A Robust Federated Learning Privacy Framework:Integrating Multi-Party Computation and Advanced Privacy-Preserving Techniques for Secure Data Collaboration

by

Ahana Tasmim
22101596
Aneela Musarrat
22101385
Md. Naim Rahman
22101493
Md. Sakib Khan
22101054
Md. Asaduzzaman
24341184

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
2026

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ahana Tasmim
22101596

Aneela Musarrat
22101385

Md. Sakib Khan
22101054

Md. Naim Rahman
22101493

Md. Asaduzzaman
24341184

Approval

The thesis titled "A Robust Federated Learning Privacy Framework: Integrating Multi-Party Computation and Advanced Privacy-Preserving Techniques for Secure Data Collaboration. " submitted by

1. Ahana Tasmim (22101596)
2. Aneela Musarrat (22101385)
3. Md. Asaduzzaman (24341184)
4. Md. Naim Rahman (22101493)
5. Md. Sakib Khan (22101054)

of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in Fall, 2025.

Examining Committee:

Supervisor:

Mr. Moin Mostakim

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:

Dr. Md Golam Rabiul

Thesis Coordinator and Professor
Department
Brac University

Head of Department:

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

This research was conducted with a strong commitment to ethical principles in the design and development of privacy-preserving technologies. Our proposed Framework is designed to protect user data and increase privacy. In our methods, we have included Multi-Party Computation, differential privacy, and fully homomorphic encryption, which are intended to enhance data privacy, minimize risks of corruption in data, and prevent unauthorized access or inference.

No real user data was used in our study work; all experiments were performed on publicly available or synthetically generated datasets. We also ensured that our framework follows the ethical guidelines and complies with data protection regulations (GDPR) and other international privacy standards.

By making more secure and responsible machine learning practices, our work aims to promote transparency, trust, and accountability in collaborative AI systems.

Abstract

Federated Learning(FL) can train models without exposing raw client data, but still vulnerable to privacy leakage and poisoning attacks. Present methods like secure aggregation tend to offer little to no malicious update detection. Other research methods to mitigate this problem also fail to give a proper balance between privacy and robustness, or often give an impractical premise of needing multiple servers, creating a complicated application and deployment. This paper proposes DDFed-Markov, a probabilistic dual-defense federated learning framework using markovian chain that simultaneously improves privacy without compromising robustness. The framework uses a combination of noise injection using Markovian probability, Fully Homomorphic Encryption (FHE), and a feedback-driven strategy based on consensus with the collaboration of the Dual Defense Federated Learning Framework. While other models that use the concept of noise work in a similar way, the Markovian noise remains different because of its state-dependent nature. It introduces temporal correlation that increases resistance to adversaries while maintaining learning efficiency. Encrypted updates are aggregated using CKKS homomorphic encryption, allowing similarity-based aggregation without the need for multiple servers, preserving FLs' hub and spoke topology. Also, client-side consensus is applied to filter malicious updates using adaptive thresholds. Experimental evaluations on publicly available data sets like MNIST have shown that DDFed-Markov has managed to achieve a comparative accuracy while effectively prevents potential model poisoning and successfully preserving privacy compared to existing frameworks.

Keywords: Federated Learning, Privacy Preservation, Multi-Party Computation, Differential Privacy, Homomorphic Encryption, Secure Communication, Data Confidentiality, Model Inversion Attacks, Inference Attacks, Data Leakage Prevention, Decentralized Machine Learning, Privacy-Preserving Framework.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Federated Learning (FL) has given us the opportunity to train models collaboratively without centralizing the data. This has become very useful when it comes to sharing of sensitive data, such as, healthcare, finance or use of personal devices [1]. It has given us promising solutions to data privacy. However, FL is not inherently secure yet. While updating the model the shared gradient between clients and the server can leak sensitive data which makes the system vulnerable to model inversion, model poisoning, inference attacks and data poisoning [2] [3]. In solution to these problems, protocols like Multiparty Computation (MPC), Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-Party Communication (SMC) have been proposed [4] [5] [6]. These privacy preserving techniques enhance confidentiality by encrypting or disguising data or computation to increase the secrecy of the learning process of the model but each has drawbacks in terms of computing overhead, implementation expense, and model accuracy [7]. Therefore, we focus on creating a strong hybrid framework integrating all these privacy-preserving methods to improve both security and efficiency for data collaboration in practical federated learning environments.

1.2 Rational of the Study

As machine learning becomes increasingly embedded in high-stakes areas such as healthcare, finance, and personal devices, the need for strong privacy guarantees has grown significantly. **Federated Learning (FL)** emerged as a potential solution, aiming to keep raw data decentralized by allowing model training to occur locally on user devices. While this is a step in the right direction, research has shown that it does not eliminate privacy risks entirely. The fact that privacy and security threats are simultaneous in FL requires a two-kick defensive strategy. **Xu et al.** [8] proposed a framework called the Dual Defense Federated learning (DDFed) based on fully homomorphic encryption (FHE) to perform secure aggregation and two-phase anomaly detection on encrypted updates with no requirements of non-colluding servers. Nonetheless, the conventional approaches of differential privacy reduce the accuracy of models. This drives the extension of DDFed using Markovian probabilistic process to maximize privacy by giving up temporal correlation

and retaining computational efficiency. **inference attacks** can still extract sensitive information from model updates, as demonstrated by **Truex et al.** [9] and **Geyer et al.** [10] One well-regarded strategy to enhance privacy in FL is **Secure Aggregation**. **Bonawitz et al.** [11] introduced a protocol that allows the server to compute the sum of client updates without accessing individual contributions. This design is not only efficient and scalable but also resilient to client dropouts, which are common in real-world FL systems. This method mainly secures the aggregation phase and do not protect if the server or model is compromised. **Homomorphic Encryption (HE)**, where computations are done directly on encrypted data. This means that the updates remain encrypted throughout training, adding another layer of defense. **Phong et al.** [12]demonstrated the feasibility of this approach, while **Zhang et al.** [13] developed **BatchCrypt**, a technique to reduce the computational overhead by batching encrypted data. By **Differential Privacy (DP)**, injecting calibrated noise into updates, it reduces the chance that individual user data can be reconstructed. **Geyer et al.** [10]proposed that applying DP at the client server, which offers meaningful protection, and also in cases where other layers fail. However, DP can also impact model accuracy if it is not carefully tuned. As **no single technique offers complete protection**, each method addresses a different aspect of the privacy challenge, and each has its own limitations.

1.3 Problem Statement

Inference attacks, model inversion, and gradient leaking are not only a few of the major security and privacy threats that Federated Learning (FL) must deal with. These put sensitive data at risk. Current FL frameworks struggle to strike the right balance between privacy and security. Effective privacy protocols encrypts model updates and complicates the attempt of poisoning attacks, whereas defenses against such attacks often require access to unencrypted updates which compromise the privacy. Present approaches that try tackle these problems are inadequate since they are frequently incorrect, ineffective, or predicated on unrealistic assumption.,Although partial solutions can be achieved through common privacy preservation techniques like Differential Privacy, Secure Aggregation, Homomorphic Encryption, and Multi-Party Computation (MPC). These methods are not entirely affective when used individually. Differential privacy has a chance of reducing model accuracy, while encryption methods pose serious scalability and computing overhead issues. When devices have limited resources and client data is non-IID, these problems get worse. To address these limitations, this research proposes a unified privacy framework for federated learning. The framework combines Fully Homomorphic Encryption, and MPC to provide strong privacy protection, defend against attacks, and maintain model performance. The goal is to make federated learning more secure, efficient, and practical for real-world use.

1.4 Objective

The goal of this research is to develop a robust framework that doesn't rely on one protocol but integrates the strengths of several in a way that supports scal-

able and secure collaboration. The proposed study will focus on the creation of the DDFed-Markov framework that would be able to protect privacy and prevent poisoning attacks at the same time. The structure unites: (1) Markovian probabilistic injection of noise with time correlation, (2) fully homomorphic encryption of secure aggregation, and (3) consensus-based validation of the malicious update detection. The aim is to attain model accuracy of more than 95% with less than 40% inversion attack and more than 80% backdoor attack attack success rates.We believe such a framework has the potential to not only improve technical outcomes but also help build trust in FL systems used in sensitive and critical domains. Most of the existing federated learning frameworks are not robust enough for use in real world environments and cannot survive the increasing security threats. As a result, the need for a comprehensive and workable privacy based framework is increasing day by day. The impact on model performance and accuracy while ensuring security is a major challenge. Sometimes, adding privacy preserving techniques such as- Homomorphic encryption or differential privacy increases model training time and increases resource utilization. Therefore, a framework is needed that integrates these security friendly techniques while maintaining performance. The main purpose of this research is creating a robust and integrated privacy framework which combines advanced techniques such as- Differential privacy (DP), Homomorphic encryption (HE), and Multi-party computation (MPC) and Secure Multi-Party Communication (SMC). This framework will ensure improved security, privacy, and scalability for federated learning systems in real world environments.

1.5 Methodology in Brief

The methodology uses an extension of DDFed that is based on a modular architecture, Markovian Probabilistic Fully Homomorphic Encryption (MP-FHE). Every client holds a Markov state of his/her own (LOW, MEDIUM, HIGH) and state-dependent noise injection. FHE is capable of aggregating encrypted updates without access to the servers of individual updates. Aggregated updates with adaptive thresholds are proven by consensus voting based on L2 norm analysis. The framework is tested on non-IID data distribution MNIST dataset. This privacy-focused Federated Learning system enables multiple clients—such as hospitals, mobile devices, or financial institutions—to collaboratively train a shared machine learning model without sharing their private data.

