FEDERAL UNIVERSITY OF MINAS GERAIS
School of Engineering
Graduate Program in Electrical Engineering

Julia Zibetti Castilhos de Mellos

A PRIVACY PRESERVING APPROACH TO SHORT-TERM HOUSEHOLD LOAD FORECASTING WITH FEDERATED LEARNING AND DIFFERENTIAL PRIVACY

Belo Horizonte
2025

Julia Zibetti Castilhos de Mellos

A PRIVACY PRESERVING APPROACH TO SHORT-TERM HOUSEHOLD LOAD FORECASTING WITH FEDERATED LEARNING AND DIFFERENTIAL PRIVACY

Dissertation presented to the Graduate Program in Electrical Engineering of the School of Engineering at the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Advisor: Prof. Frederico Gadelha Guimarães
Co-advisor: Dr. Omid Orang

Belo Horizonte
2025

Mellos, Julia Zibetti Castilhos de.
M527p A privacy preserving approach to short-term household load forecasting with federated learning and differential privacy [recurso eletrônico] / Julia Zibetti Castilhos de Mellos. – 2025.
1 recurso online (91 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.
Coorientador: Omid Orang.

Dissertação (mestrado) – Universidade Federal de Minas Gerais, Escola de Engenharia.

Inclui bibliografia.

1. Engenharia elétrica – Teses. 2. Energia elétrica – Administração – Teses. 3. Instalações elétricas domiciliares – Teses. I. Guimarães, Frederico Gadelha. II. Orang, Omid. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo bibliotecário Marcio A. A. Gomes CRB/6 2812
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

Escola de Engenharia

COLEGIADO DO CURSO DE PÓS-GRADUAÇÃO EM Engenharia Elétrica

FOLHA DE APROVAÇÃO

"A Privacy Preserving Approach To Short-term Household Load Forecasting With Federated Learning And Differential Privacy"

Julia Zibetti Castilhos de Mellos

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 24 de novembro de 2025.

Por:

Prof. Dr. Frederico Gadelha Guimarães
DCC (UFMG) - Orientador

Dr. Omid Orang
(UFMG) - Coorientador

Prof. Dr. Cristiano Leite de Castro
DEE (UFMG)

Prof. Dr. Petrônio Cândido de Lima e Silva
(IFNMG)

Prof. Dr. André Luís da Costa Mendonça
(UNILAB)

Documento assinado eletronicamente por **Frederico Gadelha Guimaraes**, **Professor do Magistério Superior**, em 27/11/2025, às 00:25, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **André Luís da Costa Mendonça**, **Usuário Externo**, em 01/12/2025, às 13:27, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Omid Orang**, **Usuário Externo**, em 01/12/2025, às 14:26, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Petrônio Cândido de Lima e Silva**, **Usuário Externo**, em 22/12/2025, às 08:28, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Cristiano Leite de Castro**, **Professor do Magistério Superior**, em 29/12/2025, às 19:34, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4753577** e o código CRC **5DA769B9**.

Referência: Processo nº 23072.270821/2025-20 SEI nº 4753577

Eu dedico esse trabalho à minha mãe, que é a força, ao meu pai, que é o sonhar, e à minha irmã, que é poesia.

Acknowledgements

Agradeço ao meu orientador, Frederico Gadelha Guimarães, pela boa vontade e abertura desde o primeiro contato, pelo apoio e pela orientação impecável. Foi uma oportunidade enorme ser orientada por alguém tão relevante na área e sou muito grata por tudo que aprendi ao longo desse processo.

Agradeço também ao meu coorientador, Omid Orang, por ter aceitado embarcar nesta jornada e por todo o suporte ao longo do caminho.

Ter vocês dois ao meu lado fez toda a diferença. Não poderia ter imaginado uma dupla melhor para me guiar neste projeto. Muito obrigada aos dois pela confiança.

Aos professores André Luís da Costa Mendonça, Cristiano Leite de Castro e Petrônio Cândido de Lima e Silva, deixo meu sincero agradecimento por aceitarem compor a banca e dedicarem seu tempo a este trabalho. A presença e a experiência de vocês engrandecem esta pesquisa.

Ao meu grande parceiro de pesquisa, Lucas Malacarne Astore, minha companhia diária, seu apoio e amor me fizeram ir muito mais longe. Foi muito especial e importante estar tão sintonizada com alguém nesse mundo da academia.

Aos meus companheiros da pós-graduação, Bruna Falcucci, João Vitor Vaz e Ester de Lima, não consigo imaginar este caminho sem vocês. Desejo que nossa parceria dure uma vida inteira.

Aos amigos que fiz no FutureLab, nossas conversas, risadas, cafés e trocas tornaram essa jornada muito mais leve. Aos nossos graduandos, fico encantada em ver vocês trilhando um caminho tão brilhante. Sempre vou torcer por vocês e estou ansiosa para ver cada um indo cada vez mais longe.

Às pessoas que amo, muito obrigada por acreditarem em mim quando eu falhei em acreditar em mim mesma. Não estaria em pé sem vocês.

À minha mãe, que não só me deu a vida como fez de tudo para que eu vivesse saudável e feliz. Ao meu pai, que me incentivou a escolher essa vida acadêmica. E à minha irmã, que sempre mostrou que a vida pode ser poesia. Obrigada pelo apoio incondicional e sincero. Sei que minha felicidade é a felicidade de vocês e vice-versa. Este trabalho é dedicado a vocês, mesmo vocês não entendendo nada rsrs.

À minha vózinha que está no céu, minha fé é você. Obrigada por olhar por mim.

Aos meus amigos, se eu fosse citar todos vocês aqui, isso viraria outra dissertação.

Vocês sabem quem são. Obrigada mil vezes. Às minhas irmãs de coração, Anna Lins, Gabriela Czarnobay e Mariana Lemos, vocês são parte imensa desta jornada. Muito obrigada.

Por fim, fica aqui minha gratidão à educação pública de qualidade. Que lutemos para que ela seja cada vez mais inclusiva.

"We suffer from an incurable illness called hope."
Mahmoud Darwish (Palestinian poet)

Resumo

A previsão de carga de curto prazo constitui um componente fundamental dos sistemas modernos de gestão de energia, ao apoiar decisões operacionais críticas, mitigar riscos de interrupções e aprimorar a confiabilidade geral da rede elétrica. Nos últimos anos, os modelos de aprendizado profundo, como as redes *Long Short-Term Memory* (LSTM) e os *Transformers*, alcançaram resultados notáveis na melhoria da precisão das previsões. No entanto, sua aplicação em ambientes de aprendizado centralizado apresenta riscos significativos à privacidade, uma vez que os dados de consumo específicos de cada usuário precisam ser coletados e processados em um servidor central, potencialmente expondo informações sensíveis. Com o objetivo de enfrentar essas limitações, esta pesquisa propõe um arcabouço de previsão com preservação de privacidade, que integra o Aprendizado Federado (*Federated Learning* - FL, em inglês) com Privacidade Diferencial Local (*Local Differential Privacy* - LDP, em inglês) e o algoritmo de otimização FedProx, a fim de viabilizar uma previsão segura e robusta da demanda elétrica residencial. Na abordagem proposta, os clientes treinam seus modelos localmente de forma colaborativa, transmitindo ao servidor central apenas atualizações de modelo criptografadas e perturbadas, eliminando assim a necessidade de compartilhamento direto dos dados brutos. A incorporação da LDP garante a preservação da privacidade mesmo sob a suposição de um servidor não confiável, pois um ruído cuidadosamente calibrado é adicionado no nível do cliente. Paralelamente, o algoritmo FedProx aumenta a robustez do modelo contra ameaças à integridade – como envenenamento de dados (data poisoning) e manipulação de gradientes – ao melhorar a convergência em cenários de distribuição heterogênea entre os clientes. O método proposto é avaliado em dois conjuntos de dados de carga residencial de referência, sob três configurações distintas: aprendizado centralizado, aprendizado federado padrão e aprendizado federado com LDP. Os resultados experimentais demonstram que o arcabouço FL-LDP alcança um equilíbrio favorável entre privacidade e utilidade, mantendo desempenho competitivo de previsão enquanto reforça significativamente a confidencialidade dos dados, tanto em cenários de dados balanceados quanto desbalanceados.
Palavras-chave: previsão de carga; aprendizado federado; privacidade diferencial; FedProx.

Abstract

Short-term load forecasting (STLF) is a fundamental component of modern energy management systems, supporting critical operational decisions, mitigating outage risks, and enhancing overall grid reliability. In recent years, deep learning (DL) models, such as Long Short-Term Memory (LSTM) and Transformers, have achieved notable success in improving forecasting accuracy. However, their deployment in centralized learning environments poses substantial privacy risks, as user-specific consumption data must be collected and processed in a central server, potentially exposing sensitive information. To address these concerns, this research introduces a privacy-preserving forecasting framework that integrates Federated Learning (FL) with Local Differential Privacy (LDP) and FedProx optimization to enable secure and robust household electricity demand prediction. In the proposed approach, individual clients collaboratively train models locally, transmitting only encrypted and perturbed model updates to a central server, thereby eliminating the need for direct data sharing. The incorporation of LDP ensures that privacy is maintained even under the assumption of an untrusted server, as carefully calibrated noise is added at the client level. Meanwhile, the FedProx algorithm enhances model robustness against integrity threats such as data poisoning and gradient manipulation by improving convergence under heterogeneous client data distributions. The proposed method is evaluated using two benchmark residential load datasets under three distinct configurations: centralized learning, standard FL, and FL with LDP. Experimental results demonstrate that the proposed FL-LDP framework achieves a favorable privacy-utility trade-off, maintaining competitive forecasting performance while significantly enhancing data confidentiality across both balanced and imbalanced data scenarios.

Keywords: load forecasting; federated learning; differential privacy; FedProx.

List of Figures

Figure 1 – Overview of the Centralized Learning (CL) architecture for Short-Term Load Forecasting (STLF). (Source: author's own elaboration) 29

Figure 2 – Overview of the Federated Learning (FL) architecture for household STLF. The figure shows the server distributing the current global model to households, local training on private smart-meter data, and the return of model updates for aggregation. (Source: author's own elaboration). . 32

Figure 3 – Unbalanced clients in FedAvg versus FedProx. Example proportions: $\alpha_1 = 0.80$, $\alpha_2 = 0.15$, $\alpha_3 = 0.05$. Left: the weighted update Δw is driven by the largest client, shifting $w^{(t+1)}$ toward its distribution. Right: FedProx constrains local updates, yielding a smaller resultant Δw_{prox}. (Source: author's own elaboration) 32

Figure 4 – High-level overview of the proposed framework combining federated learning, local differential privacy, and FedProx. In each federated round: (1) the participating clients are selected (2) the global model is broadcast by the server, (3) clients train locally on private household data, calibrated noise is added to the FedProx constrained updates before transmission, (4) updates are sent back to the server, and (5) aggregation produces a refined global model. FL ensures data remains local, LDP protects the shared updates, and FedProx stabilizes convergence under heterogeneous conditions. 39

Figure 5 – Distribution of data records for the five clients in the UK-DALE dataset (2013-2017). 47

Figure 6 – Energy consumption (kWh) distributions for the four defined client groups from the HUE dataset. **(a)** Group 1 (5 clients, balanced), **(b)** Group 2 (5 clients, unbalanced), **(c)** Group 3 (7 clients, unbalanced), and **(d)** Group 4 (10 clients, unbalanced), highlighting heterogeneity and outlier presence. 59

Figure 7 – Mean evaluation loss per round for different values of μ under balanced and unbalanced client data distributions. 60

Figure 8 – Boxplots of energy consumption (kWh) by client. 62

Figure 9 – Effect of μ on global model MSE under poisoning. FedAvg ($\mu = 0$) becomes increasingly unstable after early rounds, while $\mu \in \{0.01, 0.1\}$ maintain stable convergence. 64
Figure 10 – Detailed Analysis of Client Sampling Fraction (ϕ) on the 28-Client Set. Plot (a) shows the overall model utility trend, and Plot (b) shows the robustness by tracking the client with the highest loss in each round. . 67
Figure 11 – UK-DALE FL LSTM convergence analysis. Plot (a) shows the aggregated convergence behavior; plot (b) highlights the per-client learning curves. 69
Figure 12 – Transformer Global Convergence on UK-DALE. The shaded band represents the standard deviation across runs. 70

List of Tables

Table 1 – Clients selected from the HUE dataset for balanced and unbalanced scenarios. 46
Table 2 – Model Architecture Hyperparameters 49
Table 3 – Common Training and Privacy Hyperparameters 55
Table 4 – LSTM baseline performance on the balanced HUE group (Residences 4, 8, 9, 10, 13). Mean over 10 runs. 57
Table 5 – Transformer baseline performance on the balanced HUE group (Residences 4, 8, 9, 10, 13). Mean over 10 runs. 58
Table 6 – Aggregate heterogeneity metrics for HUE client groups (means across clients in each group). 58
Table 7 – Heterogeneity metrics for the seven-residence unbalanced HUE group. Bold values indicate extreme heterogeneity (highest/lowest). 60
Table 8 – Per-residence MAE and RMSE for the unbalanced 7-client HUE group under different μ values. Relevant best (low) and worst (high) values are highlighted. 61
Table 9 – Heterogeneity metrics per UK-DALE client. 61
Table 10 – Per-client RMSE for UK-DALE under different μ values. 63
Table 11 – Mean L2 norms of client updates under poisoning for the unbalanced HUE dataset group (Residences 17, 21, 26, 27, 28). 64
Table 12 – Final Global Model Performance Metrics (Mean Only) for all HUE Client Scenarios 65
Table 13 – Final Global Model Performance Metrics (Mean Only) on 28-Client HUE Dataset 66
Table 14 – Final Global Model Performance Metrics (Mean Only) for the Transformer Architecture on HUE Dataset 68
Table 15 – UK-DALE FL LSTM: Final Performance Metrics 68
Table 16 – UK-DALE Transformer: Final Performance Metrics 70
Table 17 – FL-LDP-LSTM across client groups at different privacy levels ($\delta = 1 \times 10^{-6}$), alternative selection. 72
Table 18 – FL-LDP-LSTM at different privacy levels ($\delta = 1 \times 10^{-5}$) when training on subsets of 28 clients. 72

Table 19 – FL-LDP-LSTM on UK-DALE at different privacy levels ($\delta = 1 \times 10^{-6}$), averaged over five repetitions. 73
Table 20 – Performance of the FL–LDP–LSTM model at different privacy levels for each HUE client group in the prior publication. Values were averaged over five repetitions. 89
Table 21 – Performance of the FL–LDP–LSTM model at different privacy levels for the UK-DALE dataset in the prior publication. Values were averaged over five repetitions. 90

Abbreviations and Acronyms

AI Artificial Intelligence. 22

ANN Artificial Neural Network. 28

ARIMA Autoregressive Integrated Moving Average. 21, 25, 26, 28

CL Centralized Learning. 11, 21, 29

CNNs Convolutional Neural Networks. 27

DL Deep Learning. 21, 25, 27, 28, 30

DP Differential Privacy. 22–24, 28, 34–36, 38, 51

DTW Dynamic Time Warping. 51

EV Electric Vehicle. 20–22

FedAvg Federated Averaging. 22, 31, 33, 39, 42, 44, 73

FedProx Federated Proximal. 22–24, 31, 33, 37, 38, 42–45, 50, 53, 73

FL Federated Learning. 11, 22–24, 28, 30–36, 38, 43, 50, 51

GRU Gated Recurrent Unit. 27

HUE Hourly Usage of Energy dataset. 46

IID Independent and Identically Distributed. 22, 23, 36, 43

LDP Local Differential Privacy. 22–24, 28, 36, 38, 40, 41, 43, 44, 50

LLM Large Language Models. 21

LSTM Long Short-Term Memory. 21, 24, 27, 28

ML Machine Learning. 26, 27, 30

RNN Recurrent Neural Network. 27

STLF Short-Term Load Forecasting. 11, 20–22, 24–34, 36, 43

SVR Support Vector Regression. 21, 25, 26

TSF Time Series Forecasting. 34

UK-DALE UK Domestic Appliance-Level Electricity dataset. 47

xLSTM Extended Long Short-Term Memory. 27

ZNE Zero Net Emissions. 20

Contents

1 Introduction 20
1.1 Background and Context 20
1.2 Research Problem 21
1.3 Motivation and Relevance 22
1.4 Objectives 23
1.4.1 General Objective 23
1.4.2 Specific Objectives 23
1.5 Contributions 23
1.6 Structure of the Dissertation 24
2 Literature Review 25
2.1 Short-Term Load Forecasting 25
2.2 Classical models 25
2.3 Deep Learning Models 27
2.4 Hybrid Models 28
2.5 Centralized Approaches and Privacy Concerns 29
2.6 Federated Learning 30
2.6.1 Federated Learning Architecture 30
2.7 Federated Optimization: FedAvg vs FedProx 31
2.8 Adversarial and Gradient Inversion Attacks 33
2.9 Differential Privacy and Local Differential Privacy 34
2.10 Research Gaps and Problem Statement 36
3 Methodology 38
3.1 Framework Overview 38
3.2 Federated Learning Architecture 38
3.3 Local Differential Privacy Mechanism 40
3.3.1 Privacy Budget and Noise Calibration 40
3.3.2 Privacy-Utility Trade-off 41
3.4 FedProx Aggregation 42
3.5 Combined FL + LDP + FedProx Approach 43
3.5.1 Algorithm and Workflow 43
3.5.2 Design Decisions 44
4 Experimental Setup 46
4.1 Datasets 46
4.1.1 HUE Dataset 46
4.1.2 UK-DALE Dataset 47

4.2 Preprocessing and Feature Engineering . . . 47
4.3 Baseline Models . . . 48
4.3.1 LSTM . . . 48
4.3.2 Transformer . . . 49
4.3.3 Federated Approach . . . 49
4.3.4 Privacy-Enhanced FL Model . . . 50
4.4 Evaluation Metrics and Privacy Accounting . . . 50
4.4.1 Forecasting Error Metrics . . . 50
4.4.1.1 Mean Absolute Error (MAE) . . . 50
4.4.1.2 Root Mean Squared Error (RMSE) . . . 50
4.4.1.3 Symmetric Mean Absolute Percentage Error (sMAPE) . . . 51
4.4.1.4 Utility . . . 51
4.4.2 Fairness and Client Heterogeneity Metrics . . . 51
4.4.2.1 Per-Client Sample Size and Target Variance . . . 51
4.4.3 Distributional Heterogeneity Metrics . . . 51
4.4.3.1 Profile Distance . . . 51
4.4.3.2 Distribution Shift . . . 52
4.4.4 Gradient Metrics . . . 53
4.4.4.1 Gradient Drift . . . 53
4.4.5 Privacy Parameters . . . 53
4.5 Implementation Details . . . 53
4.5.1 Software and Core Libraries . . . 53
4.5.2 Hardware and Execution Environments . . . 54
4.5.3 Hyperparameter Configuration . . . 54
5 Results and Discussion . . . 56
5.1 Baseline Performance: LSTM and Transformer . . . 56
5.1.1 LSTM Baseline Results . . . 56
5.1.2 Transformer Baseline Results . . . 57
5.1.3 Discussion . . . 58
5.2 Effect of FedProx on Heterogeneous Clients . . . 58
5.2.1 Heterogeneity in STLF . . . 58
5.2.1.1 HUE Dataset . . . 58
5.2.1.1.1 Seven-client unbalanced group. . . . 60
5.2.1.2 UK-DALE Dataset . . . 61
5.2.2 FedProx under Poisoning Attacks . . . 63
5.3 Federated Results . . . 65
5.3.1 HUE dataset . . . 65
5.3.1.1 LSTM . . . 65
5.3.1.2 Impact of Client Sampling (ϕ) on the 28-Client Group . . . 66

5.3.1.3 Transformer . . . 68
5.3.2 UK-DALE . . . 68
5.3.2.1 LSTM . . . 68
5.3.2.2 Transformer . . . 70
5.4 Privacy-Utility Trade-off . . . 71
5.4.1 HUE Results . . . 71
5.4.2 UK-DALE Results . . . 73
5.5 Comparison to Prior Work and Privacy Regimes . . . 74
5.6 Overall Discussion of Findings . . . 74
6 Conclusions and Future Work . . . 76
6.1 Summary of Contributions and Main Findings . . . 76
6.2 Implications and Limitations . . . 77
6.3 Future Directions . . . 77
6.4 Publications Derived from this Research . . . 78

References . . . 80

Appendix A Reference Results from Prior Publication . . . 88
A.1 HUE Results from Prior Publication . . . 89
A.2 UK-DALE Results from Prior Publication . . . 89

Chapter 1

Introduction

1.1 Background and Context

Electrical energy is one of the most fundamental resources of modern society, supporting not only industrial production but also residential comfort and digital infrastructure. In recent decades, technological advancements, global population growth, rising living standards, and the expansion of digital services have all contributed to a steady increase in electricity demand worldwide. Although there is a growing shift towards renewable resources, fossil fuels still accounted for nearly 60% of global electricity generation in 2024 [International Energy Agency, 2025a]. In parallel, global commitments to Zero Net Emissions (ZNE) over the coming decades place increasing pressure on electricity systems to decarbonize rapidly. At the same time, the electricity sector remains one of the largest markets in the world, with annual investments exceeding USD 1 trillion [International Energy Agency, 2024]. This dual challenge creates a growing urgency for energy usage optimization, not only to reduce environmental impacts but also to enhance system reliability, economic competitiveness, and consumer welfare.

