Federated Deep Learning for Enhanced Prediction of Electric Vehicle Charging Station Availability

Lydia Douaidi[1*], Sidi-Mohammed Senouci[1], Ines El Korbi[1], Fouzi Harrou[2], Ahmet Yazıcı[3]

[1]Université de Bourgogne Europe, DRIVE Lab, Nevers, France.
[2]Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi Arabia.
[3]Department of Computer Engineering, Eskisehir Osmangazi University, Eskisehir, 26048, Turkey .

*Corresponding author(s). E-mail(s): Lydia.Douaidi@u-bourgogne.fr;
Contributing authors: Sidi-Mohammed.Senouci@u-bourgogne.fr;
Ines.El-Korbi@u-bourgogne.fr; fouzi.harrou@kaust.edu.sa;
ayazici@ogu.edu.tr;

Abstract

The growing demand for public Electric Vehicle (EV) Charging Stations (CSs) is vital for promoting wider EV adoption but frequently results in congestion due to limited availability and high usage, leading to increased wait times for drivers. In response, this study proposes a predictive occupancy model utilizing a privacy-preserving Federated Learning (FL) approach, which enables multiple Charging Station Operators (CSOs) to collaborate without sharing sensitive user data. Unlike traditional centralized models, our FL framework allows CSOs to contribute to a central server that aggregates their local models, maintaining privacy and ensuring data security. A key challenge in this setup is managing non-Independently and non-Identically Distributed (non-IID) and heterogeneous data, common in real-world scenarios with diverse user behaviors and charging patterns. To address this, we evaluate various aggregation algorithms, including FedAvg, FedProx, SCAFFOLD, and FedPer, to determine their effectiveness under different conditions. Using 10-minute interval data from the Dundee City CS dataset, we predict station occupancy one hour ahead. The results show that FedPer and SCAFFOLD perform exceptionally well when handling unbalanced data, while FedAvg proves to be more effective in situations with skewed feature distributions. This FL approach not only improves the accuracy of EV charging

station occupancy predictions but also lays the groundwork for securely scaling EV infrastructure, ensuring that privacy concerns do not hinder the development of intelligent, data-driven services.

Keywords: Charging Station ; Occupancy Prediction ; Electrical Vehicle ; Federated Learning ; Data Privacy ; Non-IID Data

1 Introduction

The shift to Electric Vehicles (EVs) is rapidly progressing to achieve emission-free transportation and align with Paris Climate Agreement goals. During 2018-2019, global electric car sales surged by 6%, reaching 2.1 million [1]. As of April 1, 2020, there were 312,767 electric and plug-in hybrid vehicles in circulation in France.

Over the past decade, EVs have gained considerable attention in the literature, covering areas such as predicting energy demand [2, 3], optimizing Charging Station (CS) pricing strategies [2, 4], improving EV charging navigation [5, 6], developing efficient on-demand EV routing [7], identifying suitable locations for deploying of CSs [8, 9], and optimizing maintenance costs for EV fleets [10]. However, the growing number of EVs has increased the need for CSs to meet this high demand, resulting in queues at CSs and causing traffic congestion [11]. Hence, predicting Electric Vehicle Charging Station (EVCS) occupancy is vital for optimizing and managing efficiency and profitability. This predictive capability enhances resource management by allocating CSs more effectively, reducing waiting times for EV drivers, and maximizing CS utilization. Additionally, it empowers grid operators to manage electricity supply and demand efficiently, potentially alleviating strain on the power grid during peak charging times. Moreover, it facilitates scheduling maintenance during low-demand periods, minimizing disruptions and bolstering infrastructure reliability.

Several recent predictive models have been proposed to enhance EVCS occupancy prediction, such as the Spatial-Temporal Graph Convolutional Network (STGCN) in [12], the mixed LSTM neural network in [13], and the Deep Fusion of Dynamic and Static Information model (DFDS) in [14]. While centralized models offer the advantage of aggregating data without concerns about privacy or ownership, enabling a more efficient and straightforward training process, they also present significant limitations. The reliance on centralized architectures raises critical privacy and security concerns due to the sensitive nature of data collected from users and smart grid devices. Furthermore, these models often depend solely on limited local datasets from individual Charging Station Operators (CSOs), which constrains their effectiveness. This dependence decreases their accuracy and robustness and limits the generalizability of the predictive insights, as they fail to incorporate diverse data patterns from a broader operational context. This paper presents an effective collaborative approach using Federated Learning (FL) to tackle the challenges of EVCS occupancy prediction. FL allows multiple CSOs to collaboratively build a predictive model while preserving

