, such as Secure Multiparty Computation (SMPC) or Differential Privacy (DP). While SMPC encrypts data during transmission but may not address all privacy leaks, DP can defend against some attacks but may decrease accuracy. Combining multiple strategies is needed for comprehensive protection, raising concerns about FL in privacy-sensitive areas like healthcare [1].

The paper [2] introduces Average Accuracy Deviation Detection (AADD), addressing cybersecurity in FL. AADD compares each client's model accuracy to the average across all clients, flagging significant deviations that may indicate poisoning. This method ensures collaborative model integrity by identifying potentially malicious clients. Model poisoning in FL occurs when malicious clients manipulate local data or updates to degrade overall performance. Detecting these attacks is difficult since local data isn't visible to the server. A new framework addresses this by monitoring shared weight activations in local models, identifying unusual patterns that signal poisoning and improving FL security [3].

In paper [4], a framework enhances FL security by eliminating adversarial users, identified through their reported loss values during training. This approach helps maintain accuracy by preventing malicious users from degrading the global model's performance. FedRecover, introduced in paper [5], helps restore a global model in FL after poisoning attacks, using historical training data collected before malicious clients are detected. This ensures an accurate model while keeping computational and communication costs low. Paper [6] categorizes FL defense mechanisms into two approaches: evaluating local updates' trustworthiness and securely aggregating them into a global model. It analyzes strengths, weaknesses, and challenges like scalability and balancing accuracy with security, providing insights for improving FL defenses.

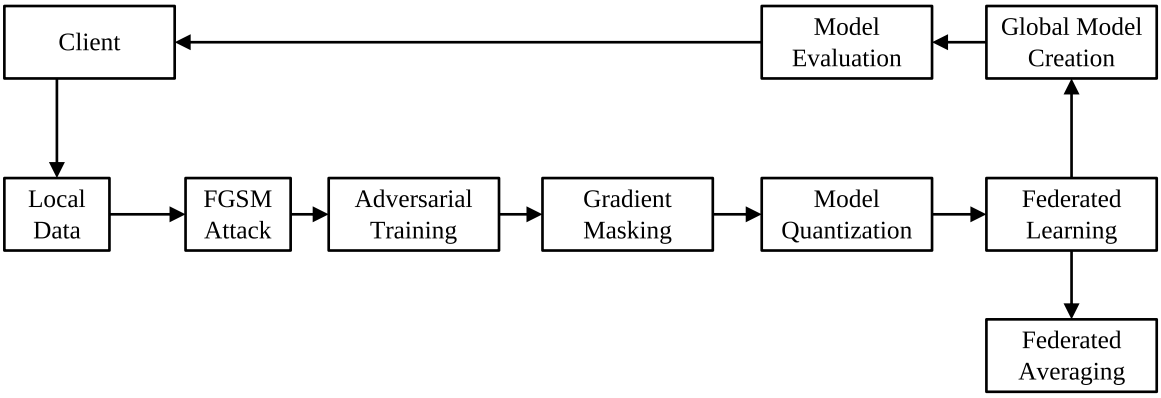

Fig.1. Block diagram of early cardiovascular disease detection with Federated Learning

The study in paper [7] explores FL's security issues, including poisoning, inference, communication, and free-riding attacks. It reviews existing defenses, highlights their limitations, and suggests future research to improve FL's resilience to emerging threats. Paper [8] discusses FL's privacy and security challenges, such as communication costs and diverse environments. It reviews defense techniques like DP and SMPC, categorizing threats by their impact on confidentiality, integrity, and availability. The study notes that while FL enhances privacy, its complexity poses challenges for widespread use. Paper [9] addresses free-rider attacks in FL, where clients submit fake updates for rewards. A high-dimensional anomaly detection method in the STD-DAGMM framework is proposed for detection. DP can mitigate these attacks but complicates the process. Robust defenses are needed to ensure FL's reliability. The study in paper [10] examines FL's security challenges, including various attack types and current defensive strategies. It highlights the limitations of these methods and suggests future research directions to improve FL's resilience. Research in paper [11] critiques the evaluation of FL security, noting unrealistic assumptions that may overstate attack effectiveness. A systematic study classifies attacks, recommending improvements for more accurate and relevant assessments in FL security research.