Over the past two decades, electricity networks have evolved into smart grids, supported by the introduction of smart meters, digital sensors, and automated control systems [International Energy Agency, 2025b, U.S. Department of Energy, 2025]. These technologies give operators near real-time information on consumption and grid conditions, which helps manage fluctuations in demand and integrate variable renewable sources. At the household level, many consumers now generate and store electricity through rooftop solar, batteries, and Electric Vehicles (EVs), which adds both flexibility and variability to the system [Manzoor et al., 2024b, Zhang et al., 2018]. This transformation produces large amounts of high-resolution consumption data that can improve planning and control, but also increases the difficulty of accurate and secure forecasting [International Energy Agency, 2023].

Short-Term Load Forecasting (STLF) models in power grid systems play a bene-

ficial role in supporting decision-making, reducing blackout risk, and enhancing supply reliability. The increase in data storage capacity, along with advances in measurement equipment technologies, has enabled the collection of large amounts of data, such as real-time energy consumption [Zhou et al., 2016]. This expansion of grid big data enables adaptive planning and operation at every level of the power system. However, it also amplifies the need for accurate and secure STLF, especially at the household level, where demand patterns have become increasingly volatile due to the combined effects of distributed generation, EVs, climate-dependent heating and cooling loads, and evolving consumer behaviors [Zhou et al., 2016, International Energy Agency, 2025b, U.S. Department of Energy, 2025].

Earlier approaches to STLF relied mainly on statistical models such as Autoregressive Integrated Moving Average (ARIMA) and exponential smoothing, as well as classical machine learning techniques such as Support Vector Regression (SVR) and decision trees [Hippert et al., 2001, Manzoor et al., 2024b]. However, these methods struggle to capture nonlinear relationships and long-term temporal dependencies, which has motivated the development of Deep Learning (DL)-based approaches. Recent models such as Long Short-Term Memory (LSTM) networks [Inala et al., 2025], Transformers [Kampezidou and Grijalva, 2016], and Large Language Modelss (LLMs) [L'Heureux et al., 2022, Orang et al., 2025] have therefore been applied to electricity demand forecasting.

1.2 Research Problem

Traditionally, most predictive models are built upon a Centralized Learning (CL) strategy, where a central server hosts the model and aggregates training data from multiple sources. This approach poses significant privacy risks and is less robust against various model attacks, such as membership inference, data reconstruction, and data injection.

In the context of load forecasting, sensitive information such as household occupancy, daily routines, and appliance ownership can be inferred from smart meter traces, enabling the reconstruction of socio-demographic characteristics and behavioral routines [Zoha et al., 2012, Beckel et al., 2014]. High-frequency smart meter data may reveal even more intimate details, such as activities of daily living relevant to health monitoring [Devlin and Hayes, 2019], or media consumption patterns (e.g., which television program is being watched) [Greveler et al., 2012].

Although regulators might already have some access to household consumption, smart meters can capture high-resolution data, while utilities and regulators do not necessarily collect or store this level of detail, as their operational needs are often limited to aggregated data for billing or tariff purposes [McKenna et al., 2012]. Moreover, with the growing adoption of distributed energy resources such as rooftop solar, batteries, and EVs, a portion of household energy flows may not be visible to utilities at all.

Despite enhancing model protection, Federated Learning (FL) is still vulnerable as attacks on the central server can be manipulated, as seen in the gradient inversion attack [Meijer et al., 2025a]. Differential Privacy (DP) and its local variant (Local Differential Privacy (LDP)) have become key tools in privacy-preserving machine learning [Dwork, 2006], enabling individual clients to perturb their sensitive local dataset. This ensures that, even in adversarial environments, no party can confidently infer sensitive user-level patterns, while also reducing the risk of appliance- and activity-level inference. While DP does not fully conceal macro-level patterns such as prolonged absence or overall consumption trends, these risks are already mitigated by the FL setup. DP therefore primarily strengthens protection against micro-level signatures that enable fine-grained inference (e.g., appliance ownership or media consumption), without entirely sacrificing forecasting performance.

Another important limitation in the context of household load forecasting is the non- Independent and Identically Distributed (IID) distribution of data across clients. Each household presents distinct demand profiles determined by appliance ownership, occupancy behavior, weather conditions, and the integration of distributed energy resources. This statistical heterogeneity leads to client updates that diverge from each other, reducing model accuracy and sometimes preventing convergence when using standard aggregation strategies such as Federated Averaging (FedAvg). To mitigate this issue, Federated Proximal (FedProx) modifies the local training objective by adding a proximal term that constrains local updates, thereby improving stability under heterogeneous conditions [Li et al., 2020b].

1.3 Motivation and Relevance

Short-term load forecasting (STLF) is widely recognized as a cornerstone of modern power system operation. As highlighted by Qiu et al. (2025), STLF "directly impacts the economic dispatch of power systems, the accommodation of renewable energy, and the efficiency of demand-side response, playing a crucial role in ensuring the stable operation of the entire power system" [Qiu et al., 2025]. Accurate short-term forecasts support critical decision-making in economic dispatch, power plant operations, grid balancing, and managing operating reserves, which are essential for reliability and market efficiency [Akhtar et al., 2023]. With the growing penetration of distributed energy resources such as rooftop solar, batteries, and electric vehicles (EVs), the need for accurate household-level forecasting has become even more pressing [Manzoor et al., 2024b, Zhou et al., 2016].

While detailed smart meter data enables this efficient management, centralizing such information for Artificial Intelligence (AI)-based forecasting raises significant concerns for consumer trust and data governance [Ramirez et al., 2024, Giaconi et al., 2020]. Given both regulatory requirements and consumer expectations, privacy-preserving forecasting approaches have increasingly become a requirement for the sustainable deployment of

smart grid analytics.

Developing FL strategies enhanced with DP mechanisms directly addresses this critical need in the context of smart grids, where both data privacy and forecasting accuracy must be balanced. A federated approach enables forecasting models to be trained directly on local data, leveraging its predictive value without requiring central collection of sensitive household records, while LDP provides an additional layer of protection by perturbing local updates before aggregation. To ensure these models remain effective under the heterogeneous conditions of real households, aggregation methods such as FedProx further enhance stability and robustness. Together, these elements directly address the growing demand for accurate yet trustworthy forecasting in modern electricity systems.

Given these motivations, the following objectives guide the development and evaluation of this dissertation.

1.4 Objectives

1.4.1 General Objective

The general objective of this dissertation is to develop and evaluate a privacy-preserving framework for short-term residential load forecasting that balances predictive accuracy with strong protection of user data.

1.4.2 Specific Objectives

To achieve this, the specific objectives are:

- Propose a federated learning (FL) model for household short-term load forecasting, enabling sensitive data to remain at the client side without central collection.
- Incorporate differential privacy (DP), in particular its local variant (LDP), to strengthen privacy guarantees and reduce reliance on a fully trusted central server.
- Address the non-IID nature of household consumption data by adopting FedProx as an aggregation method to improve stability and robustness against integrity attacks such as poisoning.
- Evaluate the proposed framework in comparison to baseline scenarios – centralized learning, FL without privacy, and FL with privacy – using the HUE [Makonin, 2019] and UK-DALE [Hebrail and Berard, 2006] datasets.

1.5 Contributions

The main expected contributions of this dissertation are:

- A novel privacy-preserving framework for residential STLF, which combines FL with LDP and addresses the challenges of data heterogeneity through FedProx.
- An approach that integrates local DP into federated forecasting without prohibitive degradation of accuracy, thereby providing stronger user-level privacy guarantees than standard FL.
- A comprehensive experimental evaluation, using the HUE and UK-DALE datasets, to assess forecasting accuracy, privacy guarantees, and robustness against adversarial threats.
- A comparative analysis of centralized, FL-only, and FL+DP scenarios, highlighting the privacy-utility trade-off and providing insights for practical deployment in smart grids.

1.6 Structure of the Dissertation

The remainder of this dissertation is organized as follows.

- Chapter 2 reviews the state of the art in short-term load forecasting, covering classical statistical methods, machine learning approaches, and recent advances in federated learning and privacy-preserving techniques. It concludes with the identification of research gaps and the formal statement of the research problem and hypotheses.
- Chapter 3 details the proposed methodology, including the federated learning architecture, local differential privacy mechanisms, and the use of FedProx as an aggregation method to handle data heterogeneity.
- Chapter 4 describes the experimental setup, including the datasets, preprocessing pipeline, centralized and federated baselines, privacy-enhanced models, evaluation metrics, and implementation details.
- Chapter 5 presents the results and discussion, including comparisons between centralized and federated baselines, the effect of FedProx on heterogeneous clients, the privacy-utility trade-off, and the evaluation of advanced architectures such as LSTM and Transformers.
- Chapter 6 concludes the dissertation, summarizing the main findings and contributions, discussing implications and limitations, and outlining directions for future research.

Chapter 2

Literature Review and Problem Formulation

2.1 Short-Term Load Forecasting

Short-Term Load Forecasting (STLF) is a class of time series prediction applications widely explored, with existing models ranging from classical statistical methods such as autoregressive integrated moving average (ARIMA) to modern DL approaches [Hong and Wang, 2022] using CNNs, Transformers, and even xLSTM [L'Heureux et al., 2022]. A general classification of those models is presented in Kuster et al. [2017], categorizing the models into time series analysis, artificial neural networks (ANN), and support vector regression (SVR).

The household STLF task can be formulated as a univariate time series prediction problem to estimate future electricity consumption over a short horizon from historical load measurements:

$$\{L_{t-k}, L_{t-k+1}, \ldots, L_t\} \rightarrow \hat{L}_{t+1} \tag{2.1}$$

where $L_t \in \mathbb{R}$ denotes the recorded power demand at time step t, and $\hat{L}_{t+1}$ is the predicted load for the next time step. The measurements are typically collected at regular intervals (e.g., hourly or every 15 minutes) via smart meters installed in each residential unit.

2.2 Classical models

Statistical models were among the earliest approaches to short-term load forecasting (STLF), relying on historical load data and assuming that future consumption patterns could be explained by linear dependencies, seasonality, and stochastic processes. These methods dominated before the rise of machine learning and deep learning due to their simplicity, interpretability, and relatively low computational requirements.

In the context of STLF, ARIMA was one of the first widely applied models. The model was first introduced by Box and Jenkins [1970] in their seminal work on time series forecasting. ARIMA combines autoregression (AR), differencing (I), and moving average (MA) terms to capture temporal dependencies and trends in time series data. An early example is the work of Hagan and Behr [1987], who applied ARIMA to electricity load forecasting and compared it to neural networks, showing competitive performance at the time. Seasonal extensions such as SARIMA have also been extensively used to account for daily and weekly seasonality in load series [Taylor, 2003]. However, ARIMA-type models face important limitations since they rely on linearity assumptions, which restrict their ability to capture complex demand patterns, and they struggle to incorporate multiple high-dimensional exogenous variables such as weather and calendar effects.

Other statistical approaches have also been applied to STLF, such as exponential smoothing methods, that generate forecasts by updating level, trend, and seasonal components with exponentially decaying weights [Taylor, 2003], and regression-based models, that, unlike the ARIMA model, allow explicit incorporation of exogenous variables such as temperature and holidays, providing straightforward interpretability [Papalexopoulos and Hesterberg, 1990]. ARIMAX and transfer function models extend ARIMA by also integrating these exogenous variables [Cho et al., 1995, Alfares and Nazeeruddin, 2002].

The reliance on linearity assumptions makes classical statistical models struggle when dealing with highly nonlinear and non-stationary data. Incorporating multiple exogenous variables is also challenging and often requires ad hoc formulations.

Although nonlinear extensions exist, such as TAR/STAR, GARCH, and nonlinear Kalman filters (EKF, UKF) [Tong, 1990, Hamilton, 1994, Engle, 1982, Julier and Uhlmann, 1997, Wan and Van Der Merwe, 2000], they are rarely used in STLF. Due to estimation complexity, limited ability to handle multi-variable data and small performance gains, they were soon overshadowed by Machine Learning (ML) approaches.

With the advance of ML, new approaches were introduced to overcome some of the limitations of purely statistical models. Techniques such as support vector regression (SVR) and decision trees demonstrated improved flexibility, particularly when incorporating exogenous variables like temperature or calendar effects. For instance, Mohamed and Kamel [2005] applied SVR to short-term load forecasting and reported gains over linear regression by better capturing nonlinear load-temperature relationships. Similarly, Chen et al. [2010] employed decision tree models, highlighting their interpretability and ability to handle categorical features such as weekdays and holidays. Ensemble variants such as gradient boosted trees have also been explored [Wang et al., 2019], showing competitive accuracy in both residential and system-level forecasting tasks.

Compared to statistical models, however, ML methods generally require larger and higher-quality datasets to achieve competitive performance, which remains a challenge in household-level load forecasting due to data scarcity. This limitation has also motivated

the exploration of hybrid models that combine statistical structure with ML flexibility (see Section 2.4).

While classical approaches offer stronger interpretability and relatively simple implementations, they remain constrained in capturing long-term temporal dependencies and highly nonlinear patterns. These limitations motivated the later shift toward deep learning architectures such as LSTMs and Transformers.

2.3 Deep Learning Models

DL techniques have emerged as powerful tools for time series forecasting due to their ability to learn complex nonlinear relationships, automatically extract hierarchical features, and capture long-range temporal dependencies. Unlike classical models that rely on predefined structures or handcrafted features, DL architectures can adapt to diverse data patterns and scale effectively with high-dimensional inputs such as weather variables, calendar effects, and household consumption data.

Because of these properties, Recurrent Neural Network (RNN) and their variants, particularly LSTM and Gated Recurrent Unit (GRU), have been widely applied to electricity load forecasting [Marino et al., 2016, Shi et al., 2017]. These architectures preserve information across time steps through hidden states, making them especially effective at capturing seasonality and short-term demand cycles such as daily or weekly patterns. At the household level, LSTM models have consistently outperformed classical baselines when sufficient high-resolution smart meter data are available [Kong et al., 2017].

Convolutional Neural Networks (CNNs) have also been adapted for load forecasting by treating consumption series as structured temporal signals. They can extract local motifs, that is, short recurring temporal patterns within a limited time window, such as morning and evening demand peaks, transient spikes caused by high power appliances, or rapid load ramps associated with device switching, which improves robustness when combined with recurrent layers [Marino et al., 2016, Shi et al., 2017].

More recently, attention-based models such as Transformers have been introduced to STLF, motivated by their ability to capture long-range dependencies without recurrent connections [Zhou et al., 2021, L'Heureux et al., 2022]. By leveraging self-attention, these models can dynamically assign weights to relevant past time steps, which is particularly advantageous for non-stationary load series where relevant historical events may occur at irregular lags. Emerging variants such as the extended LSTM (Extended Long Short-Term Memory (xLSTM)) combine recurrent structure with attention-like mechanisms, further improving the modeling of long temporal contexts [Inala et al., 2025].

However, DL methods typically require large-scale, high-quality datasets and significant computational resources. They are also prone to overfitting and often require regularization strategies such as dropout, early stopping, or data augmentation. In the con-

text of STLF, the scarcity of high-resolution household-level datasets remains a challenge [Marino et al., 2016, Shi et al., 2017, Kong et al., 2017], limiting the effective deployment of complex deep models. Additionally, the fine-grained data required by these architectures intensifies privacy concerns, and overfitting can further expose clients, especially in unbalanced settings where households with larger datasets exert disproportionate influence on the global model.

To address the data requirements, hybrid approaches that combine statistical models with DL have been explored, see Section 2.4. At the same time, mitigating risks of client exposure requires privacy-preserving methods, what has motivated the exploration of decentralized approaches such as FL and privacy-preserving mechanisms including DP and LDP, which are discussed in Sections 2.6–2.9.

2.4 Hybrid Models

Hybrid approaches have emerged in STLF as a way to combine the strengths of different modeling paradigms. The motivation is twofold: to increase forecasting accuracy and interpretability while addressing the data and modeling limitations faced by individual methods.

One line of work integrates statistical models with machine learning. For example, ARIMA models are often combined with Artificial Neural Network (ANN)s, where ARIMA captures linear and seasonal components while the ANN learns nonlinear residuals [Zhang, 2003, Hippert et al., 2001]. Similar ideas include ARIMA-SVR models that blend autoregressive structure with nonlinear regression capabilities [Nie et al., 2012]. These approaches improve flexibility while retaining interpretability from the statistical side.

Another line of research explores hybrid DL architectures. CNN-LSTM models are particularly popular: convolutional layers extract local motifs such as peak shapes or appliance patterns, while LSTM layers model long-term temporal dependencies. This combination enables the system to capture both short- and long-range structures in household load data, yielding improved accuracy over standalone CNNs or LSTMs [Wang et al., 2024, Iqbal et al., 2024].

Despite these advantages, hybrid models typically require larger and higher-quality datasets than purely statistical models, as well as greater computational resources. Their implementation is also more complex, which can hinder adoption in practice.

Still, regardless of architecture, classical, DL or hybrid, most models in the literature are trained under centralized learning, where raw household data are pooled in a single server. This reliance on centralization raises significant privacy concerns, motivating the decentralized approaches discussed in the following sections.

2.5 Centralized Approaches and Privacy Concerns

Traditional STLF have predominantly relied on CL, in which raw household data are collected and aggregated at a central server for model training. While this strategy provides straightforward implementation and has been widely adopted in early STLF research [Manzoor et al., 2024a, Akhtar et al., 2023], it raises significant privacy concerns. Smart meter traces can reveal occupancy routines, appliance ownership, or even media consumption patterns [Beckel et al., 2014, Greveler et al., 2012], enabling intrusive inferences about household behavior. Surveys such as [Giaconi et al., 2020] highlight that these risks undermine user trust and complicate large-scale deployment.

Figure 1 illustrates this centralized architecture: raw household data are transmitted to a central server, which performs model training on the aggregated dataset.

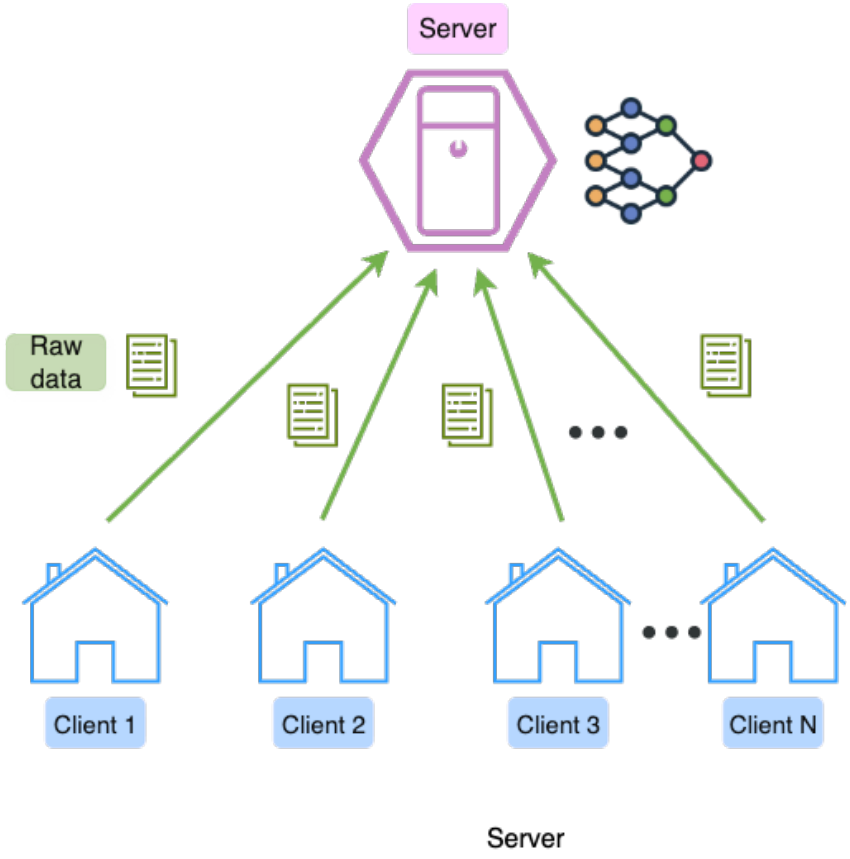


Figure 1 – Overview of the CL architecture for STLF. (Source: author's own elaboration)

While utilities and regulators may have access to some consumption information, this is usually limited to low-frequency billing data (e.g., monthly totals) or aggregated neighborhood-level measures [McKenna et al., 2013]. This is fundamentally different from high-resolution household traces sampled every few seconds or minutes. Furthermore, in scenarios with distributed energy resources such as solar panels or batteries, even regulators may only observe net-metered values rather than complete consumption profiles. These differences highlight that detailed smart meter data are not essential for regulatory or

billing tasks, yet their centralization for forecasting exposes households to unnecessary privacy risks.

The increasing availability of high-resolution smart meter data has amplified concerns regarding consumer data governance. These issues are further reinforced by the emergence of strict data protection regulations such as GDPR in Europe, LGPD in Brazil, and PIPL in China, which restrict the centralization of personally identifiable energy data.

Given these limitations, centralized solutions are increasingly viewed as impractical for household-level forecasting, motivating decentralized alternatives such as FL. In response, various protection techniques have been proposed and integrated into model design to enhance resilience against such threats.

2.6 Federated Learning

As discussed in Section 2.3, DL methods typically require large volumes of high-quality data to achieve competitive performance. In practice, however, the availability of such datasets is often constrained. Competitive pressures and strict regulatory frameworks related to data privacy and security lead to data fragmentation and isolation, limiting the effectiveness of centralized training.

To address this problem, McMahan et al. [2017] introduced FL, a decentralized training paradigm that enables multiple independent clients to collaboratively train ML models without sharing their raw data. In this way, FL allows participants to benefit from one another's data while keeping individual datasets private. Models trained with FL typically achieve higher predictive accuracy than those trained in isolation, while reducing the privacy risks inherent to centralized learning.