[1] https://www.ecologie.gouv.fr/developper-lautomobile-propre-et-voitures-electriques

data privacy, unlike centralized models that require direct data sharing. This ensures that sensitive EV user and CS data remain protected, while FL leverages a more diverse and comprehensive dataset, significantly enhancing model robustness and generalizability across different operational contexts. FL works by training ML models on decentralized data sources without the need to transfer data to a central server. In this framework, multiple CSOs act as FL clients, training local models on their data. The central server aggregates these locally trained models using specified aggregation methods to build a globally optimized model. This approach protects data privacy and ensures improved prediction accuracy by utilizing the combined knowledge of all CSOs, making FL an effective option for real-world, privacy-sensitive applications.

The present work builds upon the earlier framework by Douaidi et al. [15], which introduced a Federated Deep Learning (FDL) approach for predicting EVCS occupancy using Independent and Identically Distributed (IID) data from different CSOs. However, real-world applications present a more challenging scenario, as data collected from various CSOs often exhibit non-IID characteristics due to factors like geographical location, user behavior, and infrastructure differences. This data heterogeneity can significantly impact model performance, making it essential to explore solutions beyond the original IID-based framework. The key contributions of this paper are summarized as follows:

1. We develop an advanced predictive model for EVCS availability, which yields substantial economic benefits by helping CSOs optimize resource allocation, reduce congestion, and enhance the efficiency of EV charging infrastructure.
2. We propose a novel FDL-based framework for EVCS occupancy prediction, enabling multiple CSOs to collaborate without sharing sensitive data. This is a significant step forward in addressing privacy concerns while improving the accuracy of occupancy predictions.
3. We introduce a variety of local Deep Learning (DL) models, including Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Convolutional Neural Networks (CNNLSTM), and Transformer. These architectures are specifically chosen for their ability to capture complex temporal relationships and dynamic patterns in EVCS occupancy data. Furthermore, the inherent challenges posed by non-IID data in real-world scenarios can significantly impact the effectiveness of traditional FL aggregation methods, such as Federated Averaging (FedAvg). To address these challenges, our study conducts a comprehensive exploration and comparison of four widely-used aggregation algorithms: Federated Averaging (FedAvg)[16], SCAFFOLD [17], and Federated Personalization (FedPer) [18]. This comparative analysis addresses a significant gap in the literature, which often neglects the evaluation of multiple aggregation strategies in heterogeneous data. By examining these algorithms, we provide valuable insights into their relative strengths and weaknesses, thereby enhancing the predictive capabilities of our FDL framework for EVCS occupancy prediction.
4. The prediction performance is evaluated on a non-IID dataset from the Dundee city EVCS. Multi-step predictions are conducted one hour ahead using 10-minute interval data, offering detailed and forward-looking insights to improve planning

grid management and reduce congestion at CSs. The results indicate that FedPer and SCAFFOLD-based FL perform better with unbalanced datasets, while FedAvg produces satisfactory outcomes for skewed feature distribution data. Moreover, feature importance identification demonstrates that incorporating lagged data improves EVCS occupancy prediction. Additionally, the SHapley Additive exPlanation (SHAP) method demonstrates the contribution of individual features to the predictions.

The paper's organization is illustrated as follows: The first section reviews previous research on forecasting EVCS occupancy. Subsequently, the second section elaborates on the proposed occupancy prediction framework. The third section comprehensively presents the dataset's experimental details and discusses the obtained results. Finally, we conclude by shedding light on the research's contributions and highlighting potential future directions.

