MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
MOHAMED KHIDER UNIVERSITY OF BISKRA
Faculty of Exact Sciences, Natural and Life Sciences
Computer Science Department

N° : ………/M2/2025

Dissertation

Presented to obtain the academic master's degree in Computer Science

Option : Information systems, optimization, and decision-making(SIOD)

Title: Modeling and Optimization of Energy Distribution Networks in Smart Grids Using Federated Learning

Presented by:

- Yousra Saadaoui
- Mira Benabdelkader

Supervisor : Dr. Ilyes Naidji

Co-supervisor : Dr. Ahmed Tibermacine

Jury :

– Dr. Meftah Zouai	MCB	President
– Dr. Asma Ammari	MCB	Examiner
– Dr. Ilyes Naidji	MCB	Supervisor

Academic Year : 2024-2025

Acknowledgements

First and foremost, I thank Allah, the Most Merciful, the One who never left me even when doubt crept into my heart. His light guided me through the darkest moments, and His mercy carried me when I could no longer carry myself. Alhamdulillah for the strength, the patience, and the dream fulfilled.
Then, I turn to myself with deep, heartfelt gratitude.
Thank you For every time you woke up tired but still chose to keep going. For the long nights, the frustration, the confusion, the fear of not being enough, and pushing through anyway. For not giving up when giving up felt easier. For continuing in silence, even when no one saw you. For being your own support, your own comfort, your own encouragement. Thank you for holding on to a dream that once felt far too big for your hands and still refusing to let it go. This achievement is not just a goal, it is a testament to the strength I didnt know I had. To my beloved family, I express my deepest gratitude. You have been my foundation, my courage, and my hope. Your prayers, words of encouragement, patience, and unconditional love carried me through the hardest times. Thank you for everything you are, for everything youve taught me, and for always believing in me even in silence. To my supervisor Dr. Naidji Ilyes, I offer my sincere thanks. Your thoughtful guidance, insightful advice, and constant support greatly enriched this work. Thank you for your kindness and availability throughout this journey. Finally, to my dear friends, faithful companions along the way, thank you for your presence, comforting words, calming smiles, and inspiring energy. Your support was a breath of fresh air when I was weary, and your friendship is a treasure I deeply cherish. This journey was long. It was painful. At times, it felt impossible. I lost things along the way time, energy, pieces of myself. But I gained something far greater: resilience, wisdom, and a victory I truly earned. Thank you, from the depths of my heart, to every version of me that brought me here.

YOUSRA

Acknowledgements

First and foremost, I thank Allah for granting me the strength and determination to complete this work. I would like to express my deep gratitude to my supervisor, Dr. Naidji Ilyes, for his constant guidance, valuable advice, great availability, and his admirable human qualities. I also sincerely thank the members of the jury for the honor of accepting to evaluate this work. My thanks also go to all the professors of the Computer Science Department for the education and support they provided throughout my academic journey. Finally, I extend my heartfelt thanks to my family for their unconditional support and to my friends who kindly helped review this document.

MIRA

Abstract

The transition to smart grids is revolutionizing the way energy is generated, distributed, and consumed, introducing both opportunities and challenges. This work addresses two critical issues in modern smart grid systems: electric load forecasting and energy theft detection, both of which are essential for efficient grid operation and security. Traditional centralized machine learning approaches, while effective, raise concerns about data privacy, scalability, and robustness in distributed environments. To overcome these limitations, this project proposes a Federated Learning framework that enables collaborative model training across multiple decentralized data sources without sharing raw data. We begin by exploring the architecture and challenges of smart grids, followed by a comprehensive state-of-the-art review on forecasting techniques, theft detection mechanisms, and recent advances in FL. The proposed methodology integrates real-world consumption data, thorough preprocessing, and tailored deep learning models, including LSTM and GRU, optimized for both forecasting and classification tasks. Several federated aggregation algorithms are implemented and evaluated using the Flower framework. Experimental results demonstrate that the federated models can achieve comparable, and in some cases superior, performance to centralized models, while preserving data privacy and improving generalizability. Moreover, mitigation strategies for non-IID data and client drift are explored to enhance FL robustness. This work highlights the feasibility and effectiveness of applying Federated Learning to critical smart grid applications and paves the way for privacy-aware, distributed intelligence in future energy systems.
Keywords: Smart Grids, Federated Learning, Load Forecasting, Energy Theft Detection, Data Privacy, LSTM, GRU, Non-IID Data, Distributed Machine Learning, Flower Framework.

Résumé

La transition vers les réseaux électriques intelligents révolutionne la manière dont lénergie est produite, distribuée et consommée, en introduisant à la fois des opportunités et des défis. Ce travail aborde deux problématiques essentielles des systèmes de réseaux intelligents modernes : la prévision de la charge électrique et la détection de la fraude énergétique, toutes deux cruciales pour assurer une exploitation efficace et sécurisée du réseau. Les approches traditionnelles dapprentissage automatique centralisé, bien quefficaces, soulèvent des préoccupations liées à la confidentialité des données, à la scalabilité et à la robustesse dans des environnements distribués. Pour surmonter ces limitations, ce projet propose un cadre basé sur lapprentissage fédéré qui permet lentraînement collaboratif de modèles à partir de multiples sources de données décentralisées, sans partage des données brutes. Nous commençons par explorer larchitecture et les défis des réseaux intelligents, suivis dune revue approfondie de létat de lart concernant les techniques de prévision, les mécanismes de détection de la fraude, ainsi que les avancées récentes en apprentissage fédéré. La méthodologie proposée intègre des données réelles de consommation, un prétraitement rigoureux et des modèles dapprentissage profond adaptés, notamment les réseaux LSTM et GRU, optimisés pour les tâches de prévision et de classification. Plusieurs algorithmes dagrégation fédérée sont implémentés et évalués à laide du framework Flower. Les résultats expérimentaux montrent que les modèles fédérés peuvent atteindre des performances comparables, voire supérieures, à celles des modèles centralisés, tout en préservant la confidentialité des données et en améliorant leur capacité de généralisation. En outre, des stratégies datténuation des effets liés à la non-IID des données et à la dérive des clients sont explorées afin de renforcer la robustesse de lapprentissage fédéré. Ce travail met en évidence la faisabilité et lefficacité de lapplication de lapprentissage fédéré aux applications critiques des réseaux intelligents et ouvre la voie à une intelligence distribuée respectueuse de la vie privée dans les systèmes énergétiques du futur.

Mots-clés : Réseaux électriques intelligents, Apprentissage fédéré, Prévision de la charge, Détection de fraude énergétique, Confidentialité des données, LSTM, GRU, Données non-IID, Apprentissage automatique distribué, Framework Flower.

Contents

List of Figures

General Introduction

The global energy landscape is undergoing a significant transformation driven by the increasing need for sustainable energy solutions. Traditional energy systems are facing growing challenges, particularly as renewable energy sources become more integrated into the grid. This transition requires advanced infrastructure capable of handling both the environmental impact of energy generation and the increasing electricity demand. Smart grids have emerged as a solution to these challenges, offering a modernized approach to electricity distribution. Unlike traditional grids, which rely on centralized power generation, smart grids leverage advanced information and communication technologies (ICT) to optimize energy distribution, integrate renewable sources, and enable real-time communication between utilities and consumers. This results in a more flexible, reliable, and efficient energy system. One of the main advantages of smart grids is their ability to manage the fluctuating nature of renewable energy. Solar and wind energy, for example, are subject to variability based on environmental factors such as weather and time of day. Smart grids can accommodate these fluctuations by integrating energy storage systems and employing sophisticated forecasting methods to ensure a consistent supply of power. Energy load forecasting plays a crucial role in the operation of smart grids. Accurate forecasting is necessary to balance supply and demand, minimize energy waste, and avoid grid instability. As the proportion of renewable energy in the grid increases, the complexity of load forecasting also increases. Advanced methodologies, such as machine learning and Federated Learning, are increasingly being used to improve the accuracy of these forecasts by analyzing large datasets from diverse sources. As the energy sector continues to evolve, the role of smart grids in enhancing the efficiency, reliability, and sustainability of the electrical grid becomes increasingly important. Through the application of cutting-edge technologies, smart grids are set to play a key role in the transition toward a more sustainable and resilient energy future.

Chapter 1

Introduction of Smart Grid

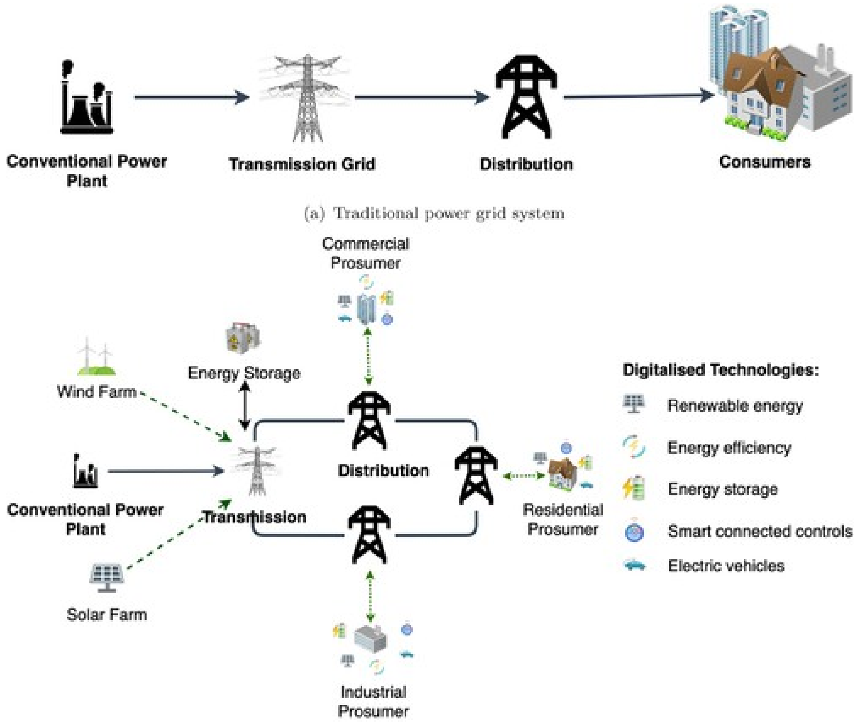
1.1 Introduction

Smart Grid represents a significant technological advancement in managing electrical networks. By integrating communication and automation technologies, they optimize energy production, distribution, and consumption, while facilitating the integration of renewable energy sources. This evolution allows for more efficient responses to demand fluctuations and addresses challenges related to the energy transition, such as the decentralization of production and improving system resilience.

1.2 From Traditional Grid to Smart Grid

- **Traditional Electrical Grid:**
  Conventional electrical grids, often referred to as centralized grids, have long been the backbone of electricity distribution. These systems rely on large-scale power plants, high-voltage transmission lines, and distribution networks to deliver electricity to end users. Operated by public or private utility companies, these grids function through a centralized model that facilitates widespread power distribution. However, this model also presents significant limitations, including vulnerability to blackouts, high maintenance costs, and minimal flexibility in managing energy production and consumption. With the growing integration of renewable energy sources and the shift toward decentralized energy systems, the traditional grid model is undergoing significant transformation [1].

- **Smart Grid:**
  Smart grid represents the modern evolution of electrical infrastructure, integrating digital technologies to enhance grid performance, reliability, and security. Through real-time monitoring and control enabled by sensors, communication systems, and advanced algorithms, smart grids allow for dynamic energy management and improved efficiency. These systems can quickly detect and respond to outages, reducing their impact, while also providing stronger protection against cyber threats through robust security measures. Unlike traditional grids, smart grids promote connectivity and communication across all components of the system, supporting decentralized energy sources and automated control. This interconnectivity facilitates adaptive, data-driven decision-making and enables automation of key grid operations, making energy distribution more intelligent and responsive to consumer demand [2].



(a) Traditional power grid system

(b) Smart grid system

Figure 1.1: Traditional Electrical Grids Vs Smart Grids

1.3 Modern Architecture of a Smart Grid

The architecture of a smart grid can be analyzed through four key functional layers:

1.3.1 Generation Level

The generation level represents the starting point of the energy flow in an electrical network. In a Smart Grid, this layer is no longer limited to large centralized power plants but also incorporates decentralized and renewable energy sources. Generation thus becomes more diverse, including solar, wind, hydro, and biomass energy. These sources, often intermittent and weather-dependent, present new challenges in terms of management and forecasting. However, they significantly reduce greenhouse gas emissions and enhance system resilience, especially when combined with energy storage solutions.

1.3.2 Transmission and Distribution Level

This level ensures the delivery of electricity from producers to consumers. In Smart Grids, transmission and distribution infrastructures are modernized through intelligent technologies that enable real-time monitoring of the grids status. Sensors and monitoring systems are deployed to quickly detect outages, balance loads, and prevent energy losses. Distribution thus becomes more dynamic and capable of adapting to instant demand and local production. The automation and digitalization of this layer improve the overall reliability and robustness of the electrical system.

1.3.3 Consumption Level

The consumption level represents the interaction between the grid and end users. In a Smart Grid, consumers become active participants in the energy system, especially with the implementation of smart meters. These devices continuously measure energy usage and transmit data to providers for real-time billing or demand-based supply adjustments. Users can also produce their own energy, for instance, via photovoltaic panels. This promotes more responsible, flexible, and grid-agnostic consumption.

1.3.4 Communication and Control Systems

Communication and control systems are the core components that interconnect all other levels. They enable fast and secure data exchange between producers, grid operators, and consumers. Technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and Cloud Computing are used to process large volumes of information, make automated decisions, and forecast grid behavior. These systems make it possible to detect anomalies, optimize production and consumption, and continually improve energy efficiency [3], [4]. Without this layer, the coordination and intelligence of the Smart Grid would not be possible.

1.4 Main Challenges of Smart Grid

1.4.1 Renewable Energy

Renewable Sources

Renewable energies, also known as green power sources, include wind, hydro, solar, biomass, as well as ocean and tidal energy. These environmentally friendly and widely available resources play a key role in the development of smart grids, which are designed to efficiently integrate and manage diverse energy sources to ensure a more sustainable and resilient power system [5].

- **Solar Energy:** Solar energy provides one of the most viable answers to meeting demand for clean, safe, reliable power. Solar radiation on the Earths land surface is more than 200 times higher than the total annual commercial energy consumption. Various incentive programs were initiated to promote the use of solar technology. For example, those who install solar panel systems on their homes can often sell excess electricity that their home generates back to the local utility grid [6].

- **Wind Energy:** Wind energy, generated by air movement caused by the uneven heating of the Earth by the sun, is an indirect form of solar power. Unlike solar panels, wind turbines can function under cloudy or rainy conditions. Typically installed on 30-meter towers to capture stronger winds, turbines are spaced 5 to 15 blade diameters apart to minimize interference. Wind energy is a clean, renewable source with no emissions or harmful byproducts, making it a sustainable solution for power generation [6].

- **Biomass Energy:** Biomass energy is derived from organic materials such as plant residues, animal dung, and municipal biodegradable waste. Biomass collects solar energy stored in chemical bonds and releases it in processes such as burning, fermentation, and anaerobic digestion. Waste is broken down by microorganisms in the absence of oxygen during digestion to give biogas, predominantly methane and carbon dioxide. It is an environmentally friendly process, cheap, and byproducts can be utilized as fertilizers [6].

- **Tidal Power:** Tidal energy captures the kinetic energy of ocean water movement to produce electricity. In optimal coastal locations, wave energy density can reach up to 65 MW per mile. One of the most efficient methods involves oscillating water columns, where incoming waves compress air in a vertical chamber to drive a turbine. Despite its potential, tidal power systems remain sensitive to extreme weather conditions such as cyclones and strong storms [6].

- **Geothermal Energy:** Geothermal energy taps into the Earth's internal heat. Ground source heat pumps can provide heating in winter and cooling in summer by exploiting the consistent underground temperature range of 10řC to 16řC (50řF to 60řF). This source is clean, sustainable, and available almost everywhere [6].

Challenges Related to Renewable Energy Integration

Integrating renewable energy into the Smart Grid (REI - Renewable Energy Integration) enhances the overall capability of the grid and helps meet growing consumer demand. Nevertheless, this integration process faces several significant challenges:

- **Storage:** Energy storage is a major challenge in modern electrical engineering. While current systems enable renewable energy integration, their efficiency and capacity need improvement to support widespread decentralized generation. Economically, the cost-effectiveness of storage depends on its role in generation, distribution, or consumer use. Renewable sources like solar and wind are intermittent, with solar energy unavailable at night. Effective energy storage systems are essential to ensure a stable and continuous power supply and prevent grid instability caused by supply gaps [7].

- **Voltage Fluctuations:** One of the major issues in integrating solar and wind energy is voltage fluctuation, which results from variations in solar radiation and wind speed. These fluctuations can lead to instability in power delivery and require advanced control mechanisms to manage effectively [8].

- **Environmental Impacts:** Although renewable sources are cleaner than traditional fuels, their large-scale deployment can still have environmental effects. For instance, land use for solar farms or the ecological impact of wind turbines must be considered during the integration process [8].

Overall, while renewable energy integration into Smart Grids presents many benefits, it also requires addressing technical, environmental, and infrastructural challenges to ensure efficiency and stability.

1.4.2 Load forecasting

Definition

Load forecasting is the process of estimating the amount of electricity that will be needed at a specific time and how this demand will impact the distribution network. It is used to ensure that a sufficient supply of energy is available to meet consumption needs while minimizing waste and inefficiency. It is a critical component of the operational planning of power systems and plays a vital role in preventing outages. Load forecasts can range from short-term to long-term. The accuracy of these forecasts directly affects the cost and reliability of the entire power supply system. Load forecasting is also part of a broader energy forecasting framework, which includes projections of the availability and cost of fuels like oil and gas, as well as renewable energy sources [9].

Differences Based on Time Scales

There are several methods for load forecasting, each analyzing historical data and relevant inputs to predict electricity demand over different time horizons.

- **Short-term load forecasting (up to one week) :**relies heavily on recent load data and weather forecasts. It supports real-time grid operations by helping system operators decide how much electricity to generate and where to route it. Accuracy is critical, as even small errors can cause energy waste or grid overloads [9].
- **Medium-term forecasting (one week to one year):**is used for maintenance planning and fuel reserve management. It considers seasonal demand variations and scheduled outages [9].
- **Long-term forecasting (beyond one year):** incorporates factors such as demographic shifts, economic growth, and energy policies. It focuses on system planning and investment decisions, especially regarding capacity expansion and balancing traditional vs. renewable energy sources [9].

Importance in Smart Grid Operation

Load forecasting is a fundamental component of smart grid operation, as it enables efficient planning, real-time decision making, and optimal resource allocation. Accurate demand forecasts, across various time horizons (short term, medium term, and long term), are essential for balancing electricity supply and demand, minimizing operational costs, and maintaining grid stability.

In traditional power systems, forecasting was primarily used for generation scheduling. However, in Smart Grids characterized by distributed energy resources and active consumer participation, load forecasting plays an even more strategic role. It supports:

- **Optimal Dispatch of Generation Units**: Precise forecasts help determine which energy sources should be activated or deactivated, reducing reliance on expensive reserve power.
- **Integration of Renewable Energy**: Since renewable sources like solar and wind are variable, accurate demand forecasting helps align generation with consumption, avoiding overproduction or shortages.

- **Demand Response and Grid Flexibility**: Smart Grids use forecasts to activate demand-side strategies, such as shifting or reducing consumption during peak hours.
- **Infrastructure Planning**: Long-term forecasts guide investments in grid expansion, reinforcement, or modernization.

By improving the accuracy and resolution of load forecasting through advanced data analytics and machine learning, Smart Grids become more intelligent, resilient, and capable of adapting to dynamic energy environments.

1.4.3 Energy theft

Unlawful Electricity Consumption and Its Impact on Smart Grid Integrity

Energy theft, often referred to as electricity or power theft, remains a critical issue for utility companies around the world. It consists of illegally accessing and using electricity without proper authorization or payment.

The motivations behind energy theft are diverse. Financial hardship often pushes individuals or businesses to steal electricity, while in other cases, organized crime exploits it for activities such as cryptocurrency mining or operating unauthorized facilities. Regardless of the reason, energy theft disrupts the integrity and proper functioning of the electrical grid.

The impacts of energy theft are far-reaching. Utility providers face significant financial losses, which hinder investments in infrastructure improvement and modernization. This degradation can lead to a decline in the reliability and resilience of power systems. Consumers ultimately bear part of the burden through increased energy prices meant to offset losses. Additionally, energy theft introduces serious safety hazards: unauthorized modifications to the electrical network can lead to fires, electrocution, and equipment failures [10].

In the context of Smart Grids, the issue becomes even more complex. The integration of advanced technologies provides new opportunities to detect and prevent theft, but it also introduces new vulnerabilities that malicious actors can exploit. Thus, energy theft remains one of the major challenges for the future of smart, sustainable, and secure power systems [11].

Techniques of Electric Energy Theft

Electric energy theft is a significant issue for energy suppliers, with reports indicating over 300 different methods used to illegally consume electricity. These techniques are often highly inventive and are employed by individuals to reduce their electricity bills. Collectors and meter readers are trained to identify these fraudulent activities, but some methods are challenging to detect. Below are the main categories of energy theft and the most common techniques [10]:

1. Classic Energy Theft

Classic energy theft primarily involves illegal connections made before the meter. This method is often seen in older buildings with outdated electrical systems. The goal is to connect additional installations to the line before the metering system. There are two types of theft in this category [12]:

- **Complete Use of an Illegal Source**: This involves consuming energy entirely without passing through the meter, which is easier to detect.
- **Partial Use of an Illegal Source**: The energy consumption is partial, making it harder to detect. This type of theft is often seasonal, such as during winter for heating purposes.

2. Modification of Meter Operation

This category includes several methods that alter the functioning of the electric meter. The most common techniques include [12]:

- **Tampering with Terminals or Bridging**: In a three-phase system, experienced consumers may disconnect the neutral wire, creating an imbalance in consumption, which causes the meter to understate the energy usage by as much as 30%. This type of fraud is difficult to detect unless specialized meters are used.
- **Magnetic Interference**: Strong magnets, often neodymium magnets, are used to slow down or stop the meters counting wheel, leading to inaccurate readings and lower bills.
- **Photographic Film**: A strip of photographic film is inserted between the back housing and the front glass of the meter. This blocks the movement of the wheel, effectively hiding the energy consumption without leaving visible traces, making detection difficult.
- **Drilling the Meter Housing**: A small hole, typically less than 1mm in diameter, is drilled into the meters housing, allowing for the insertion of elastic materials or other objects that interfere with the meters operation, reducing energy readings.

3. Modification of Digital Meter Software

This technique is more advanced and less common. It involves altering the software or variables stored in digital meters, specifically through manipulation of Object Identification System (OBIS) codes. These variables are responsible for tracking energy usage and billing. Changing these variables can affect the recorded consumption and, therefore reduce the amount billed. This form of fraud is rare because it requires high-level expertise and is time-consuming to execute [12].

4. Other Methods of Theft

In addition to the techniques mentioned above, there are other ways to steal electricity that include both physical tampering with the meter and illegal modifications to the electrical network [12]:

- **Illegal Network Modifications**: Professional thieves may install unauthorized branches on the electrical network, bypassing the meter entirely. These illegal connections are often hidden within walls or concealed in furniture. A well-known case involved a hidden socket installed in a nightstand to power a heater at night, avoiding the metered consumption.
- **Tampering with the Meters Bearings**: Some individuals unscrew the meters bearings to slow down the counting wheel, which decreases the recorded energy consumption.

5. Collusion with Employees

A more complex form of energy theft involves collusion with an employee of the energy supplier. The employee might assist in manipulating meter readings or in the collection and billing process, making this type of theft difficult to detect as it involves insider help [12].

1.4.4 Cybersecurity Threats

As smart grids become more interconnected and digitalized, they face growing cybersecurity threats. Hackers constantly seek to exploit vulnerabilities, putting the security and stability of energy networks at risk. Several forms of attacks threaten the smart grid. Hardware attacks involve tampering with smart meters or other devices to alter transmitted data, potentially leading to manipulated pricing or unauthorized access. Insider threats from employees leaking sensitive information further compromise system integrity [13].

Ransomware and malware represent another critical danger, where malicious software targets grid infrastructures, paralyzing systems until a ransom is paid. Finally, traffic-related cyberattacks can overload communication channels within the grid. Techniques such as eavesdropping on IoT communications may allow attackers to intercept sensitive information, increasing the risk of intrusion and control over critical operations. Securing the smart grid requires robust cybersecurity strategies to defend against these evolving threats [14].

1.4.5 Infrastructure Modernization

The current electrical grid infrastructure was designed with traditional, centralized energy sources in mind, such as fossil fuels like coal. These sources provide a constant and controllable flow of energy, making the grid relatively easy to manage. However, as energy systems evolve to accommodate renewable sources like wind and solar power, the grid faces significant challenges. These renewable sources are inherently intermittent and unpredictable, posing difficulties for a system originally designed for a steady, uninterrupted energy flow [14].

- **State of Current Infrastructure:** Traditional grids are optimized to handle energy generated from fossil fuels, which can be adjusted and dispatched on demand to meet consumption needs. In contrast, renewable energy generation is highly variable, often influenced by weather patterns, time of day, and seasonal changes. For example, solar energy generation stops during nighttime, and wind energy fluctuates based on wind speed. This makes it difficult for the current grid systems to balance energy supply with demand effectively, leading to potential instability or inefficiencies. During periods of high renewable energy generation, such as during sunny or windy days, the grid may face an excess of power that it cannot efficiently distribute. On the other hand, during low generation periods, the grid may experience energy shortages, requiring backup from traditional sources or external energy imports. This variability creates additional complexity in managing the grid, especially as the share of renewable energy increases [14].

- **Grid Modernization with Smart Grid:** To overcome these challenges, the concept of smart grids has been introduced. Smart grids are designed to integrate a diverse range of energy sources, including both traditional fossil fuels and renewable energy, into a unified and flexible energy network. Unlike traditional grids, smart grids leverage advanced technologies such as energy storage systems (e.g., batteries) and real-time monitoring to better handle fluctuations in energy production.

  By using energy storage, smart grids can capture surplus energy during periods of high renewable production and store it for later use when generation drops.
  This capability allows smart grids to release stored energy during times of high demand or when renewable generation is low, ensuring a steady supply of power. Additionally, smart grids use sophisticated sensors and communication technologies to monitor and manage the flow of electricity across the grid more efficiently, responding dynamically to changes in supply and demand.

  As a result, smart grids offer a more adaptable and resilient energy system compared to traditional grids. They can effectively balance the varying outputs from renewable sources, reduce the reliance on backup generation, and minimize energy waste, all while enhancing the overall reliability and efficiency of the grid [14].

1.5 Problem Statement and Objectives of the Project

- **Problem Statement:**
  In the context of the Smart Grid, the optimal management of energy demand heavily relies on the accuracy of load forecasting. Inaccurate forecasts can either overload the grid or lead to underutilization of resources, affecting both the stability of the system and its operational costs. Another major challenge in many networks is energy theft, which distorts consumption data, complicates forecasting, and results in significant economic losses for electricity providers.

  These two issues, forecasting inaccuracies and energy theft, are closely related: energy theft degrades the quality of historical data used to train forecasting models, while accurate forecasting can help detect abnormal consumption patterns potentially linked to fraud.

- **Project Objectives:**
  - Design and implement an electric load forecasting model using Federated Learning, enabling decentralized training across multiple data sources while preserving data privacy.
  - Ensure privacy-preserving training by integrating secure aggregation mechanisms and preventing raw data exposure.

- Develop a federated learning-based strategy for detecting electricity theft by identifying suspicious consumption behaviors across distributed clients without compromising confidentiality.

1.6 Conclusion

This first chapter introduced the fundamentals of Smart Grids and highlighted the major transformations they bring to traditional electrical systems. By integrating advanced digital technologies, Smart Grids enable more efficient, flexible, and sustainable management of electricity generation, transmission, and consumption. The functional architecture of a Smart Grid relies on seamless interaction between the generation, distribution, and consumption levels, along with communication and control systems. In this context, renewable energies play a central role, while also presenting technical challenges related to their integration into the grid. A key element for the proper functioning of these networks is electric load forecasting. It allows for anticipating demand, ensuring the balance between generation and consumption, and optimizing available resources. However, this task becomes more complex in the presence of anomalies such as energy theft, which degrades data quality and skews forecasting accuracy.

Chapter 2

State of Art

2.1 Introduction

This chapter presents an overview of existing work on load forecasting, electricity theft detection, renewable energy integration, and the use of Federated Learning for addressing these challenges in smart grids. While deep learning has improved forecasting accuracy, challenges remain regarding adaptability and generalization. Theft detection methods show potential but often lack scalability and privacy protection. Renewable energy integration continues to face issues due to source intermittency. Federated Learning emerges as a promising solution, enabling decentralized, privacy-preserving, and robust modeling across these domains.

2.2 Related Work on Load Forecasting

Several research efforts have been proposed in the field of load forecasting within smart grid environments, focusing on the development of accurate, scalable, and adaptive models to address the growing complexity of modern energy systems.