3. METHODOLOGY

The creation of a robust, privacy-preserving system for early cardiovascular disease detection combines Federated Learning (FL) with advanced neural network architectures. With FL, hospitals and other clients can work together to train a global model without disclosing private patient information. To lower privacy threats, each client trains a local model and only exchanges model updates, such as weights and gradients. By keeping data local, this decentralized method maintains anonymity while enhancing the functionality of the global model.

Three neural network architectures are employed: Feedforward Neural Networks (FNN), Multi-Layer Perceptrons (MLP), and Gated Residual Networks (GRU). FNN provides a simple structure for initial binary classification tasks, MLP offers enhanced resilience against adversarial attacks, and GRU uses advanced gating and residual connections for complex data patterns, making it particularly effective for understanding cardiovascular risk factors. Adversarial training is integrated using the Fast Gradient Sign Method (FGSM) to build resistance against attacks, while quantization techniques reduce the precision of model weights for efficient communication. Differential Privacy (DP) adds noise to model updates, further protecting patient data during transmission. Using the Federated Averaging (FedAvg) algorithm, local updates are combined into a global model, allowing for collaborative improvement while maintaining privacy.

An ensemble model, combining predictions from FNN, MLP, and GRU through majority voting, further enhances accuracy, leveraging the strengths of each model for better decision-making. This approach improves predictive performance by balancing accuracy with robust data privacy. This framework offers a transformative solution for early cardiovascular disease detection, maintaining high model accuracy, ensuring data security, and enabling collaboration among healthcare providers. This approach can improve patient outcomes and propel healthcare breakthroughs by enabling institutions to collaborate while protecting sensitive data.

The Table.1 describes the privacy-preserving federated learning architecture that uses an ensemble of neural networks (FNN, MLP, and GRU) to detect cardiovascular illness. By introducing noise to gradients, it integrates differential privacy approaches and Federated Averaging guarantees secure aggregation.

Federated Learning with Privacy-Preserving Ensemble Framework

Input: Local datasets D_i for N healthcare clients, Privacy budgets ϵ_i, Learning rate η, Number of clients N and Neural network models: Feedforward Neural Network, Multi-Layer Perceptron, and Gated Recurrent Unit

Output: Global Ensemble Model $M_{ensemble}$

Procedure:

1. **Initialize global models** M^{FNN},M^{MLP},M^{GRU}
2. **Local Training (at each client i): For each client i:**
    - Perform local training on data D_i for models FNN, MLP, and GRU.
    - Compute local gradients g^{GRU}, g^{FNN}, g^{MLP}
    - Reduce the precision of model weights to q-bits.
    - Add Laplace noise to gradients for Differential Privacy (DP): $g_i'=g_i+Laplace(0,\Delta/\epsilon_i)$

- Send perturbed gradients g_i'(for all models) to the central server.

3. **Global Aggregation (at the central server):**
   - Aggregate local updates for each model using Federated Averaging (FedAvg).
   - Add Gaussian noise to global gradients for additional privacy: $g'_{global}=g_{global}+Gaussian(0,\sigma^2)$.
   - Update global models: $M_{global}=M_{global}-\eta\cdot g'_{global}$
4. **Ensemble Model Construction:**
   - Collect predictions from M^{FNN},M^{MLP},M^{GRU} back to the clients.
   - Apply majority voting to combine predictions into the ensemble model:

   $M_{ensemble}(x)=\text{argmax}(Vote(M^{FNN}(x),M^{MLP}(x),M^{GRU}(x)))$
5. **Iterative Training:**
   - Send the updated global models M^{FNN},M^{MLP},M^{GRU} back to the clients.
   - Repeat Steps 2–4 until the models converge.
6. **Final Deployment:** Deploy the trained ensemble model.