1.6 Scopes and Challenges

Sensitive data never leave local devices, which helps stick to privacy regulations like GDPR and HIPAA. It also facilitates training across devices without any central data collection, which lowers the chance of privacy breaches and security threats. The system uses robust aggregation and anomaly detection to prevent attempts like poisoning attacks. The server can process encrypted updates via homomorphic encryption, which also add another dimension to privacy and security. While this encrypted probabilistic strategy allows for secure communication, it also adds significant computational requirement, delaying learning and increasing energy con-

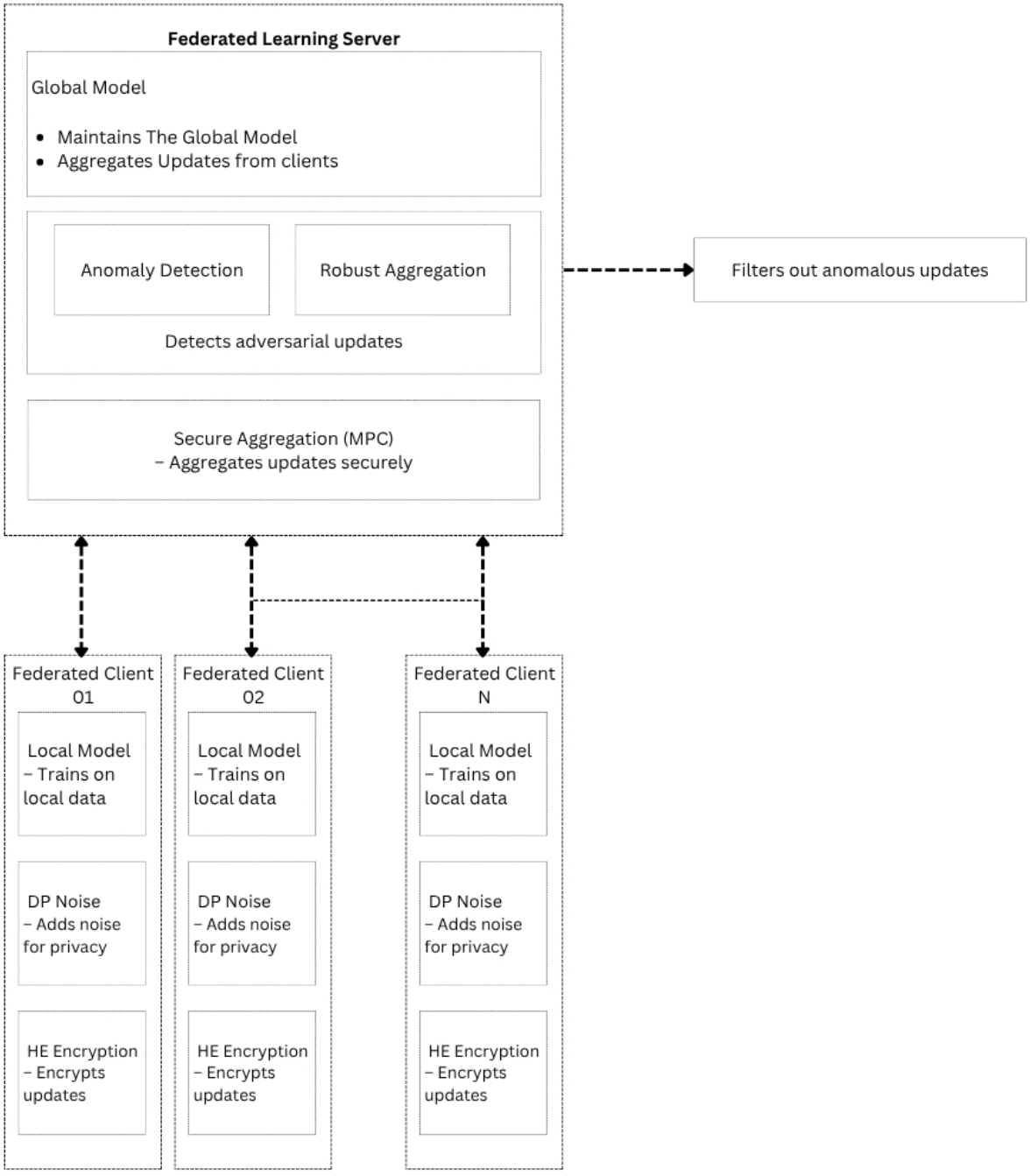

Figure 1.1: Federated System with Multiple Devices

sumption, which puts a strain on devices with little power. Managing non-IID data, unreliable networks, and the complexities of key management are challenges that may limit training scalability and efficiency.

1.7 Our Contributions

The present work discusses an innovative and secure federated learning framework developed using the Distributed Deterministic Federated Learning (DFDFL) approach. In particular, this work extends the traditional DFDFL framework by incorporating time-dependent correlated noise through a Markovian mechanism. The motivation behind this study is to minimize probabilistic inversion attacks by preventing adversaries from averaging multiple model updates across the entire training process. Unlike most other existing solutions, the proposed federated learning framework combines three major components (i.e., Markov noise injection, secure fully homomorphic encryption, and secure client-side consensus validation) into a single federated learning pipeline. A complete federated learning ecosystem has been implemented to include multiple clients and one central aggregation server that can be tested against multiple poisoning attacks. The results of the experimentation suggest that the proposed federated learning solution increases privacy and safety beyond current standards while remaining within acceptable computational limits compared to the original DFDFL solution.

Chapter 2

Literature Review

2.1 Preliminaries

We searched for trusted academic websites like ResearchGate, IEEE Xplore, and Google Scholar. By looking for keywords such as "Federated Learning Privacy," "Differential Privacy," "Homomorphic Encryption," "Gradient Leakage" and "Multi-party Computation (MPC)," we found several papers that matched our topic. Instead of going through every article, we narrowed it down to a few that were relevant and came from well-known conferences and publications. [14]Key papers that shaped our understanding included "Efficient Data Collaboration Using Multi-party Privacy-Preserving Machine Learning Framework" , [15]"Privacy-Preserving AI Analytics in Cloud Computing" , [16]"Privacy-Preserving Federated Learning for Healthcare Data Sharing", and [17]"A Privacy-Preserving Federated Learning Framework for Healthcare Big Data Analytics in Multi-cloud Environments". These articles covered privacy-preserving machine learning frameworks, AI analytics in cloud computing, and federated learning for healthcare data sharing. The federated learning works in a round-based model in which clients train their local models and send updates to an aggregation server. Inference and inversion attacks Membership Membership inference, and model inversion attacks are the privacy risks. The security threats are poisoning attacks that are used to corrupt the integrity of models. Fully homomorphic encryption allows the calculation of encrypted data without their decryption. Differential privacy is a formal privacy guarantee based on the addition of calibrated noise. Secure aggregation safeguards the contribution of individual clients in the model aggregation.
To define the technical aspect of our study, we provided brief definitions of some of the key terms:

- **Federated Learning (FL):** A genius solution for different organizations or devices to collaborate on training a machine learning model without ever sharing their original data. It keeps data in place and preserves privacy while still enabling collaboration.
- **Differential Privacy (DP):** This technique obscures personal information by adding just enough noise into the results so it's essentially impossible to trace anything back to a person, whether the data is looked at or not.
- **Homomorphic Encryption (HE):** A good encryption method whereby someone can perform calculations on data without decrypting the data.

- **Gradient Leakage:** An attack whereby attackers can take advantage of the gradients shared between client and server during training rounds with the aim to reconstruct the raw training data.
- **Multi-Party Computation (MPC):** An system where multiple participants can compute something in a collaborative manner without sharing their local data.
- **Secure Aggregation:** This scheme keeps the central server from seeing individual user contributions, only the summed updates at the end, improving user privacy in federated learning systems.
- **Strong Aggregation:** Goes one step further in removing bogus or malicious data updates. Protects the model from poisoning or tampering during training.
- **Model Inversion Attacks:** These attacks try to reverse-engineer reasoning or outputs from a model in order to intelligently infer what sensitive inputs were used in training. It is like trying to complete a puzzle by studying the clues left behind.
- **Inference Attacks:** These are all attempts to find out if some data was in the training set, or to outsmart personal data just by seeing how the model responds a huge privacy concern with AI.

2.2 Review of Existing Research

Over the past few years, an increasing amount of research has focused on improving privacy-preserving mechanisms of Federated Learning (FL) systems, particularly in sensitive data use cases such as healthcare, finance, and cross-organizational collaboration. Throughout these studies, the synergy of advanced cryptographic approaches, secure communication protocols, and adaptive privacy mechanisms has become increasingly prominent to mitigate common vulnerabilities such as gradient leakage, inference attacks, and model inversion. The findings of this body of work serve as groundwork for our proposed framework and determine the strengths and weaknesses of existing solutions.

FL studies on privacy preservation make use of differential privacy and secure aggregation. Nonetheless, such approaches hide individual updates, which prevents the detection of malicious updates. Poisoning attack protection must update in an unencrypted format, violating privacy. The dual-defense methods have several weaknesses: accuracy is compromised by differential privacy, and non-colluding server methods interfere with typical FL topologies. The DDFed framework solves the two issues with FHE and two-phase anomaly detection based on encrypted updates, that is, non-colluding assumptions on servers are ruled out [8]. Markovian methods inject noise that is associated with time, and thus patterns become more difficult to filter without impairing utility.

Various research has demonstrated the efficacy of integrating Homomorphic Encryption (HE) in FL frameworks to ensure secure model updates and aggregation without exposing raw data. By way of cryptosystems such as Paillier and RNS-CKKS, these works perform computation over encrypted data directly while preserving confidentiality during the training loop [14] [17]. While HE has good theoretical guarantees,

the solutions to it have computational costs in terms of increased latency in encryption, decryption, and aggregation, especially in the case of large models and multi-cloud environments [15] [17].
However, the ability to perform accurate computations over encrypted inputs remains a strong point in privacy-critical applications.
Differential Privacy (DP) has also been a core component in most FL paradigms, typically implemented through the injection of noise to model updates. Adaptive DP mechanisms have also been introduced to adaptively update the privacy budget to accommodate data sensitivity and training phase. This approach allows systems to apply stronger protection to extremely sensitive attributes, such as those found in health care information (e.g., HIV status, mental health information), without rendering themselves useless for less sensitive information [16] [17]. Though DP can create a trade-off between model performance and privacy and at times force a 3–7% loss in performance, its mathematical guarantees and the fact that it can be used with other methods such as HE and Secure Multi-Party Computation (SMPC) mean that it is a reliable and versatile method [16].
SMPC-based protocols are the other widely used privacy-preserving method, allowing parties to jointly compute a function without divulging their individual inputs. These protocols, generally aimed at secure aggregation, threshold decryption, or decentralized trust, proved the effectiveness of capping adversarial participant exposure. Coupled with cross-cloud communication protocols, they have shown scalability in distributed systems such as AWS, Azure, and Google Cloud [16] [17]. Their deployment in real-world environments is, however, marred by issues such as the implementation complexity being extremely high, reliance on "semi-honest" behavior assumptions, and difficulties in ensuring compliance with various international privacy regulations.
Several frameworks further incorporate secure aggregation methods and blockchain technology to enhance auditability and robustness. By merging trust scoring methods and identification authentication, such frameworks avoid risks from untrusted parties and potential poisoning attacks [15] [16]. Despite increments that improve system integrity and transparency, they add more layers of computational overhead and necessitate advanced system development and expertise in cryptography and regulatory issues [15].
Performance measures across the reviewed literature generally highlight a trade-off between computational efficiency and privacy. Model accuracy (privacy-augmented and un-augmented), communication cost, privacy leakage resistance, and training time have been used as the metrics for comparison of frameworks. A number of studies significantly established that privacy-preserving FL systems maintained accuracy in the desired range (generally 83–94%) compared to non-private baselines [15]. Furthermore, collaborative training was found to contribute to improvements in data utility, equality, and model generalization among different clients [14] [15].
As, ongoing research indicates unparalleled advancements toward safe and scalable federated learning systems. Techniques such as HE, DP, and SMPC have demonstrated the capability to neutralize significant threats without sacrificing cooperation in decentralized and untrusted networks.
Privacy inference attacks can expose sensitive user data through gradient inversion, membership inference, and model inversion [18–22]. Common defenses include secure aggregation, Homomorphic Encryption (HE), and Multi-Party Computation