The FL approach, as discussed in [Manzoor et al., 2024a], is particularly well-suited for load forecasting tasks, given the naturally distributed nature of time series data collected from individual households. In [Briggs et al., 2022], FL was applied to enhance privacy in the context of STLF, demonstrating the suitability of this architecture for the application. However, their design did not consider potential vulnerabilities at the server level.

2.6.1 Federated Learning Architecture

In a federated setting, a central server coordinates the training process while data remain stored locally on each client. At the beginning of each communication round, a subset of clients is selected to participate. The server broadcasts the current global model parameters to these clients, which then initialize their local models accordingly. Each client performs local training on its own smart meter data for a fixed number of epochs, producing updated model parameters or gradients. These local updates are sent back to

the server, where they are aggregated to form a new global model. The cycle is repeated for a pre-defined number of rounds or until convergence criteria are met.

The most widely used aggregation strategy is FedAvg, where the server computes a weighted average of client updates, with weights typically proportional to the number of local samples [McMahan et al., 2017]. FedAvg implicitly assumes that client data are independent and identically distributed (IID), so that averaging provides an unbiased approximation of centralized training. In practice, however, household load data are highly heterogeneous. Consumption patterns vary significantly across homes due to differences in appliances, occupancy schedules, and distributed energy resources, resulting in non-IID conditions. Under such heterogeneity, FedAvg can suffer from unstable convergence or reduced global accuracy, since averaging may dilute or even conflict with client-specific patterns.

To mitigate these challenges, extensions such as FedProx have been proposed. FedProx introduces an additional regularization term in the local objective, known as the proximal term, that penalizes large deviations between a client's local model parameters and the current global model, constraining local updates to remain closer to the global model. This regularization improves robustness when client distributions diverge, stabilizing training and reducing the risk of drift [Li et al., 2020a]. The formal optimization objective and mathematical formulation of FedProx are presented in Section 3.4. These considerations are particularly relevant for household STLF, where client variability is inherent and simple averaging alone is insufficient to guarantee effective collaboration.

The architecture of a federated environment for household STLF is illustrated in Figure 2. In this setup, the central server broadcasts the current global model to selected clients, each household trains a local model on private smart-meter data, and only model updates are returned to the server for aggregation using methods such as FedAvg or FedProx. Importantly, no raw consumption data ever leaves the households.

2.7 Federated Optimization: FedAvg vs FedProx

In addition to privacy and server-side vulnerabilities, prior work has also highlighted the challenge of client data heterogeneity in FL, particularly under non-IID and unbalanced conditions [McMahan et al., 2017, Briggs et al., 2022]. Li et al. [2020a] showed that FedProx achieves more stable convergence and higher accuracy when client distributions diverge. Figure 3 illustrates how unbalanced client contributions affect aggregation in FedAvg compared to FedProx.

Household load data are inherently non-IID due to several factors. The appliance mix varies widely across homes, as shown in studies of smart meter signatures [Beckel et al., 2014, Greveler et al., 2012]. The integration of distributed energy resources (DERs) such as rooftop solar, batteries, or electric vehicles introduces further variability by altering

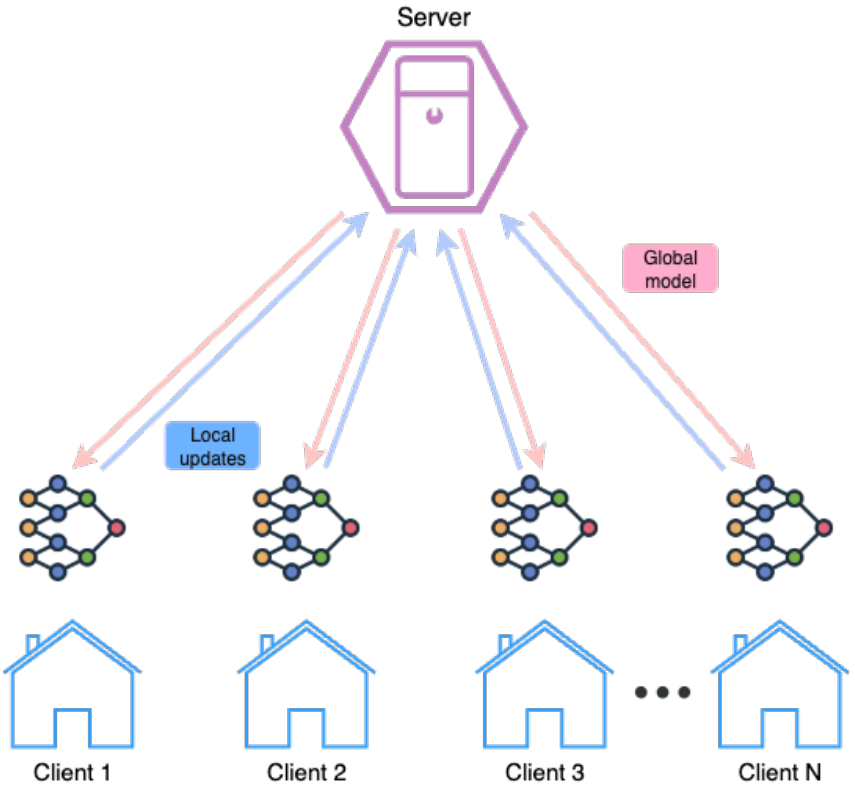


Figure 2 – Overview of the FL architecture for household STLF. The figure shows the server distributing the current global model to households, local training on private smart-meter data, and the return of model updates for aggregation. (Source: author's own elaboration).

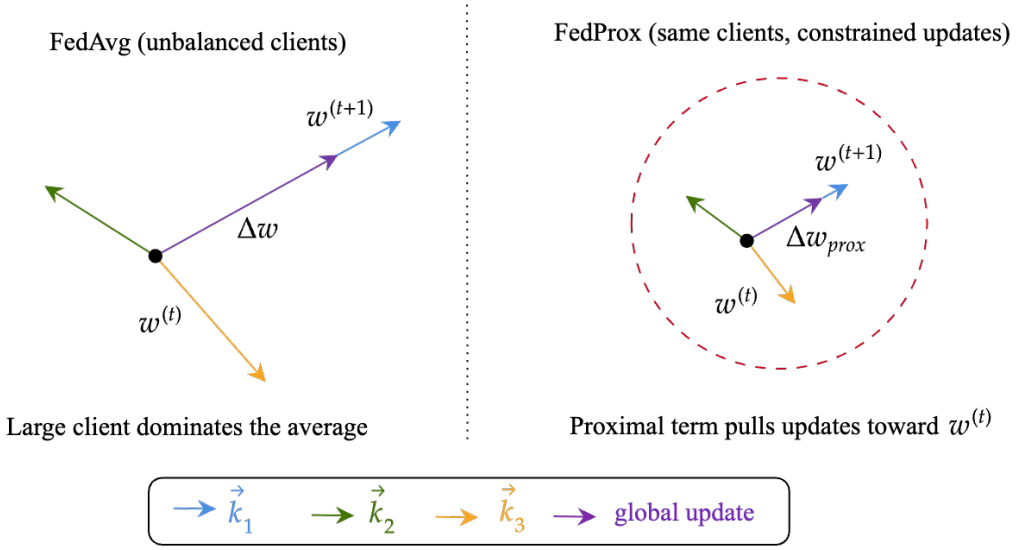


Figure 3 – Unbalanced clients in FedAvg versus FedProx. Example proportions: $\alpha_1 = 0.80$, $\alpha_2 = 0.15$, $\alpha_3 = 0.05$. Left: the weighted update Δw is driven by the largest client, shifting $w^{(t+1)}$ toward its distribution. Right: FedProx constrains local updates, yielding a smaller resultant Δw_{prox}. (Source: author's own elaboration)

net consumption profiles [Kong et al., 2017, Briggs et al., 2022]. Regional and climatic conditions also shape demand, as factors such as temperature, humidity, and daylight directly influence heating, cooling, and lighting loads [Hippert et al., 2001, Hong and Fan, 2016]. Together, these aspects generate heterogeneous client distributions that deviate significantly from the IID assumption required by FedAvg [McMahan et al., 2017].

Under such heterogeneity, FedAvg may suffer from unstable training dynamics. Large clients or those with skewed distributions can disproportionately influence the aggregated model, driving convergence toward biased solutions that reflect only part of the population. By contrast, FedProx adds a proximal term to the local objectives, constraining updates to remain closer to the global model (see Section 3.4). This reduces the influence of outlier clients and mitigates divergence, leading to more stable convergence in non-IID environments such as household STLF (Figure 3).

While FedProx was not designed for privacy, its proximal constraint reduces client drift by pulling local updates closer to the global model. This may incidentally dampen some client-specific signal in weight updates, but unlike differential privacy, it provides no formal guarantee of protection. Stronger privacy-preserving solutions are therefore required, which are discussed in Section 2.9.

2.8 Adversarial and Gradient Inversion Attacks

As already introduced in previous sections, FL alone is not sufficient to provide strong privacy guarantees. While it indeed enhances privacy by keeping raw data local, it also opens a surface for adversarial integrity attacks and remains vulnerable to privacy attacks. In fact, adversarial attacks are considered one of the most challenging problems in FL [Lyu et al., 2022], with the majority of the attacks focusing on classification tasks.

The distributed nature of FL makes it especially vulnerable to poisoning and backdoor attacks, since the server has no direct access to inspect the clients' local data. Malicious participants may manipulate their data or model updates to bias the global model, introduce hidden behaviors, or impair predictive performance [Fang et al., 2020, Bagdasaryan et al., 2020, Bhagoji et al., 2019].

As for privacy attacks, these are designed to disclose information about the clients participating in the process. Although private data never leaves the owner, the model updates exchanged with the server can still reveal sensitive patterns or properties of the underlying data. The most common privacy attacks reported in the literature include gradient inversion, membership inference, and property inference [Hitaj et al., 2017, Nasr et al., 2019, Melis et al., 2019].

Gradient inversion attacks (GIA) have received particular attention. In these attacks, an adversary exploits the gradients or weight updates shared during training to reconstruct the private training data of clients. Early methods such as Zhu et al. [2019]

demonstrated that even single gradient updates from image classifiers can reveal individual samples, sparking a wide body of follow-up research improving accuracy, speed, and robustness of inversion methods [Geiping et al., 2020, Yin et al., 2021]. While most GIA work has been developed in the vision domain, recent efforts such as Meijer et al. [2025b] extend these attacks to the time-series setting, highlighting that gradient leakage is a modality-agnostic risk in FL.

For Time Series Forecasting (TSF), there is limited research on adversarial attacks, with the majority of existing work focusing on integrity threats. Specifically in the load forecasting domain, several studies have analyzed the impact of poisoning. Fereidooni et al. [2021] and Bai et al. [2022] demonstrated that malicious clients that corrupt local training data or inject manipulated updates can bias aggregate forecasts, degrade predictive performance, or implant backdoors triggered by specific temporal patterns. More recently, Rieger et al. [2023] studied data-integrity attacks in decentralized FL for STLF, showing that targeted poisoning can significantly increase error rates and systematically shift forecasts in adversaries' favor.

On the privacy side, there are no known implementations of membership or property inference attacks for TSF. Nevertheless, membership inference has been investigated for centralized time-series models [Kulynych et al., 2022], and could, in principle, be adapted to FL. Regarding reconstruction attacks, Meijer et al. [2025b] introduced the novel *TS-Inverse* attack, showing that gradient inversion is particularly challenging in the time-series setting, since both input observations and forecast targets must be reconstructed while preserving temporal dynamics such as trends and periodicity. TS-Inverse is, to the best of our knowledge, the first work that explicitly adapts gradient inversion to time-series forecasting in a federated learning setting. The authors emphasize that such attacks can recover sensitive temporal information from various domains, including energy consumption.

Even with fewer studies than classification tasks, these findings underline that FL-based TSF inherits the dual threat surface of privacy leakage and poisoning, motivating the development of defenses such as differential privacy, secure aggregation, and robust aggregation rules tailored to sequential data.

2.9 Differential Privacy and Local Differential Privacy

Injecting noise into data before training is a common strategy for privacy preservation. DP achieves this by adding calibrated statistical noise, controlled by a privacy budget, to the data to protect individual sensitive information. The privacy budget is defined by the pair (ε, δ) [Dwork, 2006, Dwork and Roth, 2014b]. A randomized mechanism $\mathcal{M}$ is

said to satisfy (ε, δ)-differential privacy if, for all datasets D_1 and D_2 differing in at most one record, and for all measurable subsets S of outputs,

$$\Pr[\mathcal{M}(D_1) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D_2) \in S] + \delta.$$

While ε measures the magnitude of privacy loss, δ bounds the probability of this guarantee failing (see calibration details in Section 3.3.1).

The parameters ε and δ are non-negative real values that control the strength of the differential privacy guarantee. By definition, $\varepsilon \geq 0$, with smaller values corresponding to stronger privacy protection, while $\delta \in [0, 1)$ bounds the probability that the guarantee may fail. The case $\varepsilon = 0$ represents perfect privacy, whereas $\delta = 0$ yields pure ε-differential privacy; negative values for either parameter are not meaningful in the differential privacy framework [Dwork, 2006, Dwork and Roth, 2014b].

The parameters ε and δ influence privacy as follows:

- Smaller ε values provide stronger guarantees but require more noise, which can reduce model accuracy.
- Smaller δ values indicate stronger guarantees and are usually set below $1/n$, where n is the dataset size, to ensure negligible risk.

The values chosen for (ε, δ) directly determine the amount of noise that must be added: tighter budgets require stronger perturbations, while looser budgets allow lighter noise. In practice, the perturbations are usually drawn from either a Gaussian or a Laplacian distribution, depending on the chosen mechanism. Calibrating this level is a critical step, particularly in the context of FL. If the noise is too strong, it may distort the underlying load patterns and make model convergence slower or less stable. In practical terms, higher privacy guarantees usually come at the expense of forecasting accuracy, reflecting the inherent trade-off between privacy and utility.

Theoretical works recommend $\delta \ll 1/n$, where n is the dataset size, to ensure negligible risk [Dwork and Roth, 2014b]. In practice, applied studies in energy and related domains use values in the range of 10^{-5} to 10^{-6} [Eibl et al., 2018, Zhao et al., 2021], balancing feasibility with privacy protection. Recent NIST guidelines also emphasize that δ should be small enough to prevent meaningful leakage, while acknowledging trade-offs in applied settings [National Institute of Standards and Technology (NIST), 2023].

DP can be applied at different levels depending on the trust assumptions of the system. In its classical or centralized form, DP assumes the presence of a trusted server that collects raw data and adds noise only to the aggregated output (e.g., the sum of loads across households). For example, Eibl et al. [2018] applied Laplacian noise to aggregated smart metering signals, masking individual contributions while preserving forecasting utility. While this reduces disclosure risk, it still requires raw household data

to be centralized before noise is applied, which conflicts with data minimization principles and increases exposure risk.

More recent work has explored combining DP with FL, where raw data remain local. Zhao et al. [2021] proposed a DP-enhanced FL framework for STLF, injecting Gaussian noise into model updates at the server. This provides user-level DP but still allows the server to observe unprotected client updates before aggregation, leaving it as a potential attack surface.

In contrast, LDP shifts the noise addition to the client side: each household perturbs its own data or model updates before transmission. This prevents the server from ever observing raw consumption traces or unprotected gradients, offering stronger guarantees when the server cannot be fully trusted. Existing studies in smart grids [Shukla et al., 2025, Zhao et al., 2021] adopted server-side DP, where the server adds noise after receiving raw updates.

2.10 Research Gaps and Problem Statement

Despite extensive research on STLF, significant challenges remain when forecasting at the household level under modern privacy and security constraints. Traditional centralized learning exposes raw smart meter traces, conflicting with regulatory requirements such as GDPR, LGPD, and PIPL, and making centralized solutions increasingly impractical. FL has emerged as a promising alternative, since it keeps data local, but relying on this paradigm alone is insufficient: model inversion, membership inference, and gradient-based attacks still threaten user privacy, while heterogeneous household consumption patterns cause unstable convergence during training.

Prior works have attempted to mitigate these challenges with differential privacy (DP), but almost exclusively in centralized or server-side settings. In these approaches, the server continues to receive unprotected client updates before noise is added, leaving the aggregator as a single point of failure. To the best of our knowledge, no study has systematically applied local (LDP) to the household STLF context, where noise is injected directly at the client side, ensuring that the server never observes raw gradients or consumption traces. Moreover, the interaction between privacy guarantees, model utility, and convergence stability under non-IID client distributions remains underexplored.

This dissertation addresses these gaps by proposing an FL-LDP framework enhanced with FedProx aggregation. The following research hypotheses guide the work:

H1: The combined approach will maintain competitive forecasting accuracy relative to non-private baselines.

H2: The incorporation of LDP will provide stronger privacy guarantees by protecting client updates even against an untrusted server.

H3: The use of FedProx is expected to improve stability under heterogeneous household data, mitigating divergence and supporting more consistent convergence.

Together, these hypotheses motivate the methodology developed in Chapter 3.

Chapter 3

Proposed Methodology

3.1 Framework Overview

This work applies privacy protection layers to ML-assisted load forecasting through the implementation of FL with DP (FL-DP), exploring the trade-off between utility and performance. In addition, FedProx is incorporated to mitigate challenges arising from client heterogeneity.

The first protection layer is a decentralized approach based on FL, where raw data remain on the client side and are never shared with the server. FL proceeds in rounds: in each round, the global model (GM) is broadcast to a subset of clients $\mathcal{C}_i$, which train locally on their private data and return updated parameters. These updates are then aggregated by the server to refine the GM, and the process repeats for a fixed number of rounds or until convergence.

Figure 4 illustrates the proposed architecture at a high level. FL ensures local training on household devices, LDP injects calibrated noise into updates before transmission, and FedProx introduces a proximal constraint to stabilize training under heterogeneous conditions. Together, these layers form a framework that addresses data privacy, client heterogeneity, and forecasting accuracy. The following sections detail the mechanisms of each component and their integration into the full algorithm.

3.2 Federated Learning Architecture

In this work, households are modeled as clients in a federated learning environment, coordinated by a central server. Each client stores its own smart-meter data locally and never transmits raw records. Training proceeds in communication rounds, where the server orchestrates the following sequence:

1. **Client selection:** at the beginning of each communication round, a subset of available households is selected uniformly at random to participate in training.

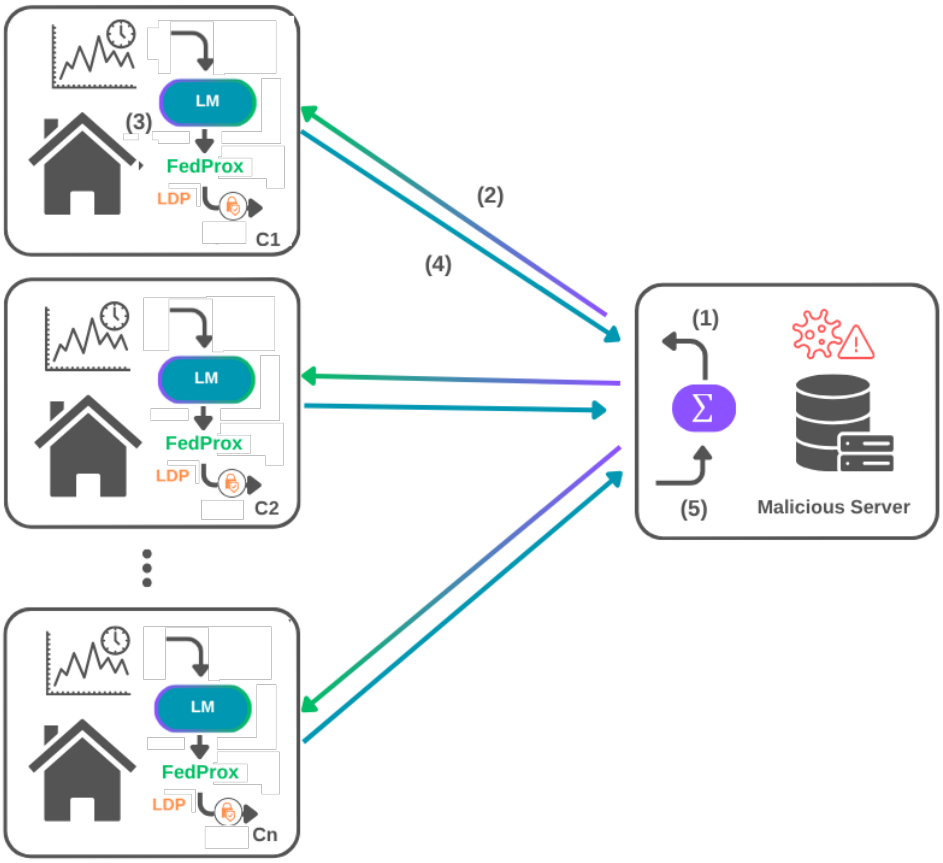


Figure 4 – High-level overview of the proposed framework combining federated learning, local differential privacy, and FedProx. In each federated round: (1) the participating clients are selected (2) the global model is broadcast by the server, (3) clients train locally on private household data, calibrated noise is added to the FedProx constrained updates before transmission, (4) updates are sent back to the server, and (5) aggregation produces a refined global model. FL ensures data remains local, LDP protects the shared updates, and FedProx stabilizes convergence under heterogeneous conditions.

2. **Broadcast:** the current global model $w^{(t)}$ is distributed to the selected clients.

3. **Local training:** each client initializes its local model with $w^{(t)}$ and performs E epochs of training on its private data, obtaining updated parameters $w_i^{(t+1)}$.

4. **Aggregation:** the server collects all local updates and combines them into a new global model.