3.1 MODELS

The project is implemented using three Models-Feedforward Neural Networks (FNN), Multilayer Perceptron (MLP) and Gated Recurrent Networks (GRU).

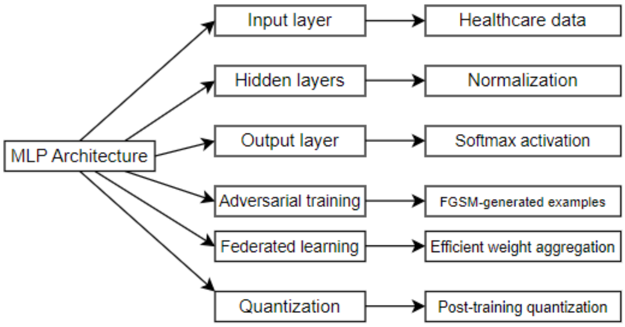

Fig.2. MLP Architecture

Multiple layers of nodes (neurons) make up MLPs, as seen in Fig.2, with each node connected to every other node in the layers above and below. An input layer, one or more hidden layers, and an output layer are often included in the design. The weighted sum of inputs received from the preceding layer is subjected to a non-linear activation function by every node in the output and hidden layers. The MLP uses backpropagation to learn the ideal set of weights in order to translate input data into output predictions. The term "multilayer" refers to MLP's stacked hidden layers, which enable the network to recognize more intricate patterns in the data than a basic perceptron. Although MLPs are quite powerful for various tasks, they are most effective in solving problems where the input data has a fixed size and is not sequential, such as image classification or tabular data.

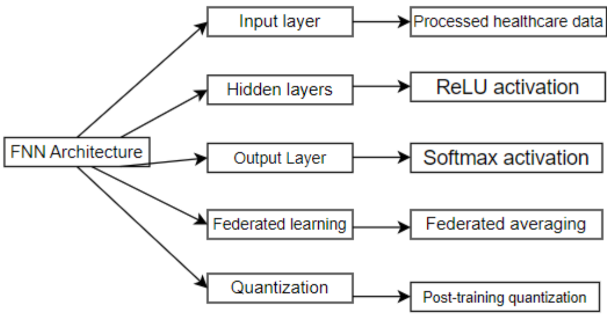

Fig.3. FNN Architecture

Generalizations of MLPs are feedforward neural networks (FNNs), as illustrated in Fig.3. Their name comes from the fact that there are no feedback loops or cycles, and that information only moves in one direction, from the input layer to the output layer via the hidden layers. For many other neural networks, such as MLPs, Convolutional Neural Networks (CNNs), and more intricate designs, FNNs can be thought of as the fundamental building block.

An FNN is essentially composed of layers where each neuron processes input data and passes it forward, and there is no interaction between neurons within the same layer. During training, FNNs optimize their parameters using algorithms like backpropagation and gradient descent. For applications including function approximation, regression, and classification, FNNs are frequently utilized. Despite their ease of use, FNNs may not be able to handle jobs involving sequential or temporal data (such as time series forecasting) because they lack a way to take input order into consideration. A particular kind of neural network called Gated Recurrent Units (GRUs), depicted in Fig.4, is made especially to process sequential data by introducing the idea of recurrence. GRUs retain a recollection of prior inputs and use that memory to guide current predictions, in contrast to MLPs and FNNs, which analyze inputs independently. Because of this, GRUs are perfect for jobs where the order of data points is important, like time series analysis, speech recognition, and language modeling. Gates, which regulate the information flow via the network, are a crucial component of GRUs. Two main gates are used by GRUs:

- The Update Gate: Establishes the proportion of historical data that should be carried forward into the future.
- The Reset Gate: Determines how much of the past data ought to be erased.