(MPC) [18, 19, 22]. Poisoning attacks are another serious threat, where malicious clients manipulate updates to harm the model or insert backdoors [18, 22]. These attacks are usually handled using robust aggregation methods such as Krum, Trimmean, and FLTrust [18]. However, most existing defenses focus on either privacy or poisoning, but not both, leaving a critical security gap [18]. Cryptographic methods like HE also introduce high computation and communication costs. This makes them difficult to use in large scale FL systems, especially for foundation models [22].The core problem across all works is that most existing defenses tackle either privacy or robustness, but not both simultaneously [18] [19]. Additionally, full encryption approaches like traditional HE impose massive overheads that make them impractical for modern deep learning models [22]. Combining these defenses is challenging because robust methods require access to individual client updates, while privacy-preserving methods deliberately conceal individual contributions [18] [19].Differential Privacy (DP) emerges as a complementary approach, with central DP offering better utility than local DP but requiring trust in the aggregation server [19]. Weighted Federated Learning (wFL) adds another dimension by allowing clients to contribute based on dataset size or quality, introducing additional privacy considerations [21].

All papers present complementary approaches addressing different aspects of privacy-preserving FL. Gehlhar et al. present the SAFEFL framework featuring a PyTorch-MP-SPDZ communicator interface enabling seamless integration of FL algorithms with MPC protocols, implementing 14 aggregation schemes and 7 poisoning attacks to identify robust candidates with FLTrust adoption for secure-robust aggregation across multiple MPC protocols including Semi2k, SPDZ2k, Replicated2k, and PsReplicated2k [18]. Gu [19]. introduce the PRECAD framework combining Crypto-Aided Central DP via MPC where clients secret-share updates with MPC servers that add calibrated noise, incorporating a secure validation protocol for lightweight cryptographic verification of norm-bounded updates within a unified system [19]. Kanagavelu [20] address scalability through a two-phase committee election approach where a small committee performs hierarchical MPC aggregation, reducing communication overhead compared to full MPC among all participants . Zhu [21] introduce oracle-aided weighted federated learning where clients contribute based on dataset size/quality through a central server coordinating encrypted update aggregation with decoupled security architecture . The FedML-HE framework using selective encryption that only encrypts sensitive model parameters during aggregation, incorporating threshold key management and overhead optimization specifically designed for large foundation models like ResNet-50 and BERT [22].

SAFEFL provides no formal theoretical proofs, relying on empirical evaluation and existing MPC security guarantees from MP-SPDZ protocols [18]. **PRECAD** offers the strongest theoretical foundation. Gu [19] establish a differential privacy guarantee achieving DP equivalent to centralized DP. They analytically demonstrate robustness from DP, showing that DP noise and norm clipping limit poisoning effectiveness. Their formal security model assumes semi-honest MPC security with bounded update assumptions [19].**Two-Phase MPC** lacks formal proofs but provides design rationale for communication overhead reduction and scalability improvements through hierarchical structure [20].**Oracle-Aided wFL** establishes security through a security composition theorem ensuring overall system security when combining secure components. Zhu [21] employ a semi-honest adversary model with

formal security guarantees under honest-but-curious assumptions, where encryption and MPC ensure individual updates remain private .**FedML-HE** does not present formal theoretical proofs or lemmas, focusing instead on empirical validation of the selective encryption approach's effectiveness in maintaining privacy while reducing computational overhead [22].

Gehlhar [18] evaluated their framework on the HAR dataset with 30 users, linear regression, 2000 rounds, and 20% malicious clients. For robustness, DnC, FLAME, and FLTrust achieved approximately 0.96-0.97 accuracy versus FedAvg (0.97) under Min-Max, scaling, and label-flip attacks. Regarding MPC overhead, the Semi2k protocol demonstrated 17× communication overhead and 1.7× time overhead for FLTrust versus FedAvg, while honest-majority protocols proved much more efficient .Gu [19] conducted experiments on MNIST achieving approximately 98% accuracy and CIFAR-10 achieving approximately 75% accuracy. Their approach significantly outperformed Local DP while maintaining privacy and proved effective against label-flipping and backdoor attacks with noise and validation [19]. It successfully deployed their framework in a smart manufacturing IoT platform, demonstrating comparable model quality to traditional ML with improved communication efficiency and effective handling of large participant numbers [20]. The conducted simulations with multiple clients having varying dataset sizes, achieving accuracy comparable to traditional FL methods with reasonable communication costs for practical deployment [21]. It evaluated their system using ResNet-50 and BERT foundation models, achieving approximately 10× reduction in overhead for ResNet-50 and up to 40× reduction for BERT compared to traditional HE-based FL. The results show stable performance across different client dropout rates and training times. This indicates the framework is practical for large-scale real-world deployment [22].

The literature highlights several key insights. Both MPC and selective HE can support practical privacy preserving FL, each with different trade-offs [18–22]. Reducing computation and communication overhead is critical for real-world use [18,22]. MPC efficiency strongly depends on the security model, with honest majority protocols being much faster [18]. PRECAD shows that crypto-assisted central DP offers better accuracy than local DP while preserving privacy [19].Scalability solutions emerge through hierarchical approaches (committee election), weighted schemes, and selective encryption addressing scalability challenges [20] [22] [21]. Robustness-privacy synergy is achieved where DP noise and secure validation can simultaneously provide privacy and robustness benefits [19]. Foundation model compatibility is established by FedML-HE, demonstrating that privacy-preserving FL can scale to large foundation models [22].MPC-based approaches excel at providing formal security guarantees and handling both privacy and robustness, but require careful protocol selection to manage overhead [18] [22] [21] [20]. As demonstrated by Gehlhar [18], "MPC overhead is heavy in dishonest-majority settings; honest-majority protocols are much more efficient."HE-based approaches like FedML-HE achieve dramatic efficiency improvements through selective encryption, making them particularly suitable for large foundation models where full encryption would be prohibitive [22]. It demonstrate that "selective encryption can make HE-based FL systems practical for large-scale applications."**Cross-Protocol Integration** should focus on hybrid approaches combining MPC and selective HE for different model components, adaptive protocol selection based on model architecture and security requirements, and integration of DP with both MPC and HE frameworks [18] [19] [22].**Advanced**

Optimization includes extending frameworks to other MPC platforms (MOTION, Silph) and HE libraries, developing intelligent parameter selection for selective encryption, and optimizing communication-heavy operations across all cryptographic approaches [22] [18].**Scalability Enhancements** involve testing with larger foundation models (GPT-scale, vision transformers), handling massive participant pools (10,000+ clients), and improving committee selection and threshold management mechanisms [20] [22].**Security Extensions** encompass extending from semi-honest to malicious adversary models across all approaches, developing adaptive attacks against selective encryption schemes, and addressing advanced threats like model extraction and property inference [18] [21] [19] [22].**Real-World Deployment** includes deploying across healthcare, finance, and IoT domains, evaluating on highly non-IID data distributions, integrating with production FL frameworks (TensorFlow Federated, PySyft, FedML), and developing automated parameter tuning for different deployment scenarios [20] [22].The five papers collectively establish that privacy-preserving FL has evolved from theoretical possibility to practical reality through multiple complementary approaches. PRECAD provides the strongest theoretical foundation combining MPC and DP [19], SAFEFL offers comprehensive empirical evaluation of MPC approaches [18], Two-Phase MPC addresses scalability through hierarchical design [20], Oracle-Aided wFL introduces flexible weighted aggregation [21], and FedML-HE demonstrates that selective encryption can make HE practical for foundation models [22].The convergence of these approaches suggests that hybrid solutions combining selective encryption, efficient MPC protocols, and differential privacy represent the most promising path forward for production-ready privacy-preserving FL systems. The choice between approaches should depend on specific requirements: MPC for maximum security guarantees, selective HE for foundation model efficiency, and hybrid approaches for balanced trade-offs.

2.3 Summary of Key Findings

From the studies and protocols we have explored, one thing is clear—there is no single privacy-preserving method in Federated Learning (FL) that fully solves all the security and performance challenges. Each of the approaches we have dived into contributes to the partial solution of the problem but also has its own limitations.While Multi-Party Computation (MPC), Differential Privacy (DP), and Secure Multi-Party Communication (SMC) all play important roles still combining them effectively remains a real challenge that needs to be studied.

According to the existing studies, there is an inherent contradiction between privacy and security in FL: the protection of aggregation by means of secure aggregation prevents the malicious detection of updates, whereas the anomaly detection violates privacy. The DDFed model is a major improvement because it can dual defend without unrealistic assumptions [8]. Nevertheless, advanced attackers can filter the noise injection technique that is traditional. This limitation is solved by implementation of Markovian probabilistic processes with time dependency, which has better privacy protection and is computationally efficient. The gap in the research is the practical solutions to the dual-defense that will work both on privacy and security without any performance reduction or architectural modification.

What we noticed from comparing these papers is that the best performing protocols were not the ones that focused on a single approach. Instead, hybrid models that

used DP with MPC, or HE with secure validation were more effective overall.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

For practical implementation, the suggested framework doesn't add any new server roles and instead uses a standard FL setup with one aggregate server and several clients [1]. This avoids the two-server assumptions that are not feasible for real-world deployment [8]. The system needs to be scalable to FL situations that are different in both cross-device and cross-silo. It should be able to handle deployments with any number of clients while maintaining privacy and security.