The work in [15], focuses on improving short-term load forecasting during the summer by proposing a model that incorporates cooling load factors. Using data from Jinan in 2016, the authors analyze daily load patterns influenced by meteorological conditions and develop a Least Squares Support Vector Machine (LS-SVM) model that accounts for the accumulated temperature effect to enhance prediction accuracy. Although the approach provides a promising solution to improving load prediction, the article assumes that daily load curves remain similar despite variations in weather patterns. The model calculates point loads based on the similarity between daily load curves and daily maximum and minimum loads. However, this method may oversimplify the complexity of real-world variations in power consumption, especially given the dynamic nature of temperature and cooling demands. The experimental results reported in the article demonstrate the effectiveness of the developed LS-SVM model, showing that it can generate relatively accurate predictions of cooling load. However, while the model succeeds in replicating daily load patterns, it could benefit from further refinement to account for a wider range of weather conditions and sudden temperature fluctuations. The study primarily focuses on the summer season in Jinan, which may limit the model's generalizability to other geographical regions with different climates or seasonal patterns. Additionally, the reliance on historical weather data and load curves raises concerns about the model's ability to adapt to unexpected events, such as extreme weather conditions or sudden changes in energy consumption due to socioeconomic factors.

In conclusion, while the LS-SVM model presented in this study makes a valuable contribution to short-term load forecasting for the summer period, there is potential for improvement. Future research could explore integrating more dynamic variables, such as real-time weather data, consumer behavior patterns, and regional and seasonal variability, to further enhance the model's robustness and applicability in broader contexts.

This paper [16], addresses the crucial role of load forecasting in energy management systems, particularly with the integration of advanced machine learning techniques. In this study, a Temporal Convolutional Network (TCN) model is proposed for short-term load forecasting, aiming to capture the temporal patterns of historical load data efficiently. Using real-world data from a region in Shanghai, the authors compare the TCN model with three traditional models: ARIMA, Artificial Neural Networks (ANN), and Long Short Term Memory (LSTM) networks. The results show that the TCN model achieves better forecasting accuracy than these baseline models. While the findings highlight the strong performance of the TCN approach, the study's comparative scope remains limited. It only includes a narrow set of baseline models and omits several other recent and competitive forecasting methods, such as Transformer-based architectures or hybrid models, which have shown promising results in time series forecasting tasks. Including a broader set of benchmarks could have provided a more comprehensive understanding of the TCNs relative performance.

Moreover, the studys reliance on a single regional dataset (Shanghai) limits the generalizability of its conclusions. Load forecasting models can be highly sensitive to regional variations in consumption behavior, climate, and socioeconomic factors. Evaluating the TCN model across multiple datasets representing diverse geographic and operational contexts would help validate its robustness and wider applicability. Integrating real-time and dynamic scenarios could also enhance the practical relevance of the results in real-world energy systems.

The study in [17], proposes a hybrid model named EEMD-LSTM for short-term electricity load forecasting. It aims to improve prediction accuracy by addressing the nonlinear and non-stationary nature of load data. Ensemble Empirical Mode Decomposition (EEMD) is used to decompose complex signals, and Long Short-Term Memory (LSTM) networks are applied for prediction. This combination enhances stability and captures temporal dependencies. The model was tested on real-world load data, showing improved forecasting performance. The study highlights the potential of hybrid techniques in power systems. However, certain methodological limitations remain. While the EEMD-LSTM model demonstrates improved forecasting accuracy, it presents several drawbacks. The approach relies heavily on the quality of EEMD decomposition, which may decline if components are highly correlated or not well separated. EEMD also increases computational costs due to repeated processing steps, and the added noise may leave residual disturbances in the resulting intrinsic mode functions (IMFs). Moreover, the model's reliability is sensitive to parameter selection, such as noise amplitude and ensemble size factors the paper does not deeply explore. Lastly, the study does not assess the computational complexity of the full pipeline, raising concerns about its feasibility for real-time applications in smart grid environments where efficiency is crucial.

The authors in [18], address the challenge of short-term electricity load forecasting in the context of rising power demand. They compare two deep learning models, Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) using historical consumption data. The models are trained and tested on the London Smart Energy Meter dataset. Forecast accuracy is evaluated using the Root Mean Square Error (RMSE) metric. The study aims to identify the most effective model for predicting electricity demand. Results highlight the predictive power of recurrent neural networks. The work contributes to improved energy management strategies. While the study provides valuable insights into the application of recurrent neural networks for load forecasting, several limitations can be identified. First, the evaluation relies solely on the RMSE metric, which, although informative, does not provide a complete picture of model performance, especially in the presence of extreme values or underprediction biases. Including complementary metrics such as Mean Absolute Percentage Error (MAPE) or R-squared would have strengthened the analysis.

Furthermore, the study limits its investigation to two models. Exploring additional architectures, such as hybrid models combining Convolutional Neural Networks (CNNs) with recurrent layers, could have offered further insights into capturing complex temporal and spatial dependencies in electricity consumption patterns.

This work [19], presents an enhanced short-term load forecasting method by addressing errors in weather forecasts and the cumulative effect of high temperatures. Traditional models often overlook these influences, reducing accuracy during extreme conditions. The authors propose a two-step correction process: adjusting weather forecasts using historical data and modeling cumulative temperature effects. These refined inputs are used to train a Backpropagation (BP) neural network. Experiments show improved performance, with an error rate of 1.61%. The approach offers a more accurate and robust solution. However, some challenges remain regarding generalization and input complexity. Despite these promising results, several limitations can be highlighted. While the weather correction process improves input quality, it introduces an additional layer of complexity and dependency on accurate historical weather datasets. In real-world scenarios, timely access to precise historical weather corrections may not always be feasible, potentially limiting the practical deployment of this approach.

Additionally, the study focuses primarily on the impact of temperature and does not account for other influential meteorological factors such as humidity, wind speed, or solar radiation, which could also affect load variations, especially in different seasons or regions.

Finally, the validation is limited to specific test cases without extensive cross-validation across diverse geographical areas or different types of load profiles. Broader testing would be essential to fully assess the generalizability of the proposed model under varied climatic and operational conditions.

2.3 Related Work on Electricity Theft

Electricity theft and other non-technical losses pose a significant global challenge, with estimated revenue losses reaching $101.2 billion annually across 138 countries . This alarming figure highlights the urgent need for effective and reliable methods to detect and prevent electricity theft, making it a critical area of research in the context of smart grid systems[20]. Several research efforts have been proposed to address electricity theft in smart grid systems, aiming to detect fraudulent behaviors and improve the reliability and security of energy distribution.

This study [21], proposes a hybrid deep learning model for electricity theft detection by combining Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU), optimized with the Manta Ray Foraging Optimization (MRFO) algorithm. CNN captures local patterns, while GRU models long-term dependencies. The MRFO is applied for hyperparameter tuning. Evaluated on a benchmark dataset, the model achieves 91.1% accuracy, approximately 6% higher than the baseline. This demonstrates the model's effectiveness in detecting anomalies in electricity consumption. However, deployment considerations remain, especially in real-world environments. Despite its strong performance, the models complexity due to the integration of CNN, GRU, and MRFO poses deployment challenges, especially in real-time or resource-limited environments like smart meters. The computational cost of metaheuristic optimization can also hinder scalability. Furthermore, the datasets limited geographic scope raises concerns about generalizability, given regional differences in consumption behavior. A more diverse dataset and broader testing across varied conditions would be crucial to evaluate how well the model adapts to different regional contexts. The models interpretability is another concern. Its black box nature may limit trust and regulatory acceptance, especially in high-stakes contexts.

Additionally, the model lacks an evaluation against simpler, more interpretable models, making it difficult to assess whether the accuracy gains justify the increased complexity. Given that transparency and trust are critical for adoption by utility companies, addressing this issue would improve the models practicality. A potential area for improvement is the real-time applicability. Although the study shows promising accuracy in controlled conditions, practical deployment in real-world smart grid environments would require optimizing the model for low latency, high performance requirements. As electricity theft detection must be timely and adaptive, reducing the model's computational load could make it more feasible for integration into operational systems. A focus on model efficiency and deployment readiness would enhance the models practical value for utility providers.

This project [22], proposes a cost-effective electricity theft detection system using Empirical Mode Decomposition (EMD) for signal processing and K-Nearest Neighbor (KNN) for classification. It utilizes real electricity consumption data from the State Grid Corporation of China (SGCC), covering 1,035 days and two primary classes normal usage and theft. Missing values are handled through interpolation. EMD decomposes the raw signals into Intrinsic Mode Functions (IMFs), from which statistically significant features are extracted for input into the KNN classifier. The system achieves 91.0% detection accuracy and is particularly suited for environments with limited computational resources due to its simplicity and efficiency. Despite its practical appeal, several limitations affect the systems scalability and robustness. EMD, although effective for nonlinear signals, is sensitive to noise, which may introduce spurious components and compromise feature reliability. KNN, while intuitive, struggles in high-dimensional spaces and with large datasets, as its performance depends heavily on feature relevance and the chosen parameters. The study also lacks cross-validation and multi-regional testing, which raises concerns about the models generalizability across different grid environments and customer profiles.

Another limitation is the manual feature selection from EMD outputs, which introduces subjectivity and limits scalability. The absence of automated feature engineering or adaptability reduces its effectiveness for dynamic smart grids. Additionally, the study does not benchmark the approach against more recent or interpretable methods, leaving unclear how it compares in terms of performance versus complexity. While effective in its specific context, broader validation and methodological refinement are necessary for wider application.

This work [**23**], proposes a novel method to detect intermittent electricity theftan irregular, hard-to-detect form of non-technical loss. The approach combines multi-scale CNNs to identify suspicious intervals and Pearson correlation to associate these with abnormal line loss. DBSCAN clustering is then used to isolate outliers without needing predefined cluster counts. Evaluated on synthetic datasets mimicking real consumption patterns, the model outperforms conventional techniques. Its ability to detect subtle manipulation patterns makes it a promising solution. However, deployment challenges and generalizability issues remain. While the results are promising, the approach has limitations. Its reliance on deep learning and unsupervised clustering makes it computationally intensive, which could hinder real-time or large-scale deployment, especially in resource-constrained settings. Real-world implementation would likely require optimized or hardware-accelerated versions. The use of synthetic data, though based on real patterns, raises concerns about generalizability, as it may not fully capture the diversity of theft behaviors across users or regions. Extensive validation with authentic, labeled datasets is essential to confirm robustness and practical viability.

Additionally, the method's black box nature limits interpretabilityan important factor for utility providers and regulators who need transparency in decision making, particularly when legal or financial consequences are involved. Finally, while effective for intermittent theft, the model does not address the wider spectrum of non-technical loss behaviors. Integrating it into a broader, adaptive detection system would enhance its practical relevance.

The authors in [**24**], address electricity theft in Bangladesh using a supervised machine learning framework trained on a real-world dataset of over 56,000 records from 2017 to 2020. To handle class imbalance, the study employs various resampling techniques including SMOTE and ENN. Classical classifiers such as KNN, SVM, and Decision Trees are tested under different sampling conditions. The study presents a detailed comparison of model performance. This work emphasizes methodological rigor in preprocessing and evaluation. Results show that proper sampling boosts detection accuracy. Still, challenges in scalability and adaptability remain. Despite the studys methodological rigor, several limitations reduce its practical applicability. First, the use of a centralized data environment does not reflect the decentralized nature of modern power grids. Centralized data handling raises privacy and security concerns; a federated learning approach would better support privacy and scalability by enabling distributed model training without raw

data transfer.
Additionally, the models are trained on known theft patterns, which limits their ability to detect emerging or sophisticated strategies. Theft methods evolve, and systems must generalize beyond historical anomalies for long-term effectiveness.

The exclusive use of classical ML models also restricts the capture of temporal patterns in consumption data, which are often sequential and seasonally influenced. Incorporating temporal modeling techniques could improve detection accuracy by addressing these dynamics.

Lastly, the geographic specificity of the dataset limited to Bangladesh raises concerns about generalizability. Electricity theft behavior varies across regions, requiring cross-domain validation for broader deployment. The study also offers limited insights into model interpretability, which is crucial for adoption by utilities and regulators who must justify flagged cases in legal and operational contexts.

In conclusion, while the framework provides valuable insights into theft detection using classical ML and effective resampling, its real-world scalability, adaptability, and explainability remain constrained, highlighting the need for more federated, temporal, and interpretable models in future work.

This study [25] ,introduces TLGRU, a hybrid electricity theft detection model combining LSTM and GRU to address limitations of traditional systems. It incorporates synthetic attacks to handle class imbalance, LSTM for extracting temporal features, and GRU for adaptive classification. Dropout and the Adam optimizer are used to enhance performance. Tested on real consumption data from China, the model achieves over 91% accuracy with a low 1% false positive rate. TLGRU shows strong predictive capacity in identifying anomalous behavior. These results highlight its robustness under controlled conditions. Yet, challenges remain for real-world application.Despite these promising outcomes, the model is developed and tested in a centralized environment, which may not reflect the data privacy and distribution realities of modern smart grids. As smart meters become more prevalent, the ability to train models across decentralized data sources without compromising user privacy becomes crucial. A federated learning approach would be better suited for such environments, enabling local training at the edge while aggregating model updates centrally, thereby enhancing both privacy and scalability.

Additionally, the reliance on synthetic attack scenarios for data balancing may not fully capture the evolving and context-specific nature of real-world theft patterns. Furthermore, the study lacks comparative analysis against simpler or more interpretable models, which limits the assessment of its performance relative to model complexity. The manual nature of data preprocessing and the absence of explainability mechanisms also pose challenges for regulatory acceptance and operational deployment. In conclusion, while TLGRU demonstrates technical innovation and high accuracy in a controlled setting, its practical adoption would benefit from decentralized training, real-world validation, and improved interpretability.

2.4 Related Work on Renewable Energy

Related work on renewable energy has focused extensively on the integration, optimization, and management of sustainable energy sources within modern power systems. Researchers have investigated various challenges such as variability, storage, grid stability, and demand forecasting, to enable efficient and reliable use of renewables in smart grid infrastructures.

This study [26] , introduces ESNCNN, a hybrid deep learning model combining Echo State Networks (ESNs) and Convolutional Neural Networks (CNNs) for forecasting renewable energy production and electricity consumption. ESNs capture temporal dependencies, while CNNs extract spatial features. Residual connections address vanishing gradients, and fully connected layers enhance predictions. Evaluated on benchmark datasets, ESNCNN outperforms existing methods across multiple metrics (MSE, RMSE, MAE, MBE). The model shows strong generalizability and applies to both RE and load forecasting tasks. Despite strong performance, the study does not provide evidence for the models claimed efficiency, omitting runtime or resource benchmarks. Its centralized training setup raises privacy and scalability concerns, which could be mitigated through federated learning. Limited interpretability and a lack of robustness testing under anomalous or extreme conditions further constrain its deployment in real-world smart grid environments.

This work [27], presents a modified stacked GRU-RNN architecture for forecasting renewable energy generation and electricity load. The method includes correlation-based feature selection, an AdaGrad optimizer with momentum, and application to both univariate and multivariate scenarios. Evaluated on wind and load datasets, the model demonstrates improved forecasting accuracy over baseline approaches, supporting its relevance for smart grid operations.

While the results are promising, the study lacks ablation analysis, leaving it unclear which components drive performance gains. Its generalizability across regions or energy systems is not assessed, limiting broader applicability. The absence of interpretability tools and the reliance on linear correlation for feature selection may hinder transparency and overlook nonlinear relationships, particularly as smart grid data grows in complexity.

This study [28] , employs a Multi Layer Perceptron (MLP) model to forecast renewable energy production in Poland, focusing on wind, solar, biogas, and biomass. Using historical data from 1990 to 2018, the model predicts substantial growth in most renewables, aligned with EU climate goals, while noting stagnation in hydro due to resource limitations.

Although MLP captures nonlinear relationships, it lacks mechanisms to model temporal dependencies critical in volatile and seasonal energy data. The absence of comparative benchmarking with time series-specific models and the lack of sensitivity analysis limit confidence in the model's robustness and adaptability to real-world dynamics. Incorporating time-aware techniques would enhance predictive reliability.

2.5 Federated Learning in Theft Detection and Load Forecasting

Federated Learning (FL) has emerged as a promising paradigm in the context of smart grids, enabling collaborative model training across distributed energy data sources while preserving data privacy. Recent studies have explored the application of FL in addressing key challenges such as electricity theft detection and load forecasting, aiming to enhance grid intelligence without compromising sensitive user data. This section reviews the most relevant contributions in this area.

This study [29] , presents a robust solution for detecting cyber and electricity theft attacks in smart grids by integrating a multi-step LSTM imputation model with a Federated Learning based Stacking Ensemble GRU (FL-SE-GRU). The approach addresses challenges like missing data, privacy concerns, and coordinated attacks. Tested on smart grid datasets, the model achieves over 95% accuracy in detection and classification. Its privacy-preserving design is well aligned with modern grid environments. Multiple evaluation metrics and feature significance analysis support its validity. This combination ensures robust preprocessing and decentralized model training. However, real-world applicability remains to be proven.

However, the study is not without limitations. While the authors claim the model is efficient, the stacking ensemble within a federated learning setup likely introduces non-trivial communication and computational overhead, which is not quantified. Additionally, the framework has not been tested in a real-time environment, raising questions about latency and scalability. Moreover, potential vulnerabilities of federated learning such as poisoning or model inversion attacks are not addressed, despite the papers focus on security.

This work [30], proposes a privacy-preserving framework for forecasting residential electricity consumption using Federated Learning (FL) combined with Long Short Term Memory (LSTM) models. It enables local training on smart meters and aggregates models globally without sharing raw data, ensuring both privacy and scalability. Temporal and weather-related features are integrated to enhance accuracy, while K-means clustering groups users by behavior. The model achieves performance comparable to centralized methods (RMSE between 0.12 and 0.14 kWh). Evaluation on the UK and Italian datasets shows strong results. Its architecture aligns with GDPR and reduces data transmission needs. This makes it a promising approach for decentralized smart grids.

While the FL model significantly reduces communication overhead, the need for periodic model updates could place stress on the communication infrastructure, particularly in large-scale networks with millions of devices. Moreover, the approach assumes that all edge devices possess sufficient computational capabilities, which might not be realistic in legacy systems. Additionally, performance drops were observed in regions with irregular consumption behavior, indicating the potential need for dynamic clustering or integrated anomaly detection mechanisms.

This project [31], introduces a novel approach that combines federated learning with client-level personalization to improve short-term electricity load forecasting while preserving data privacy. In the proposed method, a global forecasting model is trained collaboratively across multiple consumers without sharing raw data, and then personalized locally using a proximal regularization term. The approach addresses the limitations of both purely local models (prone to overfitting) and global federated models (unable to capture user heterogeneity). Experiments on real smart meter data from 100 households demonstrate improved accuracy compared to traditional methods, especially when local data is scarce.

This study presents an effective solution for load forecasting by introducing a personalized federated learning framework. However, the use of a fully connected artificial neural network (ANN) as the base model may limit the methods capacity to capture temporal dependencies inherent in electricity consumption data. Models such as Long Short-Term Memory (LSTM) networks, which are specifically designed for time series analysis, could offer significant improvements in capturing sequential patterns. Furthermore, while the personalization technique via proximal regularization is innovative, the computational cost and complexity of tuning parameters like data discrepancy and regularization weight are not deeply explored. Overall, the paper makes a strong contribution to privacy-preserving forecasting, though it leaves room for enhancement through more temporally aware models.

2.6 Conclusion

In summary, the current literature highlights substantial progress in load forecasting, electricity theft detection, and renewable energy management. However, several limitations persist, including scalability, data privacy, and adaptability to real-world constraints. These challenges underline the necessity of exploring novel paradigms such as federated learning, which offers a promising path forward by enabling collaborative, privacy-preserving, and distributed intelligence in smart grid environments.

Chapter 3

Methodology and Federated Learning Approach

3.1 Introduction

This chapter outlines the methodology adopted to develop predictive models for electric load forecasting and electricity theft detection in smart grids. It emphasizes the use of Federated Learning (FL) as a decentralized and privacy-preserving solution. The process begins with a presentation of the real-world dataset, followed by data preprocessing steps such as cleaning, normalization, and time series transformation. The architecture of the proposed FL system is then introduced, including client-server roles and model aggregation strategies. Next, we describe the modeling approaches for both forecasting and theft detection. The chapter concludes with a comparative analysis of centralized vs. federated learning approaches, highlighting the advantages of FL in smart grid applications.

3.2 Data Presentation

3.2.1 Data Origin

Two distinct datasets are utilized in this work, each serving a different objective within the project.

- **The first dataset** [32], is designed for load forecasting and originates from a real-world electrical distribution system. The data was collected throughout the year **2017** using the Supervisory Control and Data Acquisition (SCADA) system operated by Amendis, a utility company in charge of electricity distribution. It consists of 52,416 entries recorded at 10-minute intervals, offering a detailed and continuous stream of energy usage and environmental information.

  This dataset covers three distinct electricity distribution zones, namely Zone 1 (Quads), Zone 2 (Smir), and Zone 3 (Boussafou). Each observation includes nine features, summarized below:

  - **Date Time**: Timestamp of the 10 minute interval.
  - **Temperature**: Ambient temperature.
  - **Humidity**: Atmospheric humidity.
  - **Wind Speed**: Wind velocity.
  - **General Diffuse Flows**: The global diffuse solar radiation (in W/mš) received on a horizontal surface, useful for analyzing the influence of sunlight on energy usage.

- **Diffuse Flows**:The portion of solar radiation that is scattered in the atmosphere, without direct sunlight. Also relevant for understanding weather-related consumption changes.
- **Zone 1 Power Consumption**:Electrical power consumption in Zone 1 (Quads), measured in kilowatts (kW).
- **Zone 2 Power Consumption**:Electrical power consumption in Zone 2 (Smir), measured in kilowatts (kW).
- **Zone 3 Power Consumption**:Electrical power consumption in Zone 3 (Boussafou), measured in kilowatts (kW).

- **The second dataset [33]**, is designed for electricity theft detection and classification. It comprises hourly electricity consumption records collected over one year for 16 different consumer types. The dataset originates from the Open Energy Data Initiative (OEDI), a centralized platform by the U.S. Department of Energy that aggregates high-value energy research datasets.

  To simulate fraudulent behavior, six distinct types of electricity theft were synthetically injected into the original data. These types simulate common manipulation patterns encountered in smart grids, namely:

  1. **Partial consumption reduction during the day:** Consumption values are multiplied by a random factor between 0.1 and 0.8.
  2. **Random zero consumption:** Consumption drops to zero at randomly selected periods.
  3. **Random hourly scaling:** Each hourly consumption value is multiplied by a different random factor.
  4. **Random fraction of mean:** Values are replaced with a random fraction of the average consumption.
  5. **Constant mean reporting:** All consumption values are replaced with the mean consumption.
  6. **Reversed readings:** The time series order of consumption values is reversed.

  A dedicated theft generator was developed to randomly apply these six theft types to various consumer profiles, enabling robust training and benchmarking of anomaly detection and classification models.

  The dataset contains the following key features:

  - **ID:** Unique identifier for each observation.
  - **Electricity:Facility:** Total electricity consumed in the facility.
  - **Fans:Electric:** Energy consumed by ventilation fans.
  - **Cooling:Electric:** Energy used for air conditioning.
  - **Heating:Electric:** Electricity used for space heating.
  - **InteriorLights:** Electricity consumed by indoor lighting systems.
  - **InteriorEquipment:** Usage of interior electrical equipment.
  - **Gas:Facility:** Gas consumption at the facility level.
  - **Heating:Gas:** Gas used specifically for heating purposes.

- **InteriorEquipment (Gas):** Equipment that consumes gas inside the building.
- **WaterHeater:** Energy used by water heating systems.
- **Class:** Type or category of the consumer (e.g., FullServiceRestaurant, SmallOffice).
- **Theft:** Label indicating whether the record is *Normal* or represents one of the six *Theft* types.

3.2.2 Challenges Encountered and Constraints Related to Real Data

Working with real-world datasets, especially in the context of electricity consumption and theft detection, presents several challenges and constraints. One of the key issues is the **quality and completeness of the data**. Missing, inconsistent, or erroneous values are common in real-world datasets. For example, some records may have incomplete time stamps, or power consumption data may be missing for certain periods due to sensor malfunctions or communication issues. These gaps can significantly impact the accuracy of machine learning models if not properly handled during preprocessing.

Another significant challenge is the **imbalance between normal and theft-related data**. In most real-world scenarios, theft events are far less frequent than normal consumption patterns, leading to a highly imbalanced dataset. This imbalance can make it difficult for machine learning models to learn meaningful patterns for fraud detection. Techniques like oversampling (e.g., SMOTE) and cost-sensitive learning are often used to address this issue, but they come with their challenges, such as the risk of overfitting or generating unrealistic synthetic samples.

Furthermore, **temporal dynamics** pose a challenge when dealing with time series data. Electricity consumption patterns can vary seasonally, daily, and even hourly, depending on various factors such as weather, holidays, and socioeconomic conditions. This introduces significant **temporal dependencies** that need to be captured for accurate forecasting and anomaly detection. Data preprocessing, feature engineering, and model selection must account for these dependencies to ensure reliable predictions.

Finally, **privacy and data security concerns** are also important. In real-world scenarios, personal consumption data can be sensitive, and its misuse or unauthorized access could lead to privacy violations. Ensuring the protection of such data, especially when sharing across multiple organizations or using federated learning approaches, requires careful attention to data encryption, access control, and anonymization techniques.

3.3 Data Preprocessing

3.3.1 Data Cleaning

Data cleaning is an essential preprocessing step to ensure the quality and integrity of the data before feeding it into machine learning models. In this step, we focus on handling missing, inconsistent, or erroneous data entries.

- **Handling Missing Values:** All rows containing `NaN` values were removed to maintain dataset integrity and avoid biases during model training.
- **One-Hot Encoding:** To prepare the data for Federated Learning, categorical variables were encoded. The Class variable was transformed using One Hot Encoding, representing each building type as a binary vector to ensure compatibility with federated aggregation mechanisms. The target variable theft was also encoded into a binary format (0 for "Normal", 1 for "Theft") to enable classification.

3.3.2 Normalization of Values

Normalization scales numerical data to a common range, ensuring that all features contribute equally during model training. This process improves learning stability and overall model performance. It is particularly crucial in federated learning, where models are trained on heterogeneous data distributed across multiple clients, each potentially exhibiting different data distributions.

3.3.3 Transformation into Time Sequences

Sequence creation is an essential preprocessing step for time series data, particularly when using machine learning models such as recurrent neural networks (RNNs). It structures the data to enable the model to capture temporal dependencies and make accurate predictions.

- **Sequence creation:** The time series is divided into smaller subsets called sequences. Each sequence consists of a fixed number of consecutive time steps used to predict the next value in the series.
- **Time window size (Sliding Window):** A fixed-size window slides across the time series to extract overlapping sequences of successive observations.
- **Multivariate case:** When the dataset includes multiple variables (e.g., energy consumption, temperature, humidity), each input at a time step is a vector containing the values of all variables. The resulting sequences are thus multivariate, capturing correlations across features over time.

3.3.4 Separation of Training and Testing Datasets

In this work, the dataset was divided into two parts: 80% was used for training the models, and 20% was reserved for testing. This standard split allows the model to learn from the majority of the data while being evaluated on a separate, unseen portion to assess its generalization performance.

3.4 Federated Learning Approach

Federated learning is a decentralized machine learning approach that enables model training across multiple devices or data sources without exchanging raw data. This technique enhances **data privacy and security**, ensures regulatory compliance, and fosters model robustness by leveraging **diverse local datasets**. Instead of centralizing data, each node trains a model locally, and only model updates are shared with a central server, which aggregates them to improve the global model [34],[35].

3.4.1 Architecture of our Federated Learning System

A standard federated learning system comprises several distinct components, each with specific roles and responsibilities:

- **Clients (or Workers)**: These are the entities holding the local, private data used for training. Their primary responsibilities include :
    - Storing and managing their local dataset securely.
    - Receiving the current global model parameters and training instructions from the server.
    - Performing local model training on their data for one or more rounds.
    - Calculating model updates.
    - Sending the processed updates back to the server.
- **Server (or Coordinator/Aggregator)**: The server acts as the central coordinator of the federated learning process, without ever accessing the clients' raw data. Its responsibilities include:
    - Initializing the global model.
    - Selecting a subset of clients for each training round and sending them the current global model along with training configurations.
    - Collecting local model updates and evaluation metrics from the selected clients.
    - Aggregating the received model updates using strategies such as FedAvg, FedProx, or other algorithms.
    - Aggregating evaluation metrics to monitor the overall performance of the global model.
    - Updating the global model and coordinating the next round.
    - Repeating the cycle until convergence.
- **Model**: The model refers to the machine learning algorithm collaboratively trained by all participants. It can take various forms depending on the task, such as a linear regression model, a support vector machine, or more commonly, a deep neural network. The model's architecture is usually predefined and distributed uniformly to all clients.
- **Communication Protocol:** The communication protocol defines the interaction framework between clients and the central server, specifying how information is securely and efficiently exchanged during the federated training process. It includes:
    - **Transmission protocols:** Use of network protocols such as gRPC or REST APIs over HTTPS to enable reliable and scalable communication between clients and the server.
    - **Data serialization formats:** Model parameters and updates are serialized using efficient formats like Protocol Buffers (Protobuf) or FlatBuffers for compact and fast binary communication. In some cases, formats like JSON or HDF5 may be used for compatibility or readability.