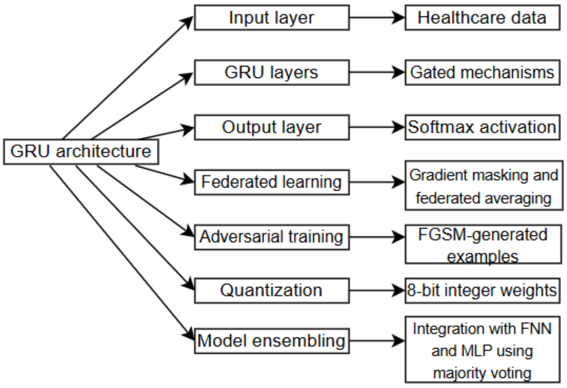

Fig.4. GRU Architecture

GRUs may concentrate on pertinent portions of the input sequence while eliminating unnecessary data thanks to these gates, which also let them to selectively keep or reject information. Unlike Long Short-Term Memory (LSTM) networks, GRUs may effectively capture long-term dependencies without requiring as much computational complexity thanks to its design. Unlike LSTMs, GRUs combine simplicity and efficiency by using fewer gates, which reduces computational overhead while maintaining performance on many sequential tasks. Compared to FNNs and MLPs, which excel in static data problems, GRUs are uniquely suited for modeling time-dependent processes and dynamic patterns in sequential data.

A major development in machine learning is Federated Learning (FL), which allows several organizations, including medical research centers and hospitals, to work together to train a common model without having to centralize sensitive data. One of the most important issues in data-driven enterprises is the security and privacy of personal data, which is addressed by this decentralized approach. Conventional machine learning techniques are frequently needed for data aggregation on a central server, which can lead to vulnerabilities and the exposure of private information.

FL, on the other hand, guarantees that the raw data stays inside the boundaries of each institution by enabling each participating body to train a local model using its own data. Every client, whether a hospital, clinic, or research facility, starts the FL process by using a particular subset of data to train its local model. Only the model parameters, such as weights and gradients, are transmitted to a central server after the local training is finished. The obtained changes are then combined by this server to improve a global model iteratively. While maintaining patient privacy during the training cycle, the aggregation procedure enables the integration of insights from many datasets, ultimately enhancing the model's performance. A significant benefit in healthcare settings where anonymity is crucial is that each client can contribute to the model's development without disclosing its raw data.

The incorporation of adversarial training methods, particularly the Fast Gradient Sign Method (FGSM), is used to strengthen the security of this cooperative training procedure. By creating adversarial examples during the training process, this technique strengthens the model's resistance to possible attacks. Each client can improve the model while protecting data privacy by encouraging collaborative learning through adversarial training, resulting in a more robust machine learning framework. To further protect personal information, Differential Privacy (DP) is used in addition to FL. When doing analyses or training machine learning models, DP, a mathematical framework, protects the privacy of each individual within a dataset.

DP reduces the possibility of disclosing private information about specific people by introducing deliberate random noise into the model's gradients. This method protects individual privacy by making sure that the inclusion or exclusion of anyone's data has no effect on the final model outputs. Gaussian noise is added to the gradients determined during model training to execute DP. It is much more difficult for adversaries to deduce information from the model updates because of the additional noise, which effectively masks the contributions of individual data points. An additional layer of security is added by using gradient masking, which randomly nullifies less significant gradients. This dual strategy, which combines gradient masking and noise injections, guarantees that the model retains high performance accuracy while simultaneously improving data security.

Federated Learning, adversarial training, and Differential Privacy work together to provide a strong solution for handling private medical data. Applications like heart disease prediction, where the analysis of private medical data needs to be done with extreme caution, benefit greatly from this architecture. This creative method creates groundwork for a time when machine learning can be used in healthcare without jeopardizing patient confidentiality by protecting data privacy and facilitating collaborative model training. In conclusion, the combination of FL and DP presents a viable technique to safely use machine learning in delicate settings, opening the door for improvements in patient care and medical research.