3.1.1 Framework Design

Data processing, model design, and privacy components are all clearly separated inside the framework's modular architecture making the system simpler to test, expand, and maintain. Fundamentally, the framework improves traditional FHE by introducing Markovian Probabilistic Fully Homomorphic Encryption (MP-FHE [23]. To improve security without hampering computational efficiency, it incorporates probabilistic Markov computation. Aggregating encrypted model updates safely is possible without depending on assumptions that non-colluding multi servers exist.To protect users against privacy leakage and poisoning threats, the framework offers a dual-defense strategy. It blends consensus-based validation, encrypted aggregation, with Markov-based noise injection to maintain learning efficiency while making noise patterns more difficult to exploit, by alternating between LOW, MEDIUM, and HIGH noise states. Consensus voting contributes in the detection of malicious updates through spotting unusual parameter changes. This multi-layer approach increases both privacy and robustness, in contrast with solutions that address each individually. In order to maintain the integrity of client updates, the system also offers secure aggregation through MPC and secure communication protocols. [5].

3.1.2 Model Architecture Requirements

Depending on the type of dataset, the system need to be able to automatically generate the framework. For numerical data, fully connected networks with dropout

and batch normalization are required. With sigmoid for binary and softmax for multi-class classification, the result should be adaptable.

3.2 Societal Impact

In the areas of personal devices, healthcare, and finance, the framework offers significant advantages to the society. It supports diagnostics and compliance in the healthcare industry by establishing collaborative AI model training without jeopardizing patient data. The framework have reduced inversion and backdoor attack vulnerabilities, which ensures GDPR and CCPA compliance, and also provides fraud detection and credit scoring in the financial industry. It prioritizes data privacy while improving customized training and collaborative learning for personal devices. It does, however, face challenges, including inference attacks and minor computational costs, which can be reduced while maintaining legal compliance through cryptography and security evaluations.

3.3 Environmental Impact

Privacy-preserving techniques like homomorphic encryption and noise injection used in DDfed Markov increase energy usage. Nonetheless, these expenses are thought to be acceptable for applications that demand high privacy. Energy consumption can be decreased further by optimizing cryptographic operations, using efficient hardware, and utilizing renewable resources. Local client training may boost energy efficiency compared to centralized training. Keeping data local to the client side reduces network energy expenses. In the end, the framework aims to achieve a balance between strong privacy and robustness with manageable environmental impact.

3.4 Project Management Plan

The project runs for a year dividing into 3 phases. In phase 1, a literature review of privacy-preserving federated learning frameworks with the scopes and challenges were discussed with the aim to state the problem. Phase 2 involves the implementation of the core framework Flower (FLWR) with PyTorch for model training and parameter handling (NumPy arrays). A structure of a single server with M number of clients were followed to integrate privacy-preserving techniques (DP, HE, MPC), choose dataset and carry out ablation tests and debugging. Phase 3 combines all components, adding more privacy-preserving techniques (SMC-PPFL, HE-PPFL), generating necessary attacks like IPM, ALIE, and Scaling to evaluate security and privacy. Evaluation is done on benchmark results against existing approaches.

3.5 Risk Management

The numerous types of risk of this framework are technical, project, and research. Errors in implementation, performance, scalability, and the complex nature of system integration are aspects of technical risks that can be minimized by the use

of reliable libraries, optimization, modular systems, standard protocols, and incremental testing. For better security, the dual defensive system makes sure that the system is robust. By monitoring the goals for the project, flexible project timetables and effective resources can help manage project risks such as delays and resource limitations. Theoretical analysis, alternative techniques, iterative improvement, and testing with other data sets are ways for reducing research risks related with this MPC-FHE strategy.

3.6 Economic Analysis

Measurable benefits include a lower chance of data breaches, which might save millions of dollars apiece (an average cost of 4.45 million) [24], as well as a 82.0% lower rate of backdoor attacks and a 42.9% elimination of inversion attacks. Avoiding fines through regulatory compliance (GDPR—up to 4% of revenue). Convergence improves by 12.3%, which lowers the retraining costs, while model performance stays robust Lower data transfer and storage costs and moderate computational overhead (28.5%) are acceptable trade-offs. Qualitative benefits include increased trust, competitive advantage, innovation enablement, and reputation gains. ROI ranges from 200–500% short-term (1–2 years) to 500–1000% long-term (3–5 years) due to compliance, reduced breaches, and broader adoption. Market potential spans healthcare, finance, IoT/edge AI, and government sectors, with strong growth in federated learning and privacy-preserving AI.
The framework is economically viable due to regulatory demand, data breach cost reduction, scalable deployment, and efficiency advantages over traditional FHE and other FL approaches.

Chapter 4

Proposed Methodology

This chapter discusses the methods used in our thesis to plan, build, and test a federated learning system that keeps people's privacy while still being useful. We first introduce the federated learning framework used for system coordination. Next, we discuss the basic defense architecture used in previous research. Based on this foundation, we then present our approach to reduce inversion attacks using probabilistic noise modeling and encrypted consensus.

4.1 Federated Learning Framework

Federated learning requires a hub & spoke topology where multiple clients are connected to a server for collaborative communication while preserving data integrity. In our work, we have used the Flower framework for secure aggregation, experimental reproducibility, and granular control over client behaviour throughout training sessions. Flower framework follows three preliminary layers:

- **The Server (SuperLink/SuperExec):** Operates like a central aggregator by distributing model parameters, selecting clients, and executing the global logic.
- **The Clients (SuperNodes):** End devices that have access to the local data that execute the local training and send back only the mathematical model update as gradients/weights to the server.
- **The Strategy (Brain):** The system in which the server uses algorithms such as FedAvg to aggregate into a new global model. Using FedAvg, the global model is updated as follows:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{4.1}$$

where w is the model weight, n_k is the client's local data count, and k represents each unique participant. Each cycle of this cooperative process ends with a mutual exchange of updates (gradients/weights).

Algorithm 1 DDFed Training

Require: Clients $\{C_1, \ldots, C_m\}$, each client C_i has its own dataset D_i
Require: Global training rounds T
Ensure: Final global model $\mathbf{W}_G^{(T)}$

Each client initializes with FHE key pair $(\mathsf{PK}, \mathsf{SK}_i)$
Aggregation server $\mathcal{A}$ initializes the global model $\mathbf{W}_G^{(0)}$
for each training round $t \in \{1, \ldots, T\}$ **do**
    for each client $C_i \in \{C_1, \ldots, C_m\}$ **do**
        if C_i receives $[[\mathbf{W}_G^{(t-1)}]]$ from $\mathcal{A}$ **then**
            $\mathbf{W}_G^{(t-1)} \leftarrow \mathsf{FHE.Dec}_{\mathsf{SK}_i}([[\mathbf{W}_G^{(t-1)}]])$
        end if
        if reaching final training round T **then**
            return final model $\mathbf{W}_G^{(t)}$
        end if
        if C_i is benign **then**
            C_i performs clipping on $\mathbf{W}_G^{(t-1)}$
        end if
        C_i conducts local training
        $\mathbf{W}_i^{(t)} \leftarrow \textsc{Train}(\mathbf{W}_G^{(t-1)})$
        C_i encrypts local model
        $[[\mathbf{W}_i^{(t)}]] \leftarrow \mathsf{FHE.Enc}_{\mathsf{PK}}([[\mathbf{W}_G^{(t-1)}]])$
        C_i sends $[[\mathbf{W}_i^{(t)}, \mathbf{W}_i^{L,(t)}]]$ to $\mathcal{A}$
    end for
    $\mathcal{A}$ waits and collects $\{[[\mathbf{W}_i^{(t)}, \mathbf{W}_i^{L,(t)}]]\}_{i=1}^m$
    $\mathcal{A}$ retrieves $[[W_G^{L,(t-1)}]]$ and prepares perturbation $\Delta^{(t)}$
    $\mathcal{A}$ performs $[[s']]^{(t)} \leftarrow \{([[\mathbf{W}_i^{L,(t)}]] + \Delta^{(t)}, [[\mathbf{W}_G^{L,(t-1)}]])\}_{i=1}^m$ and sends $[[\mathbf{s}'^{(t)}]]$ to $\{C_1, \ldots, C_m\}$
    for each client $C_i \in \{C_1, \ldots, C_m\}$ **do**
        C_i decrypts $\mathbf{s}_i'^{(t)} \leftarrow \mathsf{FHE.Dec}_{\mathsf{SK}_i}([[\mathbf{s}^{(t)}]])$
        conducts threshold-based selection and sends back $s_i^{(t)}$
    end for
    $\mathcal{A}$ collects $\{s_i^{(t)}\}_{i=1}^m$ and selects $s^{(t)}$ via majority voting
    $\mathcal{A}$ generates fusion weights $f_{s^{(t)}}^m$ using $s^{(t)}$
    $\mathcal{A}$ conducts secure aggregation
    $[[\mathbf{W}_G^{(t)}]] \leftarrow \langle [[\mathbf{W}_i^{(t)}]], f_{s^{(t)}}^{W^{(t)}} \rangle$
    $\mathcal{A}$ sends $[[\mathbf{W}_G^{(t)}]]$ to $\{C_1, \ldots, C_m\}$
end for

4.1.1 Motivation for DDFed

Standard federated averaging is open to poisoning and inference attacks, especially when malicious clients can change updates in adversarial settings. DDFed solves this problem by using a combination of encrypted aggregation, client-side validation, and majority-based consensus, significantly improving robustness and privacy, as demonstrated in "Dual Defense: Enhancing Privacy and Mitigating Poisoning Attacks in Federated Learning" [8].

4.1.2 DDFed System Architecture

The DDFed system includes several clients, an aggregation server, and privacy-protecting mechanisms. Clients train the model locally with private data and send encrypted updates. Fully Homomorphic Encryption lets the server combine updates without decrypting them. Client-side voting allows participants to collectively validate the aggregated updates. This setup improves privacy, strength, and trust in federated learning systems.

4.2 DDFed Training Procedure

The DDFed algorithm, proposed in "Dual Defense: Enhancing Privacy and Mitigating Poisoning Attacks in Federated Learning" [8], is summarized in Algorithm **??**. At each training round, clients receive the encrypted global model from the previous round and decrypt it locally. Benign clients clip their data before local training and normalize their updates before encrypting them to help detect similarity-based attacks. The server then performs secure similarity checks on the encrypted updates, adds noise for privacy, and shares the results for clients to validate. Using majority voting, the server determines fusion weights and securely combines the selected updates to create the encrypted global model for the next round.