The baseline aggregation method used in most federated learning systems is FedAvg [McMahan et al., 2017]. In FedAvg, local model updates are averaged in proportion to each client's dataset size:

$$w^{(t+1)} = \sum_{i=1}^{K} \frac{n_i}{\sum_{k=1}^{K} n_k} w_i^{(t+1)}, \tag{3.1}$$

where n_i is the number of training samples of client i, and K is the number of participating clients in the round.

This process is repeated for a pre-established number of rounds or until convergence. In household load forecasting, where client distributions are naturally heterogeneous, FedAvg serves as the reference aggregation strategy before introducing extensions such as FedProx (Section 3.4).

Algorithm 1 Baseline Federated Learning with FedAvg

1: Initialize global model $w^{(0)}$
2: **for** each round $t = 0, 1, 2, \ldots, R$ **do**
3: Server selects a subset of clients $\mathcal{C}_t$
4: **for all** $i \in \mathcal{C}_t$ **do** ▷ executed in parallel
5: Send $w^{(t)}$ to client i
6: Client i trains locally for E epochs
7: Client i sends $w_i^{(t+1)}$ to server
8: Server aggregates:

$$w^{(t+1)} \leftarrow \sum_{i \in \mathcal{C}_t} \frac{n_i}{\sum_{k \in \mathcal{C}_t} n_k} w_i^{(t+1)}$$

3.3 Local Differential Privacy Mechanism

The second layer applies LDP on each client $\mathcal{C}_i$, adding controlled statistical noise to the local updates. This ensures that even if weight updates are intercepted, sensitive household data cannot be reliably reconstructed. The privacy budget is defined by the pair (ε, δ), as described in Section 3.3.1, and determines the strength of the perturbations.

3.3.1 Privacy Budget and Noise Calibration

To enforce (ε, δ)-DP, each client proceeds as follows. First, compute the raw local update on batch B_i:

$$\Delta w_i \;=\; \sum_{x \in B_i} \nabla_w \ell(w; x), \tag{3.2}$$

where $\ell(w; x)$ denotes the per-sample training loss. The update is then clipped to have bounded L2-norm τ:

$$\Delta \bar{w}_i \;=\; \frac{\Delta w_i}{\max\Big(1,\; \|\Delta w_i\|_2 / \tau\Big)}. \tag{3.3}$$

Gaussian noise is then added to satisfy the required privacy level. Each client perturbs the clipped update with zero-mean noise of variance $(\sigma\tau)^2$:

$$\Delta \widetilde{w}_i \;=\; \Delta \bar{w}_i \;+\; \mathcal{N}\Big(0,\; (\sigma\tau)^2 I\Big). \tag{3.4}$$

The perturbed update is applied to obtain the next local model:

$$w_i^{t+1} \;=\; w_i^t + \Delta\widetilde{w}_i \tag{3.5}$$

and w_i^{t+1} is transmitted to the server, ensuring that each client's contribution satisfies (ε, δ)-differential privacy.

The clipping step bounds sensitivity, preventing any single client's update from disproportionately influencing the global model or leaking information through outlier weights. The Gaussian mechanism is widely used in private machine learning because it scales with sensitivity and composes well across multiple training steps [Dwork and Roth, 2014b, Abadi et al., 2016, Mironov, 2017b]. The variance σ^2 is calibrated so that, over all mini-batch updates and communication rounds, the cumulative privacy loss remains within the target (ε, δ) budget. This is tracked using standard composition techniques, ensuring that guarantees hold at the household level.

This work adopts user-level (ε, δ)-differential privacy, where each record corresponds to the complete load time-series of a single household Büscher et al. [2020], Yang et al. [2020], Alaqra et al. [2020]. In accordance with the standard formulation of approximate differential privacy Dwork and Roth [2014a], Mironov [2017a], the parameter δ is specified as a small constant satisfying $\delta \ll 1/N$, where N denotes the number of users contributing data. This choice follows prior work in differentially private analysis of smart-meter data, where values in the range 10^{-5}–10^{-6} are commonly employed. The specific value of δ adopted for each dataset is defined in the corresponding experimental setup section.

In this setup, each client perturbs its local weight update before transmitting it to the server. Even if the server is compromised, it only observes noisy contributions that satisfy (ε, δ)-differential privacy across the full training process. The analysis accounts for all communication rounds and mini-batch updates, ensuring that the global privacy budget is not exceeded while preserving forecasting utility.

In this way, LDP enforces user-level protection: even if all other participants collude, the contribution of a single household cannot be reliably inferred. This provides stronger guarantees than server-side differential privacy, where raw updates are exposed before noise is applied. The choice of ε and δ directly affects the balance between accuracy and privacy, as discussed in Section 3.3.2.

3.3.2 Privacy-Utility Trade-off

Differential privacy introduces a fundamental trade-off between the strength of privacy guarantees and the predictive accuracy of the trained model. In the context of load forecasting, utility is defined as the percentage of performance retained by the private model relative to its non-private federated baseline (Section 4.4).

Stricter privacy budgets, corresponding to smaller values of ε or δ, require injecting stronger perturbations into the client updates. While this provides stronger protection, it

also increases distortion of the underlying load patterns, which can reduce model accuracy and delay convergence. Conversely, larger values of ε relax the guarantees, allowing lighter noise and improving predictive utility. The calibration of (ε, δ) is therefore critical to achieving a practical balance.

This trade-off is further complicated in federated environments, where client data are non-IID and imbalanced. In such cases, the impact of noise is not uniform across clients: households with fewer samples or highly variable profiles are more vulnerable to performance degradation. As a result, the global utility depends not only on the privacy parameters but also on the degree of client heterogeneity.

In this work, the trade-off is explicitly quantified by varying ε while fixing δ to a small value (10^{-5} and 10^{-4}), following common practice in applied differential privacy for energy systems [Eibl et al., 2018, Zhao et al., 2021]. The results reported in Chapter 5 illustrate how the choice of privacy budget directly influences forecasting accuracy, with utility typically recovering as ε is relaxed. In some cases, moderate noise even improves generalization by mitigating overfitting, highlighting the dual role of differential privacy as both a protection mechanism and a regularizer.

3.4 FedProx Aggregation

Rather than employing the classical FedAvg algorithm for aggregation, this work adopts FedProx, which is an extension of FedAvg [Li et al., 2020a,b]. FedProx introduces a proximal term (μ) in the local objective function that penalizes large deviations from the global model. This is particularly relevant in smart grid applications, where the clients (households) have heterogeneous consumption patterns that can lead to unstable convergence. The local optimization problem is thus reformulated as:

$$\min_{w} \; f_i(w) + \frac{\mu}{2} \left\| w - w_t \right\|^2, \tag{3.6}$$

where $f_i(w)$ is the empirical risk on client i, w_t is the global model at round t, and μ is the proximal coefficient controlling the strength of the regularization.

After training, the clients send their model parameters to the server, which aggregates them by weighted averaging:

$$w^{(t+1)} \; = \; \sum_{i=1}^{K} \frac{n_i}{\sum_{k=1}^{K} n_k} \, \widetilde{w}_i, \tag{3.7}$$

where n_i is the number of samples of client i, and $\widetilde{w}_i$ denotes its locally updated and perturbed weights.

The effect of the proximal term is twofold. If μ is set too small, the algorithm behaves similarly to FedAvg, which may lead to divergence when clients have highly skewed data. On the other hand, if μ is set too large, local training becomes overly constrained,

limiting the ability of each client to adapt to its specific data distribution and slowing down convergence. Selecting an appropriate value of μ is therefore critical to balance global stability and local adaptability.

In the context of household load forecasting, non-IID heterogeneity arises naturally due to differences in appliance mix, occupancy routines, distributed energy resources (DERs), and climate conditions [Briggs et al., 2022, Hippert et al., 2001, Hong and Fan, 2016]. These factors can produce local updates that are highly divergent across clients, making naive averaging particularly ineffective. By constraining local updates to remain closer to the global model, FedProx reduces the risk of client drift and stabilizes training under these heterogeneous conditions.

Another practical benefit of FedProx is its ability to mitigate poisoning attacks. Malicious clients may attempt to inject adversarial updates that deviate significantly from the global trend, compromising model integrity. The proximal constraint dampens the impact of such outliers by penalizing extreme deviations, thereby improving robustness against data- or model-poisoning attempts. While this does not provide the same formal protection as differential privacy, it incidentally reduces the risk of sensitive information being amplified through abnormal updates, contributing an additional layer of resilience to the framework.

This regularization mechanism is especially beneficial when combined with LDP. The injected noise increases the variance of local updates, and the proximal term helps counteract this effect by anchoring updates around the global parameters. As shown in Chapter 5, this combination allows the framework to maintain competitive forecasting accuracy while enforcing privacy guarantees, even under unbalanced or heterogeneous client distributions.

3.5 Combined FL + LDP + FedProx Approach

This section describes the integration of FL, LDP, and FedProx into a unified framework for privacy-preserving STLF. FL ensures that household-level data remain local, with only model updates transmitted to the server. LDP introduces controlled statistical noise to further protect client privacy, while FedProx adds a proximal term during local training to stabilize convergence under heterogeneous (non-IID) data. Together, these elements aim to deliver accurate forecasting while maintaining strong privacy guarantees.

3.5.1 Algorithm and Workflow

To obtain a personalized final model for each client, FL systems execute a predetermined number of communication rounds. Figure 4 illustrates the process over a single round, with the numbered steps denoting the sequential operations. First, the server

computes a weight multiplier for each client (Algorithm 2), reflecting the (ε, δ) privacy parameters introduced in Section 3.3. The global model parameters are then broadcast to the selected clients, which initialize their local models accordingly.

Each client performs E local epochs by minimizing the proximal objective described in Section 3.4, which constrains updates to remain close to the global model and improves stability under heterogeneous data. During local optimization, calibrated Gaussian noise is injected into mini-batch updates following the (ε, δ) budget (Section 3.3), ensuring that the transmitted weights satisfy local differential privacy. The noisy updates are then returned to the server.

Finally, the server aggregates all received client updates into the new global model using the weighted averaging rule outlined in Section 3.4. This completes one federated round. The next round begins with the broadcast of the updated model and recalculated multipliers.

By layering local training with FedProx, local perturbation through LDP, and aggregation via weighted averaging, the framework integrates privacy protection, robustness to heterogeneity, and forecasting accuracy. This joint mechanism differs from plain FedAvg or server-side DP, since noise is applied locally and optimization is regularized before aggregation.

3.5.2 Design Decisions

Algorithm 2 illustrates the incorporation of a client-specific noise configuration in conjunction with the FedProx scheme, governed by a global target privacy budget (ε, δ). The noise multiplier σ is calibrated individually for each client, considering the total number of communication rounds (R) and the relative contribution of each client to the overall federated dataset. This calibration starts from a target privacy budget (ε, δ) and ensures that the reported ε values correspond to the cumulative privacy guarantee of the entire training process, rather than to a per-round budget. In other words, the final ε accounts for privacy composition across communication rounds and client participation.

Let $\mathcal{C}$ denote the set of participating clients and n_k the local dataset size of client k. The effective sampling rate of each client is defined as

$$q_k = \frac{n_k}{\sum_{j \in \mathcal{C}} n_j}, \tag{3.8}$$

and is used during noise calibration to reflect differences in data availability across clients. This allows clients with smaller datasets to be assigned higher noise levels in order to satisfy the same global privacy budget.

Noise calibration follows standard Rényi differential privacy (RDP) analysis for the subsampled Gaussian mechanism. For a given client k and Rényi order $\alpha > 1$, the

per-round RDP privacy loss is given by

$$\varepsilon_{\mathrm{RDP}}^{(k)}(\alpha) = \frac{\alpha q_k^2}{2\sigma_k^2}. \tag{3.9}$$

Privacy loss composes additively across communication rounds, yielding a total RDP budget after R rounds of

$$\varepsilon_{\mathrm{RDP,total}}^{(k)}(\alpha) = R \cdot \frac{\alpha q_k^2}{2\sigma_k^2}. \tag{3.10}$$

The final (ε, δ)-differential privacy guarantee is obtained by converting RDP to (ε, δ)-DP according to

$$\varepsilon = \varepsilon_{\mathrm{RDP,total}}^{(k)}(\alpha) + \frac{\ln(1/\delta)}{\alpha - 1}. \tag{3.11}$$

For each client, the noise multiplier σ_k is selected as the smallest value satisfying this inequality for the target (ε, δ), optimized over α.

For comparability, the same hyperparameters (learning rate, batch size, number of rounds) were applied across all experiments and datasets.

Algorithm 2 Client-specific noise configuration with FedProx and global privacy budget

Input: Clients $\mathcal{C}$, local dataset sizes $\{n_k\}$, total rounds R, target privacy (ε, δ)
Output: Noise multipliers $\boldsymbol{\sigma} = \{\sigma_k\}_{k \in \mathcal{C}}$
$N \leftarrow \sum_{k \in \mathcal{C}} n_k$
for each client $k \in \mathcal{C}$ **do**
    $q_k \leftarrow \frac{n_k}{N}$
    $\sigma_k \leftarrow \min_\sigma$ s.t. $\varepsilon \geq \min_{\alpha > 1} \left(R \cdot \frac{\alpha q_k^2}{2\sigma^2} + \frac{\ln(1/\delta)}{\alpha - 1} \right)$
return $\boldsymbol{\sigma}$

Implementation note. Algorithm 2 formalizes the analytical local differential privacy (LDP) mechanism assumed in the theoretical formulation. In the experimental implementation used in this dissertation, however, the Opacus Privacy Engine performs gradient clipping and noise injection at the mini-batch level using sample-level Rényi differential privacy accounting. As a result, the implemented mechanism is significantly stricter than the update-level mechanism described in Algorithm 2, leading to a larger number of privacy compositions and higher noise multipliers for heterogeneous or data-poor clients.

Chapter 4

Experimental Setup

4.1 Datasets

4.1.1 HUE Dataset

The Hourly Usage of Energy dataset (HUE) dataset [Makonin, 2019] provides hourly energy consumption data for 28 residential buildings in British Columbia, covering June 1, 2012, through May 20, 2020. Although it includes 13 variables related to weather and environmental factors, only the hourly energy consumption in kWh was used in this study. Aditionally to the experiments done using all clients, 4 groups differing in client heterogeneity and number were created, as summarized in Table 1.

Table 1 – Clients selected from the HUE dataset for balanced and unbalanced scenarios.

	Balanced	**Unbalanced**		
	5 clients	**5 clients**	**7 clients**	**10 clients**
1	Residential 4	Residential 17	Residential 15	Residential 7
2	Residential 8	Residential 21	Residential 17	Residential 9
3	Residential 9	Residential 26	Residential 21	Residential 15
4	Residential 10	Residential 27	Residential 22	Residential 17
5	Residential 13	Residential 28	Residential 26	Residential 20
6	-	-	Residential 27	Residential 21
7	-	-	Residential 28	Residential 22
8	-	-	-	Residential 26
9	-	-	-	Residential 27
10	-	-	-	Residential 28

4.1.2 UK-DALE Dataset

The UK Domestic Appliance Level Electricity (UK Domestic Appliance-Level Electricity dataset (UK-DALE)) [Kelly and Knottenbelt, 2015] contains appliance-by-appliance and whole-home power demand for five UK homes recorded approximately every 6s, covering approximately 4.5 years (about 53 months). In this study, we focused on whole-home demand.

The overall data distribution for the UK-DALE dataset is shown in Figure 5. Notably, it is sparse and inconsistent across users when compared to the HUE dataset. While one of the clients has readings spanning nearly 4.5 years, with 39,092 entries from 2012-11-09 to 2017-04-26, others have fewer than 6,000 readings. For example, Client 3 data covers just over one month (from February 27 until April 08, 2013).

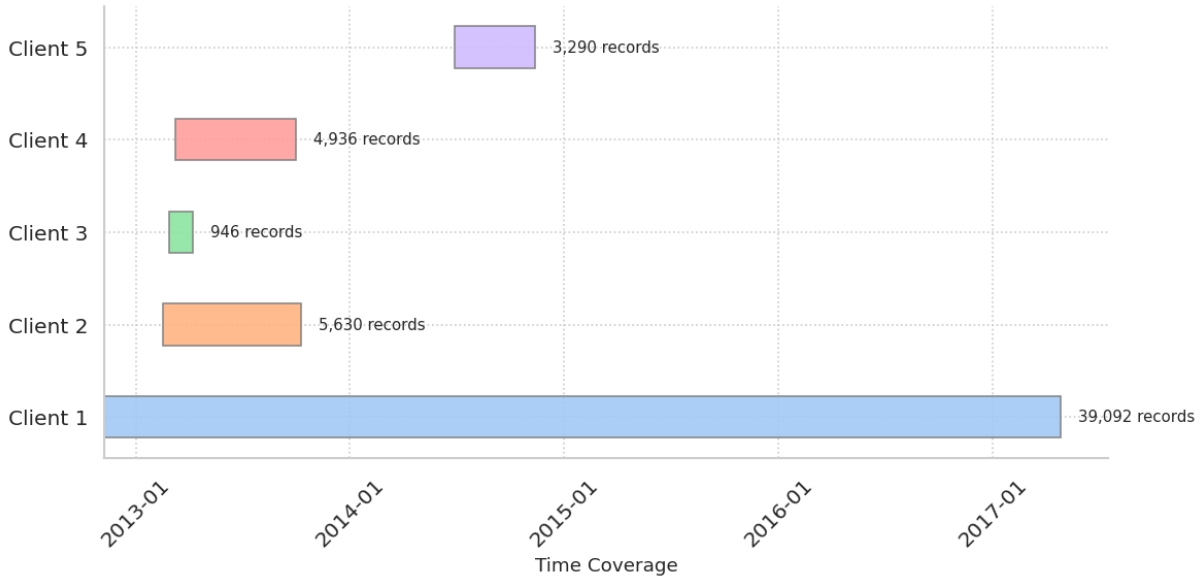


Figure 5 – Distribution of data records for the five clients in the UK-DALE dataset (2013-2017).

4.2 Preprocessing and Feature Engineering

The preprocessing pipeline was designed to produce a stable temporal structure, minimize manual feature crafting, and provide representations that are well suited for deep learning models operating under both centralized and federated training settings. Both datasets were subjected to an identical preprocessing and feature engineering pipeline, the details of which are provided below.

All time series were first aligned to a uniform hourly `DateTimeIndex`. When raw files did not contain an explicit timestamp, a datetime index was reconstructed from the available date and hour fields. Duplicate labels were removed, and the series were reindexed to a complete hourly grid, followed by linear interpolation of missing values. A five-hour

rolling mean was applied to smooth short-term fluctuations in the raw load signal, with window-induced gaps filled after smoothing.

To represent recurring consumption patterns driven by human routines and calendar cycles, cyclical encodings of hour of day and day of week were introduced using sine and cosine components. This formulation maintains continuity across temporal boundaries and preserves periodicity, improving compatibility with neural networks that rely on continuous feature spaces. The final feature set included the smoothed load value (`energy_kWh`) together with `sin_hour`, `cos_hour`, `sin_day`, and `cos_day`.

Supervised learning samples were constructed using sliding windows with a look-back of 12 time steps, predicting the value at time t given the previous 12 hours. This approach captures short-term temporal dependencies without requiring explicit lag features such as 1-hour, 24-hour, or 7-day lags. The cyclical encoding implicitly convey daily and weekly periodic structure, reducing feature dimensionality and eliminating the need for extensive handcrafted temporal features.

An 80/20 chronological split was applied to prevent temporal leakage. Feature and target normalization followed a z-score transformation fitted exclusively on the training split, with zero-variance guards applied to ensure numerical stability. The resulting sequences were exported as tensors and used consistently across all experimental configurations, including centralized training, local-only baselines, federated learning (FL), and federated learning with differential privacy (DPFL). This unified preprocessing pipeline ensured methodological consistency across all scenarios presented in Chapter 5.

4.3 Baseline Models

This work evaluates four complementary training scenarios: (i) *local-only training*, where each client trains an isolated model without collaboration; (ii) *centralized learning*, where all data are aggregated into a single model; (iii) *federated learning (FL)* with FedProx to handle heterogeneity; and (iv) *privacy-enhanced FL (DPFL)*, where model updates are perturbed with differential privacy. Using the same preprocessing pipeline and the same set of hyperparameters across all scenarios ensures that comparisons reflect only the effect of decentralization, heterogeneity management, and privacy mechanisms. All hyperparameters were tuned centrally using the training and validation partitions of the five balanced HUE clients.

4.3.1 LSTM

The LSTM baseline follows the configuration presented in Table 2. The model receives five input features, uses a single recurrent layer with 128 hidden units, and is followed by a stack of linear layers that progressively reduce dimensionality before

producing the final forecast. For centralized training, early stopping and weight decay were used to reduce overfitting, which is particularly relevant for household-level datasets with limited size. The same architecture was used in the local-only and federated settings to support a direct comparison across all scenarios.

Table 2 – Model Architecture Hyperparameters

Model	Parameter	Value
LSTM	Lookback Window	96
	Input Features	5
	Hidden Units	128
	Dropout	0.0
	Architecture	1 LSTM Layer → Linear(128, 64) → Linear(64, 32) → Linear(32, 1)
Transformer	Lookback Window	24
	Input Features	5
	d_model	64
	n_layers	2
	n_heads	4
	dim_feedforward	512
	Dropout	0.1

4.3.2 Transformer

A Transformer encoder was also evaluated as a baseline, configured according to Table 2. The architecture uses a 24-step look-back window, four attention heads, and a feedforward dimension of 512. Temporal structure is provided implicitly by the cyclical encodings introduced in Section 4.2, which replace the need for learned positional embeddings. Hyperparameters were tuned jointly with the LSTM, and the same configuration was used in all centralized and federated experiments. The Transformer serves as a point of comparison for more expressive architectures capable of capturing longer-range temporal dependencies.