- **Client management strategies:** Techniques to handle client availability, manage dropouts, and mitigate communication delays, such as asynchronous updates, client sampling, or retry mechanisms.
- **Security and privacy:** Implementation of secure communication channels using SSL/TLS encryption to ensure data confidentiality.

3.4.2 Operational Framework of Synchronous Federated Learning

In most common scenarios, particularly in cross-silo settings or simulations, a synchronous round-based interaction approach is followed. The typical flow is as follows:

1. **Initialization:** The server defines the initial global model w_0.
2. **Client Selection:** At round t, the server selects a subset of clients S_t.
3. **Broadcast:** The server sends the current global model w_t to all clients in S_t.
4. **Local Training:** Each selected client $k \in S_t$ trains the model w_t on its local dataset D_k for E epochs, resulting in a local model update Δw_k^{t+1}. This step often involves minimizing the local loss function $L_k(w)$.
5. **Update Transmission:** Each client k sends its computed update Δw_k^{t+1} back to the server.
6. **Aggregation:** The server collects updates from a sufficient number of clients and aggregates them using a selected algorithm. For example, using weighted averaging in FedAvg:
$$\Delta w^{t+1} = \sum_{k \in S_t'} \frac{n_k}{N_t} \Delta w_k^{t+1},$$
where S_t' is the set of clients that successfully returned updates, $n_k = |D_k|$, and $N_t = \sum_{k \in S_t'} n_k$.
7. **Global Model Update:** The server updates the global model as follows:
$$w^{t+1} = w_t + \eta \Delta w^{t+1},$$
where η is the server's learning rate 1.
8. **Iteration:** The process repeats from Step 2 for the next round $(t+1)$ until the termination criteria are met.

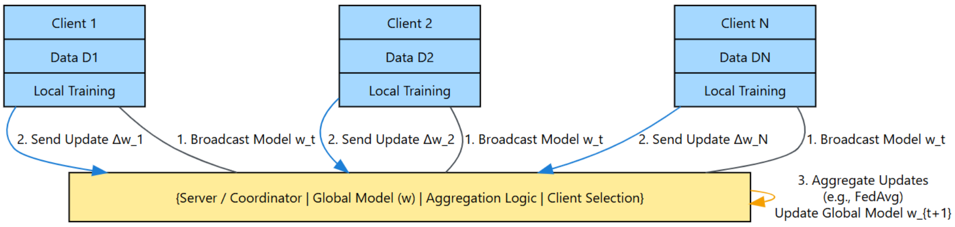


Figure 3.1: A typical client-server architecture for federated learning

3.5 Functional Architecture of the Proposed Approach

3.5.1 Presentation of the Processing Pipeline

The proposed solution is based on a federated architecture composed of multiple clients and a central server. The processing pipeline includes local data collection, normalization, creation of time sequences, local model training, and then sending the weight updates to the server. The server aggregates these updates, updates the global model, and redistributes it to the clients. This process is repeated until convergence.

3.5.2 Role of Clients: Local Processing

Each client performs the following tasks:

- **Local Data Processing:** Clients independently preprocess their data, including normalization and conversion into time sequences.
- **Model Training:** Clients train a local model using their private dataset for a specified number of local epochs.
- **Model Update Computation:** After training, clients compute model updates.
- **Communication:** Clients transmit only the model updates to the central server, ensuring that raw data remains private.

This approach maintains data privacy while enabling clients to contribute to the global learning process.

3.5.3 Role of the Server: Aggregation

The central server performs coordination and model integration:

- **Receiving updates:** The server gathers local model updates from the participating clients.
- **Aggregation:** Using algorithms such as `FedAvg, FedProx`, the server aggregates updates to create a unified global model.
- **Model distribution:** The aggregated **weights of the global model** are then shared back with the clients for the next training round.
- **Convergence monitoring:** The server evaluates the performance of the global model and controls the training process over communication rounds.

3.5.4 Model Aggregation Strategies

FedAvg

- **Definition:** Proposed by McMahan et al.(2017), **FedAvg** is an aggregation method based on the **weighted average** of local models. Each client performs local training, then the server aggregates the model weights according to the size of each clients dataset.

- **How it works:** At each federated round t, a subset of clients S_t is selected. Each client $k \in S_t$ trains its model locally on n_k samples and sends its weights w_k^t to the server. The server then computes the global weights as follows:

$$w_{global}^{t+1} = \sum_{k \in S_t} \frac{n_k}{\sum_{i \in S_t} n_i} w_k^t$$

  - w_{global}^{t+1}: global model weights at round $t+1$
  - S_t: subset of selected clients at round t
  - w_k^t: local model weights from client k
  - n_k: number of samples owned by client k
  - $\sum_{i \in S_t} n_i$: total number of samples used in aggregation at round t

- **Advantages:**
  - Reduces communication by relying on local training.
  - Simple and efficient in homogeneous environments.
  - Easy to implement.

- **Limitations:**
  - Sensitive to non-IID data distribution.
  - Vulnerable to system heterogeneity.
  - No regularization mechanism may lead to divergence among local models.

FedProx

- **Definition:** Introduced by Li et al.(2018), **FedProx** is an extension of FedAvg designed to stabilize training in non-IID settings by adding a **proximal term** that limits divergence between local and global models.

- **How it works:** Each client k minimizes the following local loss function:

$$L_k(w) = f_k(w) + \frac{\mu}{2} \|w - w_{global}^t\|^2$$

  - $L_k(w)$: regularized local loss of client k
  - $f_k(w)$: empirical loss over local data
  - w: local model weights to be optimized
  - w_{global}^t: global model weights sent by the server at round t
  - μ: regularization parameter controlling the strength of the proximal term

  The proximal term $\frac{\mu}{2}\|w - w_{global}^t\|^2$ penalizes large deviations from the global model.

- **Advantages:**
  - Improved stability on non-IID data.
  - Reduces variance among local models.
  - Supports heterogeneous environments.

- **Limitations:**
  - Requires careful tuning of μ.
  - May slow down convergence if μ is not well chosen.

FedAdam

- **Definition:** Proposed by Reddi et al. (2020), **FedAdam** is an aggregation method inspired by the Adam optimizer, applying **adaptive learning rates at the server** to improve convergence.
- **How it works:** The server first aggregates the local updates:

  $$\Delta_t = \sum_{k \in S_t} \frac{n_k}{\sum_{i \in S_t} n_i} (w_k^t - w_{global}^t)$$

  Then, it applies the Adam update rules:

  $$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\Delta_t$$

  $$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\Delta_t^2$$

  $$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

  $$w_{global}^{t+1} = w_{global}^t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

  - Δ_t: aggregated update at round t
  - m_t, v_t: first and second moment estimates
  - β_1, β_2: decay rates for moving averages
  - η: server learning rate
  - ϵ: numerical stability constant
  - $\hat{m}_t, \hat{v}_t$: bias-corrected moment estimates
- **Advantages:**
  - Faster convergence in non-IID settings.
  - Dynamically adjusts the learning rate.
  - Reduces oscillation in the global model.
- **Limitations:**
  - Increased computational complexity at the server.
  - Sensitive to hyperparameter tuning.

FedTrimmedAvg

- **Definition:** Proposed by Wang et al. (2021), **FedTrimmedAvg** is a robust aggregation method based on the trimmed mean technique. It improves the resilience of federated learning systems by discarding a certain percentage of the highest and lowest model updates, thereby mitigating the impact of outliers and potential adversarial contributions.

- **How it works:** For each model dimension j, the $\alpha\%$ highest and lowest values are discarded, and the aggregation is computed as:

$$w_{global}^{(j)} = \sum_{k \in S_\alpha^{(j)}} \frac{n_k}{\sum_{i \in S_\alpha^{(j)}} n_i} w_k^{(j)}$$

  - $w_k^{(j)}$: model weight of client k in dimension j
  - $S_\alpha^{(j)}$: set of clients after trimming $\alpha\%$ outliers in dimension j
  - n_k: number of samples held by client k

- **Advantages:**
  - Resilient to Byzantine faults and abnormal updates.
  - Improves robustness in unreliable environments.

- **Limitations:**
  - Risk of discarding useful information with excessive trimming.
  - Sensitive to the choice of α.
  - Computationally expensive due to per-dimension filtering.

3.6 Modeling of Load Forecasting

Load forecasting refers to the process of predicting future electricity demand based on historical consumption patterns and other influencing factors such as time of day, weather, and seasonality. Accurate load forecasting is critical for ensuring the stability and efficiency of smart grid operations, allowing for optimal energy generation, distribution, and consumption planning. In this work, we employ a federated learning approach to collaboratively train a deep learning model across multiple clients without sharing raw consumption data. This strategy preserves data privacy while leveraging the richness of distributed datasets.

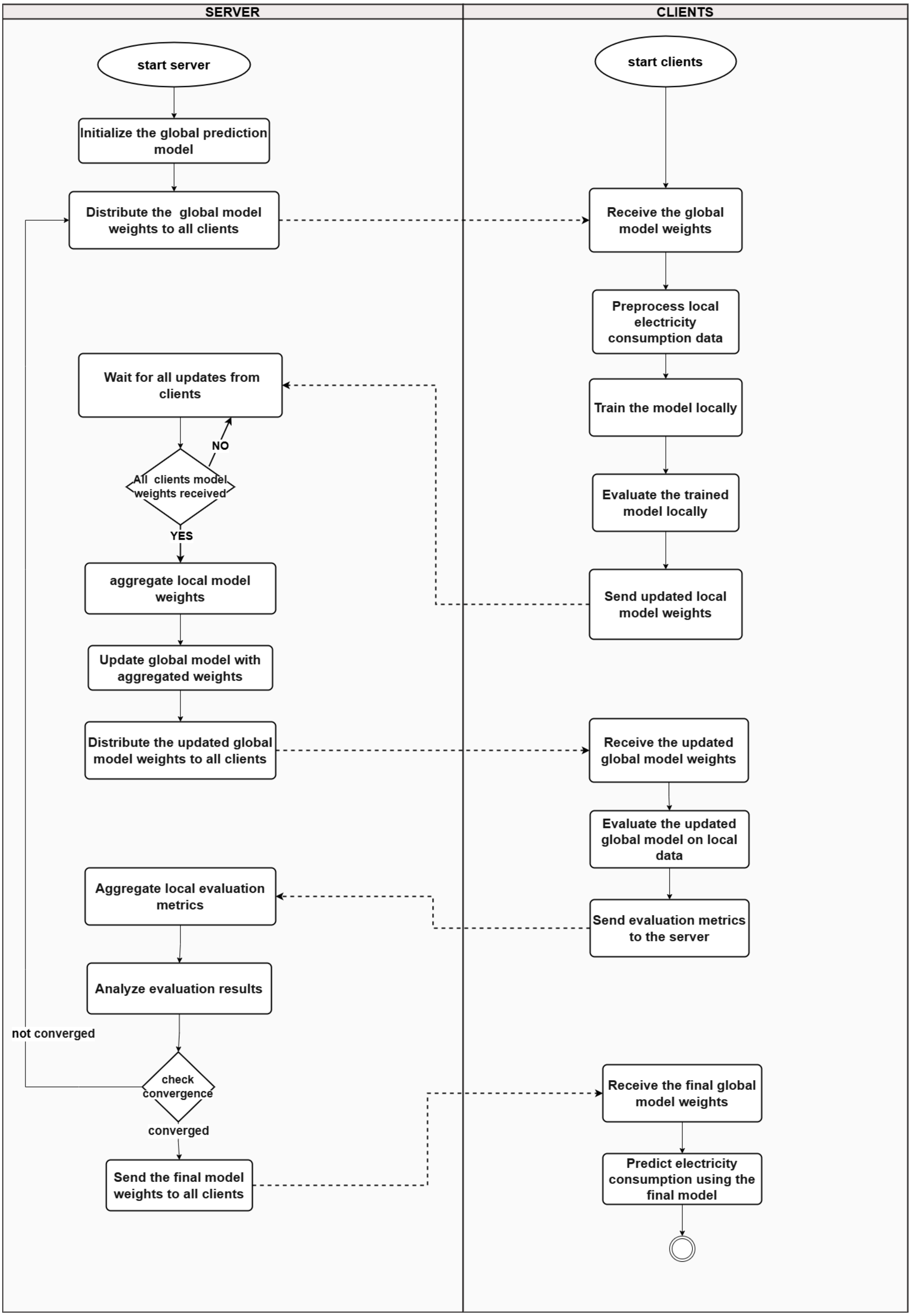

SERVER
CLIENTS
start server
Initialize the global prediction model
Distribute the global model weights to all clients
Wait for all updates from clients
NO
All clients model weights received
YES
aggregate local model weights
Update global model with aggregated weights
Distribute the updated global model weights to all clients
Aggregate local evaluation metrics
Analyze evaluation results
not converged
check convergence
converged
Send the final model weights to all clients
start clients
Receive the global model weights
Preprocess local electricity consumption data
Train the model locally
Evaluate the trained model locally
Send updated local model weights
Receive the updated global model weights
Evaluate the updated global model on local data
Send evaluation metrics to the server
Receive the final global model weights
Predict electricity consumption using the final model

Figure 3.2: BPMN 2.0 Workflow for Federated Learning Based Electricity Consumption Forecasting

3.6.1 Feature Selection using SelectKBest

To optimize the input data and improve the performance of the load forecasting model, a feature selection technique was applied using **SelectKBest** from the *scikit-learn* library. This method aims to reduce dimensionality and retain only the most informative features for the regression task.

- **Principle of Operation:** SelectKBest evaluates each feature individually based on a specified statistical test that measures its relevance to the target variable. The features are then ranked in descending order according to their scores, and the top k features are selected for model training, while the rest are discarded.
- **Scoring Function Used `f_regression`:** In this study, we used the `f_regression` scoring function, which is suitable for regression tasks. It computes the correlation between each feature and the continuous target variable, assigning an F-score that reflects the strength of this linear relationship. Features with higher F-scores are considered more relevant for predicting the target.

  By applying this method, we were able to:

  - Eliminate redundant or non-informative features,
  - Reduce the risk of overfitting,
  - Accelerate the training process,
  - Enhance the overall generalization of the model.

The value of k was chosen empirically based on model performance metrics during cross-validation.

3.6.2 Time Window Configuration

For the load forecasting task, a time window of 24 hours was selected. This means that the model uses the consumption data from the previous 24 hours to predict the next value. The 24-hour window was chosen to capture daily consumption cycles and recurring patterns that occur over a full day. This configuration enables the model to learn from broader temporal context, improving its ability to forecast demand trends while maintaining a reasonable computational cost and avoiding unnecessary noise from shorter-term fluctuations.

3.6.3 Exploration of Model Architectures

To design an effective load forecasting model, several deep learning architectures were explored and evaluated empirically. These architectures, primarily based on Recurrent Neural Networks (RNNs), are well-suited for time series forecasting due to their ability to model temporal dependencies. Each model was assessed based on prediction accuracy, training stability, and computational efficiency. The final selection was made by balancing performance with practical deployment considerations.

Unidirectional LSTM

The Long Short-Term Memory (LSTM) network was developed to address the vanishing gradient issue in standard RNNs. Its gated architecture (input, forget, and output gates) enables the modeling of long-term temporal dependencies. In load forecasting applications, the LSTM effectively modeled short-term consumption patterns and produced stable, accurate predictions. However, its relatively high computational complexity and extended training time presented limitations, especially in large-scale or resource-constrained environments.

GRU

The Gated Recurrent Unit (GRU) offers a simplified alternative to LSTM by combining the input and forget gates into a single update gate and removing the output gate. This streamlined design results in fewer trainable parameters and faster training. In our experiments, the GRU achieved competitive accuracy with reduced training time and memory usage. Its efficiency and generalization capability made it particularly well-suited for federated learning scenarios, where communication and computational resources are limited.

Hybrid LSTM-GRU

The hybrid model combines LSTM and GRU layers, either sequentially or in parallel, aiming to leverage the strengths of both architectures. Although promising in theory, the added complexity led to increased training instability and a higher risk of overfitting. In practice, it did not significantly outperform the simpler models, and its computational overhead outweighed its benefits.

Bidirectional LSTM (BiLSTM)

The BiLSTM processes sequences in both forward and backward directions, capturing richer temporal context. This made it particularly effective in short-term load prediction. However, due to its higher memory and computational demands, BiLSTM was less practical for real-time forecasting or deployment in distributed, resource-constrained environments.

Final Model Selection

Following an in-depth comparison, the GRU model was selected as the final architecture. It offered the best compromise between predictive accuracy, training speed, and resource efficiency. Its simplicity and rapid convergence made it highly compatible with federated learning frameworks, especially in settings where bandwidth and computational power are limited. Consequently, GRU was adopted for all subsequent modeling and experimentation phases.

-The following figure summarizes the comparative performance of all evaluated architectures, reinforcing the choice of the GRU model as the final selected approach.

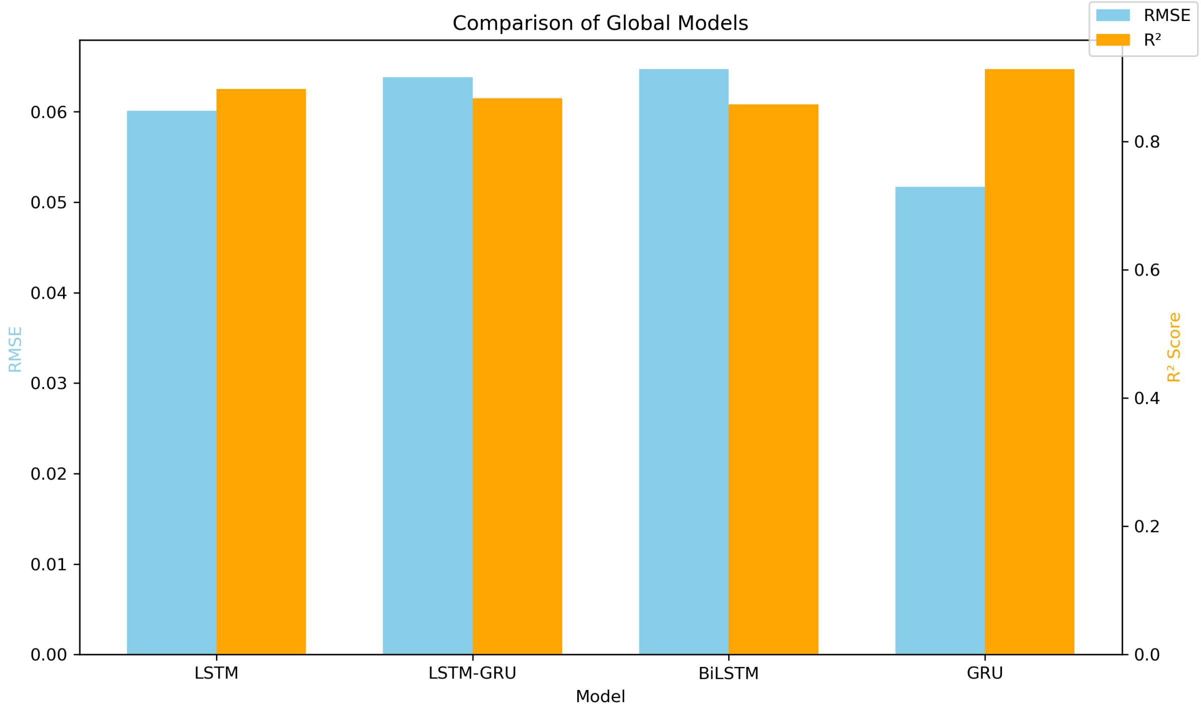


Figure 3.3: Comparative Performance of LSTM-Based Models in Federated Learning using RMSE and R^2 Metrics

3.7 Modeling for Energy Theft Detection

Energy theft represents a significant threat to smart grid operations, leading to revenue losses, grid instability, and operational inefficiencies. Detecting such illegal activities requires analyzing consumption behaviors for anomalies, often using machine learning classification models. However, due to privacy concerns and data ownership restrictions, centralized data collection is often infeasible. To address this, we adopt a federated learning approach where multiple distributed clients collaboratively train an energy theft detection model without sharing raw consumption data. Each client trains the model locally on its private dataset, and only model weights and evaluation metrics are exchanged with a central server.

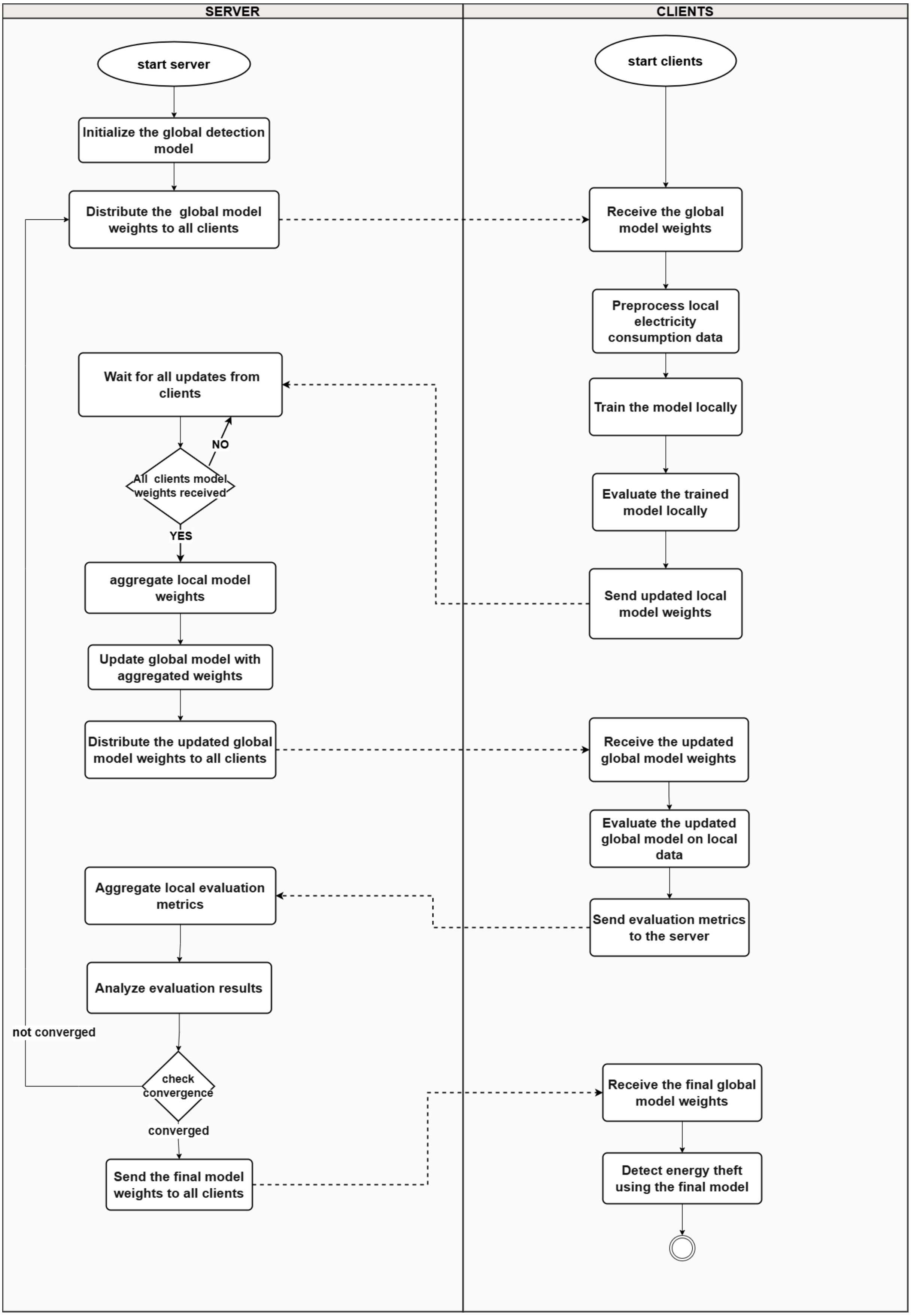

SERVER
CLIENTS
start server
Initialize the global detection model
Distribute the global model weights to all clients
Wait for all updates from clients
NO
All clients model weights received
YES
aggregate local model weights
Update global model with aggregated weights
Distribute the updated global model weights to all clients
Aggregate local evaluation metrics
Analyze evaluation results
not converged
check convergence
converged
Send the final model weights to all clients
start clients
Receive the global model weights
Preprocess local electricity consumption data
Train the model locally
Evaluate the trained model locally
Send updated local model weights
Receive the updated global model weights
Evaluate the updated global model on local data
Send evaluation metrics to the server
Receive the final global model weights
Detect energy theft using the final model

Figure 3.4: BPMN 2.0 Workflow for Federated Learning-Based Energy Theft Detection with Three Clients

3.7.1 Feature Selection Based on Permutation Importance

- **General Principle:** Permutation Importance is a model-agnostic technique that evaluates the relevance of a feature by measuring the decrease in model performance when the values of that feature are randomly shuffled. A large drop indicates high importance.
- **Methodology Steps:**
  1. **Train the model:** Train a predictive model on the complete dataset using all input features.
  2. **Compute baseline performance:** Measure the model's performance on a validation or test set to establish a baseline (e.g., accuracy, F1-score, AUC).
  3. **Permute one feature at a time:** For each feature x_j:
     - Shuffle its values randomly across all data points.
     - Evaluate the model performance using the permuted dataset.
     - Compute the drop in performance compared to the baseline.
  4. **Compute importance score:**

     $$Importance(x_j) = Performance_{baseline} - Performance_{permuted}(x_j)$$

  5. **Select features:** Rank the features based on their importance scores and retain the top ones for further modeling.

3.7.2 Time Window Configuration

In this study, a time window of 24 hours was selected for structuring the input data. This means that the model analyzes the previous 24-hour consumption values to detect potential anomalies. This configuration captures daily usage patterns, which are particularly relevant in identifying irregular behaviors related to energy theft. The 24-hour window was chosen based on domain knowledge and empirical testing, offering a good balance between capturing sufficient temporal context and ensuring model efficiency.

3.7.3 Exploration of Model Architectures

LSTM Model

The Long Short-Term Memory (LSTM) network was explored for its capacity to detect hidden sequential patterns in customer consumption behavior. Unlike its use in standard forecasting, here the model is leveraged to identify irregular temporal deviations such as sudden drops or peaks that may indicate fraudulent manipulation. The ability of LSTM to maintain temporal dependencies enables it to capture long-term correlations, which is essential when abnormal behavior develops gradually.

BiLSTM Model

The Bidirectional LSTM (BiLSTM) model was also evaluated due to its capability to access both past and future context simultaneously. In the context of energy theft detection, this dual perspective can help recognize anomalies that may not be evident when relying solely on past information. For instance, energy usage that seems typical in isolation may appear suspicious when compared to future patterns. BiLSTM thus enhances sensitivity to such contextual inconsistencies.

Final Model Selection

After experimentation and performance evaluation, the LSTM model was chosen as the final architecture for energy theft detection. Despite the theoretical advantages of BiLSTM, the LSTM model delivered better overall results in practice. It achieved a more favorable balance between complexity and accuracy, and demonstrated superior generalization in the presence of noisy consumption data typically encountered in real-world smart grid environments.
-The following figure summarizes the comparative performance of the evaluated LSTM and BiLSTM architectures, reinforcing the choice of the LSTM model as the final selected approach.