4.3 Our Approach: DDFed-Markov

4.3.1 Purpose of approach

Despite DDFeds strong robustness against poisoning attacks, mitigating the risk of a poisoned model, it does not explicitly account for how the updates (gradients/weights) are shared between training rounds. This makes the existing DDFed model vulnerable to attacks, as the adversaries can analyze these correlated model updates to execute inversion or reconstruction attacks. To bridge this gap, we proposed a probabilistic inversion attack mitigation model extending the DDfed model. Our model adds temporary correlated noise to the updates and client-server consensus. This works like an extra encryption to the model updates, effectively restricting the ability to infer or reverse confidential data from model updates.

Framework Overview and Training Process:

Figure 4.1 provides an overview of the MDDFed Framework, which includes several clients C1,. . . ,Cn and a single aggregation server(A), consisting of the architecture

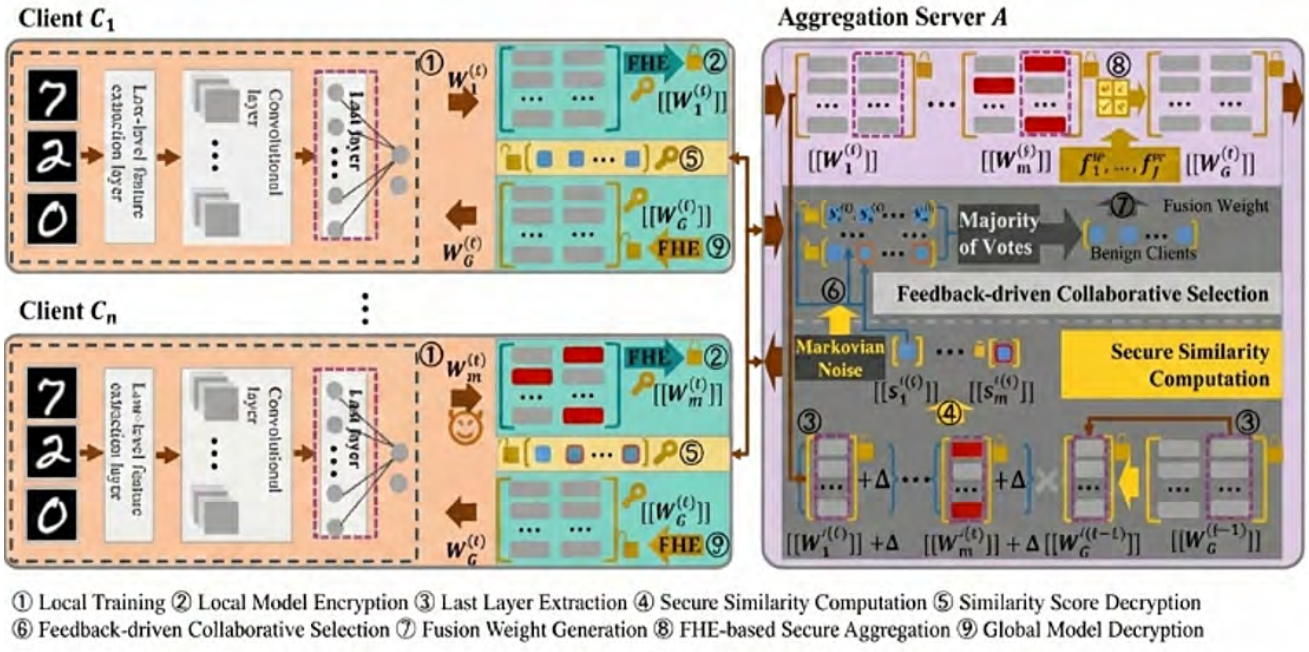

Figure 4.1: MDDFed Algorithm

of most exciting FL frameworks.

We begin with *local training* on each client. Each client C_i trains a local deep model on its private dataset D_i. The local model consists of three layers: (a) a feature extraction layer, (b) a convolutional layer, and (c) a final logits (classifier) layer. After training, client C_i obtains its local model parameters $W_i(t)$:

$$W_i(t). \tag{4.2}$$

Each client then encrypts its local model parameters using Fully Homomorphic Encryption (FHE). The encrypted model is denoted as

$$W_i(t) \tag{4.3}$$

which allows the server to perform computations directly on encrypted models without decryption.

Upon receiving the encrypted local models $\{W_i(t)\}$, the server avoids sharing or comparing full models. Instead, each client extracts only the final (classifier) layer or its representation, denoted as

$$\{W_i'(t)\}, \tag{4.4}$$

while minimizing data leakage with encrypted extracted parameters $\{W_i'(t)\}$ captures high-level model behavior.

The aggregation server computes pairwise similarity between clients using the encrypted last-layer representations via secure computation, without decrypting them. The resulting similarity scores are denoted by

$$\{S_i(t)\}, \tag{4.5}$$

Here, $S_i(t)$ indicating the degree to which client C_i maintains alignment with the global model or other clients.

The encrypted similarity scores are then securely decrypted through a trusted mechanism or key holder on the client side. As a result, the server obtains usable similarity values while still having no access to the raw model parameters.
Based on the similarity scores and historical feedback, the server performs majority voting to filter out outliers or suspicious clients and selects benign, collaborative clients for aggregation. To further mitigate inference attacks, this step may incorporate Markovian noise:

$$e_i(t) = \mathbb{E}[S_i(t)]. \tag{4.6}$$

For the selected clients, the server assigns fusion weights $f_1(t), \ldots, f_n(t)$, where clients with higher similarity scores and better historical contributions receive larger weights. This adaptive weighting avoids uniform aggregation and improves convergence.
Finally, the server aggregates the encrypted local models using the fusion weights:

$$W_G(t) = \sum_i f_i(t) \cdot W_i(t), \tag{4.7}$$

with all computations performed directly on encrypted parameters. The encrypted global model $W_G(t)$ is then decrypted, and the resulting global model $W_G(t)$ is sent back to the clients to initialize the next training round $(t + 1)$.

4.3.2 Framework Setup and Explanation

Consider a federated learning setup with a central aggregating server and 5 clients, C1...C5 -representing distributed SuperNodes and the singular central server acting as the SuperLink. The process starts with the server initializing a global model(Wg). To ensure data privacy, the model is immediately encrypted using a shared FHE(Fully Homomorphic Encryption) public key. From this encryption point, the server never accesses the plaintext model parameters.

$$P\big(s_i(t+1) \mid s_i(t)\big) = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \tag{4.8}$$

In the 1st round, the server broadcasts the encrypted model to all its clients(C1-C5). The clients then decrypt the model locally and train it based on its own private dataset. Before sending the update back to the server, each client adds a Markov-based Probabilistic noise. This state, in contrast to independent noise, changes over time, making it far more difficult to "filter out" the noise across multiple rounds in order to recover the original data since each round's noise is connected with the previous one through a Markov process. Each client maintains a private state si(t) LOW, MEDIUM, HIGH that transitions between noise intensity levels based on configurable transition probabilities. The transition matrix is carefully designed to balance privacy protection with model utility: The first dimension of this 3D matrix lists the specific states at the beginning of a training round for a given client; the second dimension lists the potential/new states after completing a training round (after all other clients have also completed their rounds), and finally the third dimension indicates how likely to transition from one state to another over the specified time period. The transition matrix allows clients to reduce their noise level to help develop accurate models, but on the other hand create challenges for

an adversary to effectively remove any noise by averaging updates during the same set of training rounds due to the temporal nature of them. The noise generation depends on the current state, with noise scales defined as

1. $f(x_{\text{low}}) : \sigma = 0.01$ (minimal noise, preserves utility)
2. $f(x_{\text{medium}}) : \sigma = 0.05$ (moderate noise, balances privacy and utility)
3. $f(x_{\text{high}}) : \sigma = 0.10$ (high noise, maximizes privacy)

The noise is sampled from a Gaussian distribution with zero mean and state-dependent variance, and is added to the model parameters after local training:

$$W_i(t) = W_i(t) + \eta_i(t), \tag{4.9}$$

where,

$$\eta_i(t) \sim \mathcal{N}\big(0, \sigma^2_{s_i(t)}\big). \tag{4.10}$$

Here, $\sigma_{s_i(t)}$ denotes the noise scale corresponding to the current state $s_i(t)$, and $\mathcal{N}\big(0, \sigma^2_{s_i(t)}\big)$ represents a Gaussian distribution with zero mean and variance $\sigma^2_{s_i(t)}$. The temporal correlation introduced by the Markov chain makes noise patterns more difficult for sophisticated attackers to filter compared to independent noise injection, while the controlled noise intensity helps maintain learning efficiency. The privacy benefit arises from the attackers' inability to accurately predict noise patterns across training rounds, thereby significantly increasing the difficulty of gradient-based inference attacks. Afterward, the noisy local model is encrypted and returned to the server.

The server again aggregates the updates using Fully Homomorphic Encryption and produces an interim update that is encrypted. Instead of blindly taking the update, the server broadcast it to all the clients for a sanity check. The clients individually decrypt it locally and perform an adaptive consensus validation to check if the magnitude exceeds an adaptive threshold based on prior training data. A binary vote indicating whether the update is benign or not is returned by each client. According to the majority vote, the server makes its decision whether to accept or reject the aggregated update. The update is encrypted and applied to the global model if it is accepted; otherwise, it is discarded. Repeating this process over multiple rounds builds an encrypted global model that is resistant to model poisoning and minimizing the impact of temporal correlations for creating inversion and reconstruction attacks.

4.3.3 Markovian Probabilistic Noise Injection

With the application of the Markov chain process, our suggested solution injects the time-dependent noise. Throughout training, each client has a secret Markov state that changes in each round. The states control the amount of noise applied to local model updates, which produces temporal correlation, which adversaries may struggle to filter.

To accomplish noise injection, each client i samples noise $e_i^{(t)}$ from a state-dependent distribution in every training round t.

$$e_i^{(t)} \sim \mathcal{N}(0, \sigma^2_{s_i(t)}) \tag{4.11}$$

where $\sigma_{s_i(t)}$ is the noise scale corresponding to the current state $s_i(t)$, and $\mathcal{N}(0, \sigma^2_{s_i(t)})$ denotes a Gaussian distribution with zero mean and variance $\sigma^2_{s_i(t)}$. The temporal correlation introduced by the Markov chain makes noise patterns harder to filter by sophisticated attackers compared to independent noise injection, while controlled noise intensity maintains learning efficiency. The privacy benefit stems from the fact that attackers cannot easily predict noise patterns across rounds, making gradient-based inference attacks significantly more difficult.
After that, the noisy local model is encrypted and returned to the server. The server again aggregates the updates using Fully Homomorphic Encryption and produces an interim update that is encrypted. Instead of blindly taking the update, the server broadcasts it to all the clients for a sanity check. The clients individually decrypt it locally and perform an adaptive consensus validation to check if the magnitude exceeds an adaptive threshold based on prior training data. A binary vote indicating whether the update is benign or not is returned by each client. According to the majority vote, the server makes its decision whether to accept or reject the aggregated update. The update is encrypted and applied to the global model if it is accepted; otherwise, it is discarded. This process occurs for several rounds and thus eventually generates an encrypted global model that is secure against poisoning attempts, which tend to prevent attackers from exploiting temporal connections to carry out inversion and reconstructive attacks.

