**Privacy-Preserving User Load Forecasting for Smart Meters Based on Federated Learning and Differential Privacy**

**1. Topic**

The smart grid, a cornerstone of future energy systems, relies on precise user electricity load forecasting for efficient operation. Smart meters provide vast amounts of high-frequency electricity consumption data essential for this task. However, this data is extremely sensitive, containing private information such as user daily routines, appliance usage patterns, and even residential occupancy status. Uploading this raw data to a central server for model training poses significant privacy risks.

This project aims to design and implement a privacy-preserving electricity load forecasting framework. The framework will leverage Federated Learning (FL) to enable multiple users (households) to collaboratively train a shared forecasting model without sharing their raw data. Concurrently, by integrating Differential Privacy (DP) mechanisms, the framework will provide rigorous, quantifiable mathematical privacy guarantees for the training process, preventing the inference of any single user's information from the aggregated model.

**2. What**

1. Data Simulation: Utilize a public electricity consumption dataset (e.g., UCI's "Individual household electric power consumption dataset") to simulate a multi-user (multi-silo) federated learning environment. The dataset will be partitioned by household or time period to be assigned to different virtual clients.

2. Forecasting Task: The task is defined as Short-Term Load Forecasting (STLF)—predicting the electricity consumption for the next hour or several hours based on historical data (e.g., the past 24 hours).

3. Model Architecture: Construct a deep learning model suitable for time-series prediction, such as a Long Short-Term Memory (LSTM) network or a Gated Recurrent Unit (GRU) network.

4. Federated Learning Framework: Build the system using a mainstream federated learning framework (e.g., Flower or TensorFlow Federated). Implement the classic Federated Averaging (FedAvg) algorithm for securely aggregating model updates from clients on the server side.

5. Differential Privacy Integration: Employ the Differentially Private Stochastic Gradient Descent (DP-SGD) algorithm in place of standard optimizers during the client-side training phase. Inject privacy protection before model updates are sent to the server by applying gradient clipping and noise addition.

6. Evaluation Metrics: Prediction Performance: Evaluate the model's forecasting accuracy using metrics like Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Privacy Protection Level: Quantify the strength of privacy protection using the privacy budget (, Epsilon). Analyze the trade-off between privacy and utility by observing the impact of different values on model performance.

**3. Why**

1. **Solves a Core Conflict:** Directly addresses the fundamental conflict between data utilization and privacy protection in the big data era, providing a paradigm for the secure use of sensitive data in fields like healthcare, finance, and energy.

2. **Technological Advancement:** This project merges cutting-edge technologies from machine learning, distributed systems, and cryptography. It aligns with current research hotspots in both academia and industry, offering immense learning and research value.

3. **Vast Application Potential:** With the growth of IoT and edge computing, this technological framework can be directly applied to numerous domains, including smart homes, connected vehicles, and digital health, indicating a promising market future.

4. **Builds High-Level Technical Skills:** Completing this project will equip you with a highly complex and sought-after skill set, creating a strong competitive advantage for career opportunities or further academic pursuits.

**4. How**

**Environment & Stack:**

1. Programming Language: Python
2. Deep Learning Framework: PyTorch or TensorFlow 2.x
3. Federated Learning Framework: Flower (Recommended for its flexibility)
4. Differential Privacy Library: Opacus (for PyTorch) or TensorFlow Privacy (for TensorFlow)

**Technical Workflow:**

1. Data Preparation: Process the public dataset into a time-series format (e.g., using a sliding window approach) and partition it to simulate local datasets for N clients.
2. Centralized Baseline Model: Train a standard LSTM model on the entire dataset to establish a performance baseline for comparison.
3. Federated Learning System Setup: Use the Flower framework to define a client-side class (including local data loading, model instantiation, and training logic) and a server-side strategy (e.g., FedAvg).
4. Differential Privacy Integration: Modify the client-side training loop to incorporate DP-SGD. Using Opacus (PyTorch) or a DPKerasAdamOptimizer (TensorFlow), apply gradient clipping and noise addition during local training.
5. Run three sets of experiments: (1) Centralized model, (2) Standard FL model, and (3) FL model with DP.
6. Compare the RMSE/MAE across the three experiments to quantify the performance impact of FL and DP.
7. Analyze the privacy-utility trade-off by varying DP parameters (like the noise multiplier) and plotting the resulting model performance against the calculated privacy budget (ϵ).

**APS Integration**

Explain how each algorithm incorporates APS concepts:

1. Federated Averaging (FedAvg): Weighted Averaging, Statistical Estimation
2. Differential Privacy (DP-SGD): Probability Distributions, Random Variables
3. Stochastic Gradient Descent (SGD): Sampling Theory, Statistical Approximation
4. Model Evaluation (RMSE): Standard Deviation of Residuals

**5. When**

|  |  |
| --- | --- |
| Date Range | Task |
| Oct 2 – Oct 15 | Literature review, finalize topic, write proposal |
| Oct 16 – Oct 31 | Data collection and preprocessing, Build and Train Centralized Baseline Model (LSTM) |
| Nov 1 – Nov 10 | Implement and Debug the Standard Federated Learning System |
| Nov 11 – Nov 20 | Integrate Differential Privacy (DP-SGD) into the Client Logic, Run Comparative Experiments, Analyze Results, and Write Final Report |
| Nov 21 – Nov 23 | Final report writing and submission |
| Nov 26 | Project presentation |

**6. Novelty, Significance, Feasibility**

1. **Novelty**: The project's innovation lies in the synergistic combination of two cutting-edge privacy-enhancing technologies (Federated Learning and Differential Privacy) and their application to a critical, real-world domain (smart grid). The outcome is not merely a predictive model but a comprehensive framework for trustworthy, decentralized machine learning.**Significance**: It provides a practical tool for students and educators to understand employment trends and make data-informed decisions.
2. **Feasibility**: Despite the conceptual complexity, the project is highly feasible due to the availability of mature, high-level open-source libraries such as Flower and Opacus/TensorFlow Privacy. These frameworks abstract away the intricate details of secure aggregation and differentially private optimization, allowing the developer to focus on the application logic. The existence of suitable public datasets for simulation further ensures that the project can be developed and validated without access to real-world sensitive data.