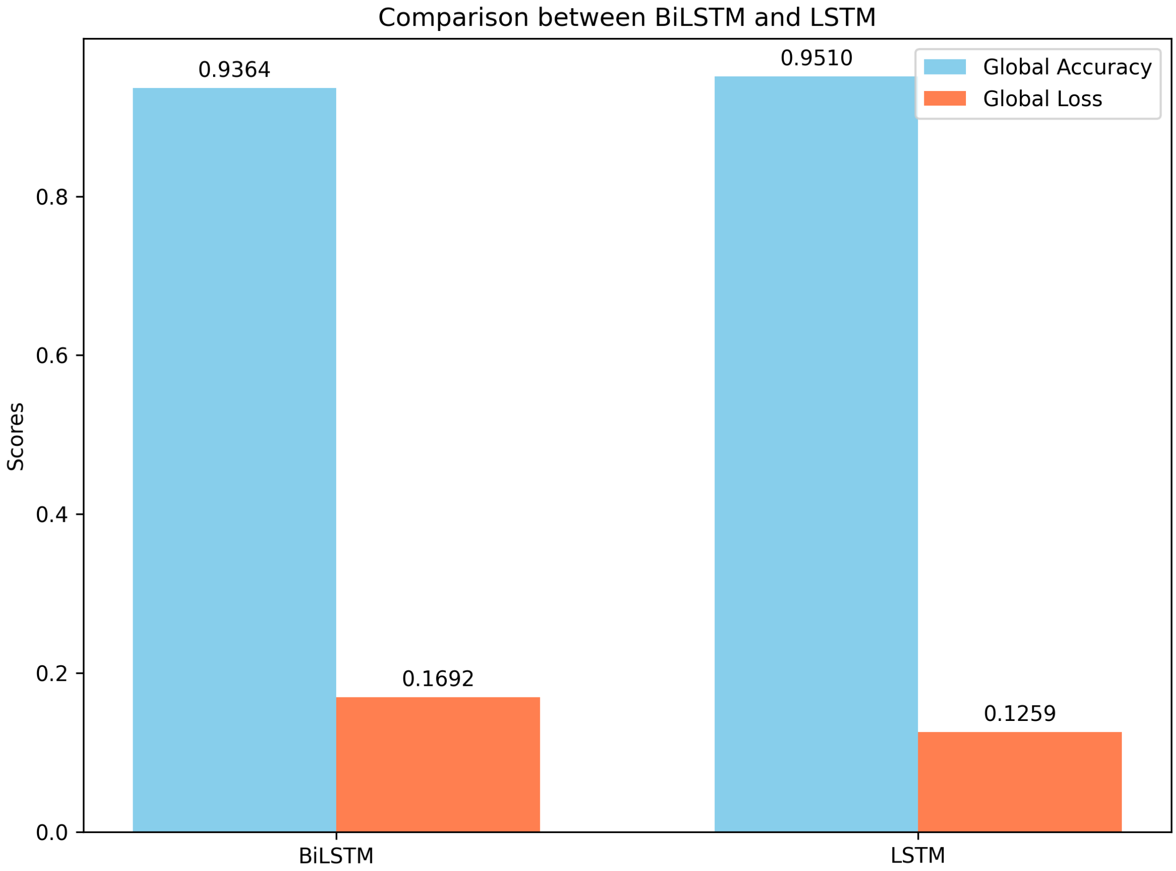


Figure 3.5: Comparison of Global Accuracy and Loss between BiLSTM and LSTM Models

3.7.4 Comparative Analysis of Federated Learning and Classical Machine Learning

Centralized Machine Learning Classifier

In the centralized learning approach, all participants connect to a **single central server** to contribute their data. The local data from each participant is uploaded directly to the cloud server, which then takes full responsibility for performing the entire model training process. This configuration offers computational efficiency for participants since they are not involved in training and do not need high-resource environments. However, this approach presents serious concerns regarding **data privacy and security**. Once the data is uploaded, it can be vulnerable to malicious activity, inference attacks, or data leaks from within the server itself. Moreover, the communication overhead increases significantly as the size of the dataset grows, creating performance bottlenecks [36].

Distributed Machine Learning Classifier: Strategies and Architectures

In large-scale machine learning tasks, especially those involving massive datasets or complex models, Distributed Machine Learning (DML) becomes essential to ensure scalability, efficiency, and performance. DML enables multiple computational nodes to collaboratively train a model. The strategies are broadly categorized into **data parallelism** and **model parallelism**, and are typically implemented through communication architectures such as **AllReduce** and the **Parameter Server**.

1. **Data Parallelism**

   Principle: The same model is replicated across multiple nodes, but each replica trains on a different subset of the data.

   Mechanism:

   - The dataset is split into balanced partitions and distributed to the nodes.
   - Each node performs local training on its own subset using an identical copy of the model.
   - After each iteration, the gradients are aggregated across all nodes commonly through AllReduce or a centralized synchronization method.
   - The global model is updated with the aggregated gradients to ensure consistent convergence.

   Advantages:

   - Highly efficient for large datasets.

2. **Model Parallelism**

   Principle: The model is partitioned across several nodes, with each node responsible for computing a portion of the model, while the full dataset is shared across all nodes.

   Mechanism:

   - The model is divided into layers, neurons.
   - During the forward pass, data flows through each segment of the model located on different nodes.
   - During the backward pass, gradients are computed in reverse order across the same nodes.
   - Communication between nodes is critical as outputs of one sub-model become inputs to the next.

   Advantages:

- Enables training of extremely large models that cannot fit into the memory of a single device.

3. **Distributed ML Using AllReduce**

   Principle: A decentralized data parallelism approach where gradient aggregation is performed using the `AllReduce` operation.

   Mechanism:

   - Each node computes local gradients after training on its data partition.
   - Gradients are averaged or summed across all nodes using AllReduce.
   - All nodes update their local models synchronously using the aggregated gradients.

   Advantages:

   - Removes the bottleneck of a centralized server.
   - High scalability in high-bandwidth environments.

4. **Distributed ML Using Parameter Server**

   Principle: A client-server architecture where one or more `parameter servers` manage and synchronize model parameters centrally.

   Mechanism:

   - The model parameters are stored on the parameter server.
   - Worker nodes train on local data and send gradients to the server.
   - The server updates the global model and returns the updated weights to the workers.
   - Can be synchronous (waiting for all workers) or asynchronous (updates occur as gradients arrive).

   Advantages:

   - Highly flexible and easy to deploy at scale.
   - Well suited to cloud and heterogeneous computing environments.

Federated Learning Classifier

Federated learning builds on the principles of distributed learning but introduces additional privacy-preserving mechanisms. In federated learning, each participant conducts the training process locally and independently, using their data and a specified number of **local epochs**. After local training, participants send only the model updates to the server. The central server then performs model aggregation, typically by averaging these updates, to form a new global model. This global model is then sent back to the participants for further training. The process continues for multiple communication rounds, enabling collaboration without ever exchanging raw data. This makes federated learning particularly suitable for privacy-sensitive applications and heterogeneous data environments, where data cannot be shared due to legal, ethical, or technical constraints [36].

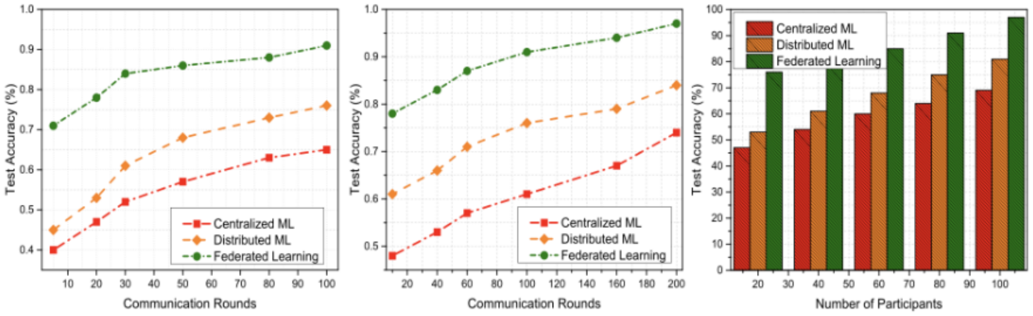
Experiments and Results

This section presents the experimental framework and the outcomes obtained through simulations, comparing classical machine learning approaches with federated learning.

- **Experimental Setup:** The experiments were carried out using two widely-used open-source datasets in the federated learning literature: **MNIST** and **CIFAR-10**. A Convolutional Neural Network (CNN) was employed for training, composed of 3 × 3 convolutional layers followed by an output layer.
  - The **MNIST** dataset includes 60,000 grayscale images of handwritten digits, each of size 28 × 28 pixels.
  - The **CIFAR-10** dataset contains 50,000 colored images, each of size 32 × 32 pixels, across 10 distinct classes.

  All experiments were conducted on a machine equipped with an Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz and 32 GB of RAM.

- **Results:** This section details the results obtained from the experiments and offers a comparative analysis between centralized, distributed, and federated learning approaches. Three experimental scenarios were considered:
  - **Scenario 1:** 50 participants were deployed with a 20% participation rate. Each of the three learning approaches was executed for 100 communication rounds on both datasets.
  - **Scenario 2:** Same configuration as Scenario 1, but using 200 communication rounds.
  - **Scenario 3:** The number of participants was varied as $p = \{20, 40, 60, 80, 100\}$ while keeping the 20% participation rate. All three learning algorithms were run for 100 communication rounds on both datasets [36].



(a) Scenario one (b) Scenario two (c) Scenario three

Figure 3.6: Convergence comparison on MNIST dataset in three different scenarios.

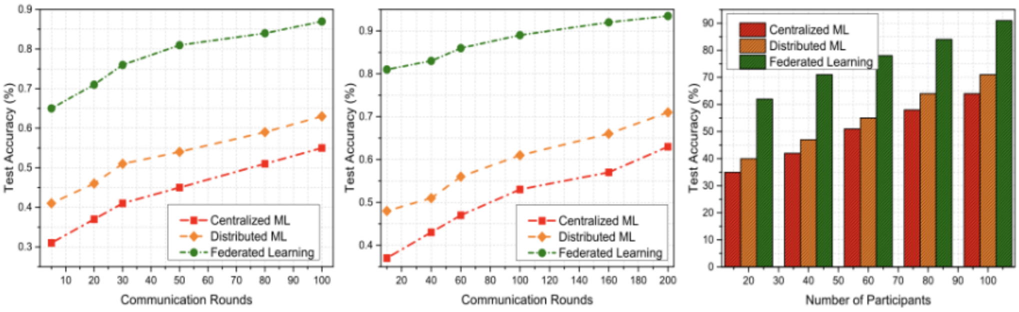


Figure 3.7: Convergence comparison on CIFAR-10 dataset in three different scenarios.

- **Performance Comparison and Convergence Analysis**

  Figures 3.8 and 3.9 illustrate the convergence behavior of centralized, distributed, and federated learning across the MNIST and CIFAR-10 datasets under the three experimental scenarios described earlier. In all distributed and federated learning experiments, the number of local training epochs was fixed at 10.

  The graphs demonstrate that decentralized approaches, especially federated learning achieve superior performance compared to centralized learning. Centralized machine learning exhibits the lowest accuracy due to the significant communication overhead it introduces. In this approach, large volumes of data must be uploaded to a central server at each communication round, which is especially problematic under limited bandwidth conditions. Specifically, centralized learning only achieved 65% and 73% accuracy on MNIST in scenarios one and two, respectively, and just 54% and 62% accuracy on CIFAR-10.

  Distributed machine learning showed moderate improvements, outperforming centralized learning but still falling short of federated learning. This improvement stems from the fact that each participant independently trains their local model before submitting updates. However, privacy concerns and reliance on a central server limited the performance, yielding 72% and 78% accuracy on MNIST, and 67% and 72% on CIFAR-10 in scenarios one and two.

  Federated learning consistently outperformed both other approaches across all scenarios. It achieved the highest accuracy: 92% and 97% on MNIST, and 86% and 94% on CIFAR-10 in scenarios one and two, respectively.

  In scenario three, where the number of participants varies, all approaches initially struggle with convergence when the participant count is low. This is due to the insufficient data available for training on each client. However, as the number of participants increases, performance improves steadily for all models. Despite the initial drop in accuracy with fewer clients, federated learning remains the most resilient, maintaining superior results even with limited participation.

  These findings highlight the strength of federated learning as a modern solution, particularly in contexts involving large-scale data and sensitive user information. Unlike traditional centralized models, federated learning offers enhanced performance while preserving data privacy an increasingly vital feature in todays digital landscape [36].

3.8 Conclusion

This chapter presented the methodological framework adopted for building a robust Federated Learning system tailored to the dual tasks of electricity load forecasting and energy theft detection. We began by addressing the characteristics and challenges of the real-world data, including necessary preprocessing steps such as cleaning, normalization, and temporal sequencing. The proposed federated architecture was then described in detail, highlighting the coordination between clients and the server, and incorporating multiple aggregation strategies to mitigate issues like client heterogeneity and malicious updates. Finally, we introduced and developed suitable model architectures for both forecasting and classification tasks, supporting them with rigorous feature selection and comparative evaluation against centralized approaches. This comprehensive methodology serves as a strong foundation for the practical implementation and performance evaluation discussed in the next chapter.

Chapter 4

Implementation and Results

4.1 Introduction

This chapter details the practical implementation of the proposed Federated Learning framework applied to electric load forecasting and electricity theft detection. It begins by describing the development environment and tools used, followed by the integration of the Flower framework to orchestrate communication between clients and the central server. Key implementation challenges such as data heterogeneity (non-IID distribution), class imbalance, and client drift are addressed through various strategies and the use of robust aggregation algorithms. The final section presents experimental results and performance evaluation for different aggregation methods, using regression and classification metrics.

4.2 Environment and development tools

In our project, we use Anaconda environment, Python language, Tensorflow, and Flower packages that are defined as follows:

- **Anaconda :** It is an open-source Python distribution designed for data science, providing a streamlined solution for package management and environment deployment. It uses the Conda package manager, which analyzes the current environment before installing new packages to prevent conflicts and ensure compatibility with existing tools and libraries [37].

- **Python :** It is an interpreted, object-oriented, high-level programming language designed to be simple, readable, and easy to learn. It supports rapid application development with powerful data structures and dynamic typing. Its fast development cycle, without a compilation step, makes debugging easier and boosts programmer productivity [38].

- **TensorFlow :** is an open source machine learning framework that provides Python and Java-based libraries and tools, specifically designed for building and training machine learning and deep learning models using data [39].

- **Flower :**is a flexible and extensible framework for building federated AI systems. Designed to support diverse use cases, it is customizable, framework agnostic (compatible with tools like PyTorch, TensorFlow, and scikit-learn), and was originally developed for AI research at the University of Oxford. Its clear and maintainable codebase encourages community contribution and adaptation [40].

4.3 Implementation of Federated Learning with Flower

The implementation of Federated Learning is carried out using the Flower framework. The architecture adopted is client-server based, allowing a decentralized model training across multiple nodes while preserving data privacy.

4.3.1 Internal Architecture and Workflow of the Flower Framework

In a federated learning (FL) setup, a typical architecture is composed of **a central coordinating server** and **multiple distributed clients**, forming what is referred to as a **federation**. The server oversees the orchestration of training rounds, while the clients independently train local models on private data and return updates to the server. This setup is commonly described as a **hub-and-spoke topology**.To enable modularity, scalability, and support for multiple concurrent FL projects, the Flower framework adopts a layered and extensible architecture, in which both the server and client sides are split into two major components: one responsible for **persistent communication** and one handling **project specific learning logic** [40].

1. Server-Side Components

SuperLink

-*Role*: A long-running, infrastructure-level process that manages all network communication between the server and the clients (SuperNodes).

-*Function*: It dispatches training tasks, receives client results, and ensures robust and efficient transmission of instructions and data. It is designed to remain active throughout the lifespan of the system.

ServerApp

-*Role*: A short-lived, user-defined component responsible for the specific logic of the federated learning project.

-*Function*: It defines key elements such as:

- Client selection strategy
- Aggregation algorithms
- Training round configuration

-*Customizable*: Developed by AI researchers or engineers, it is tailored to suit different experimental or application needs.

2. Client-Side Components

SuperNode

-*Role*: A long-running process deployed on the client device, responsible for maintaining the connection to the central server via the SuperLink.

-*Function*: It continuously listens for incoming tasks, executes them (e.g., local training or evaluation), and returns results to the server.

-*Autonomous Execution*: Ensures local operations are performed while preserving the privacy of local data.

ClientApp

-*Role*: A short-lived, project-specific component executed by the SuperNode when a training or evaluation task is received.

-*Function*: Includes the actual model training code, pre-processing and post-processing logic, and evaluation metrics.

-*Customizable*: Like ServerApp, it is developed by the user to fit the machine learning goals of the project.

4.4 Non-IID Problem

Non-IID (non-independent and identically distributed) federated learning refers to a federated learning setup where the data across different clients is heterogeneous, meaning it does not follow the same distribution. Unlike IID settings, where each clients data is assumed to be drawn from the same probability distribution, non-IID data varies significantly in terms of statistical properties, such as class distributions, feature distributions, or data sizes. This heterogeneity poses challenges in model training, as the global model must generalize across diverse local datasets while avoiding issues like model bias or poor convergence.

4.4.1 Class Imbalance in Non-IID Federated Learning

Class imbalance in non-IID federated learning refers to a scenario where the distribution of class labels is uneven across different clients or within a clients local dataset, and the data is non-independent and identically distributed (non-IID). In non-IID settings, clients have heterogeneous data distributions, meaning their local datasets may differ significantly in terms of class proportions, feature distributions, or data sizes. Class imbalance exacerbates this heterogeneity, as some clients may have an abundance of data for certain classes (majority classes) and very little for others (minority classes). This leads to challenges in training a global model that generalizes well across all clients and classes, often resulting in biased models that perform poorly on underrepresented classes.

To address class imbalance in non-IID federated learning, two commonly used techniques are:

SMOTE (Synthetic Minority Over-sampling Technique) :

SMOTE is a data-level technique used to combat class imbalance by synthetically generating new examples for the minority class. Rather than duplicating samples, it creates new instances by interpolating between a minority sample and one of its k nearest neighbors of the same class. Mathematically, for a given minority class sample x_i, a synthetic sample $\tilde{x}$ is generated as:

$$\tilde{x} = x_i + \lambda \cdot (x_{zi} - x_i)$$

where:

- x_i is a sample from the minority class,
- x_{zi} is one of the k nearest neighbors of x_i,
- $\lambda \in [0, 1]$ is a random number controlling the interpolation.

In federated learning settings, SMOTE can be applied locally on each client to ensure a more balanced representation of classes before local training, helping to reduce bias and improve generalization across heterogeneous clients.

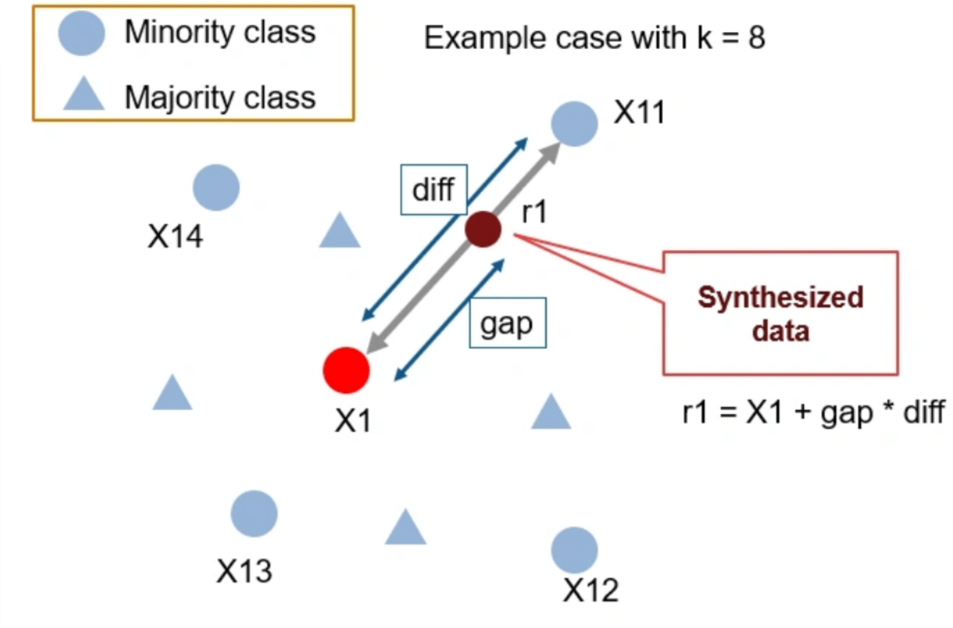


Figure 4.1: Application of SMOTE

Balanced Weighting :

Balanced weighting is a technique used to mitigate class imbalance by assigning higher weights to minority classes during model training. This ensures that their contribution to the loss function remains significant despite their low frequency in the dataset. This method helps prevent the model from being biased toward the majority class. Mathematically, in classification tasks with imbalanced classes, the weight w_c assigned to each class c can be computed as:

$$w_c = \frac{N}{n_{classes} \cdot n_c}$$

where:

- N is the total number of training samples,
- $n_{classes}$ is the total number of classes,
- n_c is the number of samples belonging to class c.

These weights are then incorporated into the loss function. For instance, in the weighted cross-entropy loss:

$$\mathcal{L} = -\sum_{c=1}^{C} w_c \cdot y_c \cdot \log(\hat{y}_c)$$

where:

- C is the number of classes,
- y_c is the true label (1 if the sample belongs to class c, 0 otherwise),
- $\hat{y}_c$ is the predicted probability for class c.

In federated learning, this weighting is applied locally on each client to adjust the models sensitivity to class distributions, allowing each client to better handle its own data imbalance.

4.5 Handling Client Drift

In federated learning, client drift refers to the phenomenon where updates generated by local client models diverge from the direction of the optimal global model. This is often due to the heterogeneous (non-IID) nature of the data across clients, where each client has access to a subset of the data that does not represent the global distribution. As a result, local training causes the model to overfit or be biased toward the clients data, and the global aggregation suffers from inconsistent updates.

4.5.1 Proposed Mitigation Strategies

To address the challenges posed by client drift, particularly in settings with highly heterogeneous data such as smart grid environments, several federated learning strategies have been developed. These methods aim to ensure more stable and coherent convergence by reducing the discrepancies between local and global updates. Below, we describe how each technique contributes specifically to mitigating client drift.

Clustering-Based Federated Learning

This approach involves partitioning clients into clusters based on shared behavioral patterns, for example, categorizing them according to **high**, **medium**, or **low** energy consumption. Since clients within the same cluster tend to exhibit similar data distributions, the local updates become more consistent and representative of their group. By training a distinct global model per cluster, this method effectively reduces internal drift and improves both accuracy and personalization, especially in environments with strong user variability [41].

FedProx

FedProx is designed to **directly address client drift** by modifying the way clients perform local training. In standard federated learning, clients train their models locally for several epochs before sending their updates to the server, often causing them to drift far from the global model, especially when data is non-IID. FedProx introduces a **proximal term** in the local objective that penalizes large deviations from the global model, thereby forcing local updates to remain closer to the global direction.
In practice, this mechanism acts like a soft constraint: it does not completely prevent adaptation to local data but discourages updates that deviate too strongly. This is particularly useful when clients have highly variable patterns, such as in energy forecasting, where one household's behavior can be vastly different from anothers. FedProx stabilizes learning and improves generalization across clients by aligning local learning trajectories with the global objective [41].

FedAdam

FedAdam tackles client drift from the **server-side aggregation** perspective. Instead of averaging raw client updates uniformly, FedAdam adapts the global model update by applying **adaptive learning rates and momentum**, similar to the Adam optimizer used in centralized deep learning. This allows the server to smooth out the effect of sudden or inconsistent updates that might come from clients with biased or noisy data. In a federated setting with heterogeneous clients, some updates can be highly erratic and harm the learning process. FedAdam reduces this risk by adjusting the contribution of each update based on the history of past gradients and their variance. This makes the global model more resilient to drift and accelerates convergence in non-IID environments.

FedTrimmedAvg

FedTrimmedAvg mitigates client drift by introducing **robust aggregation** at the server level. Instead of averaging all client updates equally, this strategy excludes a fraction of the most extreme updates, those that differ significantly from the majority before computing the global update. This mechanism is particularly effective in scenarios where a subset of clients behaves abnormally due to noisy, biased, or non-representative local data. By trimming outliers, FedTrimmedAvg limits their influence on the global model, leading to more stable and robust convergence. In energy applications, where some clients may exhibit unusual consumption patterns, this approach helps prevent their updates from derailing the global learning process.

4.6 Performance Evaluation and Results

4.6.1 Metrics Used

In order to comprehensively assess the performance of our models, we employ a combination of classification and regression evaluation metrics. These metrics are chosen based on the nature of each task and provide both general and fine-grained insights into model behavior.

Regression

For regression tasks such as load forecasting, we use the following metrics:

- **Mean Squared Error (MSE)**
  MSE is the average of the squared differences between predicted and actual values. It emphasizes larger errors and is sensitive to outliers.

- **Mean Absolute Error (MAE)**
  MAE calculates the average of the absolute differences between predicted and actual values. It provides an easily interpretable error magnitude.

- **Root Mean Squared Error (RMSE)**
  RMSE is the square root of MSE. It retains the same unit as the original data and penalizes larger errors more strongly.

- **R-squared (R^2 Score)**
  R^2 indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. A value close to 1 implies a good fit between prediction and ground truth.

Classification

For classification tasks such as energy theft detection, we use the following evaluation metrics:

- **Confusion Matrix**
  The confusion matrix provides a summary of prediction results in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It enables a visual and quantitative understanding of model performance.

- **Accuracy**
  The ratio between the number of correctly predicted instances and the total number of instances. Defined as:
  $$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- **Precision**
  The ratio between the number of true positives and the number of predicted positive instances. Defined as:
  $$Precision = \frac{TP}{TP+FP}$$

- **Recall**
  The ratio between the number of true positives and the number of actual positive instances. Defined as:
  $$Recall = \frac{TP}{TP+FN}$$

- **F1 Score**
  The harmonic mean of precision and recall. Defined as:
  $$F1Score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$$

- **AUC (Area Under the ROC Curve)**
  A scalar value that represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate.

- **Log Loss (Logarithmic Loss)**
  A loss function that measures the performance of a classification model where the predicted output is a probability value between 0 and 1. Defined as:
  $LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$

4.6.2 Results of Load Forecasting

The evaluation results presented below correspond to the performance of the load forecasting model for each local client, as well as for the aggregated global model. These results illustrate the models ability to capture clientspecific consumption patterns and reveal the impact of data heterogeneity on local predictions. Moreover, the comparison between local and global performances highlights the balance between personalization and generalization achieved by the approach.

Performance of the FedAvg Aggregation Method

Client 1:

- **MSE: 0.0021**
  The model yields a notably low Mean Squared Error, reflecting a high level of overall predictive accuracy. This indicates that the model performs well in minimizing the squared differences between the predicted and actual energy consumption values for Client 1.

- **MAE: 0.0334**
  The Mean Absolute Error remains low, suggesting that, on average, the models predictions deviate only slightly from the true values. This highlights the models reliability in providing consistently accurate estimations.

- **RMSE: 0.0454**
  The Root Mean Squared Error further confirms the models robustness by penalizing larger errors more heavily. The low RMSE value demonstrates that extreme deviations are rare, reinforcing the models stability on local data.

- **R^2: 0.9408**
  With a coefficient of determination of approximately 0.94, the model successfully explains 94.08% of the variance observed in the actual consumption data. This indicates a strong correlation between the predicted and real values and affirms the models effectiveness in capturing the underlying patterns.

- **MedAE: 0.0253**
  The Median Absolute Error is particularly low, implying that at least half of the prediction errors are below this threshold. This further emphasizes the models precision and robustness, even in the presence of potential outliers.

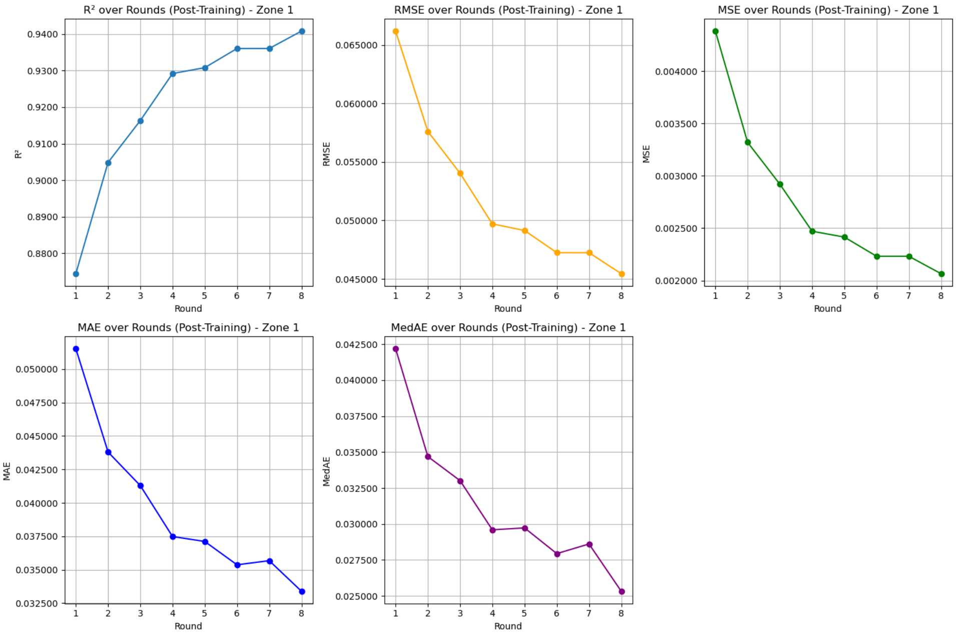


Figure 4.2: Performance metrics visualization for Client 1

Client 2:

- **MSE: 0.0022**
  The model maintains a low Mean Squared Error, indicating a good level of predictive accuracy. It effectively minimizes the average squared differences between predicted and actual energy consumption values for Client 2.

- **MAE: 0.0347**
  The Mean Absolute Error remains low, suggesting that the model provides predictions with only slight deviations from the actual values on average, thereby ensuring reliable performance.

- **RMSE: 0.0469**
  The Root Mean Squared Error remains close to that of Client 1, reflecting the models consistent ability to limit larger prediction errors and maintain robust forecasting performance.

- **R^2: 0.9301**
  With an R-squared value of approximately 0.93, the model explains 93.01% of the variance in the actual data for Client 2. This demonstrates a strong fit between predicted and observed values.

- **MedAE: 0.0265**
  The Median Absolute Error is low, indicating that at least half of the predictions differ from the actual values by less than 0.0265. This supports the model's robustness and precision in handling Client 2's data.