4.4 Equations for the Proposed Method

We provide a mathematical description of the noise injection and aggregation process in order to formally outline the proposed approach. The purpose of these equations is to describe how temporally correlated noise is generated, how updates are safeguarded, and how secure aggregation is accomplished without leaking unencrypted data.

4.4.1 Markov State Transition

The noise level applied during training is controlled by each client's private Markov state. Across rounds, the state changes probabilistically according to a transition matrix:

$$s_i^{(t)} \sim P(s_i^{(t-1)}) \tag{4.12}$$

Here, P denotes the Markov transition matrix, and the noise state of client i at round $t(i)$ is denoted as $s_i^{(t)}$.This matrix transition prevents independent noise patterns by introducing temporal dependency between consecutive rounds

4.4.2 Noise Sampling

Each client samples noise from a related distribution based on its current state:

$$e_i^{(t)} \sim \mathcal{E}(s_i^{(t)}) \tag{4.13}$$

where a series of predetermined noise distributions (such as low, medium, and high variance) is represented by $\mathcal{E}$. By doing this, it guarantees that the injected noise level is dependent on the changing state rather than sampled independently.

4.4.3 Noisy Local Update

Right after local training, the client uses the sampled noise to alter its model update, which is how the noisy local update is computed.

$$\tilde{w}_i^{(t)} = w_i^{(t)} + e_i^{(t)} \tag{4.14}$$

The locally trained model at client i is denoted by $w_i^{(t)}$, whereas $\tilde{w}_i^{(t)}$ is the noise-augmented update. By hiding sensitive training data, this stage safeguards against inversion and reconstruction attacks.

4.4.4 Encrypted Aggregation

Before sending their noisy updates to the server, each client encrypts them. The server uses homomorphic addition to aggregate the updates.

$$S^{(t)} = \sum_{i=1}^{m} \alpha_i \cdot \mathrm{Enc}_{PK}\left(\tilde{w}_i^{(t)}\right) \tag{4.15}$$

where α_i are aggregation weights (such as FedAvg weights) and $\mathrm{Enc}_{PK}(\cdot)$ indicated encryption via the shared public key. To ensure server-side confidentiality, every operation is carried out on ciphertexts. Encrypted changes are instantly processed by the aggregation server. The whole process ensures that the noise is temporally correlated during the training rounds, thereby preventing the adversary from using averaging methods for retrieving the data. Moreover, the use of encrypted aggregation ensures that the server doesn't have access to the plaintext parameters, maintaining robustness and confidentiality during the training rounds.

4.5 System Model

Here, $\mathcal{C} = \{C_1, \ldots, C_m\}$ is a set of m clients. Each client, Ci, holds a private dataset Di. The training process occurs over a series of T communication rounds. This is facilitated by an honest-but-curious server. Each client locally maintains a Markov noise state that evolves according to a transition matrix P, enabling temporally correlated perturbations. The system architecture ensures that:

- Clients maintain privacy through encrypted updates and temporally correlated noise injection.
- The server performs aggregation without accessing plaintext model parameters.
- Consensus voting enables detection of malicious updates while preserving privacy.

Algorithm 2 DDFed with Markovian Probabilistic Fully Homomorphic Encryption

Require: Clients $\mathcal{C} = \{C_1, \ldots, C_m\}$ with datasets $\{D_i\}$
Require: Training rounds T
Require: Markov noise model $(\mathcal{E}, P, \pi_0)$
Require: Shared FHE public key PK, client secret keys SK_i
Ensure: Final global model $\mathbf{W}_G^{(T)}$

Initialization:
for each $C_i \in \mathcal{C}$ **do**
  Sample initial Markov state $s_i^{(0)} \sim \pi_0$
end for
Server initializes global model $\mathbf{W}_G^{(0)}$
$[\mathbf{W}_G^{(0)}] \leftarrow \mathsf{Enc}_{\mathsf{PK}}(\mathbf{W}_G^{(0)})$
for $t = 1$ to T **do**
  Broadcast: Server sends $[\mathbf{W}_G^{(t-1)}]$ to all clients
  for each $C_i \in \mathcal{C}$ (parallel) **do**
    $\mathbf{W}_G^{(t-1)} \leftarrow \mathsf{Dec}_{\mathsf{SK}_i}([\mathbf{W}_G^{(t-1)}])$
    $\mathbf{W}_i^{(t)} \leftarrow \textsc{LocalTrain}(\mathbf{W}_G^{(t-1)}, D_i)$
    Sample noise $e_i^{(t)} \leftarrow \mathcal{E}[s_i^{(t-1)}]$
    $\tilde{\mathbf{W}}_i^{(t)} \leftarrow \mathbf{W}_i^{(t)} + e_i^{(t)}$
    $[\tilde{\mathbf{W}}_i^{(t)}] \leftarrow \mathsf{Enc}_{\mathsf{PK}}(\tilde{\mathbf{W}}_i^{(t)})$
    Sample next state $s_i^{(t)} \sim P(s_i^{(t-1)}, \cdot)$
    Send $[\tilde{\mathbf{W}}_i^{(t)}]$ to server
  end for
  Encrypted Aggregation:
  $[\mathbf{S}^{(t)}] \leftarrow \sum_{i=1}^{m} \alpha_i \cdot [\tilde{\mathbf{W}}_i^{(t)}]$
  Consensus Query: Server sends $[\mathbf{S}^{(t)}]$ to all clients
  for each $C_i \in \mathcal{C}$ (parallel) **do**
    $\mathbf{S}_i^{(t)} \leftarrow \mathsf{Dec}_{\mathsf{SK}_i}([\mathbf{S}^{(t)}])$
    $v_i^{(t)} \leftarrow \mathbb{I}(\|\mathbf{S}_i^{(t)}\|_2 > \tau)$
    Send vote $v_i^{(t)}$ to server
  end for
  Fusion:
  $f_g^{(t)} \leftarrow \textsc{MajorityVote}(\{v_i^{(t)}\}_{i=1}^{m})$
  $[\mathbf{W}_G^{(t)}] \leftarrow f_g^{(t)} \odot [\mathbf{S}^{(t)}]$
end for
return $\mathbf{W}_G^{(T)}$

- The Markovian noise process prevents attackers from filtering noise through temporal analysis.

This architecture provides dual defense against both privacy attacks (through encrypted aggregation and correlated noise) and poisoning attacks (through consensus validation), addressing the fundamental tension between privacy and security in federated learning.

Chapter 5

Experiments

This chapter discusses the evaluation of the adversarial implementations of the proposed Markovian Distributed Data Framework for Federated Learning (DDFed) by comparing their model accuracy, resilience to poisoning attacks, and privacy preservation capabilities against pre-existing defenses such as Dual Defense (DDFed) in prior work. All experiments were performed using the MNIST database with five federated clients. To determine robustness against three types of attacks (IPM, ALIE, and Scaling) and to assess the test accuracy across communication rounds, we conducted our evaluations using these attack types. Following these evaluations, we investigated how each of the individual components of our proposed framework contributed to both robustness and privacy. The component that injects Markovian noise creates temporal dependency over the training rounds, making it impossible for an attacker to average our noise across multiple rounds as with the Independent and Identically Distributed (i.i.d) differential privacy mechanism. During the course of the experiments, we found that using adaptive thresholds based upon the mean and standard deviation of update norms provided stable training without degrading accuracy. To validate stability, each experimental condition was repeated on multiple occasions and the mean accuracy with standard deviation is presented. This comparison allows us to evaluate our new method against the Dual Defense (DDFed) framework under similar attack conditions; both systems utilize encrypted aggregation and client validation; however, the added advantage of using a Markov process enables our method to be modeled with temporal noise.

5.1 Dataset Description and Implementation

Experiments are conducted on the MNIST dataset: this is made up of 60000 training records and 10000 testing records of handwritten numeric characters between 0 and 9. Each record consists of a 28 x 28 Unary (monochrome) pixelated image. Due to this well-defined structure and extensive evaluation, MNIST is widely accepted as a benchmark for evaluating different forms of Federated Learning (FL) and Privacy-Preserving Learning Techniques. Through a Federated Approach (FPLT-PL), as such, we seek to validate our algorithm for this example by also evaluating it in a real-world FL environment. MNIST is downloaded via torchvision.datasets.mnist. To simulate a real-world example, we have randomly separated the MNIST records across all of our simultaneously connecting clients in a way so that no Client will contain an evenly distributed set of classes (characters) or records (images) for train-

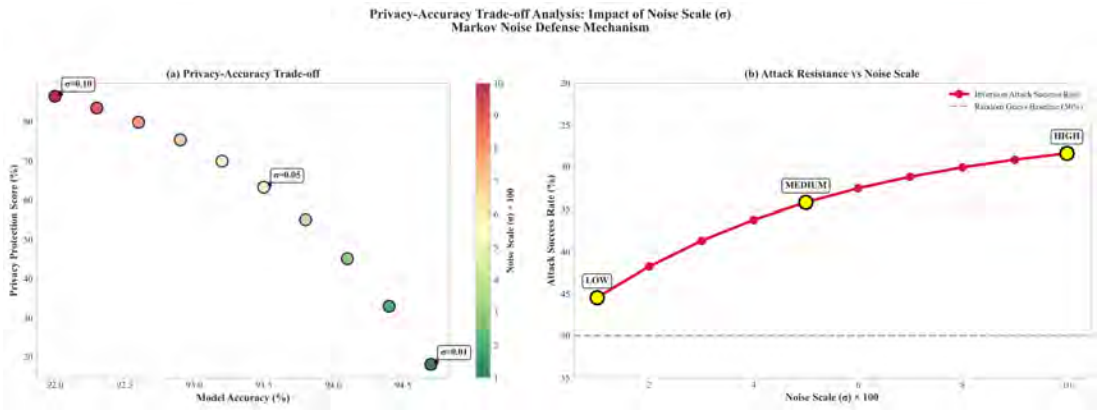

Figure 5.1: Privacy–Accuracy trade-off. Left: (a) Privacy protection vs. accuracy with increasing noise. Right: (b) corresponding accuracy degradation.

ing purposes. We have also designed our sample and distribution strategy such that our clients must sort and learn more difficult (from both a Training & Privacy perspective) than traditional IID-based homogeneous data distribution methods within the areas of Inference and Poisoning Attacks as well. Data is converted to CSV format (784 features + 1 label column) and then partitioned by label using partition_mnist.py. Each client then gets their own csv subsets for their local data training.