3.2 GRADIENT MASKING AND QUANTIZATION:

A defense technique called gradient masking is used to shield models from hostile attacks, particularly those that use gradient-based techniques like the Fast Gradient Sign Method (FGSM). In these types of assaults, the adversary uses the model's gradients to produce adversarial perturbations that can trick the model into producing inaccurate predictions. By hiding or changing the gradients, gradient masking prevents this and makes it harder for attackers to create useful adversarial samples. In this research, the main technique is to introduce Gaussian noise into the gradients as they are being trained. The gradients lose some of their informational value when noise is added. In federated learning scenarios, where several clients independently train their local models and communicate updates to a central server, this method is particularly helpful. Gradient masking aids in defending against hostile influences on both the local and global models.

Adding noise to the gradients' during training is a technique known as gradient masking. As a result, gradient-based adversarial attacks become less accurate, making it more difficult for adversaries to create effective perturbations. It is a defense mechanism that is easy to incorporate into the training pipeline and is both lightweight and computationally efficient. The quantity of noise introduced must be balanced, though, as too much noise can hinder model convergence or impair performance in general.

The process of quantization lowers the precision of a model's weights, usually turning 32-bit floating-point numbers into 8-bit integers. Smaller model sizes, less transmission capacity, and quicker inference periods are the results of this decrease in precision. Quantization is very useful in federated learning since it reduces the overhead of communication between clients and the central server. This is particularly crucial when expanding the clientele that participates. In addition to increasing performance, quantization guarantees that the model will continue to work well for extensive deployments in distributed systems by decreasing the precision of weights.

3.3 ADVERSARIAL TRAINING

By using adversarial examples-intentionally altered inputs intended to make the model produce inaccurate predictions-adversarial training is a defensive technique that increases the robustness of machine learning models. The model gains the ability to withstand such adversarial attacks in the future by being subjected to these manipulations during the training phase. The Fast Gradient Sign Method (FGSM) was used in this study to create adversarial instances. A white-box attack called FGSM modifies the input data in a way that maximizes the prediction error of the model.

3.3.1 Mathematical Representation of FGSM:

- Objective: Using a model $f_\theta(x)$ with parameters θ, an input x, and the true label Y, we want to introduce a minor perturbation to X in order to maximize the loss function $J(\theta,x,y)$.
- Gradient of the Loss: The gradient of the loss function J with respect to the input X is used to determine the perturbation direction. This gradient indicates how much the loss would change with respect to each feature in the input x: $\nabla x_J(\theta,x,y)$
- Adversarial Perturbation: In the direction of the gradient's sign, FGSM applies a perturbation of size ϵ (a tiny constant): $\delta=\epsilon\cdot sign(\nabla x_J(\theta,x,y))$. In this case, the sign function, represented by $\epsilon\cdot sign(\nabla x_J(\theta,x,y))$, indicates whether each gradient component is positive or negative.
- Perturbed (Adversarial) Input: The perturbation δ is added to the initial input (X) to calculate the adversarial example. Finally, this adversarial input is clipped to stay within the valid data range, typically between 0 and 1:

$$x_{adv}=x+\delta=x+\epsilon\cdot\mathrm{sign}(\nabla x_J(\theta,x,y))$$

To make the FNN model more resilient to attacks, the fgsm_attack function in TensorFlow computes the gradient of the loss with respect to the input (($\nabla x_J(\theta,x,y)$) and then introduces a perturbation using the sign of the gradient scaled by ϵ.

For the MLP model**,** the *create_mlp_model* function defines an architecture with two hidden layers. The *fgsm_attack* function generates adversarial inputs, and *train_local_model_with_adversarial* trains the model on these examples, improving its resilience. The *federated_learning_with_mlp* function coordinates local training across clients, combining their updates into a global model.