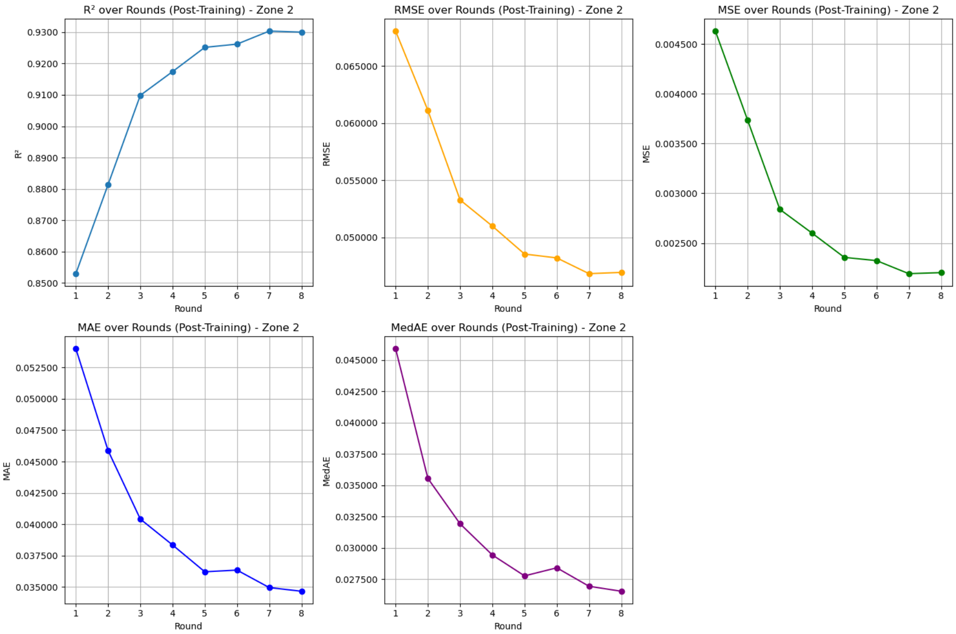


Figure 4.3: Performance metrics visualization for Client 2

Client 3:

- **MSE: 0.0012**
  The model achieves the lowest Mean Squared Error among the clients, indicating excellent predictive accuracy and minimal average squared differences between the predicted and actual energy consumption values for Client 3.

- **MAE: 0.0242**
  The Mean Absolute Error is also the lowest across all clients, suggesting that the model provides highly accurate predictions with very small average deviations.

- **RMSE: 0.0345**
  The Root Mean Squared Error confirms the models strong stability on Client 3's data, with very few large prediction errors, further reinforcing the models reliability.

- **R^2: 0.9542**
  With an R-squared value of 0.9542, the model explains 95.42% of the variance in the actual energy consumption data. This reflects a very strong fit and the best performance in terms of variance explanation among the three clients.

- **MedAE: 0.0173**
  The Median Absolute Error is the lowest, indicating that at least half of the models predictions are within 0.0173 units of the actual values. This demonstrates high precision and robustness on Client 3s data.

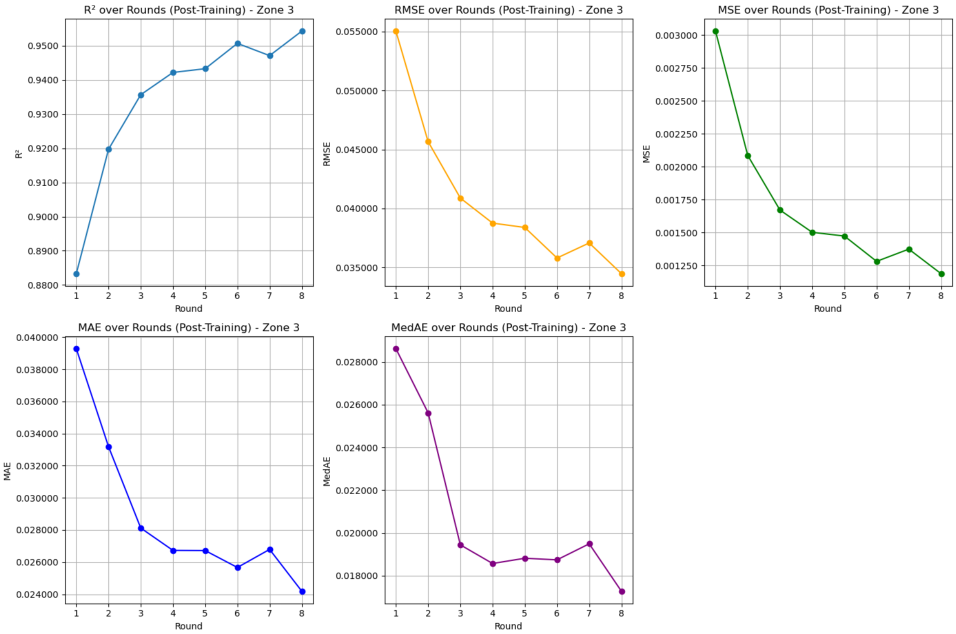


Figure 4.4: Performance metrics visualization for Client 3

Global Model:

- **MSE: 0.0027**
  The global model yields a low Mean Squared Error, indicating good overall predictive performance across all clients. While slightly higher than individual client models, it still reflects acceptable precision in forecasting energy consumption in a federated setting.

- **MAE: 0.0384**
  The Mean Absolute Error remains moderate, suggesting that the aggregated model maintains consistent accuracy across diverse local data distributions.

- **RMSE: 0.0517**
  The Root Mean Squared Error, while slightly elevated compared to individual clients, demonstrates that the model handles variability reasonably well, with no extreme prediction errors dominating the results.

- **$\mathbf{R^2}$: 0.9125**
  With an R-squared value of 0.9125, the global model explains over 91% of the variance in the actual consumption data across all clients. This indicates a strong overall fit, despite the data heterogeneity inherent in federated learning.

- **MedAE: 0.0287**
  The Median Absolute Error confirms the robustness of the global model, showing that at least half of the predictions deviate by less than 0.0287 units from the actual values, even in a non-IID data context.

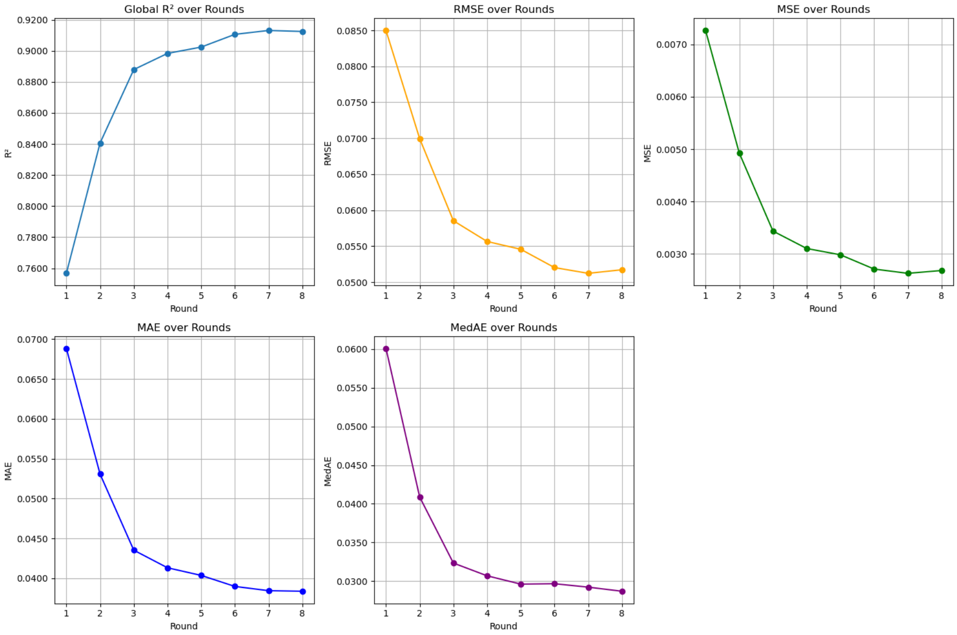


Figure 4.5: Performance metrics visualization for Global Model

Performance of the FedProx Aggregation Method

Client 1:

- **MSE: 0.0024**
  The model produces a low Mean Squared Error, indicating good overall prediction accuracy for Client 1. It shows that the model effectively minimizes the average squared differences between predicted and actual consumption values.

- **MAE: 0.0366**
  The Mean Absolute Error suggests that, on average, the predictions deviate only slightly from the actual values, reinforcing the models reliability on local data.

- **RMSE: 0.0486**
  The Root Mean Squared Error supports the models ability to control larger errors, confirming its stability and consistency in forecasting for Client 1.

- **R^2: 0.9322**
  With an R-squared value of 0.9322, the model explains 93.22% of the variance in the actual consumption data, indicating a strong fit and effective learning of underlying patterns.

- **MedAE: 0.0290**
  The Median Absolute Error further highlights the models robustness, showing that at least half of the prediction errors are below 0.0290, even in the presence of potential outliers.

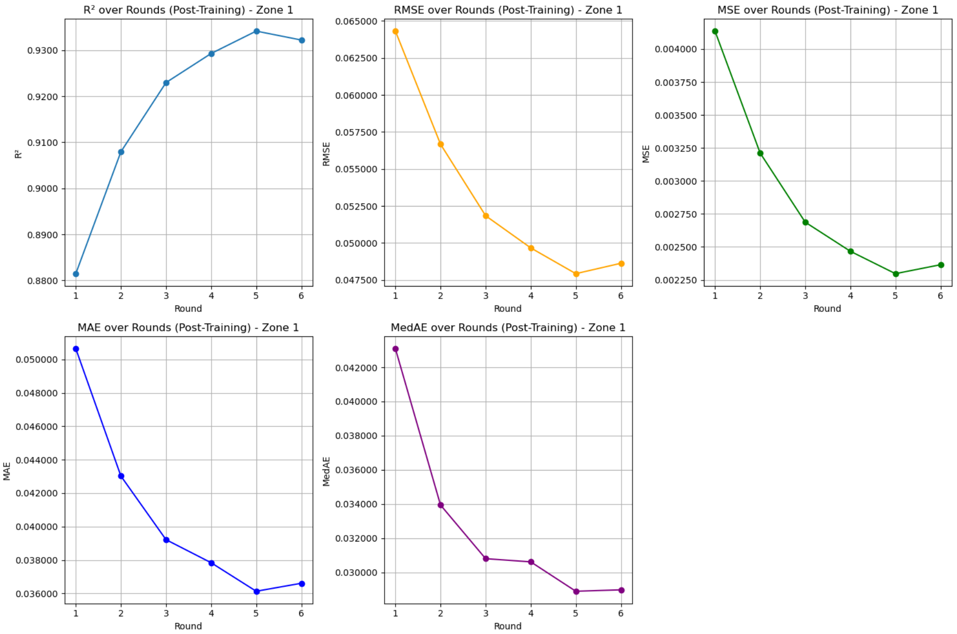


Figure 4.6: Performance metrics visualization for Client 1

Client 2:

- **MSE: 0.0023**
  The model achieves a low Mean Squared Error, reflecting a strong ability to minimize the average squared differences between predicted and actual energy consumption values for Client 2.

- **MAE: 0.0361**
  The Mean Absolute Error confirms that the model provides accurate predictions on average, with relatively small deviations from the true values.

- **RMSE: 0.0484**
  The Root Mean Squared Error shows that the model controls larger errors effectively, reinforcing its robustness and reliability in handling Client 2s data.

- **R^2: 0.9255**
  With an R-squared value of 0.9255, the model is able to explain 92.55% of the variance in the actual consumption data, indicating a good fit to the local data distribution.

- **MedAE: 0.0277**
  The low Median Absolute Error highlights the consistency of the model's predictions, with at least half of the prediction errors falling below 0.0277, even in potentially noisy conditions.

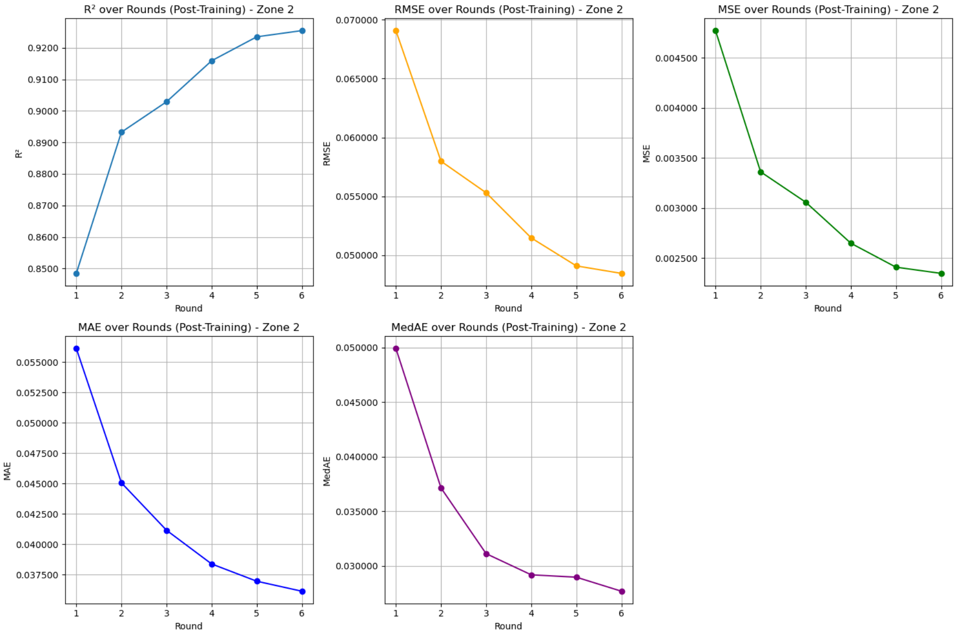


Figure 4.7: Performance metrics visualization for Client 2

Client 3:

- **MSE: 0.0014**
  The model records the lowest Mean Squared Error among the clients, indicating high prediction accuracy and a strong ability to minimize the average squared error in Client 3s energy consumption data.

- **MAE: 0.0262**
  The Mean Absolute Error is also notably low, reflecting minimal average deviation between the predicted and actual values, and highlighting the models precision.

- **RMSE: 0.0373**
  The Root Mean Squared Error supports the conclusion that the model maintains strong performance even when penalizing larger deviations, confirming its reliability for Client 3.

- **R^2: 0.9464**
  With an R-squared value of 0.9464, the model explains 94.64% of the variance in the actual energy consumption data. This demonstrates a very strong fit and excellent generalization on local data.

- **MedAE: 0.0187**
  The very low Median Absolute Error indicates that at least half of the prediction errors are under 0.0187, underscoring the models robustness and consistent accuracy, even in the presence of outliers or fluctuations.

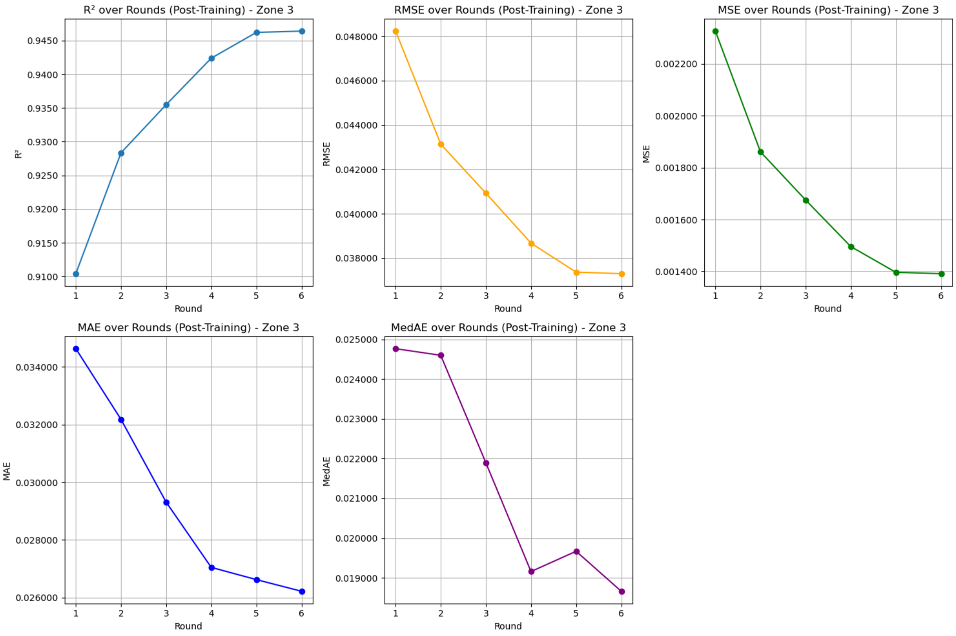


Figure 4.8: Performance metrics visualization for Client 3

Global Model:

- **MSE: 0.0031**
  The global model exhibits a slightly higher Mean Squared Error compared to local client models, which is expected due to the challenge of generalizing across heterogeneous data. Nevertheless, the value remains low, indicating acceptable prediction accuracy at the global level.

- **MAE: 0.0410**
  The Mean Absolute Error shows that, on average, predictions deviate modestly from the actual values, reflecting reasonable accuracy across diverse consumption patterns.

- **RMSE: 0.0554**
  The Root Mean Squared Error, while higher than that of individual clients, remains within an acceptable range. It confirms that large prediction errors are infrequent, despite the variability introduced by data heterogeneity.

- **R^2: 0.8993**
  With an R-squared value of 0.8993, the global model explains nearly 90% of the variance in energy consumption data. This indicates a strong overall fit, although slightly lower than that of the local models due to the need to generalize across all clients.

- **MedAE: 0.0302**
  The Median Absolute Error remains low, showing that at least half of the prediction errors are under 0.0302. This highlights the models robustness and stability across diverse client datasets.

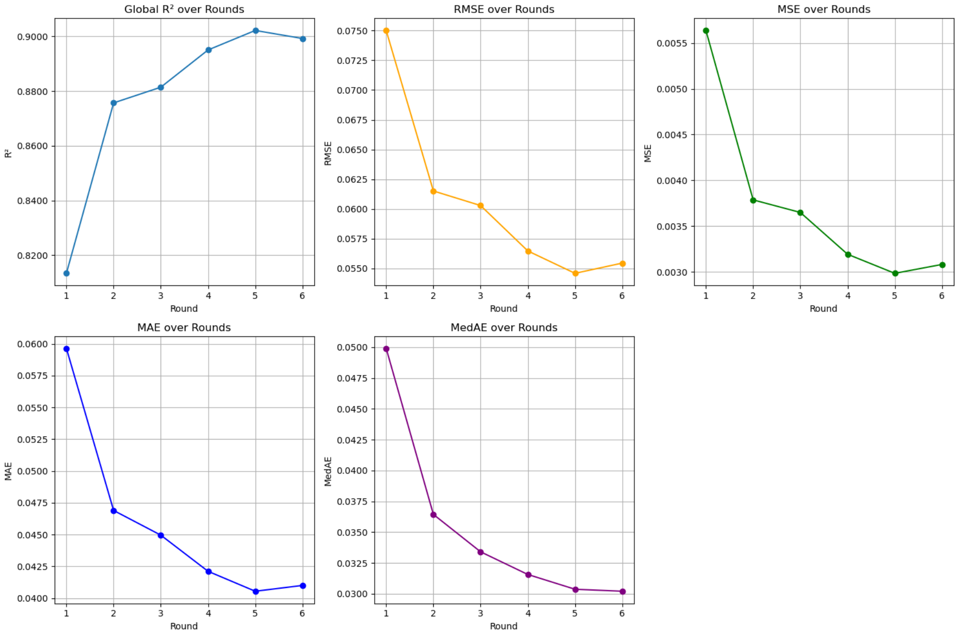


Figure 4.9: Performance metrics visualization for Global Model

Performance of the FedTrimmedAvg Aggregation Method

Client 1:

- **MSE: 0.0020**
  The model achieves a low Mean Squared Error, indicating a strong ability to minimize the average squared difference between predicted and actual energy consumption values for Client 1.

- **MAE: 0.0332**
  The Mean Absolute Error reflects minimal average deviation from the true values, suggesting high reliability and consistency in the models local predictions.

- **RMSE: 0.0449**
  The Root Mean Squared Error supports the models robustness by penalizing larger deviations more heavily, with this low value confirming stable performance on local data.

- **R^2: 0.9422**
  With an R-squared score of 0.9422, the model is able to explain 94.22% of the variance in actual consumption data, showing strong predictive power and excellent alignment with real-world patterns.

- **MedAE: 0.0252**
  The low Median Absolute Error indicates that at least half of the models predictions have an error below 0.0252, highlighting its robustness and resistance to outliers.

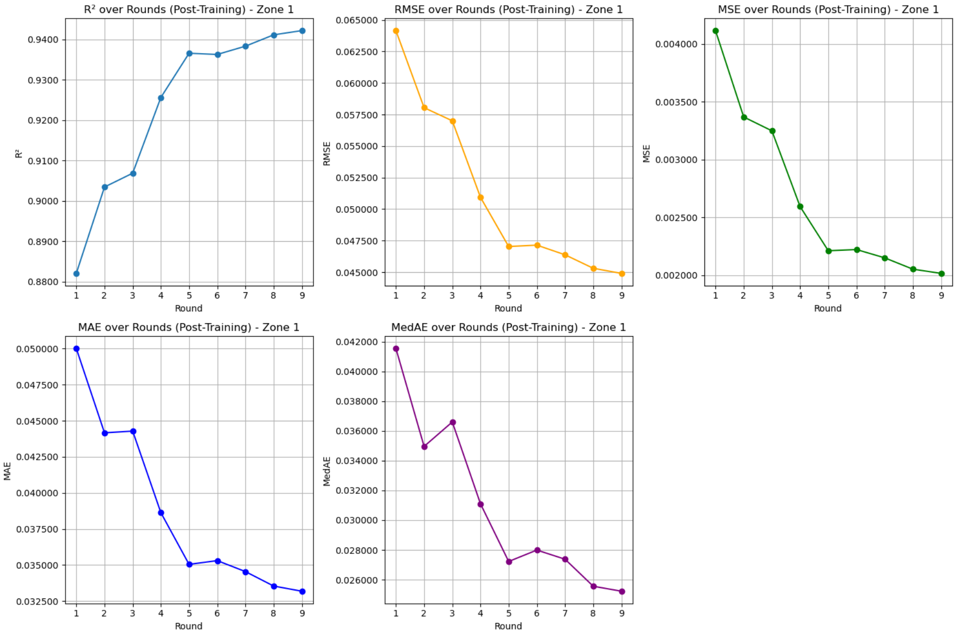


Figure 4.10: Performance metrics visualization for Client 1

Client 2:

- **MSE: 0.0020**
  The model achieves a low Mean Squared Error, indicating accurate predictions with minimal average squared deviations from actual energy consumption values for Client 2.

- **MAE: 0.0333**
  The Mean Absolute Error remains low, suggesting that the models predictions deviate only slightly from the actual values on average, thus ensuring local reliability.

- **RMSE: 0.0444**
  The Root Mean Squared Error confirms the models ability to limit significant deviations, reinforcing its robustness and stability on the local dataset.

- **R^2: 0.9375**
  With an R-squared value of 0.9375, the model explains over 93% of the variance in the actual data, showing that it effectively captures the underlying patterns in Client 2s consumption behavior.

- **MedAE: 0.0259**
  The low Median Absolute Error indicates that at least half of the prediction errors are under 0.0259, highlighting the models consistency and resilience to outliers or anomalies in the data.

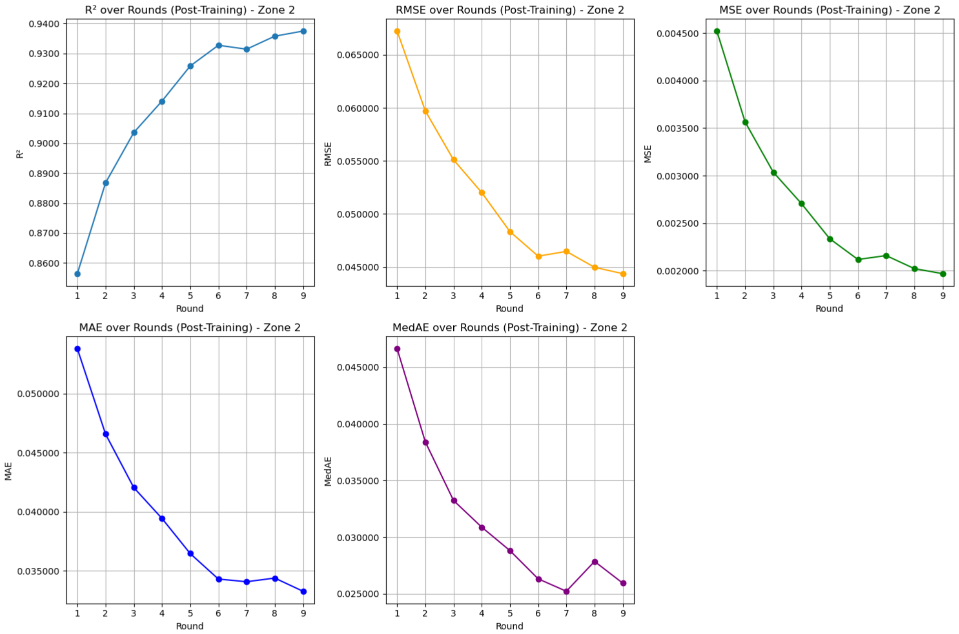


Figure 4.11: Performance metrics visualization for Client 2

Client 3:

- **MSE: 0.0013**
  The model achieves the lowest Mean Squared Error among all clients, indicating exceptional accuracy in predicting Client 3s energy consumption with minimal average squared error.

- **MAE: 0.0251**
  The low Mean Absolute Error shows that the models predictions are, on average, very close to the actual values, highlighting high precision and reliability.

- **RMSE: 0.0354**
  The Root Mean Squared Error is also the lowest across all clients, confirming that large deviations are rare and that the model generalizes well for Client 3.

- **R^2: 0.9518**
  With an R-squared value of 0.9518, the model explains 95.18% of the variance in the actual data, demonstrating excellent fit and strong predictive capability.

- **MedAE: 0.0175**
  The very low Median Absolute Error indicates that at least half of the model's predictions deviate from the true values by less than 0.0175, confirming both accuracy and consistency even in the presence of data variability.

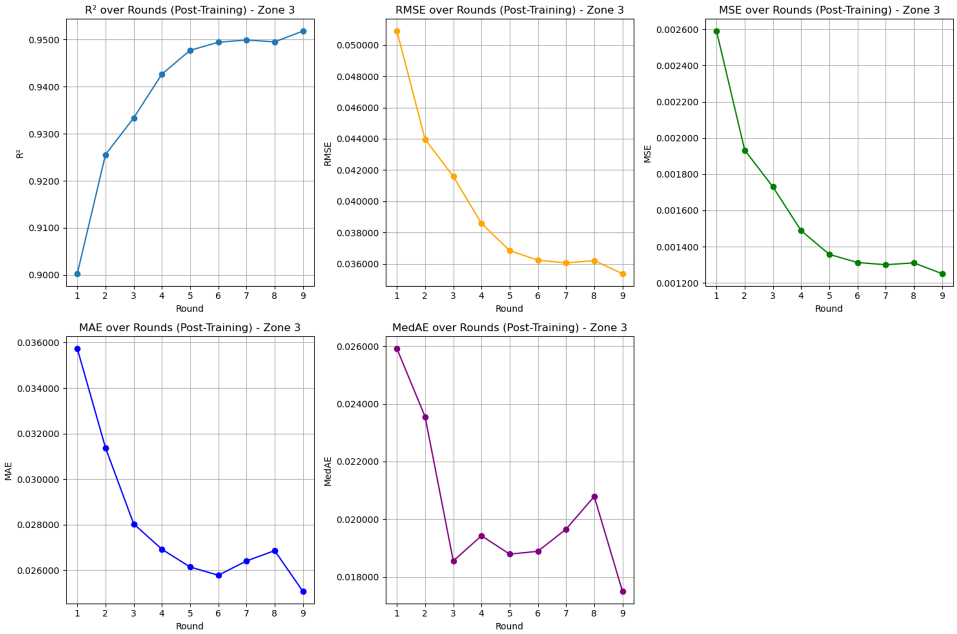


Figure 4.12: Performance metrics visualization for Client 3

Global Model:

- **MSE: 0.0025**
  The global model achieves a very low Mean Squared Error, indicating highly accurate predictions with minimal average squared differences between predicted and actual energy consumption values across federated clients.

- **MAE: 0.0376**
  The Mean Absolute Error suggests that, on average, the model's predictions deviate by only 0.0376 units from the actual valuesdemonstrating reliable performance across diverse client datasets.

- **RMSE: 0.0025**
  The Root Mean Squared Error confirms a consistent minimization of large deviations in prediction errors, reinforcing the models stability in a federated setting.

- **R^2: 0.9170**
  An R-squared value of 0.9170 indicates that the model explains approximately 91.7% of the variance in energy consumption, reflecting strong generalization capability despite the heterogeneity of client data.