4.3.3 Federated Approach

The federated baseline adopts the same LSTM architecture used in centralized training. Training is orchestrated with Flower [Beutel et al., 2020] and follows the FedProx aggregation strategy, as summarized in Table 3. Clients perform two local epochs per communication round, with a batch size of 32, and Adam is used as the local optimizer. Five clients participate in total, and two are sampled per round. The proximal term introduced by FedProx stabilizes convergence under heterogeneous (non-IID) household

profiles, constraining local updates and reducing client drift. This setting represents the non-private federated baseline against which DP-enhanced models are compared.

4.3.4 Privacy-Enhanced FL Model

The privacy-enhanced model extends the FL configuration by incorporating client-side differential privacy through Opacus [Yousefpour et al., 2021]. Each client clips gradients and adds Gaussian noise calibrated to a global (ε, δ) budget. Following common practice in privacy-preserving smart-meter analysis, the parameter δ was set to 10^{-6} for both datasets. Noise is applied locally, before transmission, ensuring that the server never observes unprotected updates. All remaining hyperparameters were kept identical to the non-private FL configuration to guarantee a fair evaluation of the privacy–utility trade-off. The DPFL model therefore isolates the contribution of local differential privacy to convergence behaviour and predictive performance.

4.4 Evaluation Metrics and Privacy Accounting

To evaluate the performance of the proposed FL-LDP with FedProx framework, we adopt both standard forecasting error metrics and complementary measures to assess fairness across clients, training stability, distributional heterogeneity, and gradient dynamics.

4.4.1 Forecasting Error Metrics

Forecasting accuracy was evaluated using standard error measures widely applied in load forecasting, each capturing different aspects of predictive performance.

4.4.1.1 Mean Absolute Error (MAE)

The MAE measures the average magnitude of forecast errors without considering their direction:

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.1}$$

It is less sensitive to outliers than squared-error metrics and provides a general indication of overall error size.

4.4.1.2 Root Mean Squared Error (RMSE)

The RMSE penalizes larger errors more heavily by squaring them before averaging:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{4.2}$$

This property makes RMSE useful for highlighting models that produce occasional but significant deviations, which is particularly relevant in energy forecasting where peak errors can have a disproportionate impact.

4.4.1.3 Symmetric Mean Absolute Percentage Error (sMAPE)

The sMAPE expresses forecast error as a scale-independent percentage while avoiding the instability of MAPE when the actual value approaches zero. It is defined as:

$$\text{sMAPE} = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \tag{4.3}$$

This symmetric formulation bounds the denominator away from zero and provides a more stable percentage error measure for household load forecasting, where consumption values may occasionally be close to zero.

4.4.1.4 Utility

To assess the impact of DP, *utility* is defined as the percentage of performance retained by the differentially private model relative to the non-private FL baseline:

$$\text{Utility} = \left(1 - \frac{\text{RMSE}_{\text{DP}(\varepsilon,\delta)} - \text{RMSE}_{\text{noDP}}}{\text{RMSE}_{\text{noDP}}}\right) \times 100\% \tag{4.4}$$

This relative measure has been used in prior work to quantify the privacy-utility trade-off in smart grid forecasting [Eibl et al., 2018, Shukla et al., 2025].

4.4.2 Fairness and Client Heterogeneity Metrics

4.4.2.1 Per-Client Sample Size and Target Variance

The number of samples and the variance of the target load (y) are reported for each client. This information highlights imbalance in data availability across clients and provides context for interpreting heterogeneous forecasting performance [Briggs et al., 2022].

4.4.3 Distributional Heterogeneity Metrics

To capture heterogeneity across clients, two complementary measures are employed: (i) DTW, which focuses on temporal alignment of daily load profiles, and (ii) Wasserstein distance, which captures broader distributional differences in consumption patterns.

4.4.3.1 Profile Distance

The distance between each client's daily average load profile and the global mean profile was measured using Dynamic Time Warping (DTW) [Berndt and Clifford, 1994].

DTW aligns time series by allowing shifts in the time axis, which makes it more robust than Euclidean distance when comparing load profiles that have similar shapes but peaks and troughs occurring at different times. This property is particularly relevant for household electricity consumption, where daily routines can vary in timing even if overall patterns are comparable. It is also important in this context because the client series were collected over different calendar periods, meaning that peaks and routines may not be synchronized across households.

Formally, for two time series $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_m)$, the DTW distance is defined as:

$$DTW(X, Y) = \min_{p \in \mathcal{P}} \left(\sum_{(i,j) \in p} d(x_i, y_j) \right), \tag{4.5}$$

where $\mathcal{P}$ is the set of all possible warping paths aligning X and Y, and $d(x_i, y_j)$ is the local distance between points x_i and y_j. In this work, the squared Euclidean distance will be used.

By quantifying the temporal adjustments required to optimally align each client's daily load curve to the global mean, DTW highlights heterogeneity across households. Small distances indicate clients with consumption patterns closely aligned with the population average, while larger distances reflect differences in timing or intensity of daily routines.

4.4.3.2 Distribution Shift

The Wasserstein distance was adopted to quantify the difference between each client's load distribution (and selected features) and the global distribution, defined as the empirical distribution aggregated across all clients [Peyré and Cuturi, 2019].

Formally, for two probability distributions P and Q over a metric space $(\mathcal{X}, d)$, the p-Wasserstein distance is defined as:

$$W_p(P, Q) = \left(\inf_{\gamma \in \Gamma(P,Q)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^p \, d\gamma(x, y) \right)^{1/p}, \tag{4.6}$$

where $\mathcal{X}$ is the domain of the random variables (e.g., load values), $d(x, y)$ is the ground distance between two points x and y, $p \geq 1$ is the order of the metric, and $\Gamma(P, Q)$ is the set of all couplings (joint distributions) whose marginals are P and Q.

Intuitively, the Wasserstein distance quantifies the dissimilarity between two distributions by considering both the magnitude and the location of their probability mass. As a true metric, it captures not only shifts in support but also changes in shape, which makes it suitable for comparing heterogeneous load distributions across households. In practice, this work employs the 1-Wasserstein distance (Earth Mover's Distance), which provides an interpretable measure of distributional shift in energy load forecasting.

In this context, smaller Wasserstein distances mean a client's load distribution is closer to the overall population, while larger distances indicate households with more distinct consumption patterns. This helps highlight heterogeneity across clients.

4.4.4 Gradient Metrics

4.4.4.1 Gradient Drift

At each round t, for client i with local update $\Delta w_i^{(t)}$ and the global aggregated update $\Delta w^{(t)}$, the drift is defined as:

$$D_i^{(t)} = \left\| \Delta w_i^{(t)} - \Delta w^{(t)} \right\|_2 \tag{4.7}$$

The mean and variance of $D_i^{(t)}$ across rounds and clients are reported. Lower drift under FedProx indicates that client updates remain closer to the global update direction, reflecting improved stability in heterogeneous settings [Li et al., 2020a].

4.4.5 Privacy Parameters

The privacy budget is reported in terms of (ε, δ)-differential privacy, where smaller values of ε correspond to stronger privacy guarantees. For the UK–DALE dataset ($N = 5$) and the HUE dataset ($N = 28$), the global parameter δ is instantiated as 10^{-6} in both cases, consistent with the user-level privacy model adopted in this work. To evaluate the privacy–utility trade-off, the global privacy budget is varied across $\varepsilon \in \{1, 4, 7, 10\}$.

4.5 Implementation Details

4.5.1 Software and Core Libraries

The experimental setup was developed in Python (v3.10). The core stack was built upon several key open-source libraries:

- **PyTorch (v2.6.0):** The primary deep learning framework for model definition and training.
- **Flower (v1.18.0):** The federated learning framework used to simulate and orchestrate experiments via its Ray-based simulation engine.
- **Opacus (v1.4.1):** Integrated for sample-level Differential Privacy (DP), converting PyTorch models and optimizers to their private counterparts.
- **Scikit-learn (v1.0.0):** Used for data preprocessing (`MinMaxScaler`) and evaluation (RMSE), as described in Sections 4.2 and 4.4.
- **NumPy & Pandas:** Used for general data manipulation and loading.

4.5.2 Hardware and Execution Environments

Experiments were conducted on two distinct hardware backends to test the implementation:

- **Cloud (Google Colab):** Used NVIDIA T4 (16GB VRAM) and A100 (40/80GB VRAM) GPUs. PyTorch was configured for the `cuda` backend. Client simulations on a single GPU were managed by allocating fractional resources (e.g., `num_gpus = 0.2`).
- **Local (Apple M3):** Used a MacBook Air (M3) with a 10-core GPU and 16GB Unified Memory. PyTorch was configured for **Metal Performance Shaders (MPS)**. The `mps` backend does not support fractional allocation, so all client processes requested the single GPU, with scheduling handled by macOS.

4.5.3 Hyperparameter Configuration

All model and training hyperparameters were tuned beforehand in a centralized environment using a standard training and validation split. The final values were then used for all federated experiments.

The experiments were conducted across several scenarios on the HUE dataset to test heterogeneity and sampling: 5 balanced clients; 5, 7, and 10 unbalanced clients; and all available clients with sampling fractions of 1.0, 0.5, and 0.2.

Training loop parameters were varied by experimental goal:

- **FedAvg vs. FedProx Comparison:** Trained for 40 server rounds and 5 local epochs.
- **Differential Privacy (DP) Experiments:** Trained for 20 server rounds and 2 local epochs to manage the privacy budget.

Common hyperparameters for training and privacy are listed in Table 3. The specific model architectures are detailed in Section 4.3.

Table 3 – Common Training and Privacy Hyperparameters

Parameter	**Value**
Local Training	
Optimizer	Adam
Learning Rate	5×10^{-4}
Batch Size	32
FedProx Parameter (μ)	0.01
Differential Privacy	
Target Epsilon (ϵ)	$\in \{1, 4, 7, 10\}$
Target Delta (δ)	1×10^{-6}
Max Gradient Norm (C)	1.0

Chapter 5

Results and Discussion

5.1 Baseline Performance: LSTM and Transformer

To establish a reference point for the federated experiments, this section evaluates the LSTM and Transformer architectures under three training regimes: centralized, local-only, and federated (FedAvg). These baselines are computed exclusively on the balanced five-residence subset of the HUE dataset (Residences 4, 8, 9, 10, and 13), which serves as the core benchmark throughout this work. Using this controlled group allows a clean comparison across architectures while avoiding the confounding effects of heterogeneity present in larger or unbalanced configurations.

All models follow the preprocessing pipeline in Section 4.2 and were trained using hyperparameters tuned on this same five-residence group. Federated training uses 40 communication rounds and 5 local epochs per round.

The centralized setting provides an empirical upper bound, since the model benefits from the full aggregated dataset. The local-only setting captures the opposite limit, where each residence trains independently on its own data. FedAvg sits between these two configurations, combining decentralized updates with cross-residence aggregation.

5.1.1 LSTM Baseline Results

Table 4 reports the LSTM baseline performance. As expected, the centralized model achieves the strongest accuracy (mean RMSE 0.09059), confirming that broader data coverage improves short-term load forecasting. Local-only training produces slightly higher errors (mean RMSE 0.09172) despite being evaluated on the same household distribution as training. This small gap reflects the limited sample size and reduced exposure to cross-residence temporal variability.

FedAvg reaches a mean RMSE of 0.09800, introducing a modest penalty relative to both baselines. This behaviour aligns with the expected effect of simple parameter averaging in the presence of behavioural differences across households. Some residences

(e.g., 8 and 9) remain close to their local-only performance, while others (e.g., 4 and 13) exhibit larger gaps, indicating that their consumption patterns differ more from the global mean learned through aggregation.

Table 4 – LSTM baseline performance on the balanced HUE group (Residences 4, 8, 9, 10, 13). Mean over 10 runs.

	Centralized		Local-Only		FedAvg	
Residence	RMSE	MAE	RMSE	MAE	RMSE	MAE
4	**0.08921**	**0.05938**	0.08980	0.06091	0.10943	0.07552
8	**0.08422**	0.04697	0.08598	**0.04638**	0.09202	0.05089
9	**0.07084**	0.04499	0.07127	**0.04367**	0.08710	0.05526
10	**0.09579**	**0.05555**	0.09654	0.05581	0.10338	0.05847
13	**0.11283**	**0.06606**	0.11340	0.06808	0.12521	0.07631
Average	**0.09059**	**0.05460**	0.09172	0.05535	0.09800	0.05929

5.1.2 Transformer Baseline Results

Table 5 presents the corresponding Transformer baselines. The centralized Transformer underperforms the centralized LSTM (mean RMSE 0.1245 vs. 0.0906), a difference consistent with the nature of this task. Hour-level forecasting on short look-back windows favours recurrence, while the Transformer's larger capacity and attention mechanism do not translate into measurable gains given the limited size of each household's dataset.

Local-only Transformers show improved performance relative to their centralized counterpart, reflecting the reduced distributional shift. However, FedAvg introduces a larger degradation than observed for the LSTM: the mean RMSE rises to 0.1293, and the performance spread increases across residences. This behaviour highlights the Transformer's greater sensitivity to cross-residence variability and its higher-dimensional parameter space, which amplifies inconsistencies during aggregation.

Despite this, the Transformer retains stable convergence across all runs, and its behaviour mirrors the structural conclusions drawn from the LSTM experiments: centralized remains strongest, local-only benefits from distribution alignment, and FedAvg introduces a predictable performance penalty shaped by the underlying heterogeneity.

Table 5 – Transformer baseline performance on the balanced HUE group (Residences 4, 8, 9, 10, 13). Mean over 10 runs.

	Centralized		Local-Only		FedAvg	
Residence	RMSE	MAE	RMSE	MAE	RMSE	MAE
4	0.1251	0.0921	**0.1056**	**0.0746**	0.1378	0.1023
8	0.1141	0.0713	**0.0986**	**0.0615**	0.1129	0.0698
9	0.0997	0.0650	**0.0863**	**0.0591**	0.1115	0.07585
10	0.1304	0.0805	**0.1103**	**0.0684**	0.1259	0.0787
13	0.1532	0.1092	**0.1306**	**0.0827**	0.1583	0.1080
Average	0.1245	0.0836	**0.1063**	**0.0693**	0.1293	0.0869

5.1.3 Discussion

Across both architectures, the same structure emerges. Centralized training provides the strongest upper bound. Local-only models benefit from perfect distribution alignment but are limited by per-residence data volume. FedAvg introduces a controlled performance penalty that reflects natural behavioural variability across households. The LSTM adapts more effectively to this setting, while the Transformer is more sensitive to cross-residence inconsistencies. These baselines establish the foundation for evaluating structured aggregation (FedProx) and privacy-preserving federated learning (LDP-FL) in the following sections.

5.2 Effect of FedProx on Heterogeneous Clients

5.2.1 Heterogeneity in STLF

5.2.1.1 HUE Dataset

The HUE dataset exhibits marked differences in distributional heterogeneity across the predefined client groups. Table 6 summarizes these differences using variance, DTW, and Wasserstein distances. The boxplots in Figures 6a–6d complement these metrics by showing the raw consumption distributions for each residence within the groups.

Table 6 – Aggregate heterogeneity metrics for HUE client groups (means across clients in each group).

Group	N Clients	Target Variance	DTW	Wasserstein
1	5	0.2447	2.8814	0.1918
2	5	0.4731	10.1293	0.5742
3	7	0.6919	12.1721	0.6156
4	10	0.5432	9.0612	0.4927
5	28	0.3702	7.2223	0.4005

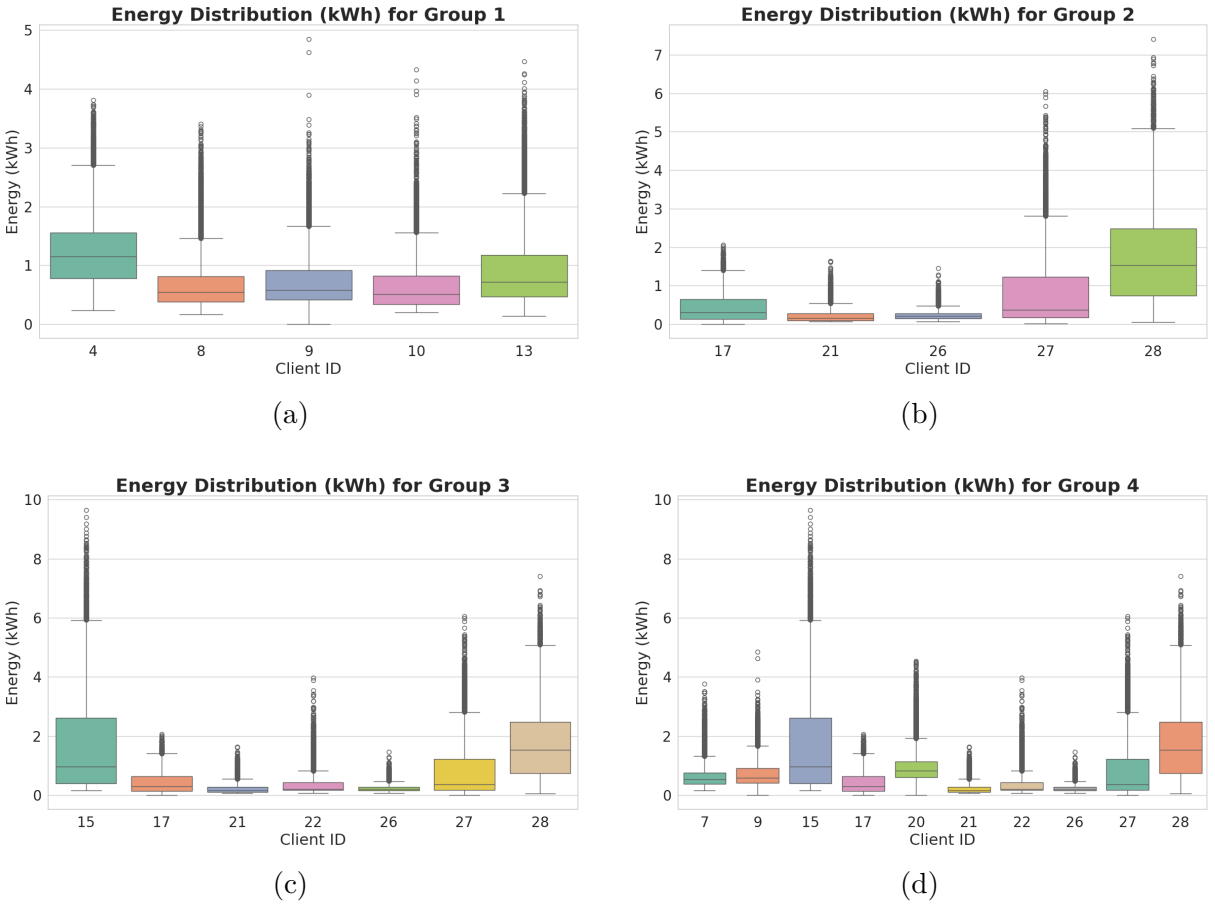
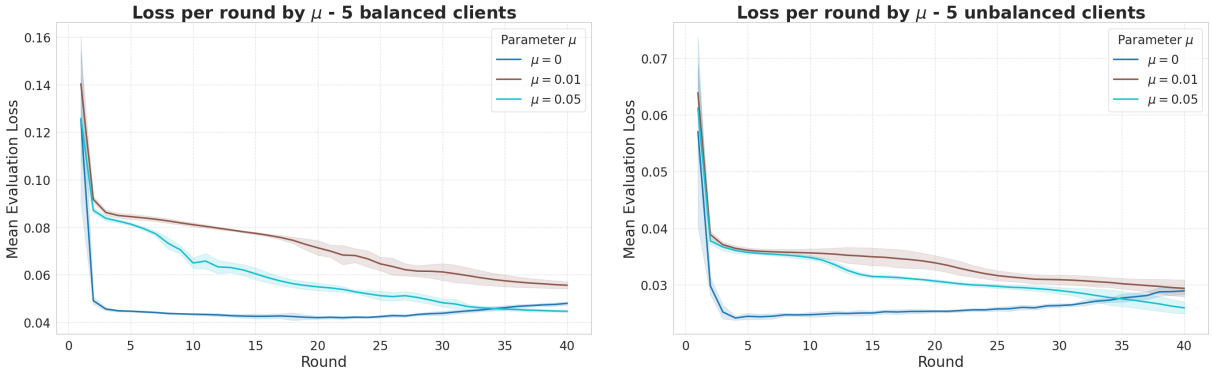


Figure 6 – Energy consumption (kWh) distributions for the four defined client groups from the HUE dataset. **(a)** Group 1 (5 clients, balanced), **(b)** Group 2 (5 clients, unbalanced), **(c)** Group 3 (7 clients, unbalanced), and **(d)** Group 4 (10 clients, unbalanced), highlighting heterogeneity and outlier presence.

Group 1 (the balanced five-client set) displays relatively homogeneous behaviour, with similar medians, interquartile ranges, and outlier patterns across households. This internal consistency explains why all μ configurations in Figure 7a converge smoothly and maintain narrow variance bands: clients generate compatible gradient directions and the optimization landscape is well aligned.

By contrast, Groups 2 and 3 display substantially wider within-group heterogeneity. Residences 15 and 28 repeatedly appear as heavy-outlier households, combining high variance with large temporal irregularity, while others (such as Residences 21 and 26) exhibit low-variance, highly concentrated profiles. The resulting mismatch in both scale and temporal structure produces uneven optimization dynamics. This is visible in Figure 7b, where the unbalanced five-client set shows higher initial loss, slower stabilization, and greater sensitivity to the choice of μ.