The train_local_model function in the GRN model calculates gradients of the loss with respect to model parameters (∇θL(y,y′). These gradients are made random by Gaussian noise and masking, which preserves privacy during training while guaranteeing efficient learning.

4. MODEL ENSEMBLING

A machine learning technique called model ensembling mixes several models to increase prediction accuracy, generalization, and resilience to assaults like overfitting. This method is especially useful in delicate industries like healthcare, where precision and dependability are crucial. By combining predictions without centralizing sensitive data, ensembling improves model security and privacy in federated learning (FL) frameworks. Even in situations where training data is still decentralized, predictions are made more accurate by integrating local model outputs through an ensemble, such as by majority voting.

This project ensembles models such as the Multi-Layer Perceptron (MLP), Feedforward Neural Network (FNN), and Gated Recurrent Unit (GRU). The central server uses majority voting to aggregate predictions from each local model's training on its own data subset. By choosing the most common label across models as the final output, this technique improves robustness and accuracy.

Better generalization, increased resistance to model extraction attempts, and increased accuracy are the driving forces behind model ensembling in FL. By adding noise to gradients, Differential Privacy (DP) further secures data and stops sensitive information from leaking. Ensembling has disadvantages like higher complexity, higher computing costs, and longer training durations, despite its advantages like fault tolerance and privacy preservation. Nevertheless, this strategy provides a robust, secure framework for healthcare applications, ensuring adherence to privacy laws like GDPR and HIPAA while maintaining model performance.

5. SIMULATED ATTACKS

A data leakage attack occurs when sensitive information is unintentionally exposed during training or model deployment. In federated learning (FL), while raw data is not directly shared, model parameters exchanged between clients can still reveal private details.

For instance, rare cases might be captured in model updates, risking the exposure of sensitive healthcare data. To address this, randomized smoothing is used, adding controlled noise to model updates, reducing the risk of overfitting to specific data. Additionally, post-ensemble quantization compresses updates, obscuring detailed information and minimizing leakage risks.

Particularly troubling in the healthcare industry, training data inference attacks use models to ascertain whether particular data was used during training. To counter this, differential privacy with enhanced noise mechanisms makes individual data contributions indistinguishable, preventing such inferences. Randomized smoothing and quantization further obscure model behavior. Model Inference Attacks (MIA) involve analyzing a model's outputs to infer if particular data points were used in training. This is a threat when models are accessed via APIs, as attackers use specific queries to gather insights into training data.

Attackers can steal proprietary algorithms without retraining by using model extraction attacks, which try to recreate a model's functionality through querying it. When models are used as black-box services in cloud-based AI applications, this poses a risk.

Feature Extraction Attacks focus on revealing the internal representations learned by a model. Attackers exploit these intermediate representations to understand how a model makes predictions, potentially exposing sensitive attributes or proprietary information.

6. RESULT ANALYSIS

6.1 MODEL PERFORMANCE ANALYSIS

- FNN: Achieved 91.89% accuracy, demonstrating robustness under adversarial conditions such as FGSM and PGD attacks. The model exhibited fast convergence, maintaining stable accuracy with low communication costs (3.32 MB over 10 rounds) and consistent loss reduction. Accuracy variations across rounds suggest sensitivity to non-iid client data and differential privacy noise.

- MLP: Reached 91.97% accuracy, showing resilience to adversarial attacks and early convergence. It demonstrated minimal communication overhead (1.05 MB over 10 rounds) and a gradual loss decline, reflecting efficient performance optimization.
- GRU: Achieved 91.76% accuracy with strong adversarial performance. The model stabilized rapidly, with a communication cost of 5.50 MB over 10 rounds, and a consistent loss decrease from 2.6245 to 2.5322. Minor accuracy fluctuations were observed, indicating iterative learning in a federated environment. Because of its sophisticated design and capacity to accurately capture intricate temporal correlations in sequential data, the GRU model has greater communication costs than FNN and MLP. This trade-off guarantees strong iterative learning and excellent adversarial performance—both essential in federated environments.