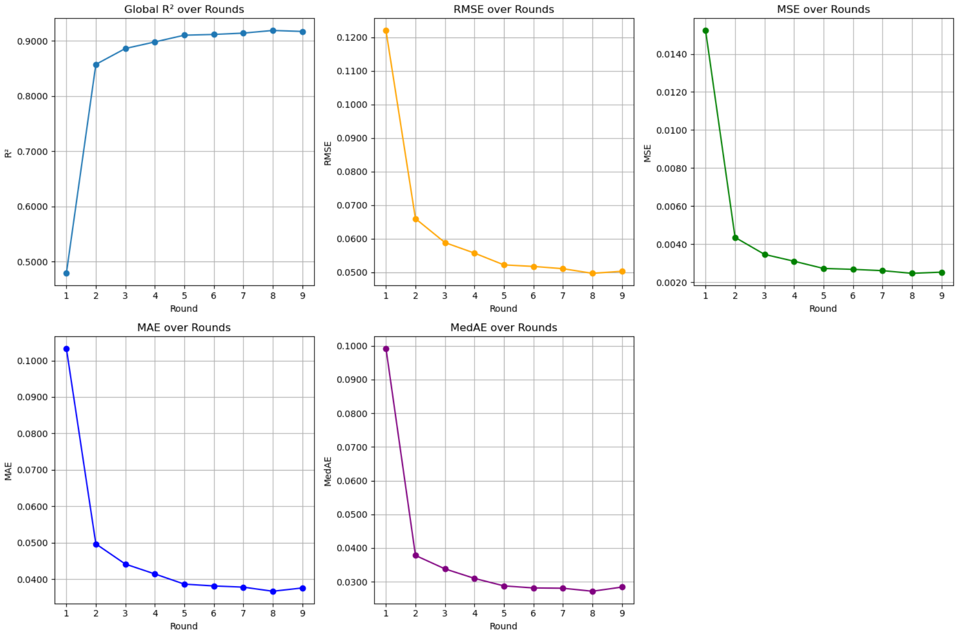


Figure 4.13: Performance metrics visualization for Global Model

4.6.3 Results of Theft Detection

The evaluation results presented below correspond to the performance of the energy theft detection model for each local client (Client 1, Client 2, and Client 3), as well as for the aggregated global model. These results highlight the impact of data heterogeneity and client-specific distributions on local performance and how they contrast with the generalized global model.

Performance of the FedAvg Aggregation Method

Client 1:

- **Confusion Matrix**
  - **True Negatives (TN = 18,786)**: The model accurately identified the majority of legitimate energy consumption cases, confirming its ability to minimize false positive alarms.
  - **True Positives (TP = 12,753)**: A high proportion of theft cases were correctly detected, demonstrating the model's effectiveness in recognizing fraudulent behavior.
  - **False Positives (FP = 590)**: A slightly higher number of normal instances were incorrectly flagged as theft, compared to previous evaluations, suggesting a minor decline in precision and the need for further tuning.
  - **False Negatives (FN = 907)**: These undetected theft cases represent missed detections, but the reduced number compared to previous figures indicates improvement in the model's recall performance.

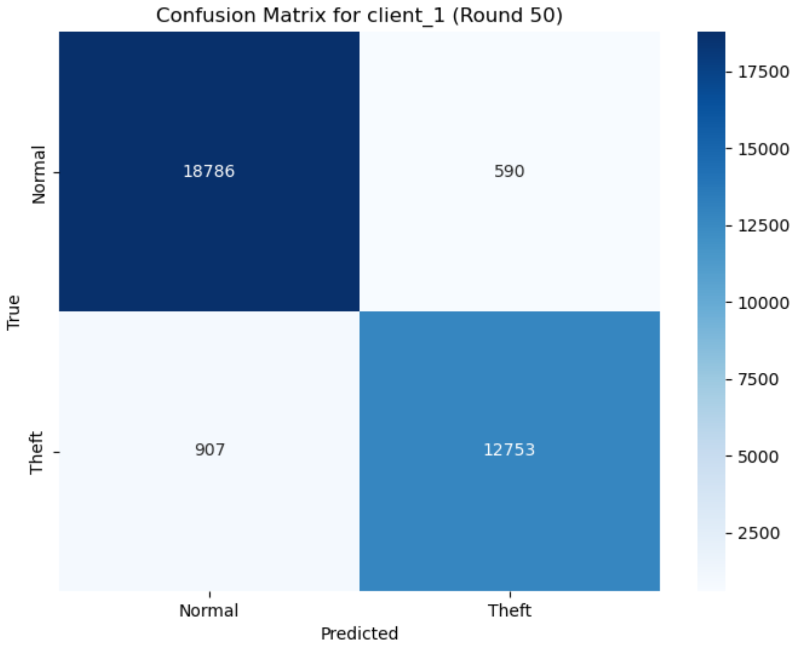


Figure 4.14: Confusion Matrix for Client 1

- **Accuracy: 0.96**
  The model achieves a high overall classification accuracy, correctly predicting 96% of the instances for Client 1, indicating strong performance on local data.

- **Precision (weighted): 0.95**
  The model accurately identifies theft cases with a weighted precision of 95%, meaning that most predicted thefts correspond to actual fraudulent instances, while accounting for class imbalance.

- **Recall (weighted): 0.95**
  With a weighted recall of 95%, the model is able to detect the vast majority of true theft cases, minimizing the number of undetected frauds even across uneven class distributions.

- **F1-Score (weighted): 0.95**
  This score reflects a well-balanced trade-off between precision and recall, confirming consistent and robust classification performance across both normal and fraudulent cases.

- **AUC-ROC: 0.99**
  The model demonstrates excellent discriminative power, effectively distinguishing between normal and fraudulent consumption patterns across various classification thresholds.

- **Log Loss: 0.12**
  The low log loss indicates high confidence and well-calibrated probability estimates, suggesting that the models predictions are both accurate and reliable for Client 1.

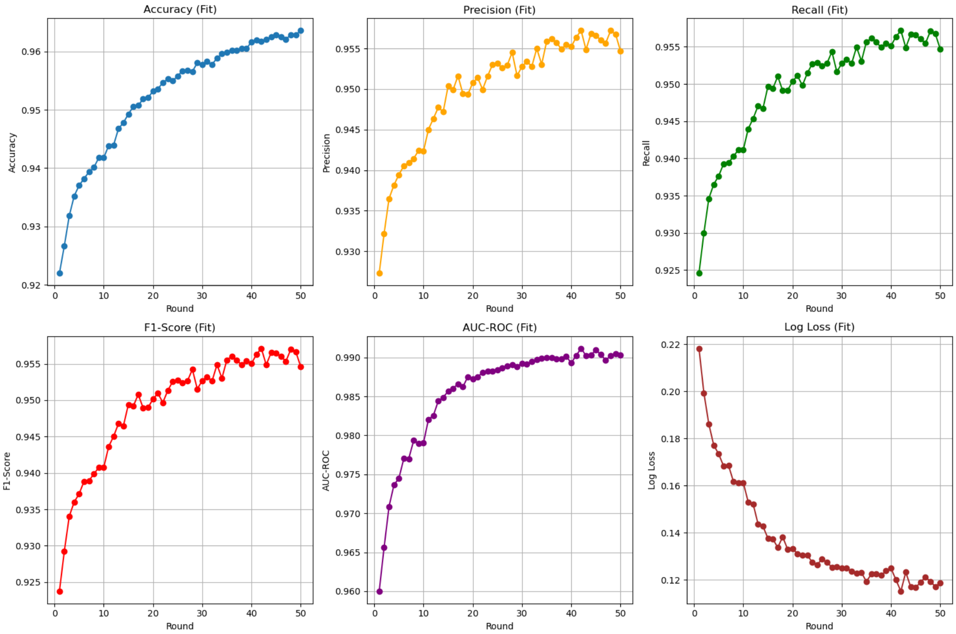


Figure 4.15: Performance metrics visualization for Client 1

Client 2:

- **Confusion Matrix**

  - **True Negatives (TN = 19,156)**: The model accurately identified the vast majority of legitimate energy consumption cases, demonstrating strong capability in reducing false alarms.
  - **True Positives (TP = 12,364)**: A substantial number of theft cases were correctly detected, showing the models effectiveness in identifying fraudulent activity.
  - **False Positives (FP = 590)**: A moderate number of normal instances were incorrectly classified as theft, indicating room for improving precision to avoid unnecessary follow-up actions.
  - **False Negatives (FN = 926)**: These represent missed theft cases; while relatively low, further enhancement of the model's sensitivity could help minimize such errors.

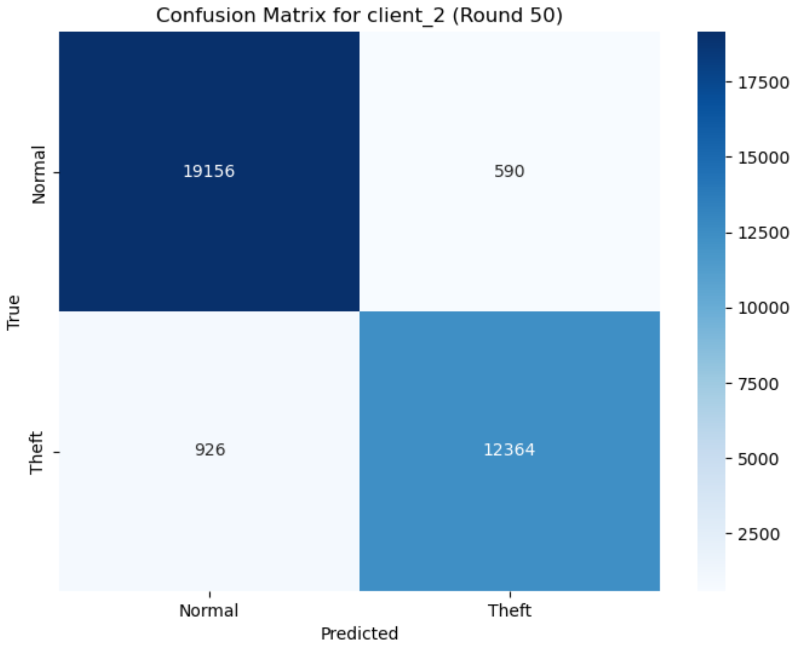


Figure 4.16: Confusion Matrix for Client 2

- **Accuracy: 0.96**
  The model achieves a high overall classification accuracy, correctly predicting 96% of the instances for Client 2, indicating strong and consistent local performance.

- **Precision (weighted): 0.95**
  The model accurately identifies theft cases with a weighted precision of 95%, indicating that most of the theft predictions correspond to actual fraudulent activities.

- **Recall (weighted): 0.95**
  With a weighted recall of 95%, the model effectively detects the vast majority of theft cases, limiting the risk of undetected fraudulent behavior.

- **F1-Score (weighted): 0.95**
  This metric confirms a well-balanced trade-off between precision and recall, reflecting the models stability and reliability in handling imbalanced class distributions.

- **AUC-ROC: 0.99**
  The model exhibits outstanding discriminative capability, with near-perfect separation between normal and fraudulent instances across decision thresholds.

- **Log Loss: 0.12**
  The low log loss value reflects high confidence and good calibration of the predicted probabilities, indicating reliable and sharp prediction quality on Client 2s data.

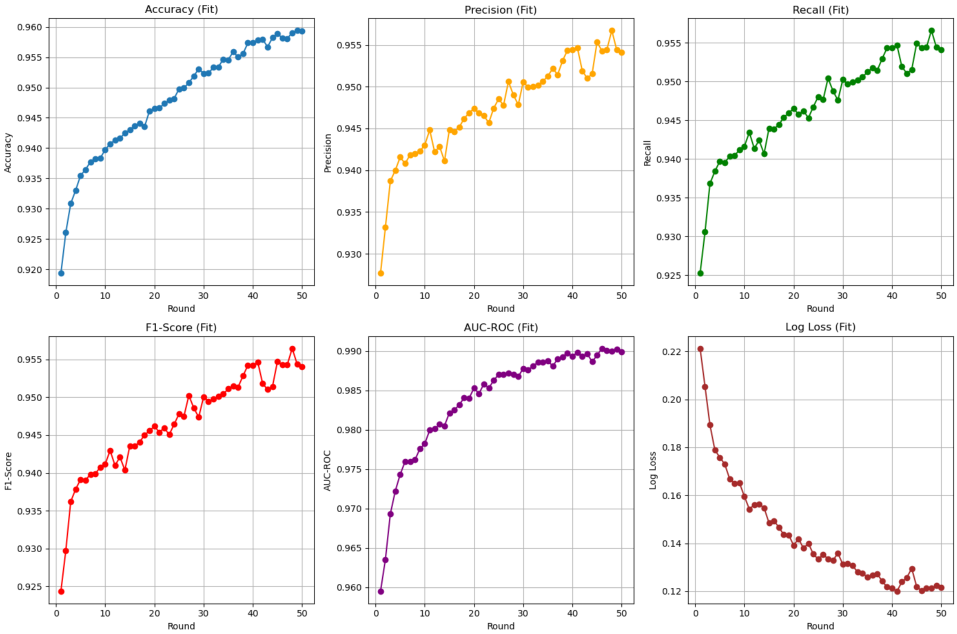


Figure 4.17: Performance metrics visualization for Client 2

Client 3:

- **Confusion Matrix**

  - **True Negatives (TN = 22,865)**: The model successfully identified a large majority of legitimate energy consumption cases, confirming its effectiveness in minimizing false alarms.
  - **True Positives (TP = 15,173)**: A significant number of theft cases were correctly detected, reflecting the models strong ability to recognize fraudulent patterns.
  - **False Positives (FP = 576)**: A moderate number of normal instances were misclassified as theft, suggesting the need for slight improvements in precision to reduce unnecessary interventions.
  - **False Negatives (FN = 1,030)**: These missed theft cases highlight areas where the model could improve its sensitivity to further reduce undetected fraud.

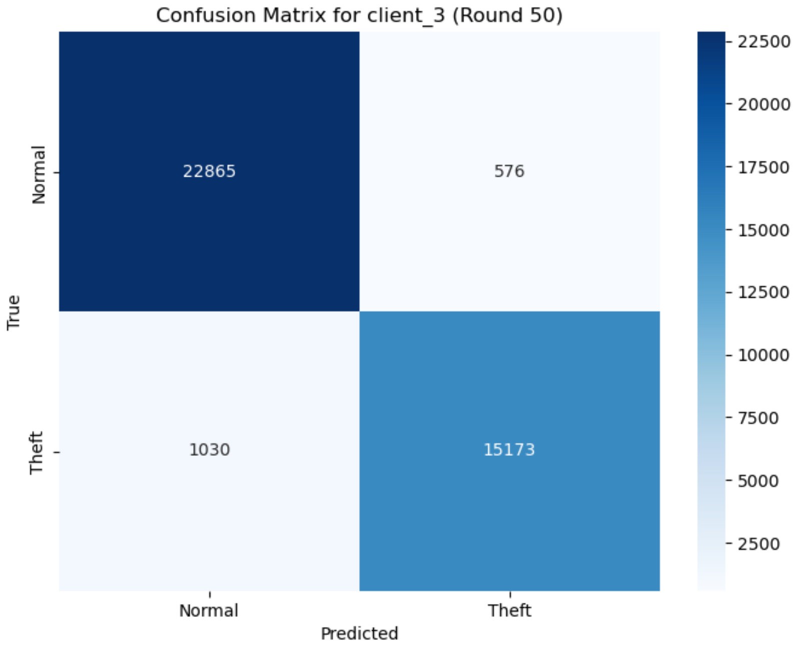


Figure 4.18: Confusion Matrix for Client 3

- **Accuracy: 0.96**
  The model achieves a high overall classification accuracy, correctly predicting 96% of the instances for Client 3, indicating strong and consistent local performance.

- **Precision (weighted): 0.96**
  The model accurately identifies theft cases with a weighted precision of 96%, meaning that the majority of its theft predictions correspond to actual fraudulent activities.

- **Recall (weighted): 0.96**
  With a weighted recall of 96%, the model effectively detects most of the actual theft cases, minimizing the risk of missed fraudulent behavior.

- **F1-Score (weighted): 0.96**
  This value confirms a well-balanced trade-off between precision and recall, demonstrating the models robustness in managing class imbalance in Client 3s data.

- **AUC-ROC: 0.99**
  The model exhibits excellent discriminative capability, nearly perfectly distinguishing between normal and fraudulent consumption instances.

- **Log Loss: 0.11**
  The very low log loss reflects both strong confidence and good calibration of predicted probabilities, indicating highly reliable performance on Client 3s data.

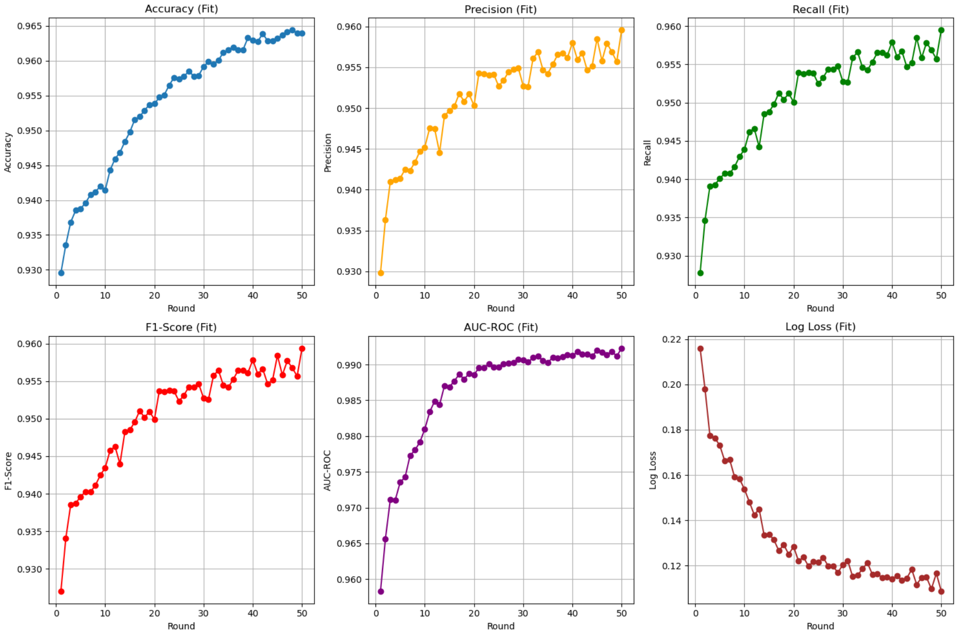


Figure 4.19: Performance metrics visualization for Client 3

Global Model:

- **Accuracy: 0.95**
  The global model correctly classifies 95.1% of the instances, demonstrating a high level of reliability in distinguishing between normal and fraudulent cases across all clients.

- **Precision: 0.95**
  With a precision of 95.1%, the model is highly effective in minimizing false theft alarms, ensuring that most flagged cases genuinely represent fraudulent activity.

- **Recall: 0.95**
  The model successfully detects 95.1% of actual theft cases, highlighting its strong ability to identify fraudulent consumption across distributed datasets.

- **F1-Score: 0.95**
  This balanced metric confirms an excellent trade-off between precision and recall, reflecting robust and consistent classification performance on global data.

- **AUC-ROC: 0.99**
  The model exhibits near-perfect discriminative ability across various thresholds, effectively separating normal from fraudulent behavior on a global scale.

- **Log Loss: 0.13**
  The low log loss (0.1259) indicates that the model is not only accurate but also confident and well-calibrated in its probability estimates, despite variations in client data distributions.

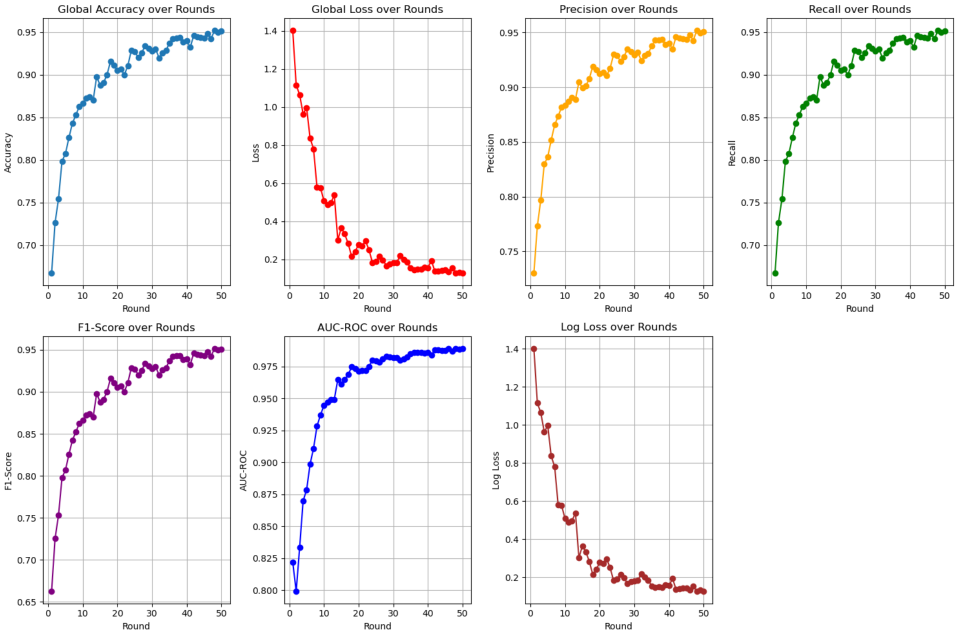


Figure 4.20: Performance metrics visualization for Global Model

Performance of the FedProx Aggregation Method

Client 1:

- **Confusion Matrix**
  - **True Negatives (TN = 18,840)**: The model accurately identified a large number of legitimate energy consumption cases, confirming its reliability in minimizing false alarms.
  - **True Positives (TP = 12,709)**: A substantial number of theft cases were correctly detected, demonstrating the model's continued effectiveness in recognizing fraudulent behavior.
  - **False Positives (FP = 536)**: A moderate number of normal instances were incorrectly flagged as theft, suggesting a need for fine-tuning to improve precision.
  - **False Negatives (FN = 951)**: These missed theft cases reflect potential for improving recall, though the overall number remains within a reasonable range.

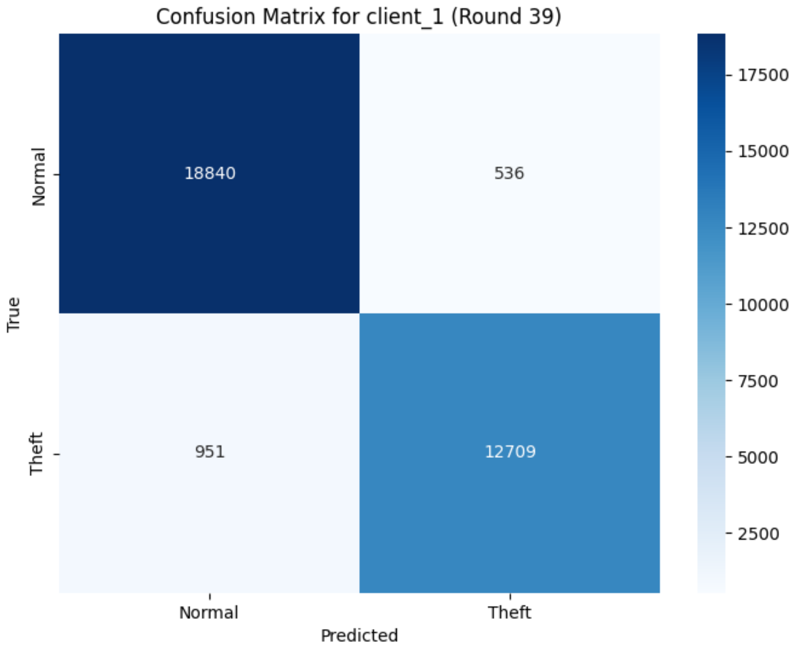


Figure 4.21: Confusion Matrix for Client 1

- **Accuracy: 0.96**
  The model achieves a high overall classification accuracy, correctly predicting 96% of the instances for Client 1, indicating strong performance on local data.

- **Precision (weighted): 0.95**
  The model accurately identifies theft cases with a weighted precision of 95%, meaning that most predicted thefts correspond to actual fraudulent instances, while accounting for class imbalance.

- **Recall (weighted): 0.95**
  With a weighted recall of 95%, the model can detect the vast majority of true theft cases, minimizing the number of undetected frauds even across uneven class distributions.

- **F1-Score (weighted): 0.95**
  This score reflects a well-balanced trade-off between precision and recall, confirming consistent and robust classification performance across both normal and fraudulent cases.

- **AUC-ROC: 0.99**
  The model demonstrates excellent discriminative power, effectively distinguishing between normal and fraudulent consumption patterns across various classification thresholds.

- **Log Loss: 0.13**
  The low log loss indicates high confidence and well-calibrated probability estimates, suggesting that the models predictions are both accurate and reliable for Client 1.

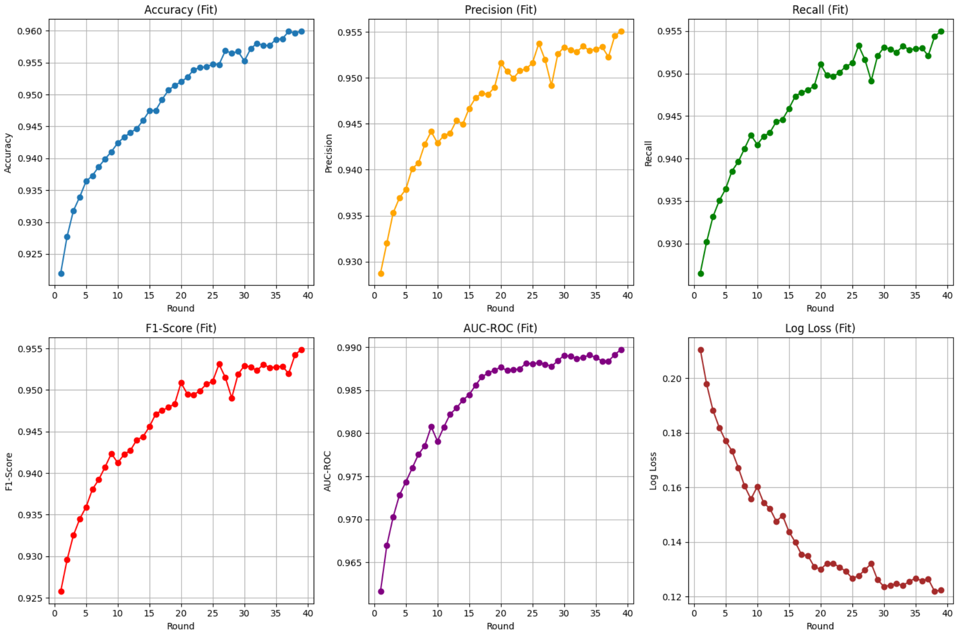


Figure 4.22: Performance metrics visualization for Client 1

Client 2:

- **Confusion Matrix**
  - **True Negatives (TN = 19,057)**: The model correctly identified a significant number of legitimate energy consumption cases, highlighting its effectiveness in reducing false positive alarms.
  - **True Positives (TP = 12,334)**: A substantial portion of actual theft cases were accurately detected, confirming the models competence in identifying fraudulent activities.
  - **False Positives (FP = 689)**: A moderate number of normal cases were misclassified as theft, indicating potential to refine the model for better precision and to reduce unnecessary interventions.
  - **False Negatives (FN = 956)**: These missed theft cases point to opportunities for improving recall, though the figure remains acceptably low in the context of overall model performance.

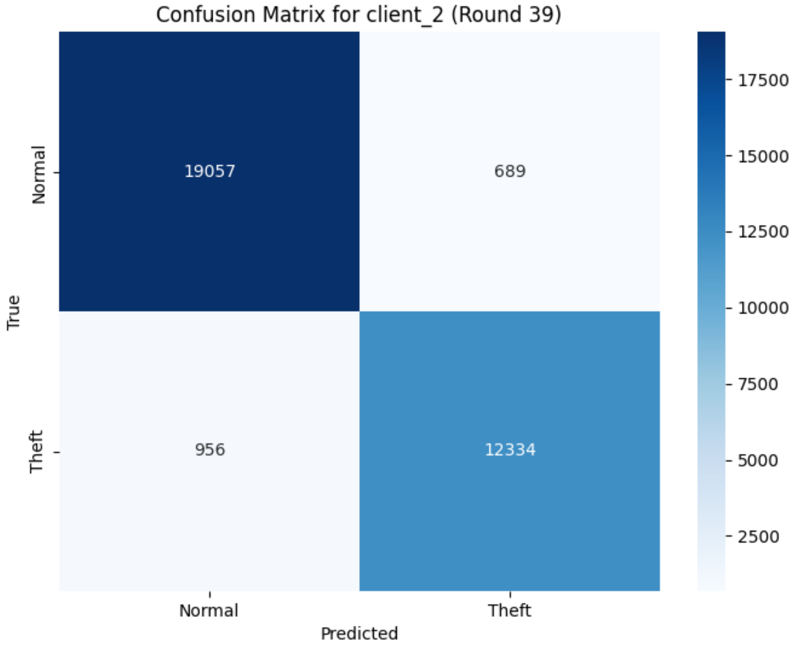


Figure 4.23: Confusion Matrix for Client 2

- **Accuracy: 0.96**
  The model achieves a high overall classification accuracy, correctly predicting 96% of the instances for Client 2, indicating strong and consistent local performance.

- **Precision (weighted): 0.95**
  The model accurately identifies theft cases with a weighted precision of 95%, meaning that the majority of theft alerts correspond to actual fraudulent activity, while minimizing false positives.

- **Recall (weighted): 0.95**
  With a recall of 95%, the model successfully captures most of the actual theft cases, reducing the likelihood of missed fraudulent behavior.

- **F1-Score (weighted): 0.95**
  This value represents a balanced compromise between precision and recall, confirming the models robustness in managing both detection accuracy and error rates.