Group 4 (10 clients) occupies an intermediate position: heterogeneity is still present, but the diversity of clients dilutes the effect of extreme profiles. This property is consistent with the privacy-utility results reported later for the larger-group settings.



(a) Loss per round with 5 balanced clients (b) Loss per round with 5 unbalanced clients

Figure 7 – Mean evaluation loss per round for different values of μ under balanced and unbalanced client data distributions.

5.2.1.1.1 Seven-client unbalanced group.

The results for the seven-client unbalanced HUE group highlight the decisive role of client-level heterogeneity in shaping the behaviour of FedProx. The heterogeneity metrics in Table 7 reveal substantial structural divergence across households: two clients (Residences 15 and 28) exhibit extremely high temporal dissimilarity to the population (DTW values close to 20) and comparatively large Wasserstein distances, while Residence 27 is the most aligned with the global distribution (DTW $\approx$ 1.64). This unevenness is directly reflected in forecasting performance.

Table 7 – Heterogeneity metrics for the seven-residence unbalanced HUE group. Bold values indicate extreme heterogeneity (highest/lowest).

Residence	N Samples	Target Variance	DTW	Wasserstein
15	21021	**2.3540**	**20.4624**	0.6261
17	7988	0.1328	6.7974	0.5741
21	5203	**0.0375**	13.1356	**0.7742**
22	16964	0.1234	9.8344	0.6447
26	4171	**0.0221**	12.8922	**0.7640**
27	10603	0.8668	**1.6405**	**0.1885**
28	15788	**1.3066**	**20.4421**	0.7374

Under the unregularized configuration ($\mu = 0$), the system attains its lowest mean RMSE (0.1190), yet this value masks wide dispersion across clients: the most heterogeneous household reaches an RMSE of 0.3242, whereas the best-aligned residence achieves 0.0361. Introducing a proximal term modifies this landscape. A moderate regularization level ($\mu = 0.005$) increases the global mean RMSE only marginally (0.1245, approximately +4.6%), but affects clients asymmetrically. While the worst-case residence experiences a degradation of roughly 12%, the best-performing residences show small improvements. This pattern shows how FedProx pushes local models toward a common direction, and

Table 8 – Per-residence MAE and RMSE for the unbalanced 7-client HUE group under different μ values. Relevant best (low) and worst (high) values are highlighted.

	$\mu = 0$		$\mu = 0.01$		$\mu = 0.005$	
Residence	MAE	RMSE	MAE	RMSE	MAE	RMSE
15	0.1875	**0.3242**	0.2247	**0.3812**	0.2084	**0.3627**
17	0.0603	0.0841	0.0651	0.0888	0.0641	0.0872
21	0.0150	0.0388	0.0156	0.0396	0.0150	0.0382
22	0.0532	0.0922	0.0563	0.0937	0.0536	0.0913
26	**0.0190**	**0.0361**	**0.0181**	**0.0358**	**0.0172**	**0.0344**
27	0.0606	0.1017	0.0676	0.1117	0.0641	0.1052
28	0.1170	0.1558	0.1210	0.1574	0.1163	0.1523
Mean	0.0732	0.1190	0.0812	0.1297	0.0770	0.1245

that direction ends up being shaped by the clients with cleaner and more stable data. Since this dataset is smoothed and uses a one-hour horizon, it hides a good part of the volatility we would see in a real short-interval forecasting scenario. With that in mind, maintaining a non-zero μ is a deliberate stability lever, especially important when the system faces higher-frequency and noisier conditions. In environments with higher temporal variability or additional noise sources (including the Gaussian perturbations introduced in later experiments), a non-null proximal term is likely to mitigate client drift and improve robustness. Thus, although $\mu = 0$ delivers the lowest error in this controlled setting, the evidence from heterogeneity metrics and the expected increase in data volatility supports adopting a moderate and non-zero proximal term to provide an appropriate balance between predictive accuracy and stable behaviour under heterogeneous and noise-sensitive conditions.

5.2.1.2 UK-DALE Dataset

Table 9 – Heterogeneity metrics per UK-DALE client.

Client ID	N Samples	Target Variance	DTW	Wasserstein
1	39088	0.0547	0.7990	0.0083
2	5626	0.0200	2.3401	0.1226
3	942	0.2000	1.7844	0.1277
4	4932	0.0425	1.3411	0.0673
5	3286	0.0529	3.6839	0.2455

Figure 8 displays the distribution of raw *energy_kwh* values per client. Consistent with the variance results, Client 3 shows a wider interquartile range and a larger spread of consumption values. High-consumption outliers in Clients 1, 2, 4, and 5 contribute to the wider distributional divergences captured by the Wasserstein distance.

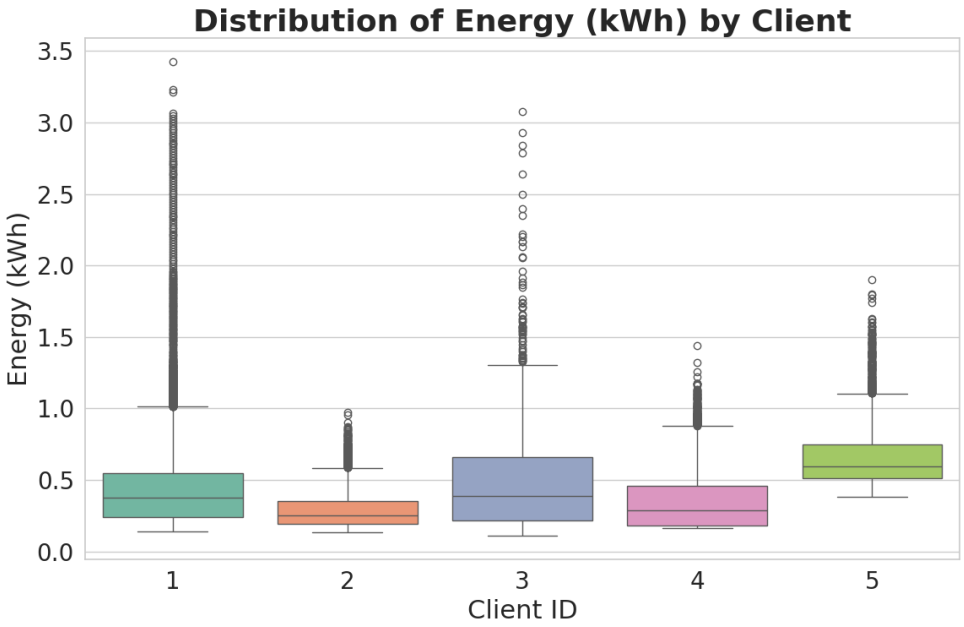


Figure 8 – Boxplots of energy consumption (kWh) by client.

The UK-DALE group includes five clients with markedly uneven data volumes and distributional characteristics. Residence 1 contributes almost forty thousand samples and shows minimal temporal and marginal divergence (DTW 0.799; Wasserstein 0.0083). In contrast, Residences 3 and 5 combine small sample counts with high variance and the largest distributional mismatch. This imbalance directly translates into the heterogeneous forecasting performance observed across clients.

Introducing a proximal term modifies this behaviour in predictable ways. The unregularized model ($\mu = 0$) achieves a mean RMSE of 0.1041, while a very small $\mu = 0.005$ improves the performance of all clients and reduces the mean error to 0.1001. In this dataset, such a light constraint is sufficient to smooth irregular local updates without suppressing useful client-specific structure. Increasing the proximal weight to $\mu = 0.01$ pushes the system into a more restrictive regime and uniformly increases RMSE, indicating a level of regularization that is too strong relative to the noise and volatility present in this version of UK-DALE.

Overall, these results indicate that, for this dataset and noise regime, smaller μ values achieve the most favourable balance between stability and flexibility. At the same time, they highlight that the effect of μ is inherently tied to the operating conditions of the system. In later sections, where differential privacy noise and adversarial robustness are evaluated, a larger proximal term can become beneficial by providing an additional stability mechanism against noisy or malicious updates. The UK-DALE analysis therefore serves as a baseline for interpreting those subsequent design choices.

Table 10 – Per-client RMSE for UK-DALE under different μ values.

	RMSE ($\mu = 0$)	RMSE ($\mu = 0.01$)	RMSE ($\mu = 0.005$)
C1	0.1094	0.1208	**0.1025**
C2	0.0920	0.0976	**0.0903**
C3	0.0871	0.0954	**0.0823**
C4	0.1034	0.1093	**0.0999**
C5	0.1284	0.1392	**0.1256**
Mean	0.1041	0.1124	**0.1001**

5.2.2 FedProx under Poisoning Attacks

To assess the robustness of FedProx in adversarial settings, a poisoning attack was evaluated in the five-client unbalanced HUE dataset scenario (Residences 17, 21, 26, 27, and 28). One client (Residence 17) was designated as the attacker, while the remaining clients behaved normally. The objective of the attacker was to manipulate the aggregation dynamics and deviate the global model from its natural convergence path.

Specifically, the attacker employed a label-offset poisoning attack, in which the local training targets were systematically shifted by a fixed offset during local optimization, inducing biased gradients while preserving plausible update magnitudes. The attack was applied to 70% of the attacker's local training batches and executed for two local epochs per round, using a reduced learning rate of 10^{-3} to avoid abrupt deviations. The offset magnitude was set to $k = 3.0$, scaled by the empirical standard deviation of the target variable ($\sigma_y = 1.0$), ensuring that the poisoned labels remained within a realistic range. This configuration produces poisoned updates that are difficult to distinguish from benign ones based solely on magnitude, while gradually steering the global model away from its natural convergence trajectory.

To assess the robustness of FedProx in adversarial settings, we evaluated a poisoning attack in the five-client unbalanced HUE dataset scenario (Residences 17, 21, 26, 27, and 28). One client (Residence 17) was designated as the attacker, while the remaining clients behaved normally. The objective of the attacker was to manipulate the aggregation dynamics and deviate the global model from its natural convergence path.

Figure 9 summarizes the effect of the proximal term. Under FedAvg ($\mu = 0$), the global model initially follows a normal convergence trajectory, but as rounds progress, the attack begins to dominate the aggregation. The global MSE rises steadily after approximately 8–10 rounds and continues to drift upward, ultimately failing to stabilize. This slow divergence trajectory reflects the cumulative impact of the poisoned updates in an unconstrained optimization regime.

In contrast, both FedProx configurations ($\mu = 0.01$ and $\mu = 0.1$) prevent this deterioration entirely. The global MSE remains controlled throughout training and con-

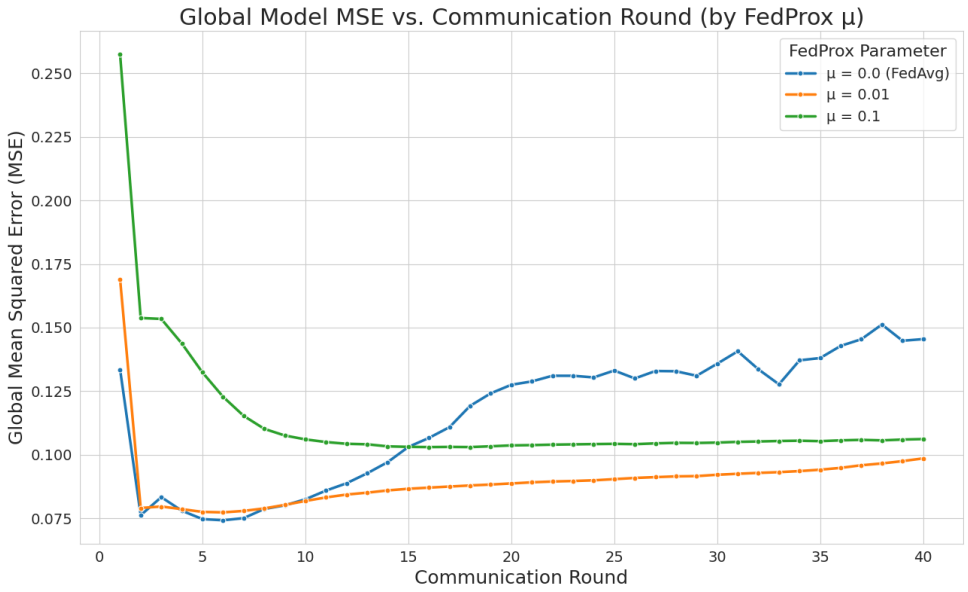


Figure 9 – Effect of μ on global model MSE under poisoning. FedAvg ($\mu = 0$) becomes increasingly unstable after early rounds, while $\mu \in \{0.01, 0.1\}$ maintain stable convergence.

verges to stable values. The proximal term effectively limits the influence of abnormal local updates and prevents the attack from steering the global model away from its natural descent path.

This stability advantage is reflected directly in the final per-client metrics. The FedAvg run, which suffered from model drift, exhibits the worst average performance (RMSE 0.2149, MAE 0.1902). Introducing a proximal term substantially improves robustness: $\mu = 0.01$ achieves the best overall results (RMSE 0.1762, MAE 0.1600), and $\mu = 0.1$ maintains comparable accuracy (RMSE 0.1827, MAE 0.1643). These results indicate that a small proximal penalty is sufficient to mitigate poisoning without overly constraining local optimization.

Table 11 – Mean L2 norms of client updates under poisoning for the unbalanced HUE dataset group (Residences 17, 21, 26, 27, 28).

μ	Attacker L2 Norm	Median Benign L2 Norm
0.00	0.2891	3.7101
0.01	0.3353	0.5644
0.10	0.3266	0.3146

Table 11 quantifies this effect: while the attacker's update remains small across all settings, the benign updates collapse from an L2 norm of 3.71 under FedAvg to 0.56 and 0.31 under $\mu = 0.01$ and $\mu = 0.1$, respectively, illustrating how the proximal term compresses variance and prevents poisoned updates from steering the global model.

It is important to note that FedProx is not a Byzantine-robust defense and does

not provide formal protection against poisoning. An attacker unconstrained by the proximal objective can still degrade the system. The experiment presented here shows that, in this specific scenario, the proximal term incidentally limits the impact of poisoned updates by reducing variance and stabilizing the aggregation. Even a modest value such as $\mu = 0.01$ is sufficient to preserve convergence and prevent drift, which makes a non-zero proximal term a practical choice for the differentially private and attack-sensitive configurations examined in the following sections.

5.3 Federated Results

For the federated experiments, models were trained for 40 communication rounds with 5 local epochs per client per round. Given the previous discussion, the federated optimizer was configured as FedProx with a proximal term of $\mu = 0.01$, as motivated in Section 5.2.

5.3.1 HUE dataset

5.3.1.1 LSTM

Table 12 summarizes the final global performance for all federated configurations evaluated on the HUE dataset. The results follow a clear pattern: smaller groups tend to achieve lower errors, and structural imbalance consistently makes the task harder. The balanced five-client group delivers the strongest MAE, while the unbalanced five-client and seven-client settings underperform due to the combination of limited data and uneven client distributions. The ten-client group performs better than both, not because it is more homogeneous, but because the additional clients introduce more useful signal without amplifying the extremes present in the smaller unbalanced sets. The full 28-client configuration remains competitive and serves as a realistic reference point for deployments where scale, variability, and generalization across diverse households are central requirements.

Table 12 – Final Global Model Performance Metrics (Mean Only) for all HUE Client Scenarios

Scenario	**RMSE**	**MAE**	**sMAPE**
Federated Learning Scenarios			
N=5 Balanced	0.0984	**0.0526**	8.21
N=5 Unbalanced	0.1171	0.0709	13.31
N=7	0.1226	0.0766	12.99
N=10	0.1197	0.0759	11.78
N=28 ($\phi = 1.0$)	0.1172	0.0851	7.23

5.3.1.2 Impact of Client Sampling (ϕ) on the 28-Client Group

Table 13 and Figure 10 evaluate the effect of sampling only a fraction of clients at each round. All configurations converge quickly and remain stable, confirming that the system maintains reliability even when participation is reduced.

Table 13 – Final Global Model Performance Metrics (Mean Only) on 28-Client HUE Dataset

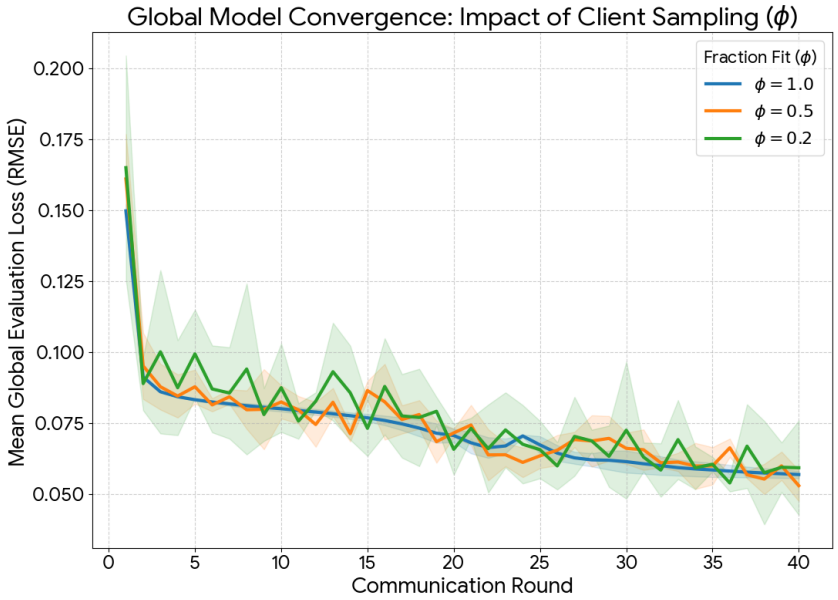
Fraction Fit (ϕ)	**RMSE**	**MAE**	**sMAPE**
1.0	0.1172	0.0851	7.23
0.5	0.1251	0.0806	12.50
0.2	0.1210	0.0770	7.73

Figure 10a shows that $\phi = 1.0$ reaches the lowest final RMSE, although the difference relative to $\phi = 0.2$ is small. This is a practical result: reducing communication does not materially degrade global utility. The $\phi = 0.5$ curve stabilizes slightly above the others but remains within the same overall band.

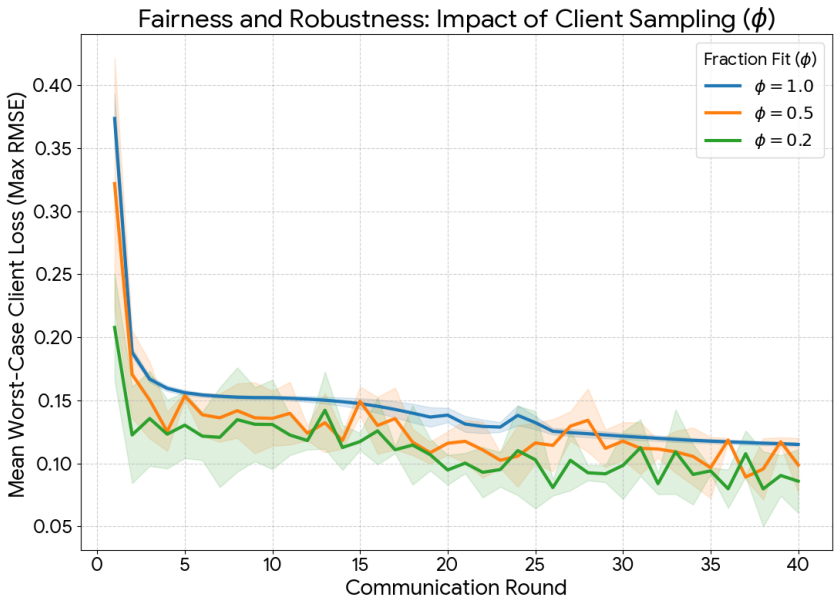
The fairness analysis in Figure 10b brings an additional layer of insight. The lowest sampling rate ($\phi = 0.2$) often yields the best outcomes for the most challenging clients, resulting in the lowest worst-case RMSE across a large portion of training. This indicates that full participation is not necessarily optimal. Subsampling can reduce the repeated influence of difficult or noisy local updates and prevent a small subset of clients from continually dragging the global model toward adverse directions.

Across all sampling fractions, the tight confidence bands demonstrate that federated optimization remains stable in the 28-client setting. These results confirm the feasibility of operating with substantially reduced communication budgets while preserving both utility and fairness.

The results in Figure 10 demonstrate the trade-off in client sampling. As shown in Figure 10a, while all fractions converge rapidly, the highest participation rate ($\phi = 1.0$) achieves the best final mean performance (lowest RMSE in Table 13), followed closely by $\phi = 0.2$. This confirms that reducing communication costs by sampling clients does not severely degrade the final utility. Figure 10b illustrates the model's stability. Intriguingly, the lowest sampling fraction ($\phi = 0.2$) often results in the best performance for the most difficult clients (lowest max loss). This suggests that selective participation might prevent bad local updates from consistently harming the performance of the worst-performing clients. The narrow standard deviation bands across all curves confirm the high stability of the federated training process.



(a) Global Model Convergence (Mean Loss)



(b) Worst-Case Client Performance (Max Mean Loss ± STD)

Figure 10 – Detailed Analysis of Client Sampling Fraction (ϕ) on the 28-Client Set. Plot (a) shows the overall model utility trend, and Plot (b) shows the robustness by tracking the client with the highest loss in each round.

5.3.1.3 Transformer

The final Transformer results for the HUE dataset are presented in Table 14. Unlike the LSTM, where the balanced five-client group consistently delivered the strongest performance, the Transformer behaves differently under varying client configurations. The lowest absolute errors (RMSE and MAE) occur in the five-client unbalanced setting, suggesting that the Transformer can capitalize on additional signal even when the distribution is uneven, as long as the group remains small. For relative error (sMAPE), however, the full 28-client configuration with $\phi = 1.0$ provides the best outcome, indicating that a larger and more diverse set of households improves the model's ability to generalize proportional deviations.