2 Related work

Predicting EVCS availability has recently gained increased attention from engineers and researchers, resulting in the development of various methods broadly classified into two categories: statistical models and Machine Learning (ML) models. Within statistical methods, Luo et al. [19] introduce a method for predicting the charging load of plug-in electric vehicles (PEVs) in China, employing a monte carlo simulation to determine the initial charging point based on probability distributions of starting charging time. In [20], Gaussian mixture models are employed to gain insights into EV user behavior and predict individual charging sessions' duration and energy demand. This analysis uses an open dataset of workplace EV charging with over 30,000 sessions. In [21], Gruosso et al. focus on forecasting the impact of EVCSs, proposing a model based on markov chains. This model considers vehicle distribution, average plug time, and energy consumption to create an occupancy distribution for individual stations and their consumption profiles. Dastpak et al. in [22] propose a Markov Decision Process (MDP) solution and a heuristic to estimate waiting times at CSs. Their method reduces waiting times by 23.7% to 95.4% and total trip duration by 1.4% to 18.5% compared to a benchmark that observes the status of CSs without an occupancy indicator. However, traditional statistical models, such as ARIMA, struggle with multi-step occupancy predictions for CSs due to their limitations in capturing non-linear relationships and reliance on strict data assumptions. In contrast, DL models, particularly LSTM networks, excel at identifying complex non-linear dependencies and long-term temporal patterns, significantly enhancing forecasting accuracy [23, 24]. By leveraging their ability to model intricate relationships in occupancy data, LSTMs consistently deliver superior performance in multi-step predictions compared to traditional methods.

In recent years, ML models, especially DL ones, have become a promising solution for time-series prediction, resulting in significant electric mobility advancements. For example, Straka et al. [25] introduced a data-driven approach to predict popular charging pool locations using classification techniques, such as logistic regression with L_1 regularization, random forests, and gradient-boosted decision trees. Results show

that strategically positioning CSs in high-traffic areas with convenient services attracts more users. The study in [26] explored the importance of features and CS characteristics that can influence occupancy using logistic regression models. The evaluation used data from 52 CSs with over 24,000 charging events. Sao et al. [14] introduced an approach by combining dynamic features such as time and day of the week with static attributes like historical occupancy patterns at specific times. Specifically, Gated Recurrent Unit (GRU)-based decoding is used to forecast the future occupancy trend of individual charging points over various periods, ranging from 10 minutes to several hours. The study in [27] focused on short-term forecasting of EVCS occupancy using extensive data streaming analysis. The authors proposed an architecture that leverages historical data and real-time data streams from CSs to forecast station availability for the next 15 minutes. They used a streaming logistic regression model that outperforms models trained solely on historical data. In [13], Ma et al. introduced Hybrid LSTM neural networks that merge historical charging occupancy sequence data with occupancy rates, achieving high performance for both short-term (10 minutes) and long-term occupancy predictions (2 hours). In [28], Luo et al. proposed an Attribute-Augmented Spatiotemporal Graph Informer (AST-GIN) that combines Graph Convolutional Network (GCN) and Informer layers to capture both spatiotemporal dependencies and external factors (like points of interest and weather). Tested in the Dundee City EV charging dataset, it outperformed three baseline models in predicting CS availability when the prediction horizon is 90 minutes and 120 minutes.

Table 1 summarizes some of the recent relevant works in the literature, as discussed in the related work section. From this table, we observe that previous studies on CS occupancy prediction have predominantly used centralized training approaches which raises privacy concerns. The data required for predicting occupancy includes sensitive information collected from CSs. This information can reveal user behaviors and patterns potentially leading to privacy breaches under regulations like GDPR. Sharing such data not only risks violating user privacy but also exposes proprietary information to competitors and could be misused for profiling charging habits. Additionally, transferring large amounts of data from all stations of all CSOs to a central server may be impractical due to bandwidth limitations and can be resource-intensive. To address these limitations and challenges, FL is an alternative solution that offers a decentralized approach to model training, where data remains localized and only model updates are shared, thus mitigating privacy concerns and reducing the volume of data transmitted over the network [29]. The use of FL represents a balanced approach that seeks to address the competing priorities of data privacy and model accuracy.

FL has demonstrated successful applications in the electric mobility domain. These applications include recommendation systems for EVCSs [30], energy demand prediction [3, 31–33], mobile CS placements [34], and competitive EV charging market analysis [35]. In our previous work [15], we introduced a FL framework for predicting CS occupancy using Independently and Identically Distributed (IID) data. In that study, we demonstrated that individual CSOs with insufficient data could not train models that converge well or accurately predict station occupancy, thereby necessitating the use of FL to collaboratively leverage data from multiple CSOs. However, it is important to acknowledge that the assumption of IID data may not hold in real-world

Table 1: Comparison of relevant works on EVCS occupancy prediction.