- **AUC-ROC: 0.99**
  The model demonstrates exceptional discriminative power, with the ability to differentiate effectively between normal and fraudulent instances across a range of thresholds.

- **Log Loss: 0.12**
  The low log loss suggests that the models predicted probabilities are well-calibrated and confident, reinforcing its reliability in real-world deployment for Client 2.

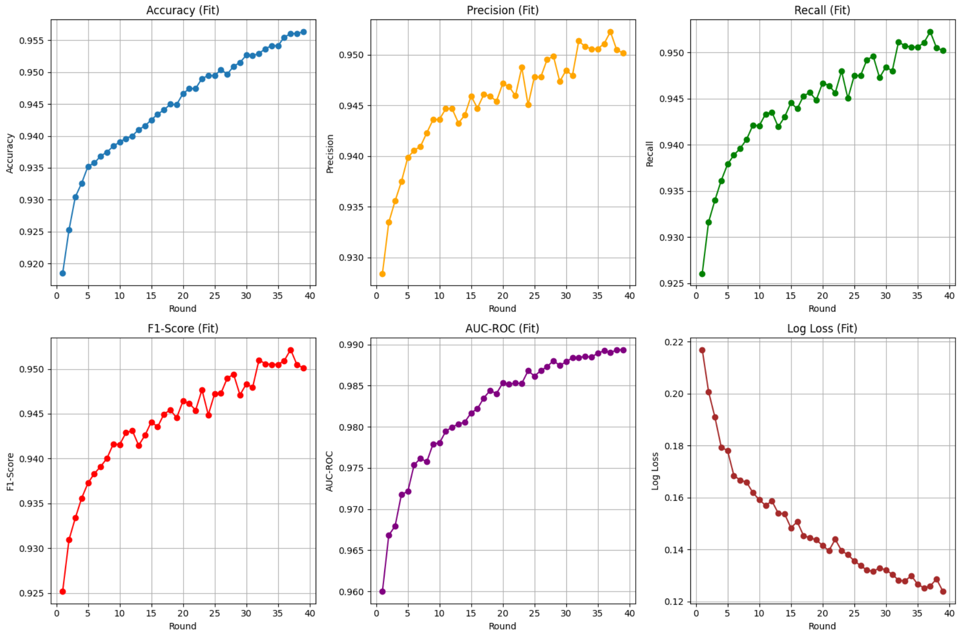


Figure 4.24: Performance metrics visualization for Client 2

Client 3:

- **Confusion Matrix**

  - **True Negatives (TN = 22,753)**: The model successfully identified a large majority of legitimate energy consumption cases, confirming its effectiveness in minimizing false alarms.
  - **True Positives (TP = 15,144)**: A significant number of theft cases were correctly detected, reflecting the models strong ability to recognize fraudulent patterns.
  - **False Positives (FP = 688)**: A moderate number of normal instances were misclassified as theft, suggesting the need for slight improvements in precision to reduce unnecessary interventions.
  - **False Negatives (FN = 1,059)**: These missed theft cases highlight areas where the model could improve its sensitivity to further reduce undetected fraud.

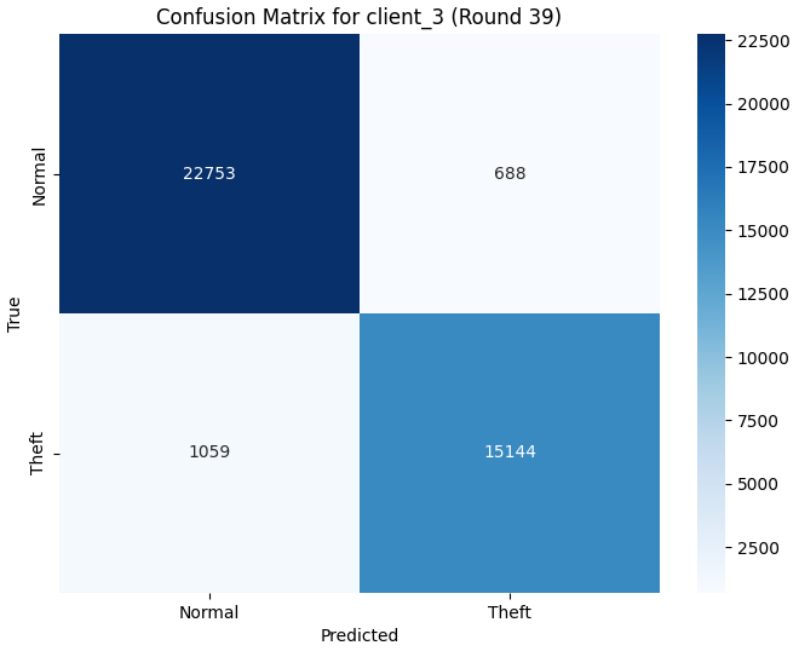


Figure 4.25: Confusion Matrix for Client 3

- **Accuracy: 0.96**
  The model achieves a high overall classification accuracy, correctly predicting 96% of the instances for Client 3, indicating strong and consistent local performance.

- **Precision: 0.96**
  The model accurately identifies theft cases with a weighted precision of 96%, meaning that the majority of its theft predictions correspond to actual fraudulent activities.

- **Recall: 0.96**
  With a weighted recall of 96%, the model effectively detects most of the actual theft cases, minimizing the risk of missed fraudulent behavior.

- **F1-Score: 0.96**
  This value confirms a well-balanced trade-off between precision and recall, demonstrating the models robustness in managing class imbalance in Client 3s data.

- **AUC-ROC: 0.99**
  The model exhibits excellent discriminative capability, nearly perfectly distinguishing between normal and fraudulent consumption instances.

- **Log Loss: 0.11**
  The very low log loss reflects both strong confidence and good calibration of predicted probabilities, indicating highly reliable performance on Client 3s data.

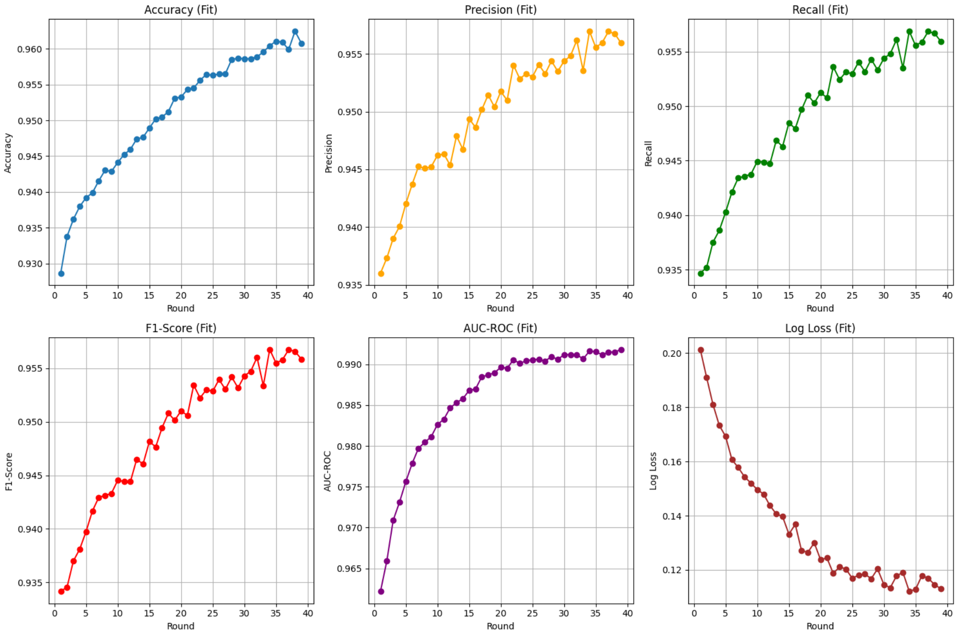


Figure 4.26: Performance metrics visualization for Client 3

Global Model:

- **Accuracy: 0.9451**
  The global model correctly classifies 94.51% of the instances, demonstrating a high level of reliability in distinguishing between normal and fraudulent cases across all clients.

- **Precision: 0.9451**
  With a precision of 94.51%, the model is highly effective in minimizing false theft alarms, ensuring that most flagged cases genuinely represent fraudulent activity.

- **Recall: 0.9451**
  The model successfully detects 94.51% of actual theft cases, highlighting its strong ability to identify fraudulent consumption across distributed datasets.

- **F1-Score: 0.9450**
  This balanced metric confirms an excellent trade-off between precision and recall, reflecting robust and consistent classification performance on global data.

- **AUC-ROC: 0.9863**
  The model exhibits near-perfect discriminative ability across various thresholds, effectively separating normal from fraudulent behavior on a global scale.

- **Log Loss: 0.1452**
  The log loss of 0.1452 indicates that the model is not only accurate but also confident and well-calibrated in its probability estimates, despite variations in client data distributions.

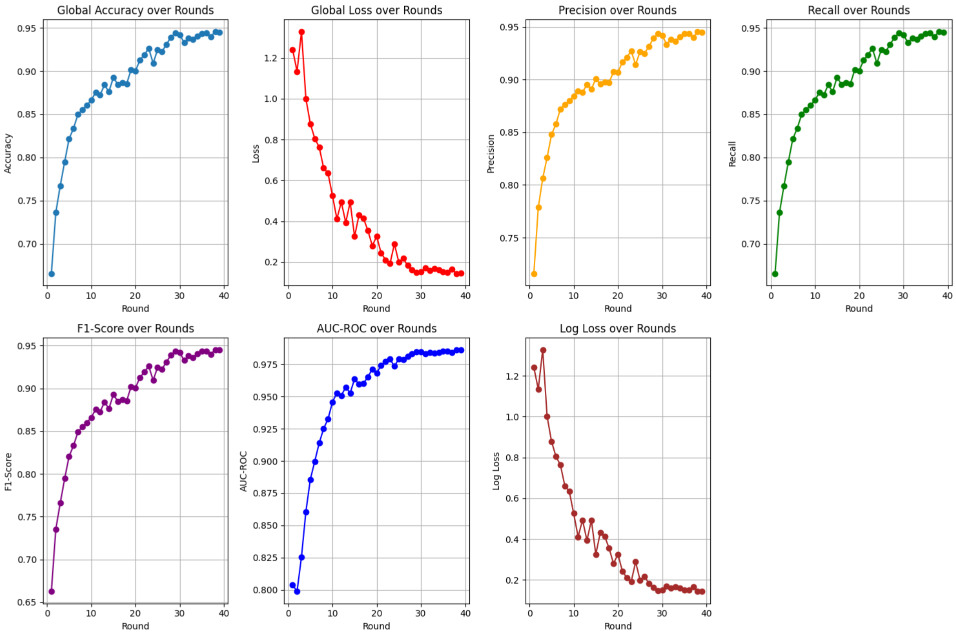


Figure 4.27: Performance metrics visualization for Global Model

Performance of the FedAdam Aggregation Method

Client 1:

- **Confusion Matrix**
  - **True Negatives (TN = 18,976)**: The model correctly identified the majority of legitimate energy consumption cases, reflecting good control over false positive rates.
  - **True Positives (TP = 12,165)**: A significant number of theft cases were accurately detected, confirming the models capability in spotting fraudulent activity.
  - **False Positives (FP = 400)**: A relatively small number of normal instances were incorrectly flagged as theft, which may cause minor inconvenience but remains within acceptable limits.
  - **False Negatives (FN = 1,495)**: These missed theft cases suggest there is room for improvement in sensitivity to ensure fewer fraudulent behaviors go undetected.

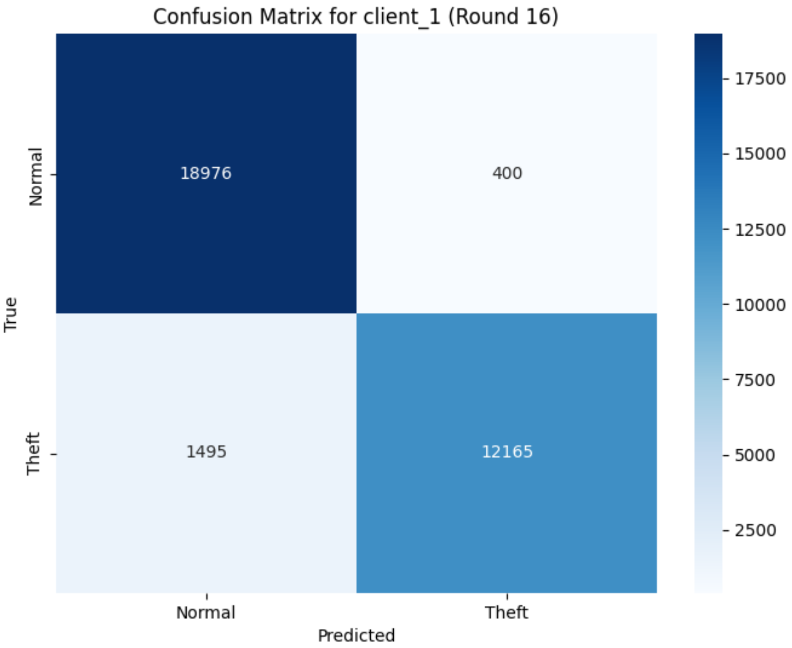


Figure 4.28: Confusion Matrix for Client 1

- **Accuracy: 0.94**
  The model correctly classifies 94% of the instances, indicating a high level of reliability in distinguishing between normal and fraudulent cases on Client 1's data.

- **Precision: 0.94**
  A weighted precision of 94% means the model effectively minimizes false alarms, ensuring that most theft predictions are truly fraudulent while considering class imbalance.

- **Recall: 0.94**
  With a recall of 94%, the model captures a significant proportion of theft cases, showing strong sensitivity and the ability to detect fraudulent consumption.

- **F1-Score: 0.94**
  This value confirms a solid balance between precision and recall, reflecting stable and consistent performance across both classes.

- **AUC-ROC: 0.98**
  The high AUC-ROC score demonstrates the models excellent ability to distinguish between legitimate and fraudulent energy usage across all decision thresholds.

- **Log Loss: 0.16**
  A log loss of 0.16 indicates that the models predicted probabilities are reasonably confident and well-calibrated, though with slight room for improvement.

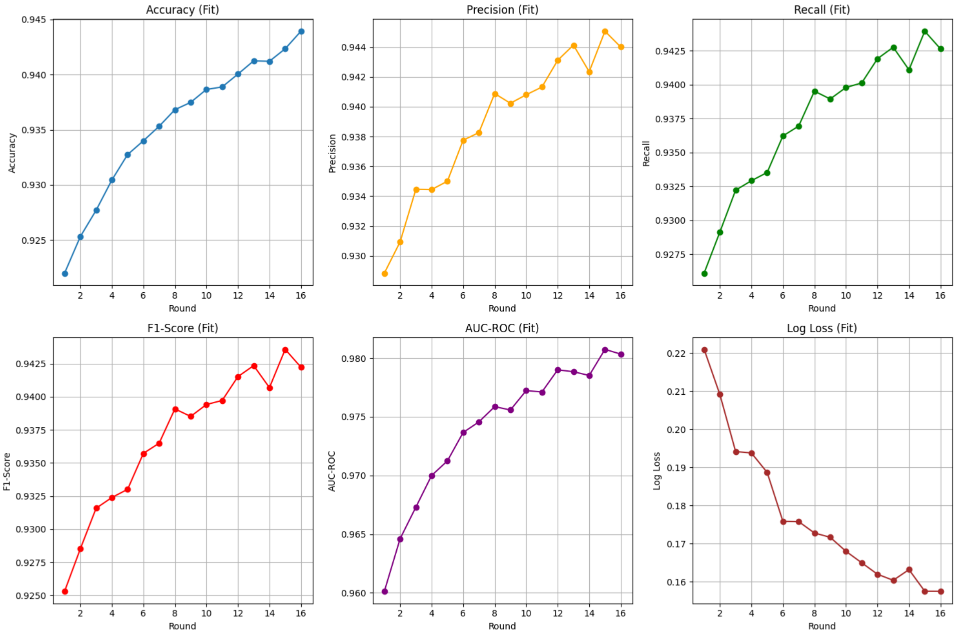


Figure 4.29: Performance metrics visualization for Client 1

Client 2:

- **Confusion Matrix**
  - **True Negatives (TN = 19,366):** The model correctly identified the vast majority of normal usage instances, demonstrating strong reliability in avoiding false alarms.
  - **True Positives (TP = 11,757):** A substantial number of actual theft cases were accurately detected, highlighting the models effective sensitivity.
  - **False Positives (FP = 380):** A relatively small number of legitimate consumption instances were misclassified as theft, suggesting minor yet manageable misclassification.
  - **False Negatives (FN = 1,533):** These missed theft cases are critical from a fraud detection standpoint and indicate potential areas for improving the models recall.

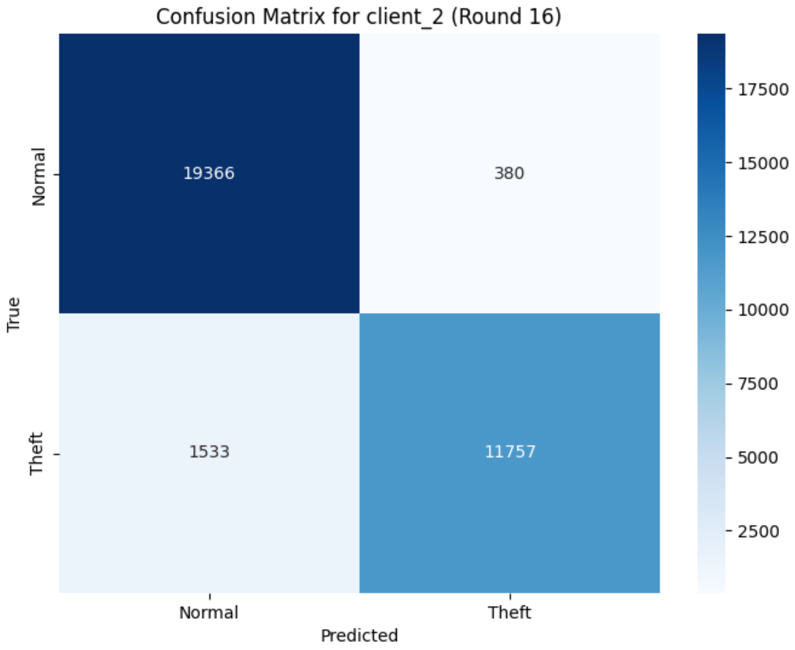


Figure 4.30: Confusion Matrix for Client 2

- **Accuracy: 0.94**
  The model achieves a high correct classification rate of 94%, indicating strong overall performance on Client 2s data.

- **Precision (weighted): 0.94**
  The model maintains a high weighted precision of 94%, ensuring that most theft predictions correspond to actual fraudulent cases while accounting for class imbalance.

- **Recall (weighted): 0.94**
  A weighted recall of 94% indicates the model's effectiveness in identifying the majority of true theft cases, minimizing missed fraud instances.

- **F1-Score (weighted): 0.94**
  This balanced metric confirms solid performance in terms of both detecting fraud and limiting false positives, even with skewed data distributions.

- **AUC-ROC: 0.98**
  The model performs very well in distinguishing between legitimate and fraudulent behavior, with excellent separability across decision thresholds.

- **Log Loss: 0.16**
  The log loss value reflects good confidence in predictions, with well-calibrated probability estimates across all instances.

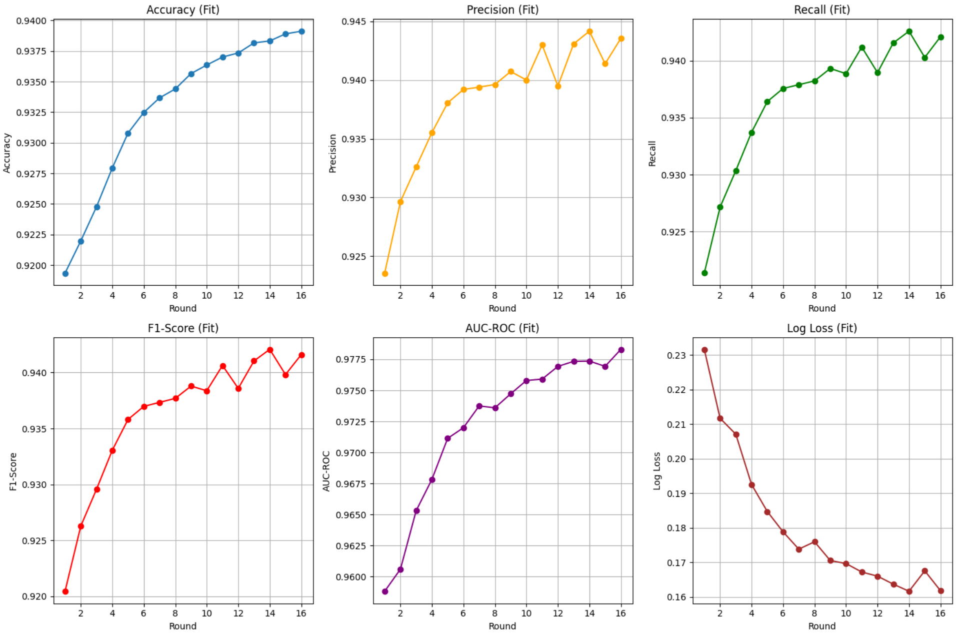


Figure 4.31: Performance metrics visualization for Client 2

Client 3:

- **Confusion Matrix**

  - **True Negatives (TN = 22,997):** The model effectively recognized the vast majority of legitimate energy consumption instances, maintaining a low false alarm rate.
  - **True Positives (TP = 14,429):** A substantial number of theft cases were correctly identified, reflecting the models strength in detecting fraudulent behavior.
  - **False Positives (FP = 444):** A small number of normal consumption cases were mistakenly flagged as theft, which may lead to minor operational inefficiencies but remains within an acceptable range.
  - **False Negatives (FN = 1,774):** These undetected fraud cases highlight opportunities for improving the models sensitivity and reducing potential losses due to missed detections.

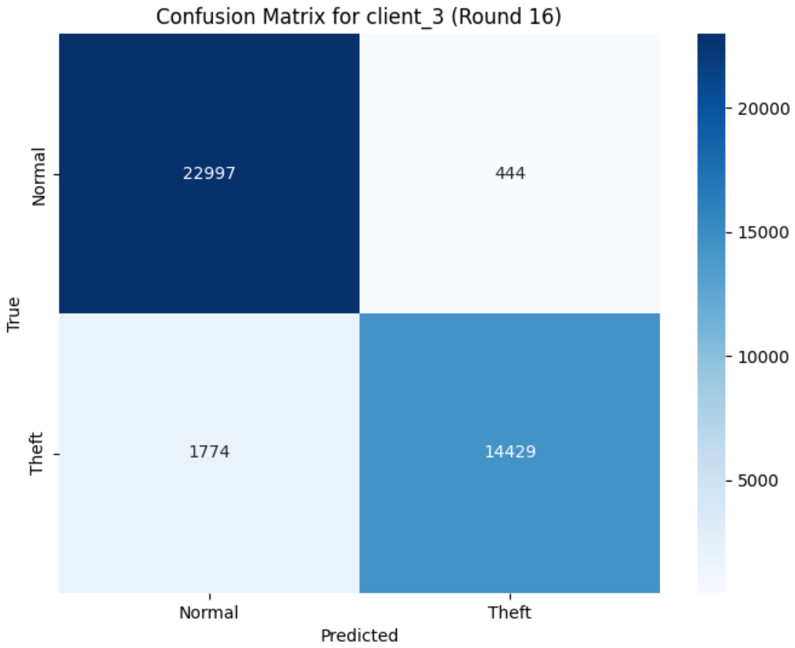


Figure 4.32: Confusion Matrix for Client 3

- **Accuracy: 0.94**
  The model demonstrates strong accuracy, correctly classifying 94% of instances, which indicates dependable performance across diverse consumption behaviors.

- **Precision (weighted): 0.95**
  With a weighted precision of 95%, the model effectively limits false positives, ensuring that most theft alerts are indeed true fraudulent cases.

- **Recall (weighted): 0.94**
  A high recall of 94% confirms the models solid ability to identify the majority of actual theft cases, minimizing undetected fraud.

- **F1-Score (weighted): 0.94**
  This balanced score reflects the models consistent performance in handling both precision and recall trade-offs under class imbalance.

- **AUC-ROC: 0.98**
  The model exhibits excellent discriminative ability, accurately distinguishing between normal and fraudulent behavior over a wide range of thresholds.

- **Log Loss: 0.15**
  A log loss of 0.15 suggests well-calibrated probability predictions, with low uncertainty and high confidence in the models classification outputs.

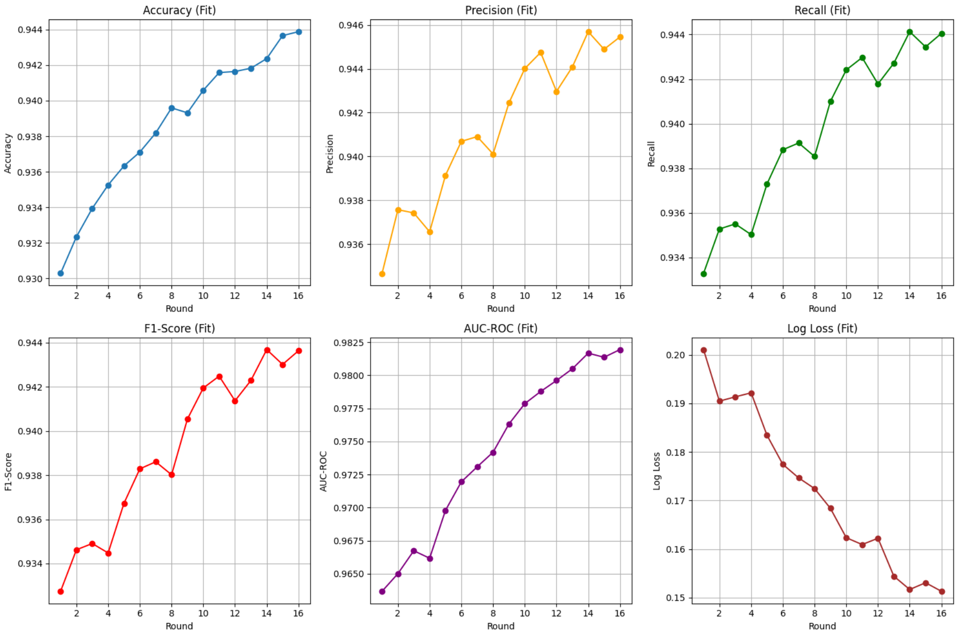


Figure 4.33: Performance metrics visualization for Client 3

Global Model:

- **Global Accuracy: 0.93**
  The global model correctly classifies 93.26% of instances across all clients, demonstrating strong generalization capability despite data heterogeneity.

- **Precision: 0.93**
  With a precision of 93.39%, the model reliably identifies theft cases while minimizing false positives on a global scale.

- **Recall: 0.93**
  The model detects 93.26% of actual thefts across the federated dataset, confirming its effectiveness in capturing fraudulent activities.

- **F1-Score: 0.93**
  An F1-score of 0.9321 indicates a balanced and robust performance, with consistent trade-offs between precision and recall across diverse clients.

- **AUC-ROC: 0.97**
  The AUC-ROC of 0.9717 highlights excellent discriminative power, reflecting the models ability to separate fraudulent from legitimate behavior over various thresholds.

- **Log Loss: 0.19**
  A log loss of 0.1899 reflects well-calibrated probability predictions, with reasonably high confidence and reliability across client distributions.

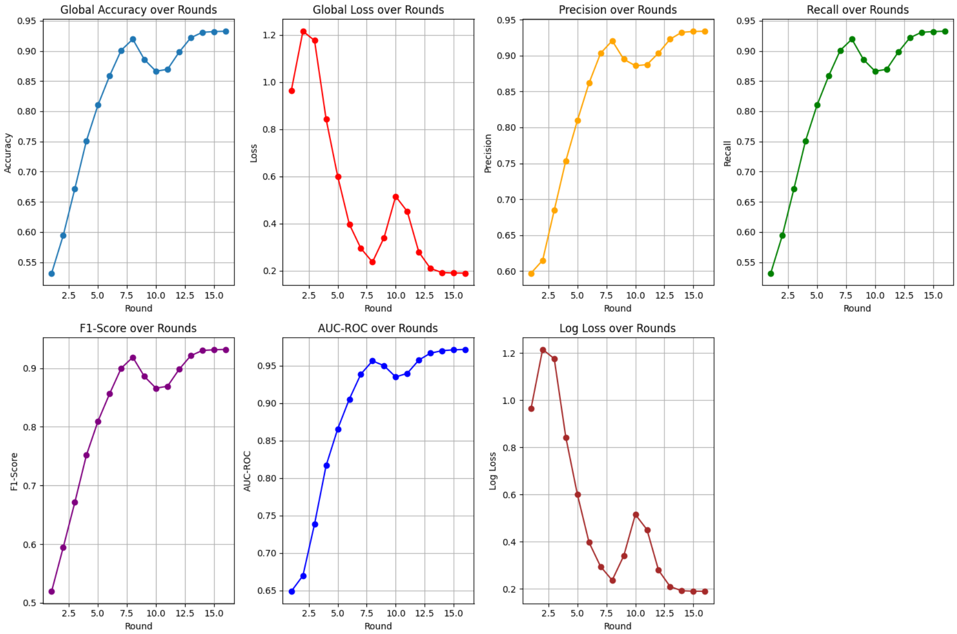


Figure 4.34: Performance metrics visualization for the Global Model

Performance of the FedTrimmedAvg Aggregation Method

Client 1:

- **Confusion Matrix**
  - **True Negatives (TN = 18,729)**: The model successfully identified a large number of legitimate energy consumption instances, indicating solid control over false positive rates.
  - **True Positives (TP = 12,772)**: A high number of actual theft cases were correctly detected, demonstrating the models effectiveness in fraud recognition.
  - **False Positives (FP = 647)**: Some normal consumption cases were incorrectly flagged as fraudulent, which could result in operational noise but is relatively contained.
  - **False Negatives (FN = 888)**: These missed thefts highlight opportunities to further enhance the models sensitivity and reduce undetected fraudulent behavior.