The Fig.6 and Fig.7 represent the following data points as a column graph for better understanding.

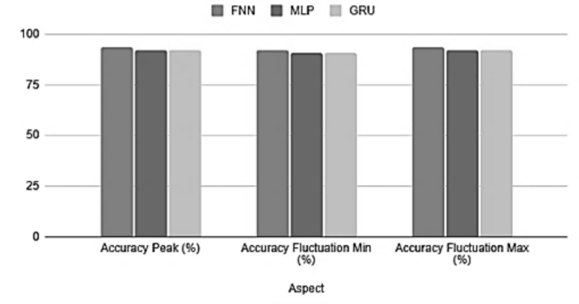

Fig.6. Accuracy Comparison of FNN, MLP, and GRU Models

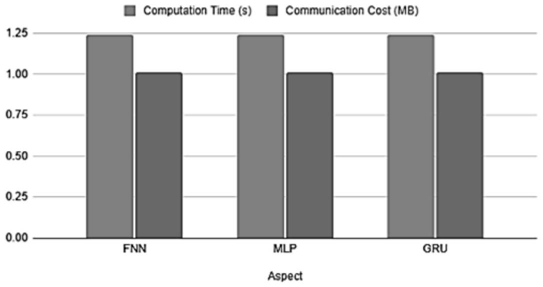

Fig.7. Computation Time and Communication Cost of FNN, MLP and GRU

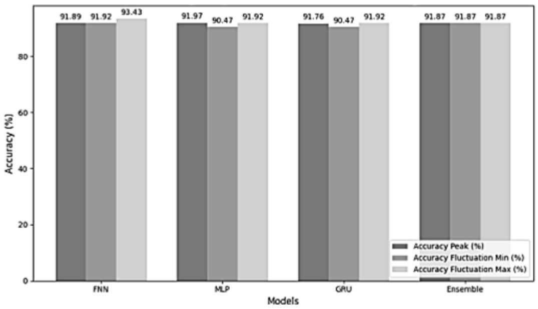

Fig.8. Accuracy Comparison Before and After Ensembling Attack Resilience Analysis

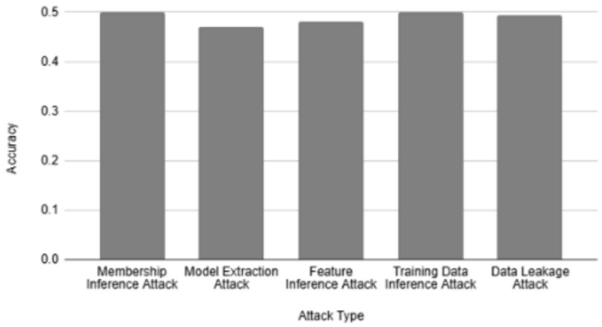

Fig.9. Accuracies of MIA, Model extraction attack, Feature inference attack, Training data inference attack, and Data leakage attack

The Fig.6 compares the Accuracy Peak and Accuracy Fluctuation Range across FNN, MLP, and GRU models. FNN achieves the highest peak accuracy of 93.43% in Round 2, while MLP and GRU both stabilize at 91.92% by Round 4. The Accuracy Fluctuation Range is smallest for FNN (91.92% to 93.43%), with MLP and GRU showing a slightly narrower range of 90.47% to 91.92%.

1) Membership Inference Attack: Accuracy and AUC of 0.5000 indicate robust privacy mechanisms, effectively preventing inference of training data membership.
2) Model Extraction Attack: With an accuracy of 0.4700 and almost zero precision and recall, the stolen model appears to have had limited success in reproducing the original model.
3) Feature Inference Attack: The attack resulted in an accuracy of 0.4800, close to random guessing, indicating poor attacker performance.
4) Training Data Inference Attack: An accuracy of 0.5000 highlights effective protection against determining the inclusion of specific records in the training dataset.
5) Data Leakage Attack: Achieved an accuracy of 0.4930, reflecting minimal success and demonstrating strong privacy protections.