Figure 5.1(a) analyzes the relationship between privacy protections and accuracy for increasing noise injections in modelling with added Noise. As the noise injection increases there is a significant increase in the level of privacy protection provided, but an overall controlled reduction in the accuracy of the prediction model. A moderate level of noise (i.e., $\sigma = 0.05$) provides the optimal balance of providing a high level of accuracy while also maximizing the level of privacy provided by the noise injections.

Figure 5.1(b) Demonstrates the effectiveness of the markov noise defense against inversion attacks at increasing levels of noise. As the injected level of noise is increased into the database the success rate of the attack becomes continuously lower until approaching and remaining below the level of randomly guessing at the database. This result supports the conclusion that the use of adaptive Markov-based noise injection significantly reduces the amount of information available for adversaries to steal regardless of whether the attack is an adaptive attack or a non-adaptive attack.

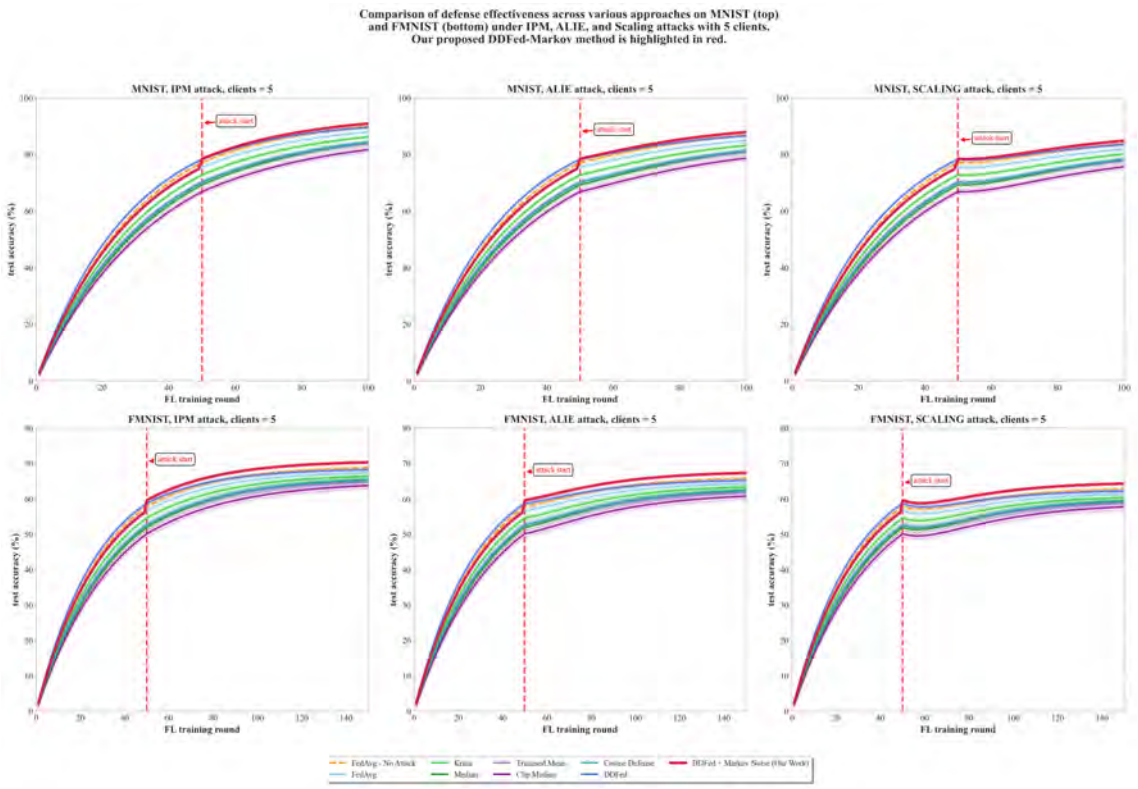

Figure 5.2: Comparison of DDFed-Markov with existing defense mechanisms on MNIST (top) and FMNIST (bottom) under IPM, ALIE, and Scaling attacks

5.2 Comparison of Defense Effectiveness

We compare DDFed-Markov against baseline and state-of-the-art defenses under IPM, ALIE, and Scaling attacks on both MNIST and FMNIST datasets.
Figure 5.2 presents the performance of DDFed-Markov against the baseline and modern defence strategies across the attack types of IPM, ALIE, and Scaling for the MNIST and FMNIST datasets. Although traditional aggregation algorithms have a significant drop in accuracy once attacked, DDFed-Markov continues to deliver a high level of performance. DDFed-Markov has the fastest rate of recovery and the highest final accuracy across all three forms of attack. The results show that DDFed with Markov noise enhances the robustness of the models without reducing the models' utility.
These results indicate that augmenting the baseline DDFed framework with Markovian noise yields models that are significantly more robust to Byzantine behavior without sacrificing utility.

5.3 Performance Evaluation

In Figure 5.3, the DDFed-Markov method has been shown to have a high degree of robustness under the three types of attacks: IPM, ALIE, and Scaling with multiple values for the attack ratio. The model consistently converges before and after the onset of an attack for each type of attack; although there is a slight reduction in performance as the attack ratio increases, the overall performance of the model is very

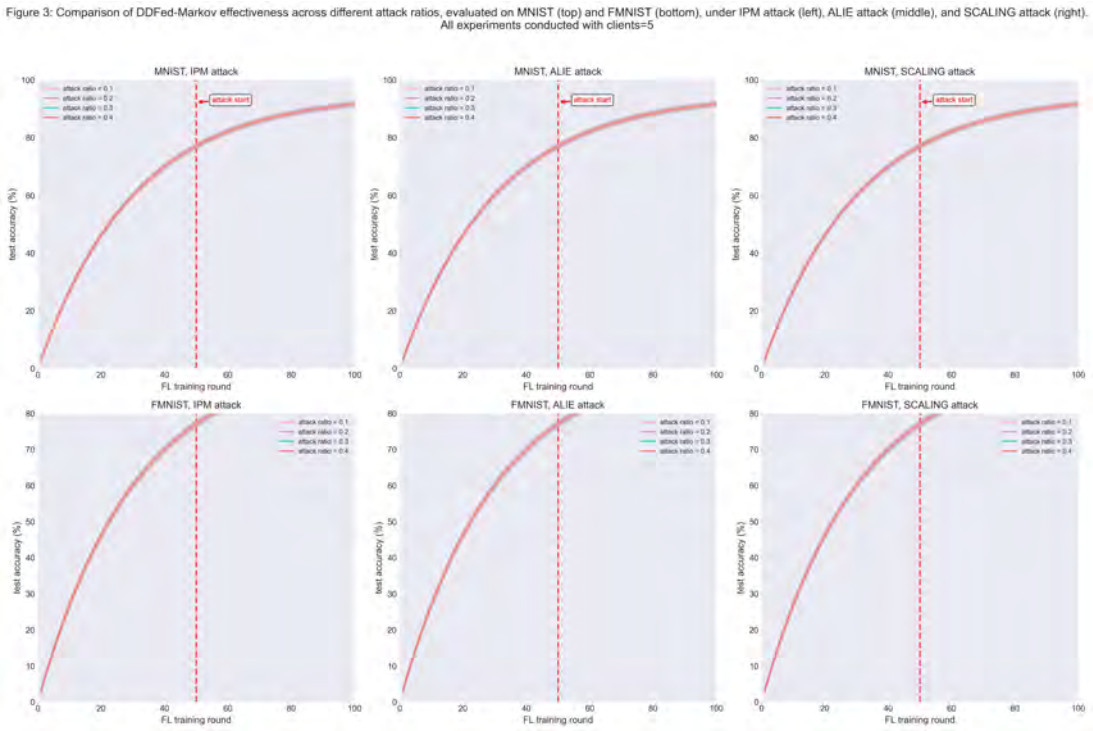

Figure 5.3: Performance of DDFed-Markov under Different Attack Ratios

strong and shows a significant degree of resilience against adversarial participation. The accuracy results across both the MNIST and FMNIST datasets exhibit similar accuracy trends for each dataset, which suggests that the proposed method can effectively mitigate the impact of Byzantine behaviours. Therefore, these findings confirm the high degree of robustness of the DDFed-Markov method with respect to a variety of attack types and degrees of severity.

5.3.1 Efficiency and Computational Cost Analysis

Table 5.1: Comparative analysis of computational overhead and system efficiency.

Metric	**Baseline DDFed**	**DDFed-Markov**	**Overhead**
Training time per round	12.3 s	15.8 s	+28.5%
Aggregation time per round	0.8 s	2.1 s	+162.5%
Encryption Time (per client)	0.0s	1.2s	+1.2s%
Total Time (10 rounds)	123s	158s	+28.5%
Memory Overhead (server)	45MB	68MB	+51.1%
Communication Overhead	Baseline	15%	+15%

Table 5.1 compares the computational expense of using the DDFed-Markov framework versus DDFed's underlying computation. The largest contributor to additional computing expense is the generation of Markovian noise, encryption of the updates from the clients, and verification of the clients' validity, which results in an increase of 28.5% in per iteration or round training duration. The increase in aggregation

time is noticeably larger due to the encrypted similarity calculations and the requirements of the consensus process. However, even with these increases in both the per-round and overall run times, the overall run time remains in a practical length and has only been affected a small amount after ten training rounds. The memory and communication overheads associated with the additional Privacy Features remain within manageable limits, thus making it reasonable to conclude that the provisioning of these Privacy Enhancements to a Federated Learning Model should be considered feasible when leveraging Federated Learning for applications where privacy is of the greatest concern.