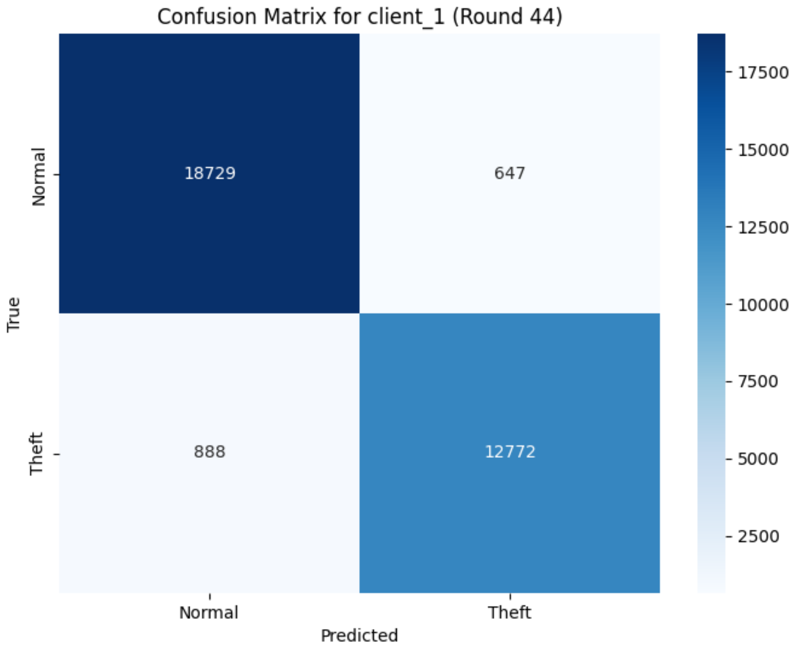


Figure 4.35: Confusion Matrix for Client 1

- **Accuracy: 0.96**
  The model correctly classifies 96% of the instances, indicating a very high level of reliability in distinguishing between normal and fraudulent cases on Client 1's data.

- **Precision: 0.96**
  A weighted precision of 96% demonstrates that the model generates very few false alarms, ensuring that most theft predictions correspond to actual fraudulent cases.

- **Recall: 0.96**
  With a recall of 96%, the model effectively detects the majority of theft cases, showcasing strong sensitivity to fraudulent behavior.

- **F1-Score: 0.96**
  This score indicates a well-balanced performance between precision and recall, reflecting the models robustness even in the presence of class imbalance.

- **AUC-ROC: 0.99**
  The model exhibits exceptional discriminative capability, nearly perfectly separating fraudulent from legitimate consumption instances across all thresholds.

- **Log Loss: 0.12**
  The low log loss reflects highly confident and well-calibrated probability estimates, reinforcing the models dependable predictive behavior.

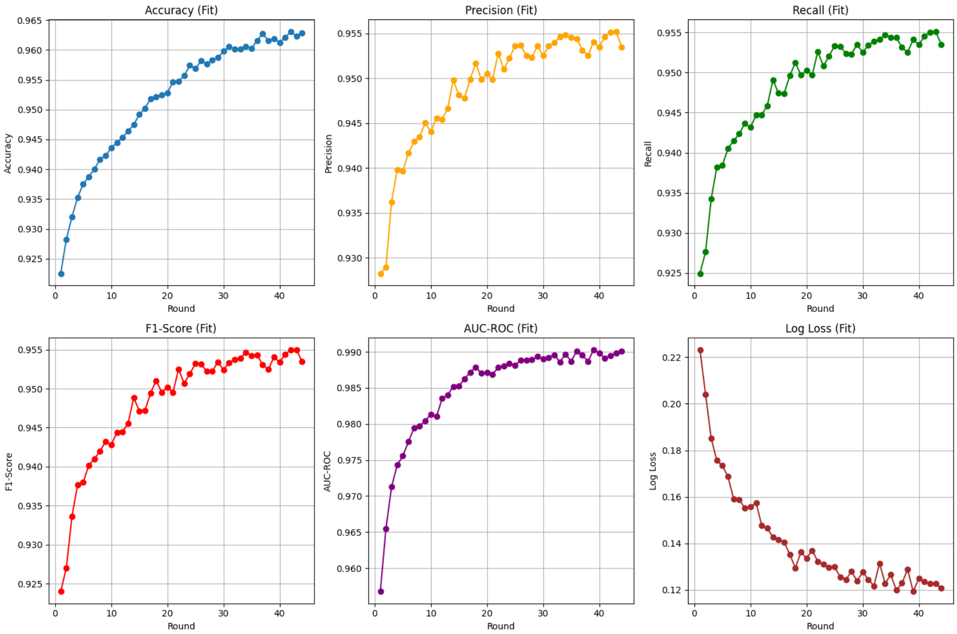


Figure 4.36: Performance metrics visualization for Client 1

Client 2:

- **Confusion Matrix**

  - **True Negatives (TN = 19,272):** The model accurately recognized the vast majority of legitimate consumption instances, confirming its strength in minimizing false alarms.
  - **True Positives (TP = 12,268):** A significant portion of actual theft cases were correctly identified, reflecting strong sensitivity to fraudulent behavior.
  - **False Positives (FP = 474):** A modest number of normal usage instances were incorrectly flagged as theft, which could lead to unnecessary inspections but remains within acceptable margins.
  - **False Negatives (FN = 1,022):** These represent undetected fraud cases, which, while relatively limited, indicate the need for continued improvement in recall.

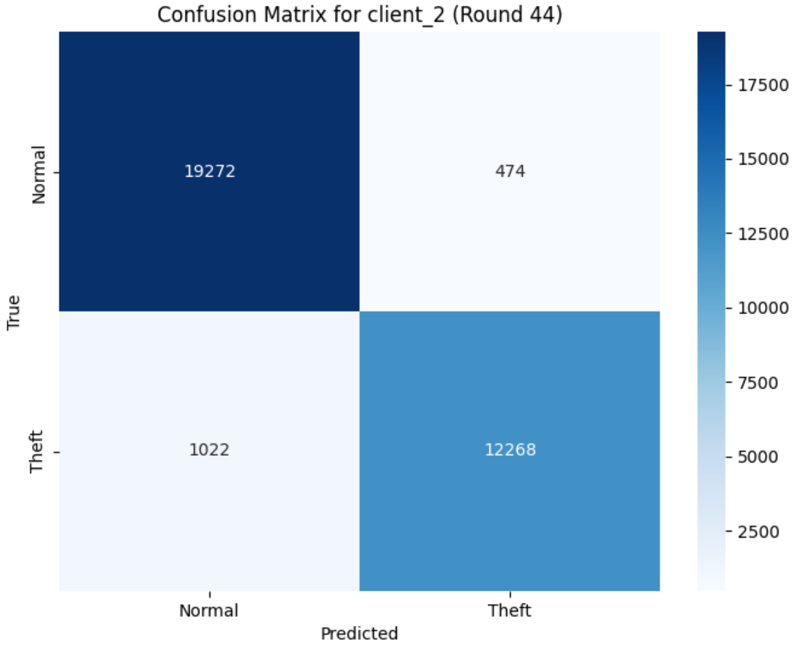


Figure 4.37: Confusion Matrix for Client 2

- **Accuracy: 0.96**
  The model achieves a strong correct classification rate of 96%, demonstrating excellent overall performance on Client 2s dataset.

- **Precision : 0.95**
  A high weighted precision of 95% shows that the model effectively minimizes false alarms, with most theft predictions being accurate even under class imbalance.

- **Recall: 0.95**
  The model correctly identifies 95% of actual fraud cases, highlighting its strong sensitivity and reduced likelihood of missing thefts.

- **F1-Score: 0.95**
  This balanced score indicates robust performance in both detecting fraudulent behavior and limiting false positives, ensuring reliable classification.

- **AUC-ROC: 0.99**
  The near-perfect AUC-ROC confirms the model's exceptional ability to separate fraudulent from legitimate usage across different threshold values.

- **Log Loss: 0.12**
  The low log loss value demonstrates high confidence and calibration in the models probability predictions, even across diverse consumption patterns.

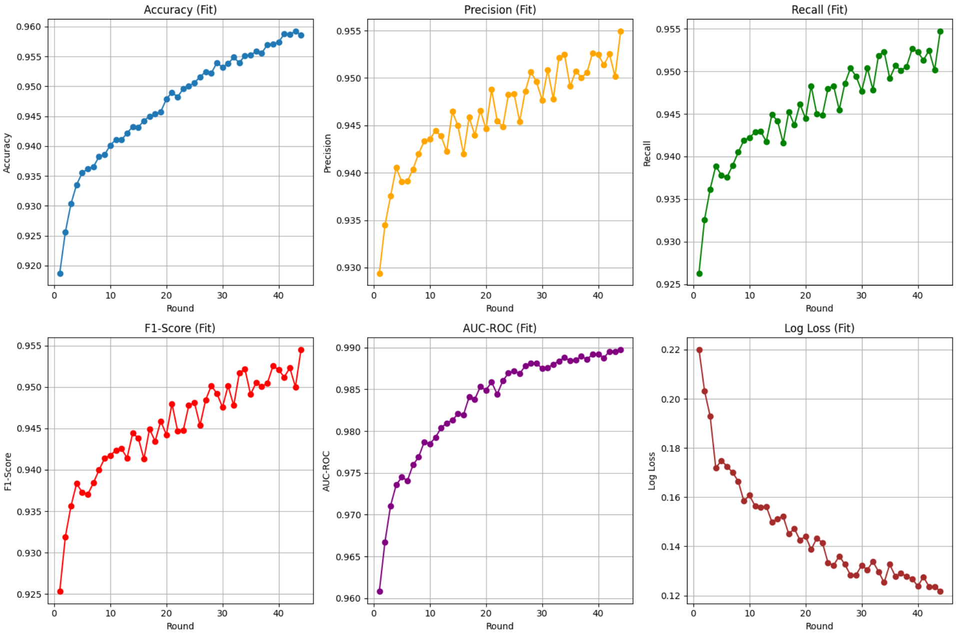


Figure 4.38: Performance metrics visualization for Client 2

Client 3:

- **Confusion Matrix**
  - **True Negatives (TN = 22,928):** The model successfully identified the majority of legitimate consumption instances, maintaining a low rate of false alarms.
  - **True Positives (TP = 15,071):** A large number of theft cases were accurately detected, confirming the models effectiveness in recognizing fraudulent behavior.
  - **False Positives (FP = 513):** A moderate number of legitimate cases were incorrectly flagged as theft, which could result in minor operational disruptions but remains within a tolerable margin.
  - **False Negatives (FN = 1,132):** These missed fraud instances indicate areas for further improvement in recall to enhance overall fraud detection performance.

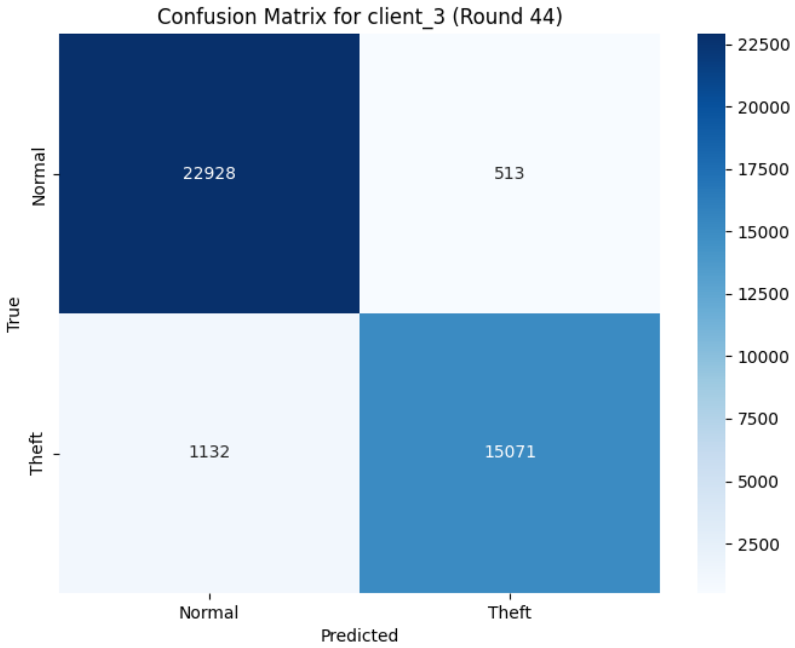


Figure 4.39: Confusion Matrix for Client 3

- **Accuracy: 0.96**
  The model achieves a high accuracy rate of 96%, indicating reliable performance in correctly classifying both normal and fraudulent consumption behaviors.

- **Precision (weighted): 0.96**
  With a weighted precision of 96%, the model effectively minimizes false positives, ensuring that theft alerts are mostly accurate and trustworthy.

- **Recall (weighted): 0.96**
  A recall of 96% confirms the models strong ability to detect actual thefts, reducing the likelihood of missed fraud cases.

- **F1-Score (weighted): 0.96**
  The balanced F1-score reflects the models excellent trade-off between precision and recall, reinforcing its overall robustness under imbalanced class conditions.

- **AUC-ROC: 0.99**
  The very high AUC-ROC score demonstrates the models outstanding capacity to distinguish between legitimate and fraudulent consumption across various thresholds.

- **Log Loss: 0.12**
  A low log loss indicates that the models predicted probabilities are highly confident and well-calibrated, supporting accurate and stable decision-making.

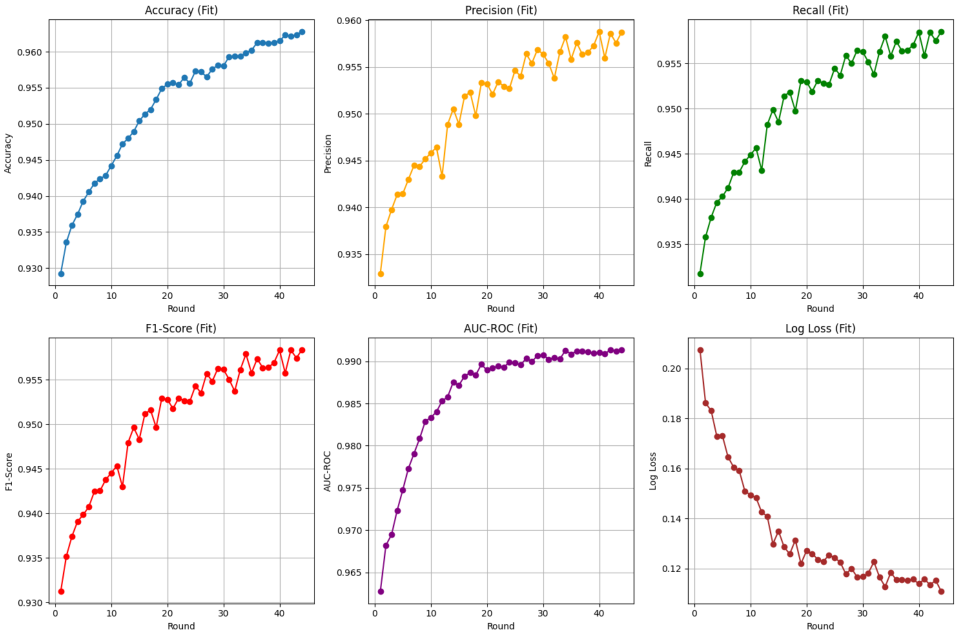


Figure 4.40: Performance metrics visualization for Client 3

Global Model:

- **Global Accuracy: 0.95**
  The global model achieves a strong accuracy of 95.00% across all clients, confirming its ability to generalize effectively despite varying local data distributions.

- **Precision: 0.95**
  With a precision of 95.00%, the model reliably identifies theft cases while maintaining a low rate of false positives on a federated scale.

- **Recall: 0.95**
  The model successfully detects 95.00% of actual fraud cases, demonstrating strong sensitivity across the aggregated dataset.

- **F1-Score: 0.95**
  An F1-score of 0.9498 reflects excellent balance between precision and recall, ensuring stable and reliable detection performance globally.

- **AUC-ROC: 0.99**
  The very high AUC-ROC score of 0.9881 confirms exceptional class separability, indicating the models robustness in distinguishing fraudulent from legitimate behavior under various decision thresholds.

- **Log Loss: 0.13**
  A low log loss of 0.1346 indicates that the models probability outputs are confident and well-calibrated, contributing to informed and reliable classification decisions.

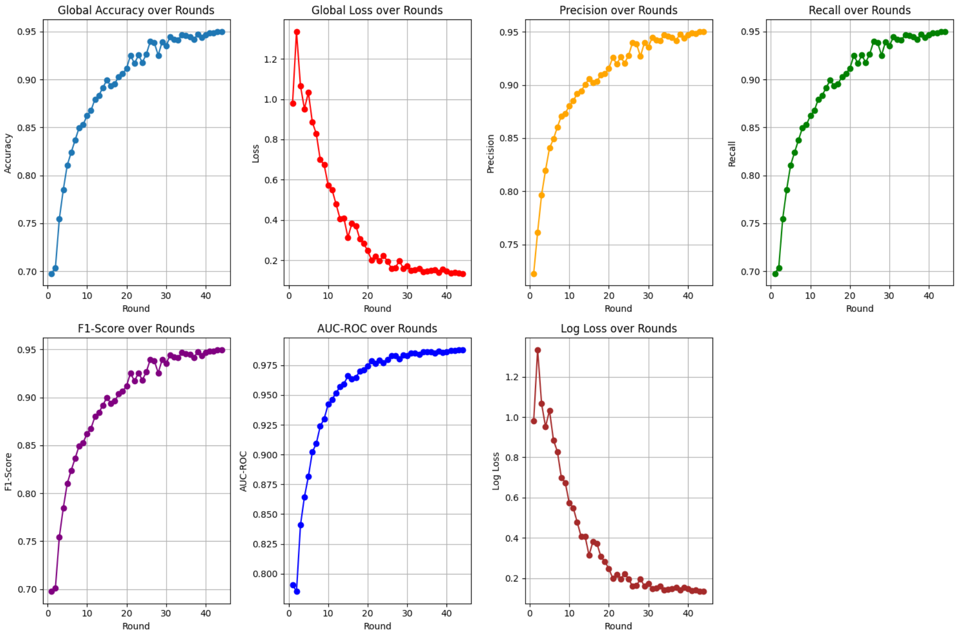


Figure 4.41: Performance metrics visualization for the Global Model

4.7 Conclusion

This chapter presented the implementation of our federated learning framework for electric load forecasting and electricity theft detection. Using the Flower framework, we simulated a decentralized environment while addressing real-world challenges such as data heterogeneity, class imbalance, and client drift. Various aggregation methods were implemented and evaluated. The results demonstrated the effectiveness of federated learning in achieving strong performance while preserving data privacy and enhancing robustness. These findings validate our approach and lay the groundwork for further improvements and real-world deployment.

General Conclusion

The transition to smart grids necessitates a profound transformation of traditional energy management, forecasting, and security methods. In this context, federated learning emerges as an innovative and promising solution to address contemporary challenges posed by data decentralization, privacy preservation, and the increasing complexity of energy systems. This study has successfully explored and demonstrated the feasibility of this approach in critical applications such as load forecasting and the detection of anomalous consumption behaviors. Through a rigorous methodology and the integration of tailored federated learning architectures, we achieved a balance between performance, robustness, and the confidentiality of distributed data. The insights gained from this project underscore the importance of designing intelligent, resilient, and collaborative solutions capable of adapting to the technical and ethical constraints of the modern energy sector. They also pave the way for future research aimed at further optimizing aggregation strategies, enhancing the management of heterogeneous data, and strengthening model resilience in dynamic, distributed environments. Ultimately, this work lays the foundation for a new generation of intelligent energy systems, where cooperation among local entities, guided by principles of data sovereignty and algorithmic security, establishes itself as a cornerstone of sustainable performance and energy resilience. By fostering scalable, privacy-preserving, and adaptive frameworks, federated learning holds the potential to redefine the future of smart grids, ensuring their efficiency and robustness in an increasingly interconnected and data-driven world.

Bibliography

[1] S. K. Salman, "Evolution of conventional power systems to smart grids," in *2019 54th International Universities Power Engineering Conference (UPEC)*, 2019, pp. 1–6.

[2] I. Naidji, M. B. Smida, M. Khalgui, and A. Bachir, "Non cooperative game theoretic approach for residential energy management in smart grid," in *The 32nd Annual European Simulation and Modelling Conference*, Ghent, Belgium, 2018, pp. 164–170.

[3] I. Naidji, O. Mosbahi, M. Khalgui, and A. Bachir, "Two-stage game theoretic approach for energy management in networked microgrids," in *Software Technologies*, M. van Sinderen and L. A. Maciaszek, Eds. Cham: Springer International Publishing, 2020, pp. 205–228.

[4] I. Naidji, M. Ben Smida, M. Khalgui, A. Bachir, Z. Li, and N. Wu, "Efficient allocation strategy of energy storage systems in power grids considering contingencies," *IEEE Access*, vol. 7, pp. 186 378–186 392, 2019.

[5] I. Naidji., O. Mosbahi., M. Khalgui., and A. Bachir., "Cooperative energy management software for networked microgrids," in *Proceedings of the 14th International Conference on Software Technologies - ICSOFT*, INSTICC. SciTePress, 2019, pp. 428–438.

[6] N. Alrikabi, "Renewable energy types," *Journal of Clean Energy Technologies*, vol. 2, no. 1, pp. 61–64, 2014.

[7] M. Guizani and M. Anan, "Smart grid opportunities and challenges of integrating renewable sources: A survey," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2014, pp. 1098–1105.

[8] S. Paul, M. S. Rabbani, R. K. Kundu, and S. M. R. Zaman, "A review of smart technology (smart grid) and its features," in *2014 1st International Conference on Non Conventional Energy (ICONCE 2014)*. IEEE, 2014, pp. 200–203.

[9] A. McGrath and A. Jonker. (2024, Mar.) Quest-ce que la prévision de charge ? IBM. Consulté le 23 avril 2025. [Online]. Available: https://www.ibm.com/fr-fr/topics/load-forecasting

[10] I. Naidji, C. E. Choucha, and M. Ramdani, "Electricity theft detection techniques using artificial intelligence: A survey," in *2024 IEEE International Conference on Advanced Systems and Emergent Technologies (IC*$_A SET), 2024, pp.1 - -6.$

[11] Sustainability Directory. (2024) How can smart grids reduce energy theft? Accessed: 2025-04-27. [Online]. Available: https://energy.sustainability-directory.com/question/how-can-smart-grids-reduce-energy-theft/

[12] R. Czechowski, “The most frequent energy theft techniques and hazards in present power energy consumption,” 04 2016, pp. 1–7.

[13] I. Naidji, M. B. Smida, M. Khalgui, and A. Bachir, “Multi agent system-based approach for enhancing cyber-physical security in smart grids,” in *Proceedings of the the 33rd Annual European Simulation and Modelling Conference*, pp. 177–182.

[14] Montel Energy. (2024) Smart grid challenges: Cybersecurity, renewable energy integration, and financial barriers. Accessed: 2025-04-27. [Online]. Available: https://montel.energy/resources/blog/smart-grid-challenges-cybersecurity-renewable-energy-integration-and-financial-barriers

[15] L. Hu, L. Zhang, T. Wang, and K. Li, “Short-term load forecasting based on support vector regression considering cooling load in summer,” in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 5495–5498.

[16] K. Gu and L. Jia, “Temporal convolutional network based short-term load forecasting model,” in *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, 2020, pp. 584–589.

[17] H. Xu, Y. Zhang, and Y. Zhao, “Short-term electricity load forecasting based on ensemble empirical mode decomposition and long short-term memory neural network,” in *2023 IEEE International Conference on Energy Internet (ICEI)*, 2023, pp. 271–275.

[18] P. K.S., P. Amitasree, G. R. Vamshi, and V. S. K. Devi, “Expression of concern for: Deep learning based load forecasting for futuristic sustainable smart grid,” in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, 2022, pp. 1–1.

[19] K. Li, X. Yang, S. Zhao, Q. Fan, Y. Huang, D. Zhou, and J. Wu, “Load forecasting based on weather forecast correction and cumulative temperature index,” in *2024 IEEE PES 16th Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2024, pp. 1–5.

[20] H. Fang, J.-W. Xiao, and Y.-W. Wang, “A machine learning-based detection framework against intermittent electricity theft attack,” *International Journal of Electrical Power & Energy Systems*, vol. 150, p. 109075, 2023.

[21] N. Ayub, K. Aurangzeb, M. Awais, and U. Ali, “Electricity theft detection using cnn-gru and manta ray foraging optimization algorithm,” in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1–6.

[22] S. Aziz, S. Z. Hassan Naqvi, M. U. Khan, and T. Aslam, “Electricity theft detection using empirical mode decomposition and k-nearest neighbors,” in *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 2020, pp. 1–5.

[23] J. Hu, Z. Liu, X. Zhou, and T. Zhang, “Cnn-dbscan intermittent electricity theft detection method for high loss station areas based on correlation analysis,” in *2024 China Automation Congress (CAC)*, 2024, pp. 5282–5287.

[24] M. Ahammad and D. M. Farid, "Electricity theft detection using binary-class imbalanced classification," in *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 2023, pp. 1–6.

[25] I. U. Khan, N. Javaid, C. J. Taylor, and X. Ma, "Robust data driven analysis for electricity theft attack-resilient power grid," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 537–548, 2022.

[26] Z. A. Khan, T. Hussain, I. U. Haq, F. U. M. Ullah, and S. W. Baik, "Towards efficient and effective renewable energy prediction via deep learning," *Energy Reports*, vol. 8, pp. 10 230–10 243, 2022.

[27] M. Xia, H. Shao, X. Ma, and C. W. de Silva, "A stacked gru-rnn-based approach for predicting renewable energy and electricity load for smart grid operation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7050–7059, 2021.

[28] J. Brodny, M. Tutak, and S. A. Saki, "Forecasting the structure of energy production from renewable energy sources and biofuels in poland," *Energies*, vol. 13, no. 10, p. 2539, 2020.

[29] M. I. Ibrahem, M. Mahmoud, M. M. Fouda, B. M. ElHalawany, and W. Alasmary, "Privacy-preserving and efficient decentralized federated learning-based energy theft detector," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 287–292.

[30] M. Savi and F. Olivadese, "Short-term energy consumption forecasting at the edge: A federated learning approach," *IEEE Access*, vol. 9, pp. 95 949–95 969, 2021.

[31] X. Qu, C. Guan, G. Xie, Z. Tian, K. Sood, C. Sun, and L. Cui, "Personalized federated learning for heterogeneous residential load forecasting," *Big Data Mining and Analytics*, vol. 6, no. 4, pp. 421–432, 2023.

[32] F. Esoriano, "Electric power consumption," 2025. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/electric-power-consumption?resource=download

[33] S. Zidi, A. Mihoub, S. M. Qaisar, M. Krichen, and Q. A. Al-Haija, "Theft detection in smart grid environment," 2022. [Online]. Available: https://data.mendeley.com/datasets/c3c7329tjj/3

[34] I. Naidji., C. Choucha., and M. Ramdani., "Decentralized federated learning architecture for networked microgrids," in *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO*, INSTICC. SciTePress, 2023, pp. 291–294.

[35] Micron Technology Inc., "What is federated learning?" n.d., accessed: 2025-05-01. [Online]. Available: https://www.micron.com/about/micron-glossary/federated-learning

[36] M. Asad, A. Moustafa, and T. Ito, "Federated learning versus classical machine learning: A convergence comparison," *CoRR*, vol. abs/2107.10976, 2021. [Online]. Available: https://arxiv.org/abs/2107.10976

[37] D. D. Lab, “Anaconda,” 2025, accessed: 2025-05-10. [Online]. Available: https://domino.ai/data-science-dictionary/anaconda

[38] P. S. Foundation, “What is python? executive summary,” 2025, accessed: 2025-05-10. [Online]. Available: https://www.python.org/doc/essays/blurb/

[39] Spiceworks Editorial Team, “What is tensorflow?” 2025, accessed: 2025-05-11. [Online]. Available: https://www.spiceworks.com/tech/devops/articles/what-is-tensorflow/

[40] T. F. Authors, “Flower: A friendly federated learning framework,” 2025, accessed: 2025-05-11. [Online]. Available: https://pypi.org/project/flwr/

[41] APXML Learning Team. (2024) Handling non-iid data in federated learning. APXML. Accessed on June 4, 2025. [Online]. Available: https://apxml.com/courses/federated-learning/chapter-4-addressing-heterogeneity-personalization/handling-non-iid-data