The Fig.8 graphically represents the accuracy of the attack resilience of the model.

7. CONCLUSION

The study of the results demonstrates the correctness, convergence, and communication efficiency of both the ensemble and individual models. The system also demonstrates resilience to attacks, guaranteeing model security and data privacy. Because of this, the deployment technique is appropriate for real-world healthcare applications where model performance and privacy preservation are crucial.

8. FUTURE WORK

Future studies can concentrate on improving the GRU model’s communication efficiency by investigating sophisticated compression methods including gradient pruning and quantization to lower overhead without sacrificing accuracy. The robustness and security of the framework can be further increased by implementing adaptive adversarial defense mechanisms and hybrid privacy-preserving techniques that combine cryptographic and differential privacy techniques. Another interesting approach is to use hierarchical or personalized federated learning to address scalability issues in diverse environments. Deeper understanding and useful advantages may result from actual validation in healthcare environments and broadening the framework's application to a variety of medical use cases. Lastly, the framework's scalability and performance in distributed systems may be maximized by using cutting-edge technologies like edge computing for real-time processing and blockchain for secure model aggregation.

REFERENCES

[1] D. Enthoven and Z. Al-Ars, "An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies", *Studies in Computational Intelligence*, Vol. 965, pp. 1-6, 2021.

[2] V. Valadi, M. Englund, M. Spanier and A. O'Brien, "Detection and Prevention Against Poisoning Attacks in Federated Learning", *Proceedings on International Conference on Artificial Intelligence*, pp. 1-6, 2022.

[3] A. Raza, S. Li, K.P. Tran and L. Koehl, "Using Anomaly Detection to Detect Poisoning Attacks in Federated Learning Applications", *Proceedings on International Conference on Machine Learning*, pp. 1-12, 2022.

[4] N. Galanis, "Defending against Data Poisoning Attacks in Federated Learning via User Elimination", *Cryptography and Security*, pp. 1-7, 2024.

[5] X. Cao, J. Jia, Z. Zhang and N.Z. Gong, "FedRecover: Recovering from Poisoning Attacks in Federated Learning using Historical Information", *IEEE Symposium on Security and Privacy*, pp. 1366-1383, 2023.

[6] Z. Wang, Q. Kang, X. Zhang and Q. Hu, "Defense Strategies toward Model Poisoning Attacks in Federated Learning: A Survey", *Proceedings on International Conference on Wireless Communications and Networking*, 2022, pp. 548-553, 2022.

[7] X. Lyu, Y. Han, W. Wang, J. Liu, B. Wang, J. Liu and X. Zhang, "Poisoning with Cerberus: Stealthy and Colluded Backdoor Attack against Federated Learning", *Proceedings on International Conference on Conference on Artificial Intelligence*, pp. 1-6, 2023.

[8] Y. Chen, Y. Gui, H. Lin, W. Gan and Y. Wu, "Federated Learning Attacks and Defenses: A Survey", *Proceedings on International Conference on Conference on Artificial Intelligence*, pp. 1-7, 2022.

[9] J. Lin, M. Du and J. Liu, "Free-Riders in Federated Learning: Attacks and Defenses", *Proceedings on International Conference on Conference on Machine Learning*, pp. 1-6, 2019.

[10] M. Benmalek, M.A. Benrekia and Y. Challal, "Security of Federated Learning: Attacks, Defensive Mechanisms and Challenges", *Revue d'Intelligence Artificielle*, pp. 49-59, 2021.

[11] A. Wainakh, E. Zimmer, S. Subedi, J. Keim, T. Grube, S. Karuppayah, M. Mühlhäuser, "Federated Learning Attacks Revisited: A Critical Discussion of Gaps, Assumptions and Evaluation Setups", *Sensors*, Vol. 23, No. 1, pp. 1-6, 2021.