Table 14 – Final Global Model Performance Metrics (Mean Only) for the Transformer Architecture on HUE Dataset

Scenario	RMSE	MAE	sMAPE
N=5 Balanced	0.1293	0.0869	11.23
N=5 Unbalanced	0.1060	0.0712	15.54
N=7 Unbalanced	0.1649	0.0986	17.62
N=10 Unbalanced	0.1924	0.1226	13.96
N=28 ($\phi = 1.0$)	0.1335	0.0899	11.51
N=28 ($\phi = 0.5$)	0.1363	0.0919	11.77
N=28 ($\phi = 0.2$)	0.1386	0.0941	12.31

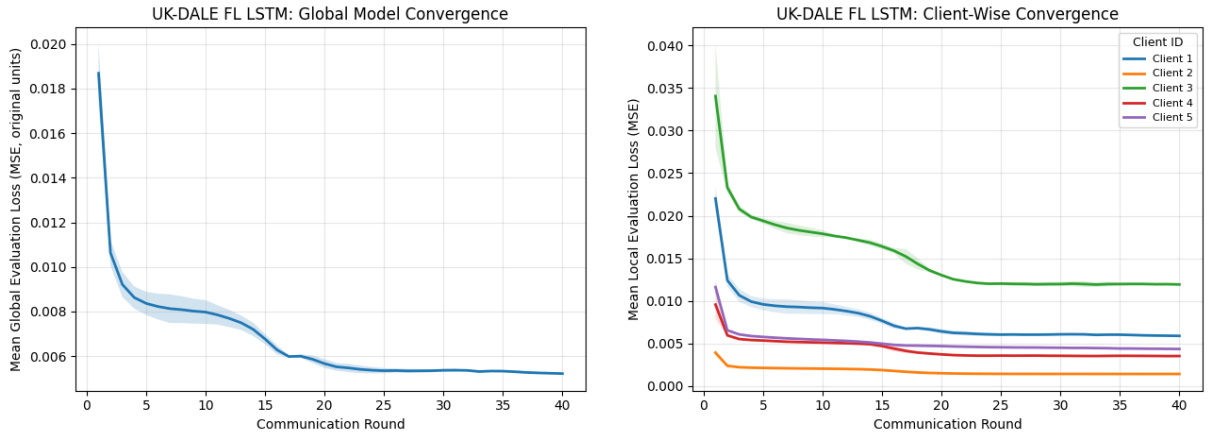
Overall, the Transformer shows higher sensitivity to structural imbalance and does not match the LSTM's performance in this short-term load forecasting task. This reproduces the behaviour observed in the centralized experiments, where the LSTM consistently outperformed the Transformer across all metrics.

5.3.2 UK-DALE

5.3.2.1 LSTM

Table 15 – UK-DALE FL LSTM: Final Performance Metrics

Client ID	RMSE	MAE	sMAPE
C1	0.0768	0.0407	8.20
C2	0.0378	0.0225	7.51
C3	**0.1092**	**0.0656**	**14.62**
C4	0.0594	0.0391	9.87
C5	0.0660	0.0404	5.95
Mean	0.0699	0.0417	9.23



(a) Global Model Convergence (Mean Loss ± STD)

(b) Client-Wise Convergence (Mean Loss ± STD)

Figure 11 – UK-DALE FL LSTM convergence analysis. Plot (a) shows the aggregated convergence behavior; plot (b) highlights the per-client learning curves.

Figure 11 shows the FL LSTM performance for the five UK-DALE households. The global curve converges smoothly and stabilizes early, with most of the loss reduction occurring in the first 10–15 rounds. Variance across runs remains narrow, indicating that the optimization process is stable despite the uneven sample counts and consumption profiles in Table 9.

The client-wise trajectories in Figure 11b make the effects of data heterogeneity explicit. Client 3, as discussed in Section 5.2.1.2, has the smallest number of samples, combined with high target variance and the second largest distributional mismatch. This client consistently represents the most challenging case: its evaluation loss remains the highest throughout the entire training process and stabilizes at a substantially worse level than the other clients.

Client 1 contains the largest volume of data and exhibits strong temporal variability. While it converges to a lower loss than Client 3, it converges more slowly and stabilizes at higher error levels when compared to the remaining clients. Clients 4 and 5 fall in an intermediate regime, reflecting moderate dataset sizes and distributional differences. In contrast, Client 2, which has the second largest number of samples and the lowest target variance among the group, converges to the lowest loss values across all rounds.

These performance differences closely mirror the heterogeneity metrics reported in Table 9. While the federated optimization process remains stable for all clients, the observed performance gap is driven by disparities in local dataset size and distributional mismatch, rather than by optimization instability.

Overall, the LSTM model exhibits stable convergence and competitive accuracy on UK-DALE, with Client 2 effectively defining a lower bound on achievable error for the shared global model in this setting.

5.3.2.2 Transformer

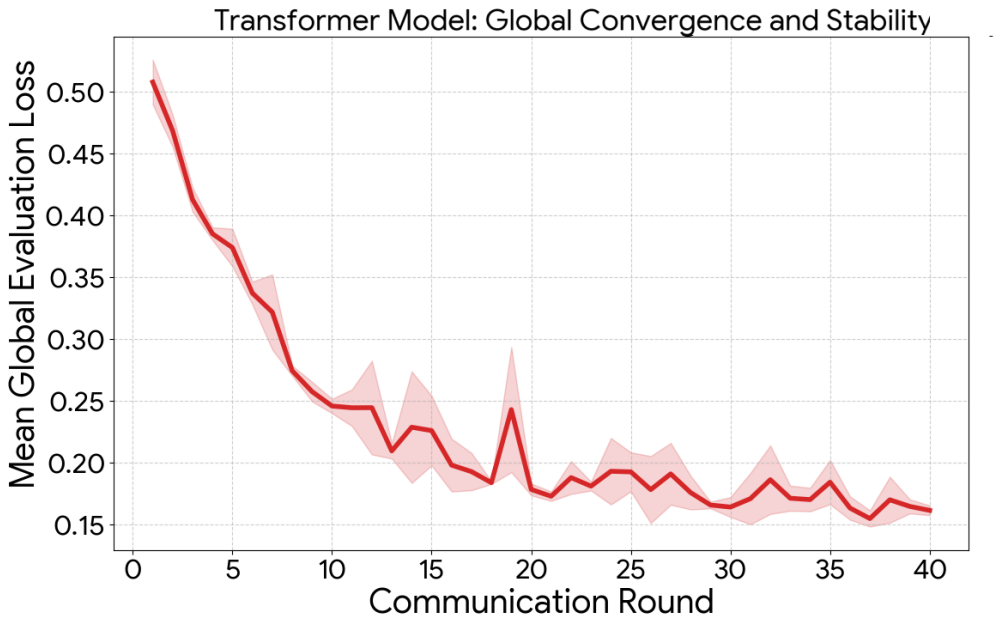


Figure 12 – Transformer Global Convergence on UK-DALE. The shaded band represents the standard deviation across runs.

Table 16 – UK-DALE Transformer: Final Performance Metrics

Client ID	RMSE	MAE	sMAPE
C1	0.0768	0.0407	8.20
C2	**0.0378**	**0.0225**	7.51
C3	**0.1092**	**0.0656**	**14.62**
C4	0.0594	0.0391	9.87
C5	0.0660	0.0404	**5.95**
Mean	0.0699	0.0417	9.23

The Transformer model also converges reliably on UK-DALE, following a similar pattern to the LSTM. The model achieves most of its loss reduction in the initial rounds, then stabilizes with narrow variance bands across runs. The per-client results mirror the LSTM behavior: Residences 2 and 5 are the easiest, while Residence 3 remains the dominant source of error.

Across all metrics, the Transformer underperforms the LSTM in this application. Given the short forecasting horizon and limited temporal complexity of the UK-DALE setup, the LSTM remains the most effective architecture for this dataset.

5.4 Privacy-Utility Trade-off

This section evaluates the interaction between differential privacy and model utility across all federated LSTM configurations on the HUE dataset. The analysis integrates the empirical with the privacy accounting method implemented in this work. Particular attention is given to the effect of group size, client subsampling, and the implications of applying local differential privacy (LDP) at every SGD step. For the FL-LDP implementation, the LSTM model was chosen and 10 federation rounds were executed with 2 local epochs for each round.

5.4.1 HUE Results

The results in Table 17 reveal a consistent pattern across all configurations: small or structurally unbalanced groups are the most sensitive to strong privacy budgets, while larger groups maintain substantially higher utility even under aggressive noise levels. In the balanced five-client case, utility remains above 90% once $\varepsilon \geq 4$, and approaches non-private performance at $\varepsilon = 7$ and $\varepsilon = 10$. In contrast, the unbalanced five- and seven-client groups deteriorate sharply at $\varepsilon = 1$, with the unbalanced five-client setting dropping to negative utility.

This behaviour is expected under local differential privacy. These unbalanced groups include households whose dataset sizes are one order of magnitude smaller than the largest client in the same group. Because LDP injects noise at *every* local update, clients with fewer samples accumulate all privacy noise over a very small number of SGD steps, making their local models significantly noisier than those of larger clients. When aggregated, these unstable updates distort the global model and amplify variance round after round. The effect is particularly pronounced at $\varepsilon = 1$, where noise levels are already high and the smaller clients effectively operate under a stricter privacy regime than intended. In contrast, the balanced five-client group (with more homogeneous data volumes) avoids this failure mode and remains stable even at $\varepsilon = 1$.

Once the system scales, the behaviour changes entirely. Table 18 shows that the 28-client configuration is consistently resilient. Even under the strongest privacy setting tested ($\varepsilon = 1$, $\delta = 10^{-5}$), the system maintains more than 83% of its non-private utility and recovers fully as ε relaxes. The 14-client and 6-client subsampled variants follow the same trend: utility remains above 81% at $\varepsilon = 1$ and converges almost perfectly for $\varepsilon \geq 5$. Larger populations absorb DP noise more effectively and stabilize aggregation, which prevents the noise amplification observed in small groups.

Client subsampling further helps preserve performance. Reducing the number of active clients per round cuts communication and computation, but it also limits the number of DP-noised updates generated during training. This introduces a natural amplification effect: clients participate less often and therefore accumulate smaller privacy loss, while the

Table 17 – FL-LDP-LSTM across client groups at different privacy levels ($\delta = 1 \times 10^{-6}$), alternative selection.

Group	Privacy level (ε, δ)	MAE	RMSE	Utility (%)
5 Clients (Balanced)	No DP ($\varepsilon = \infty$)	0.089	0.137	100.0
	$(1, 1 \times 10^{-6})$	0.121	0.180	68.37
	$(4, 1 \times 10^{-6})$	0.099	0.150	90.37
	$(7, 1 \times 10^{-6})$	0.092	0.142	96.08
	$(10, 1 \times 10^{-6})$	0.090	0.138	99.58
5 Clients (Unbalanced)	No DP ($\varepsilon = \infty$)	0.062	0.093	100.0
	$(1, 1 \times 10^{-6})$	0.151	0.219	-35.08
	$(4, 1 \times 10^{-6})$	0.108	0.157	32.00
	$(7, 1 \times 10^{-6})$	0.102	0.139	51.23
	$(10, 1 \times 10^{-6})$	0.097	0.131	60.00
7 Clients (Unbalanced)	No DP ($\varepsilon = \infty$)	0.091	0.144	100.0
	$(1, 1 \times 10^{-6})$	0.192	0.301	-9.64
	$(4, 1 \times 10^{-6})$	0.132	0.205	57.02
	$(7, 1 \times 10^{-6})$	0.122	0.190	67.83
	$(10, 1 \times 10^{-6})$	0.117	0.181	73.92
10 Clients (Unbalanced)	No DP ($\varepsilon = \infty$)	0.093	0.143	100.0
	$(1, 1 \times 10^{-6})$	0.134	0.212	51.45
	$(4, 1 \times 10^{-6})$	0.106	0.164	85.21
	$(7, 1 \times 10^{-6})$	0.099	0.154	91.75
	$(10, 1 \times 10^{-6})$	0.095	0.149	95.33

Table 18 – FL-LDP-LSTM at different privacy levels ($\delta = 1 \times 10^{-5}$) when training on subsets of 28 clients.

Group	Privacy level (ε, δ)	MAE	RMSE	Utility (%)
100% (28 clients)	No DP ($\varepsilon = \infty$)	0.095	0.143	100.0
	$(1, 1 \times 10^{-5})$	0.108	0.166	83.73
	$(5, 1 \times 10^{-5})$	0.095	0.148	96.94
	$(7, 1 \times 10^{-5})$	0.093	0.145	98.83
	$(10, 1 \times 10^{-5})$	0.094	0.144	99.13
50% (14 clients)	No DP ($\varepsilon = \infty$)	0.096	0.144	100.0
	$(1, 1 \times 10^{-5})$	0.113	0.171	81.06
	$(5, 1 \times 10^{-5})$	0.097	0.150	96.04
	$(7, 1 \times 10^{-5})$	0.094	0.145	99.26
	$(10, 1 \times 10^{-5})$	0.093	0.144	100.14
20% (6 clients)	No DP ($\varepsilon = \infty$)	0.094	0.144	100.0
	$(1, 1 \times 10^{-5})$	0.110	0.168	85.71
	$(5, 1 \times 10^{-5})$	0.095	0.147	99.86
	$(7, 1 \times 10^{-5})$	0.093	0.145	99.16
	$(10, 1 \times 10^{-5})$	0.093	0.144	99.70

global model still benefits from diverse updates across rounds. Empirically, all subsampled configurations outperform the small unbalanced groups at $\varepsilon = 1$, and every setting with $\varepsilon \geq 5$ achieves stable high utility.

As a whole, the results demonstrate that LDP is feasible for short-term load forecasting when applied at scale. Small or highly uneven groups struggle under strong privacy, but once the system reaches 10-28 clients, the model absorbs DP noise efficiently and maintains accuracy across all sampling rates. The 28-client results, in particular, show that strong privacy and high utility can coexist when training is distributed over a sufficiently large and diverse population.

5.4.2 UK-DALE Results

The UK-DALE results in Table 19 show that the federated LDP setup maintains relatively strong utility across all privacy levels. Even under the strictest setting tested ($\varepsilon = 1$), the system retains more than 60% of its non-private performance, and utility approaches the non-private baseline once $\varepsilon \geq 7$. This behaviour differs from the degradation observed in the small unbalanced HUE groups, where $\varepsilon = 1$ led to substantial loss in accuracy. Understanding this contrast helps clarify how dataset structure interacts with the LDP mechanism.

Table 19 – FL-LDP-LSTM on UK-DALE at different privacy levels ($\delta = 1 \times 10^{-6}$), averaged over five repetitions.

Privacy level (ε, δ)	MAE	RMSE	Utility (%)
No DP ($\varepsilon = \infty$)	0.0537	0.0837	100.00
$(1, 1 \times 10^{-6})$	0.0706	0.1148	62.89
$(4, 1 \times 10^{-6})$	0.0547	0.0932	88.65
$(7, 1 \times 10^{-6})$	0.0523	0.0886	94.23
$(10, 1 \times 10^{-6})$	0.0513	0.0870	96.07

Although UK-DALE is also unbalanced and includes a very small client (Client 3 with only 942 samples), the privacy mechanism produces a smaller impact on the aggregation than in HUE. One reason is the weighting imposed by the aggregation mechanism of FedProx (same as FedAvg): the smallest UK-DALE client contributes only a minor portion of the global update, so the DP-perturbed gradients from this client have limited effect on the global trajectory. In the unbalanced HUE subsets, the smallest clients are much larger (around 4,000 samples) and therefore hold enough aggregation weight for their noisy updates to influence the global parameters, which is especially visible when $\varepsilon = 1$.

A second factor is the gradient behaviour of the datasets. UK-DALE's hourly profiles, although heterogeneous, tend to produce smoother gradients that hit the clipping threshold less frequently. This reduces the relative influence of DP noise. In the unbalanced

HUE subsets, several clients have high-variance or irregular load curves; their gradients are clipped more often, which amplifies the effect of the added noise and destabilizes the optimization.

A third factor is the overall distribution of client sizes. Despite being unbalanced, UK-DALE includes four clients with more than 3,000 samples, meaning most participants perform enough DP-SGD steps to average the noise across updates. In HUE, the ratio between the largest and smallest clients is considerably wider, causing DP noise to accumulate unevenly and harming the global training dynamics.

Together, these elements explain why the UK-DALE configuration maintains higher utility under local differential privacy than the small unbalanced HUE groups. The weighting in aggregation, smoother gradient structure, and less extreme client-size distribution collectively reduce the sensitivity of the system to DP noise, enabling stable performance even under strong privacy budgets.

5.5 Comparison to Prior Work and Privacy Regimes

The earlier publication derived from this research (de Mellos et al. [2025]) reported substantially higher utilities across both datasets, but the methodological configuration used there differs in ways that make the results not directly comparable. That work applied an update-level LDP mechanism, injecting noise once per communication round and composing privacy at the granularity of rounds. The present dissertation adopts a stricter regime: Opacus' sample-level Privacy Engine clips and perturbs gradients at every mini-batch, which increases the number of compositions, tightens effective privacy budgets, and results in larger noise multipliers for the same (ε, δ) guarantee.

These differences, combined with the updated feature pipeline and the more explicit treatment of heterogeneity, fully explain the lower utilities observed here at small ε. The earlier results are included in Appendix A for historical context, but the present findings should be interpreted under this more conservative and granular privacy regime.

5.6 Overall Discussion of Findings

Across all experiments, centralized training establishes the empirical upper bound, reflecting the benefits of pooled data and full visibility of consumption variability. Under decentralization, FedAvg consistently recovers part of this performance, demonstrating the value of collaborative learning even in the presence of heterogeneity. FedProx improves stability in settings where local objectives diverge, reducing client drift and mitigating the effect of poisoned or noisy updates.

Under FL-LDP, sharper privacy budgets reduce accuracy as expected, but the results also show that moderate noise can act as a regularizer. In several configurations,

especially among larger client groups, utilities for $\varepsilon \in [4, 7]$ remain close to the non-private baselines, and in some cases exceed them within the same configuration due to reduced overfitting at the client level. Subsampling amplifies privacy while lowering computational and communication costs, and the 28-client HUE experiments illustrate that accuracy can remain stable even when only 20-50% of clients participate per round.

The comparison between HUE and UK-DALE highlights the dependence of privacy-utility behaviour on data volume, distributional properties, and temporal resolution. UK-DALE clients, despite being unbalanced, maintain high utility under strict privacy because even the smallest households contribute substantially more hourly data than the minority of volatile HUE clients. This difference underscores the role of data availability in how privacy noise spreads across updates.

Overall, the findings show that FL-LDP is feasible for STLF when applied at scale. Stability depends on heterogeneity, data volume, and the choice of optimizer, with FedProx and moderate privacy budgets offering the most reliable trade-offs. These considerations are essential when interpreting performance and when designing privacy-preserving forecasting systems for real deployments.

Chapter 6

Conclusions and Future Work

6.1 Summary of Contributions and Main Findings

This dissertation examined the feasibility of privacy-preserving FL for STLF under realistic data heterogeneity and strict user-level privacy guarantees. A complete sample-level local differential privacy (LDP) mechanism was implemented using Opacus' Privacy Engine, applying clipping and noise at every mini-batch to enforce conservative user-level protection. In parallel, the study introduced a structured analysis of FedProx across multiple client configurations, enabling a controlled investigation of its stabilizing effect under heterogeneous conditions.

A unified preprocessing pipeline ensured comparability across centralized, local-only, FL, and FL-LDP scenarios, while two deep learning architectures (LSTM and Transformer) were benchmarked under the same setup to establish upper-bound and lower-bound performance before introducing decentralization and privacy. Heterogeneity across households was quantified using DTW and Wasserstein distances, supporting a clear interpretation of client drift and distributional divergence.

Three overarching findings emerged from the empirical analysis. First, across both datasets, centralized training achieved the strongest accuracy, while local-only models highlighted the limitations of models trained independently, without benefiting from shared information across households. FedAvg provided intermediate performance, and FedProx improved stability in cases where households differed substantially in data volume or temporal structure.

Second, the privacy-utility trade-off depended strongly on group size and balance. Experiments were conducted across balanced 5-client groups, unbalanced 5-, 7-, and 10-client groups, and the full 28-client HUE population, as well as sampling fractions of 20%, 50%, and 100%. Small or highly unbalanced groups demonstrated the largest degradation at stricter privacy budgets because data-poor clients generated many DP-accounted updates relative to their sample size. In contrast, larger groups maintained high utility even at

$\varepsilon = 1$ and converged to the non-private baseline for $\varepsilon \geq 7$. Subsampling amplified privacy without harming accuracy, consistent with theoretical expectations.

Third, the poisoning experiment showed that even a modest proximal term counteracts model drift by constraining abnormal updates, maintaining stability relative to FedAvg. Although FedProx is not a Byzantine-robust defense, its regularization effect interacted positively with both heterogeneity and DP noise.

Overall, the results demonstrate that FL with LDP is technically viable for STLF when deployed at sufficient scale, and that a moderate proximal term provides an effective mechanism for stabilizing heterogeneous and noise-sensitive configurations.

6.2 Implications and Limitations

The findings have practical implications for privacy-preserving energy analytics. The results show that strong user-level privacy budgets can be achieved without substantial loss of accuracy, provided that the participating population is sufficiently large and not dominated by households with very small datasets. Sample-level DP ensures conservative guarantees aligned with regulatory expectations. The structured analysis across multiple group configurations emphasizes that both the number of clients and their balance critically influence convergence, stability, and the resulting utility.