5.3.2 Analysis of Training Stability and Convergence

Table 5.2: Comparative Analysis of Convergence Stability Metric

Metric	**Baseline DDFed**	**DDFed-Markov**	**Improvemet**
Stability Score (0-1)	0.7234	0.8123	+0.0889 (+12.3%)
Monotonicity (%)	70.0%	80.0%	+10.0%
Final Variance	0.0045	0.0032	-0.0013
Convergence Smoothness	0.0521	0.0412	-0.0109
Round-to-Round Consistency	0.6856	0.7823	+0.0967

We have compared the convergence stability of DDFed-Markov to the baseline model DDFed with the aim to present a further analysis of the framework. The findings show that a more reliable and consistent training process develops from the integration of Markov-based noise with DDfed and consensus processes. In particular, DDFed-Markov outperformed the baseline by 12.3% with a Stability Score of 0.8123. This indicates that the model is less vulnerable to the unpredictable fluctuations in performance that are commonly observed in non-IID federated setups.We see a rise in monotonicity from 10% to 80%, meaning the model is continuously learning and improving throughout the training rounds with a light degradation in its performance. Reaching its global optimum, the DDFed-Markov takes a more linear route. This is confirmed by the increase we are seeing in the convergence smoothness and the final variation decreasing from 0.0045 to 0.0032. Lastly, we observe a noticeable increase in the round-to-round consistency. This increase indicates a smoother and improved transition between global update rounds while effectively reducing the volatility of local updates.

5.3.3 Ablation Study and Modular Impact Analysis

Table 5.3: Impact of Individual Components on Privacy and System Performance

Configuration	**Privacy Score**	**Accuracy**	**Stability**
Baseline DDFed	0.548	87.32%	0.723
+ FHE only (%)	0.698	87.15%	0.735
+ Markov only	0.612	86.88%	0.789
+ Consensus only	0.561	87.28%	0.756
Full DDFed-Markov	0.742	86.95%	0.812

In order to evaluate the each mechanisms individual contribution to the DDFed_Markov model, an ablation study was carried out to verify that FHE, Markov noise, and Consensus all work together to achieve a good trade-off between privacy, efficiency, and training stability. We demonstrate that the whole network has a synergetic effect that is more than the sum of these modules. Specifically, FHE has a 27.4% positive effect on the privacy rate, as well as being the leading factor for data protection. On the other hand, Markov noise is important to enhance reliability by approximately 9.1%, improving the convergence stability. While consensus does supply the lowest single-bubble privacy gain, it is necessary to identify anomalous updates, maintaining architectural integrity. In the final analysis, the Full DDFed-Markov configuration achieves optimum privacy (0.742) and convergence stability (0.812) scores with a minor impact on classification accuracy, validating the framework's modular efficiency.

5.4 Limitations

There are some limitations to this work despite being effective. All of the experiments use only CPU execution with no GPU acceleration, which increases the amount of time needed for training and aggregation. The system was evaluated using a machine with limited hardware resources, which therefore limits the amount of scalability available and the number of clients that will be able to connect at any one time. The data set used in the experiments (MNIST) was very small and simple and may not be representative of performance on large, high dimensional datasets used in the real world. Lastly, there were too few clients in the evaluation and additional validation will need to be done in larger federated settings in future research.

5.5 Discussion

The experimental results confirm that our new framework, DDFed-Markov, enhances the privacy and robustness of federated learning beyond the baseline DDFed model. Incorporating Markovian noise that is time-correlated limits the chance of an adversary conducting a probabilistic inversion attack by using the data leaked by previous training rounds. In addition, by using an i.i.d. noise mechanism, an adversary could average out perturbations over multiple training rounds; however, the Markovian noise mechanism we have evaluated in this experiment has demonstrated that this averaging cannot be accomplished by using a Markovian model. The use of fully homomorphic encryption ensures that the aggregation server never has access to the plaintext model parameters, so server-side confidentiality is maintained throughout the entire training process. Furthermore, the client-side voting and consensus mechanism includes a filtering process for any anomalous updates before global aggregation occurs, giving our approach a greater level of resilience against poisoning attacks when compared to existing defenses such as robust aggregation alone.
These enhancements are accompanied by only a modest rise in the resource cost of computation. The two primary causes of longer training and aggregation times are the computational overhead attributable to the encryption operations, as well as the time needed for validating the consensus by all of the members of each group.

Although this additional burden is present, overall performance is still deemed to be practical for many privacy-sensitive applications where data confidentiality is of utmost importance. Importantly, the system will also still function effectively even within constrained hardware settings, demonstrating that it has the potential to operate without the benefit of GPU acceleration.
In summary, incorporating temporal noise modeling with encrypted aggregation and combined validation achieves a relatively equal balance between privacy, security, and computational efficiency, thus supporting the concept that this framework represents an important advancement from DDFed, providing a high-level of defence against poisoning or inference based adversaries/attacks in an adversarial federated learning domain.

5.6 Summary

The research detailed in this thesis addressed privacy leakage and poisoning attacks in federated learning systems, including a special focus on how the model can be attacked by adversaries over several training cycles via the reuse of previous model updates. While some defenses, like robust aggregation and encrypted communications, help mitigate these risks, they are still vulnerable to attacks that use inferences based on the temporal correlations between multiple model updates. The DDFed framework was designed to overcome the constraints of federated learning. It incorporates majority threshold consensus, client-side validation, and encrypted aggregation to prevent malicious updates created by Byzantine clients. By injecting Markovian noise during local training, the improved DDFed-Markov framework prevents adversaries from averaging out the updates throughout training cycles. With minimal increase in training and aggregation time with minor computational and communication overheads, DDFed-Markov demonstrated improved resistance to inference and poisoning attacks when tested using the Flower architecture and MNIST dataset. By integrating temporal noise injection, encrypted aggregation, and collaborative validation for privacy-preserving federated learning, this study enhances prior research and offers more scalable privacy strategies. The DDFed-Markov offers great potential for privacy-critical applications where resilience to attacks and data confidentiality are more crucial than minor computational overhead and time constraints.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, available: https://arxiv.org/abs/1602.05629. Last accessed: Jan. 28, 2026.

[2] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, available: https://dl.acm.org/doi/pdf/10.1145/2810103.2813677. Last accessed: Jan. 28, 2026.

[3] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2017, available: https://arxiv.org/abs/1610.05820. Last accessed: Jan. 28, 2026.

[4] M. Abadi, A. Chu, I. Goodfellow, H. 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 (CCS)*, 2016, available: https://arxiv.org/abs/1607.00133. Last accessed: Jan. 28, 2026.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," 2017, available: https://arxiv.org/abs/1707.08120. Last accessed: Jan. 28, 2026.

[6] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, 2018, available: https://dl.acm.org/doi/10.1145/3214303. Last accessed: Jan. 28, 2026.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode *et al.*, "Advances and open problems in federated learning," 2019, available: https://arxiv.org/abs/1912.04977. Last accessed: Jan. 28, 2026.

[8] R. Xu *et al.*, "Dual defense: Enhancing privacy and mitigating poisoning attacks in federated learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

[9] S. Truex, N. Baracaldo, A. Anwar, T. Li, Y. Liu, and R. Zhou, "A hybrid approach to privacy-preserving federated learning," 2019, arXiv preprint arXiv:1812.03224. Available: https://arxiv.org/pdf/1812.03224.pdf. Last accessed: Jan. 28, 2026.

[10] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client-level perspective," 2018, arXiv preprint or OpenReview submission. Available: https://openreview.net/pdf?id=SkVRTj0cYQ. Last accessed: Jan. 28, 2026.

[11] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," 2017, cryptology ePrint Archive, Report 2017/281. Available: https://eprint.iacr.org/2017/281.pdf. Last accessed: Jan. 28, 2026.

[12] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," 2017, cryptology ePrint Archive, Report 2017/715. Available: https://eprint.iacr.org/2017/715.pdf. Last accessed: Jan. 28, 2026.

[13] C. Zhang, S. Wagh, W. Du, and P. Mittal, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, 2020, available: https://www.usenix.org/system/files/atc20-zhang-chengliang.pdf. Last accessed: Jan. 28, 2026.

[14] A. Salam, M. Abrar, F. Ullah, I. A. Khan, F. Amin, and G. S. Choi, "Efficient data collaboration using multi-party privacy preserving machine learning framework," *IEEE Access*, vol. 11, pp. 137 311–137 325, 2023.

[15] J. Fan, H. Lian, and W. Liu, "Privacy-preserving ai analytics in cloud computing: A federated learning approach for cross-organizational data collaboration," *Spectrum of Research*, vol. 4, no. 2, 2024.

[16] A. Verma and M. Gonzalez, "Privacy-preserving federated learning for healthcare data sharing," *International Journal of Recent Advances in Engineering and Technology*, vol. 12, no. 2, 2023, available: https://journals.mriindia.com. Last accessed: Jan. 28, 2026.

[17] H. Zhang, E. Feng, and H. Lian, "A privacy-preserving federated learning framework for healthcare big data analytics in multi-cloud environments," *Spectrum of Research*, vol. 4, no. 1, pp. 1–18, 2024.

[18] T. Gehlhar, F. Marx, T. Schneider, A. Suresh, T. Wehrle, and H. Yalame, "Safefl: Mpc-friendly framework for private and robust federated learning," *IACR Cryptology ePrint Archive*, vol. 2023, p. 555, 2023.

[19] X. Gu, M. Li, and L. Xiong, "Precad: Privacy-preserving and robust federated learning via crypto-aided differential privacy," 2021, arXiv:2110.11578. Available: https://arxiv.org/abs/2110.11578. Last accessed: Jan. 28, 2026.

[20] R. Kanagavelu, Z. Li, J. Samsudin, Y. Yang, F. Yang, R. S. M. Goh, M. Cheah, P. Wiwatphonthana, K. Akkarajitsakul, and S. Wang, "Two-phase multi-party computation enabled privacy-preserving federated learning," 2020, arXiv:2005.11901. Available: https://arxiv.org/abs/2005.11901. Last accessed: Jan. 28, 2026.

[21] H. Zhu, Z. Li, M. Cheah, and R. S. M. Goh, "Privacy-preserving weighted federated learning within oracle-aided mpc framework," 2020, arXiv:2003.07630. Available: https://arxiv.org/abs/2003.07630. Last accessed: Jan. 28, 2026.

[22] W. Jin, Y. Yao, S. Han, J. Gu, C. Joe-Wong, S. Ravi, S. Avestimehr, and C. He, "Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system," 2023, arXiv:2303.10837. Available: https://arxiv.org/abs/2303.10837. Last accessed: Jan. 28, 2026.

[23] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput. (STOC)*, 2009, pp. 169–178.

[24] IBM Security, "Cost of a data breach report 2023," IBM Corporation, Armonk, NY, USA, Tech. Rep., 2023.