FedProx proved to be a simple and computationally lightweight mechanism for mitigating client drift, especially important when temporal volatility or privacy noise is present. Its stabilizing effect suggests that practical FL deployments would benefit from proximal regularization even when data heterogeneity is mild.

Several limitations remain. All experiments used hourly and smoothed load profiles, which reduce volatility and may understate the challenges of high-frequency forecasting under privacy constraints. The models were univariate and therefore did not exploit exogenous variables such as weather, temperature, or occupancy. The privacy mechanism relied on a standard Gaussian mechanism without adaptations for temporal dependence, and only a single poisoning attack was evaluated. These choices define a controlled but simplified environment intended to isolate the effects of heterogeneity, privacy noise, and aggregation dynamics.

6.3 Future Directions

Several directions emerge naturally from this work:

- **Multivariate forecasting.** Incorporating exogenous signals such as weather, occupancy patterns, or seasonal structure would enable more accurate forecasting and offer insight into how auxiliary variables interact with privacy noise.

- **Alternative architectures.** Testing a wider range of sequence models, including Temporal Convolutional Networks, patch-based encoders, attention-free architectures, and new long-horizon forecasters, may improve robustness to clipped gradients and DP noise.
- **Time-series-aware privacy mechanisms.** Adapting DP to temporal dependencies, such as structured noise injection or sliding-window DP mechanisms, could reduce the privacy cost while maintaining strong guarantees.
- **Adversarial robustness.** Extending evaluation to privacy-related attacks, specially inference attacks, such as TS-Inverse Meijer et al. [2025a], to provide a more complete assessment of privacy leakage and help validate the framework under stronger adversarial conditions.
- **Higher-frequency data.** Testing the framework on sub-hourly or near real-time data would better represent practical smart-meter deployments and allow the study of privacy noise under high temporal volatility.
- **Scaling and sampling strategies.** Further investigation into how sampling fractions interact with privacy accounting and convergence may provide principled guidelines for large-scale deployments in utility networks.

Taken together, the experimental analysis across balanced and unbalanced groups, small and large client populations, and multiple sampling regimes establishes a foundation for future research in privacy-preserving federated energy forecasting and highlights several promising avenues for advancing model design, privacy mechanisms, and real-world applicability.

6.4 Publications Derived from this Research

The research developed throughout this dissertation has led to two peer-reviewed publications documenting distinct stages and extensions of the work:

- *A Federated Learning Approach for Distributed Solar Irradiance Forecasting* Camandaroba et al. [2025]. This paper extends the federated learning framework initially designed in this research to the domain of solar irradiance prediction, exploring its applicability to distributed renewable-energy forecasting. It was accepted and presented at the XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025).
- *Federated Short-Term Household Load Forecasting with FedProx and Differential Privacy* de Mellos et al. [2025]. This earlier publication reported an alternative

implementation of the proposed framework using an update-level local differential-privacy (LDP) mechanism. It was presented at the *3rd IEEE International Conference on Federated Learning Technologies and Applications (FLTA 2025)*, held in Dubrovnik, Croatia. The paper is referenced in Appendix A for historical completeness, as it employs a less stringent privacy regime and therefore yields results not directly comparable to those of the present dissertation.

References

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 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318. ACM, 2016. doi: 10.1145/2976749.2978318.

M. Akhtar et al. Short-term electricity load forecasting using hybrid deep learning models. *Energies*, 16(10):4060, 2023. doi: 10.3390/en16104060. URL `https://www.mdpi.com/1996-1073/16/10/4060`.

A. Alaqra, D. Gruss, and F. Skopik. Differential privacy for time series forecasting. *Journal of Information Security and Applications*, 54:102–529, 2020.

H. Alfares and M. Nazeeruddin. Electric load forecasting: Literature survey and classification of methods. *International Journal of Systems Science*, 33(1):23–34, 2002.

E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2938–2948, 2020.

J. Bai, Z. Liu, and Y. Li. Poisoning attacks in federated learning-based load forecasting. *IEEE Transactions on Smart Grid*, 13(6):5231–5244, 2022.

C. Beckel, L. Sadamori, and S. Santini. Revealing household characteristics from smart meter data. *Energy*, 78:397–410, 2014. doi: 10.1016/j.energy.2014.10.025.

D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases*, pages 359–370, 1994.

D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 634–643, 2019.

G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

C. Briggs, Z. Fan, and P. Andras. Federated Learning for Short-Term Residential Load Forecasting. *IEEE Open Access Journal of Power and Energy*, 9:573–583, 2022. ISSN 2687-7910. doi: 10.1109/OAJPE.2022.3206220.

G. Büscher, D. Düing, and A. Buchmann. Differentially private smart metering: Sufficiency and utility trade-offs. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–23, 2020.

C. G. M. Camandaroba, S. S. G. Werneck, J. Z. C. de Mellos, C. D. de Jesus, M. Malta, F. G. Guimarães, and E. P. de Aguiar. A federated learning approach for distributed solar irradiance forecasting. In *Proceedings of the XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025)*, Belo Horizonte, Brazil, 2025. Sociedade Brasileira de Computação. Accepted and presented at CBIC 2025.

C.-C. Chen, J.-L. Wang, and C.-J. Huang. Short-term load forecasting using decision tree based models. *Energy*, 35(9):3886–3891, 2010. doi: 10.1016/j.energy.2010.05.028.

M. Cho, J. Hwang, and B. Chen. Electric load forecasting by statistical and neural approaches. *IEEE Transactions on Power Systems*, 10(4):1892–1898, 1995.

J. Z. C. de Mellos, L. M. Astore, O. Orang, E. de Lima Pontes Andrade, and F. G. Guimarães. Federated short-term household load forecasting with fedprox and differential privacy. *The International Conference on Federated Learning Technologies and Applications (FLTA 2025)*, 2025.

M. A. Devlin and B. P. Hayes. Non-intrusive load monitoring and classification of activities of daily living using residential smart meter data. *IEEE Transactions on Consumer Electronics*, 65(3):339–348, 2019. doi: 10.1109/TCE.2019.2908952.

C. Dwork. Differential privacy. In *Proceedings of the 33rd international conference on Automata*, ICALP'06, pages 1–12, Berlin, Heidelberg, jul 2006. Springer-Verlag. ISBN 978-3-540-35907-4. doi: 10.1007/11787006_1.

C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers, 2014a.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014b. doi: 10.1561/0400000042.

G. Eibl, K. Bao, P.-W. Grassal, D. Bernau, and H. Schmeck. The influence of differential privacy on short term electric load forecasting. *Energy Informatics*, 1(1):48, oct 2018. ISSN 2520-8942. doi: 10.1186/s42162-018-0025-3.

R. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

M. Fang, X. Cao, J. Jia, and N. Z. Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*, pages 1605–1622, 2020.

H. Fereidooni, S. Marchal, and et al. Baffle: Backdoor detection and mitigation in federated learning for load forecasting. In *Proceedings of the 2021 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2021.

J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. Inverting gradients – how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 16937–16947, 2020.

G. Giaconi, D. Gunduz, and H. V. Poor. Smart meter data privacy: A survey. *IEEE Communications Surveys & Tutorials*, 22(4):2477–2510, 2020. doi: 10.1109/COMST.2020.3008655. URL `https://arxiv.org/abs/2009.01364`.

U. Greveler, B. Justus, and D. Loehr. Multimedia content identification through smart meter power usage profiles. In *2012 International Conference on Information and Communication Systems (ICICS)*, pages 1–6, 2012. doi: 10.1109/IACS.2012.6216922.

M. Hagan and S. Behr. The time series approach to short term load forecasting. *IEEE Transactions on Power Systems*, 2(3):785–791, 1987.

J. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

G. Hebrail and A. Berard. Individual Household Electric Power Consumption, 2006.

H. S. Hippert, C. E. Pedreira, and R. C. Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44–55, 2001. doi: 10.1109/59.910780.

B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 603–618, 2017.

T. Hong and S. Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016. doi: 10.1016/j.ijforecast.2015.11.011.

T. Hong and P. Wang. Artificial Intelligence for Load Forecasting: History. *IEEE Power and Energy Magazine*, pages 14 – 23, apr 2022. URL `https://ieeexplore.ieee.org/document/9761176`.

K. P. Inala, S. Gaddam, S. Etti, J. Karangula, and P. Kashetty. Enhancing the Power Load Prediction using LSTM in a Smart Grid Scenario for Household Data. In *2025 IEEE 1st International Conference on Smart and Sustainable Developments in Electrical Engineering (SSDEE)*, pages 01–06, feb 2025. doi: 10.1109/SSDEE64538.2025.10968113.

International Energy Agency. Unleashing the benefits of data for energy systems, 2023. URL `https://www.iea.org/commentaries/unleashing-the-benefits-of-data-for-energy-systems`. Accessed: 2025-08-29.

International Energy Agency. World energy investment 2024. Technical report, International Energy Agency, 2024. URL `https://www.iea.org/reports/world-energy-investment-2024`.

International Energy Agency. Global energy review 2025. Technical report, International Energy Agency, 2025a. URL `https://www.iea.org/reports/global-energy-review-2025`.

International Energy Agency. Smart grids, 2025b. URL `https://www.iea.org/energy-system/electricity/smart-grids`. Accessed: 2025-08-29.

M. S. Iqbal, M. Nadeem, M. Niazi, M. Ahmad, and M. W. Aslam. A hybrid deep learning framework for short-term load forecasting. *Energy Reports*, 10:3523–3535, 2024. doi: 10.1016/j.egyr.2024.03.157.

S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193. SPIE, 1997.

S. I. Kampezidou and S. Grijalva. Distribution transformers short-term load forecasting models. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–5, jul 2016. doi: 10.1109/PESGM.2016.7741174.

J. Kelly and W. Knottenbelt. The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific Data*, 2:150007, 2015. doi: 10.1038/sdata.2015.7.

W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2017. doi: 10.1109/TSG.2017.2753802.

B. Kulynych, M. Yaghini, G. Cherubin, M. Veale, and C. Troncoso. Disparate vulnerability to membership inference attacks. *Proceedings on Privacy Enhancing Technologies*, 2022 (1):460–480, 2022. doi: 10.2478/popets-2022-0023.

C. Kuster, Y. Rezgui, and M. Mourshed. Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, 35:257–270, nov 2017. ISSN 2210-6707. doi: 10.1016/j.scs.2017.08.009.

T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 2, pages 429–450, 2020a. URL `https://proceedings.mlsys.org/paper/2020/file/38af86134b65d0f10fe33d30dd76442e-Paper.pdf`.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020b. URL `https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf`.

L. Lyu, H. Yu, and Q. Yang. Threats to federated learning: A survey. *Information Fusion*, 79:337–357, 2022.

A. L'Heureux, K. Grolinger, and M. A. M. Capretz. Transformer-Based Model for Electrical Load Forecasting. *Energies*, 15(14):4993, jan 2022. ISSN 1996-1073. doi: 10.3390/en15144993.

S. Makonin. HUE: The hourly usage of energy dataset for buildings in British Columbia. *Data in Brief*, 23:103744, apr 2019. ISSN 2352-3409. doi: 10.1016/j.dib.2019.103744.

H. U. Manzoor, S. Hussain, D. Flynn, and A. Zoha. Centralised vs. decentralised federated load forecasting in smart buildings: Who holds the key to adversarial attack robustness? *Energy and Buildings*, 324:114871, dec 2024a. ISSN 03787788. doi: 10.1016/j.enbuild.2024.114871.

H. U. Manzoor, S. Hussain, D. W. Flynn, and A. Zoha. Centralised vs. decentralised federated load forecasting in smart buildings: Who holds the key to adversarial attack robustness? *Energy and Buildings*, 324:114871, 2024b. doi: 10.1016/j.enbuild.2024.114871.

D. L. Marino, K. Amarasinghe, and M. Manic. Building energy load forecasting using deep neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 418–425. IEEE, 2016. doi: 10.1109/IJCNN.2016.7727224.

E. McKenna, I. Richardson, and M. Thomson. Smart meter data: Balancing consumer privacy concerns with legitimate applications. *Energy Policy*, 41:807–814, 2012. doi: 10.1016/j.enpol.2011.11.049.

E. McKenna, I. Richardson, and M. Thomson. Privacy for smart meters: Towards undetectable appliance load signatures. *IEEE Transactions on Smart Grid*, 4(2):920–928, 2013. doi: 10.1109/TSG.2012.2229645.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*, pages 1273–1282, 2017.

C. Meijer, J. Huang, S. Sharma, E. Lazovik, and L. Y. Chen. TS-Inverse: A Gradient Inversion Attack Tailored for Federated Time Series Forecasting Models, mar 2025a. URL `https://arxiv.org/abs/2503.20952v1`. Accessed: Aug. 21, 2025.

D. Meijer et al. Ts-inverse: Gradient inversion attack against federated time-series forecasting. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, 2025b.

L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*, pages 691–706, 2019.

I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, 2017a.

I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017b. doi: 10.1109/CSF.2017.11.

A. Mohamed and M. Kamel. Short-term load forecasting using support vector regression. *Electric Power Systems Research*, 73(2):169–176, 2005. doi: 10.1016/j.epsr.2004.06.006.

M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

National Institute of Standards and Technology (NIST). Guidelines for evaluating differential privacy guarantees. Technical report, NIST, 2023. URL `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-226.ipd.pdf`.

H. Nie, G. Liu, X. Liu, and Y. Wang. Hybrid of arima and svms for short-term load forecasting. In *Energy Procedia*, volume 16, pages 1455–1460, 2012. doi: 10.1016/j.egypro.2012.01.230.

O. Orang, P. O. Lucas, G. I. F. Paiva, P. C. L. Silva, F. A. R. da Silva, A. A. Veloso, and F. G. Guimaraes. Causal graph fuzzy llms: A first introduction and applications in time series forecasting, 2025. URL `https://arxiv.org/abs/2507.17016`.

A. Papalexopoulos and T. Hesterberg. A regression-based approach to short-term system load forecasting. *IEEE Transactions on Power Systems*, 5(4):1535–1547, 1990.

G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

X. Qiu, Y. Liu, and W. Zhang. A short-term load forecasting model for smart grids based on temporal convolutional networks. *Mathematics*, 13(5):813, 2025. doi: 10.3390/math13050813. URL https://www.mdpi.com/2227-7390/13/5/813.

V. Ramirez et al. Smart grid data management: Ensuring security and privacy. Technical Report PNNL-36774, Pacific Northwest National Laboratory (PNNL), 2024. URL https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-36774.pdf.

J. Rieger, M. Scherbaum, and C. Wietfeld. Robustness against data integrity attacks in decentralized federated learning for short-term load forecasting. *Applied Energy*, 335: 120776, 2023.

H. Shi, M. Xu, and R. Li. Deep learning for household load forecasting—a novel pooling deep rnn. *IEEE Transactions on Smart Grid*, 9(5):5271–5280, 2017. doi: 10.1109/TSG.2017.2686012.

S. Shukla, S. Rajkumar, A. Sinha, M. Esha, K. Elango, and V. Sampath. Federated learning with differential privacy for breast cancer diagnosis enabling secure data sharing and model integrity | Scientific Reports, apr 2025. URL https://www.nature.com/articles/s41598-025-95858-2. Accessed: Aug. 21, 2025.

J. W. Taylor. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8):799–805, 2003.

H. Tong. *Non-linear Time Series: A Dynamical System Approach*. Oxford University Press, 1990.

U.S. Department of Energy. Grid modernization and the smart grid, 2025. URL https://www.energy.gov/oe/grid-modernization-and-smart-grid. Accessed: 2025-08-29.

E. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

C. Wang, W. Li, H. Zhang, and et al. Load forecasting method based on cnn and extended lstm. *Energy Reports*, 2024. doi: 10.1016/j.egyr.2024.XXXXXX. In press.

Y. Wang, Q. Chen, C. Kang, and Q. Xia. Ensemble learning based forecasting of residential short-term load with consideration of demand response. *Applied Energy*, 236:54–69, 2019. doi: 10.1016/j.apenergy.2018.11.045.

K. Yang, T. Fan, T. Chen, and Q. Yang. Federated learning with differential privacy: Algorithms and experiments. In *Proceedings of the 2020 IEEE International Conference on Big Data*, pages 685–694, 2020.

H. Yin, Z. Wang, G. Xu, L. Zhang, J. Xu, Z. Li, and P. Xie. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16337–16346, 2021.

A. Yousefpour, K. Ganju, C. Schumann, A. Deb, M. Khodak, K. Talwar, and A. Thakurta. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

C. Zhang, J. Xu, J. Zhou, et al. A review on electric vehicle load forecasting and charging demand management for sustainable power systems. *Renewable and Sustainable Energy Reviews*, 81:2879–2892, 2018. doi: 10.1016/j.rser.2017.06.009.

G. P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003. doi: 10.1016/S0925-2312(01)00702-0.

Y. Zhao, W. Xiao, L. Shuai, J. Luo, S. Yao, and M. Zhang. A Differential Privacy-enhanced Federated Learning Method for Short-Term Household Load Forecasting in Smart Grid. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 1399–1404, dec 2021. doi: 10.1109/ICCC54389.2021.9674514.

H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

K. Zhou, C. Fu, and S. Yang. Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews*, 56:215–225, apr 2016. ISSN 1364-0321. doi: 10.1016/j.rser.2015.11.050.

L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 14747–14756, 2019.

A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012. doi: 10.3390/s121216838.

Appendix A

Reference Results from Prior Publication

This appendix reproduces the results reported in the earlier publication associated with this research, "Federated Short-Term Household Load Forecasting with FedProx and Differential Privacy," presented at the International Conference on Federated Learning Technologies and Applications (FLTA 2025) de Mellos et al. [2025]. These results are included solely for historical completeness and should not be construed as baselines for the present dissertation. As discussed in Chapter 5, the prior study employed an update-level local differential privacy (LDP) mechanism, wherein each client submitted a single clipped and perturbed update per communication round. Noise was injected once per round, and privacy composition was performed at the level of communication rounds rather than mini-batches.

The earlier work also differed methodologically in the following key aspects:

- A distinct preprocessing and feature engineering pipeline: missing values were interpolated, a rolling mean filter with a window size of 5 was applied, lagged features (1h, 24h, 7d) and 7-day moving averages were generated, time-series windows were created using a 12-step look-back, an 80/20 chronological train-test split was adopted, and both features and targets were standardized using z-score normalization;
- Use of an update-level LDP mechanism (one clipped and noised update per communication round);
- Significantly fewer privacy compositions;
- Absence of sample-level differentially private (DP) training;
- A less stringent noise calibration procedure.

In contrast, the experiments presented in Chapter 5 adopt a substantially stricter privacy regime, employing Opacus' sample-level Privacy Engine with mini-batch gradient

clipping and per-batch Gaussian noise injection. This configuration yields a markedly different privacy-utility trade-off. Consequently, the numerical results reproduced below are not directly comparable to those reported in Chapters 4 and 5.

A.1 HUE Results from Prior Publication

Table 20 reproduces the performance of the FL–LDP–LSTM model on the HUE client groups as reported in the prior publication. Values correspond to average MAE, RMSE, and utility over five repetitions.

Table 20 – Performance of the FL–LDP–LSTM model at different privacy levels for each HUE client group in the prior publication. Values were averaged over five repetitions.

Group	Privacy level (ε, δ)	MAE	RMSE	Utility (%)
(1)	No DP ($\varepsilon = \infty$)	0.069	0.114	100.0
	(1, 1×10^{-5})	0.108	0.144	73.8
	(5, 1×10^{-5})	0.101	0.140	77.3
	(7, 1×10^{-5})	0.083	0.138	79.3
	(10, 1×10^{-5})	0.090	0.136	80.7
(2)	No DP ($\varepsilon = \infty$)	0.064	0.087	100.0
	(1, 1×10^{-5})	0.019	0.120	61.8
	(5, 1×10^{-5})	0.012	0.101	84.4
	(7, 1×10^{-5})	0.012	0.099	86.4
	(10, 1×10^{-5})	0.012	0.099	86.0
(3)	No DP ($\varepsilon = \infty$)	0.065	0.106	100.0
	(1, 1×10^{-5})	0.100	0.161	47.5
	(5, 1×10^{-5})	0.092	0.146	62.3
	(7, 1×10^{-5})	0.098	0.145	62.7
	(10, 1×10^{-5})	0.093	0.145	62.6
(4)	No DP ($\varepsilon = \infty$)	0.088	0.137	100.0
	(1, 1×10^{-5})	0.091	0.142	96.3
	(5, 1×10^{-5})	0.095	0.144	94.9
	(7, 1×10^{-5})	0.096	0.144	95.0
	(10, 1×10^{-5})	0.091	0.139	98.8

A.2 UK-DALE Results from Prior Publication

Table 21 reproduces the UK-DALE FL–LDP–LSTM results from the former study. These were computed under the same update-level DP mechanism described above and are likewise included only for archival completeness.

Table 21 – Performance of the FL–LDP–LSTM model at different privacy levels for the UK-DALE dataset in the prior publication. Values were averaged over five repetitions.

Privacy Level (ε, δ)	**MAE**	**RMSE**	**Utility (%)**
No DP ($\varepsilon = \infty$)	0.0708	0.1157	100.00
$(1, 1 \times 10^{-5})$	0.0594	0.1012	112.54
$(5, 1 \times 10^{-5})$	0.0525	0.0883	123.64
$(7, 1 \times 10^{-5})$	0.0559	0.0902	122.01
$(10, 1 \times 10^{-5})$	0.0537	0.0881	123.83