Authors	Dataset	non-IID	Prediction Methods	Federated	Privacy
Luo et al. [19]	PEV dataset, China	✗	Monte Carlo simulation	✗	✗
Lee et al. [20]	ACN-data EV charging	✗	Gaussian mixture models	✗	✗
Gruosso et al. [21]	Data from Teinvan project	✗	Markov chains model	✗	✗
Dastpak et al. [22]	Open EV dataset[1]	✗	Markov chains model	✗	✗
Straka et al. [25]	EVnetNL dataset	✗	Logistic regression, random forests, gradient boosted regression trees	✗	✗
Motz et al. [26]	ACN-data EV charging	✗	Logistic regression	✗	✗
Sao et al. [14]	EVCS dataset in Germany	✗	Encoder-decoder neural architecture	✗	✗
Soldan et al. [27]	Dataset from 1,724 EV charges	✗	Logistic regression	✗	✗
Ma et al. [13]	EVCS Dundee city dataset	✗	Hybrid LSTM neural network	✗	✗
Luo et al. [28]	EVCS Dundee city dataset	✗	Graph Convolutional Network	✗	✗
Douaidi et al. [15]	EVCS Dundee city dataset	✗	LSTM-based neural network	✓	✓
This study	Generated EVCS dataset	✓	LSTM, BiLSTM, CNNLSTM, Transformer	✓	✓

scenarios. In practice, CSOs' data is often non-IID due to factors such as varying user behaviors, geographic locations, and temporal usage patterns. The presence of non-IID data poses significant challenges and can limit the effectiveness of traditional FL aggregation methods like FedAvg.

3 The proposed EVCS occupancy prediction framework

As discussed in the previous section, centralized methods pose significant privacy and security risks in the context of EVCS data. The sensitive nature of user charging information, combined with regulatory requirements and commercial competition, necessitates a privacy-preserving approach. To address this shortcoming, our contribution focuses on flexible FL-based prediction models to enhance EVCS occupancy prediction.

In this section, we provide a comprehensive overview of our contribution to advancing EVCS occupancy prediction. Our framework is built around two key components.

First, we introduce an innovative system architecture that facilitates distributed collaboration among federated CSOs, with coordination managed by a central server. This architecture ensures secure and efficient collaboration while maintaining data privacy. Second, our framework integrates a variety of FDL-based prediction models, individually trained by each CSO, which are then aggregated using FL methods at the server level. This collaborative approach enhances EVCS availability prediction by leveraging the collective insights and diverse datasets of multiple CSOs, improving model accuracy and overall system performance. By combining decentralized data with FL techniques, we maximize the predictive power while safeguarding sensitive information, providing a robust and scalable solution for EVCS management.

3.1 System architecture for EVCS occupancy prediction

The EVCS system is composed of several key components that ensure the efficient management of EV charging processes [36]. Through an in-depth literature review, we have identified the core elements that make up the EVCS system. These components include EV Users, CSOs, and e-Mobility Service Providers (eMSPs). Together, these stakeholders play essential roles in the successful operation and management of EV charging infrastructure [15] (see Figure 1):

- **EV User:** Individuals who drive electric vehicles and rely on charging infrastructure,
- **CSO:** Entities responsible for the installation, management, and maintenance of charging points. CSOs retain ownership of all data related to the usage of their CSs,
- **eMSP:** Providers of EV charging services, facilitating access to CSs and handling payment processes through agreements with one or more CSOs.

In today's competitive environment, CSOs are often hesitant to share data with competitors due to concerns around privacy and security. Centralized approaches that depend on extensive data sharing face significant challenges, particularly in maintaining robust data privacy and managing data transmission overhead efficiently. This study introduces a decentralized approach using FL) to address the privacy and security challenges inherent in EV charging systems. FL allows individual CSOs to collaborate while keeping their data local, eliminating the need for sharing sensitive information about CS usage [37]. By keeping data local, FL also minimizes the need for large-scale data transfers over the network, reducing data transmission overhead. Through collaboration within the federated framework, CSOs can benefit from the diverse datasets of other operators, improving the accuracy and reliability of occupancy predictions without direct data sharing—an essential feature in a competitive landscape. A central server, potentially operated by a neutral entity, orchestrates the model training process across all CSOs (Figure 1). This server aggregates their local models within a collaborative system using a defined aggregation method.

In the following, we provide a comprehensive overview of the main steps in the FL training process:

- *Step 1:* The central server initiates the process by initializing a DL model, denoted as M^0, and transmits this initial model to all participating CSOs,

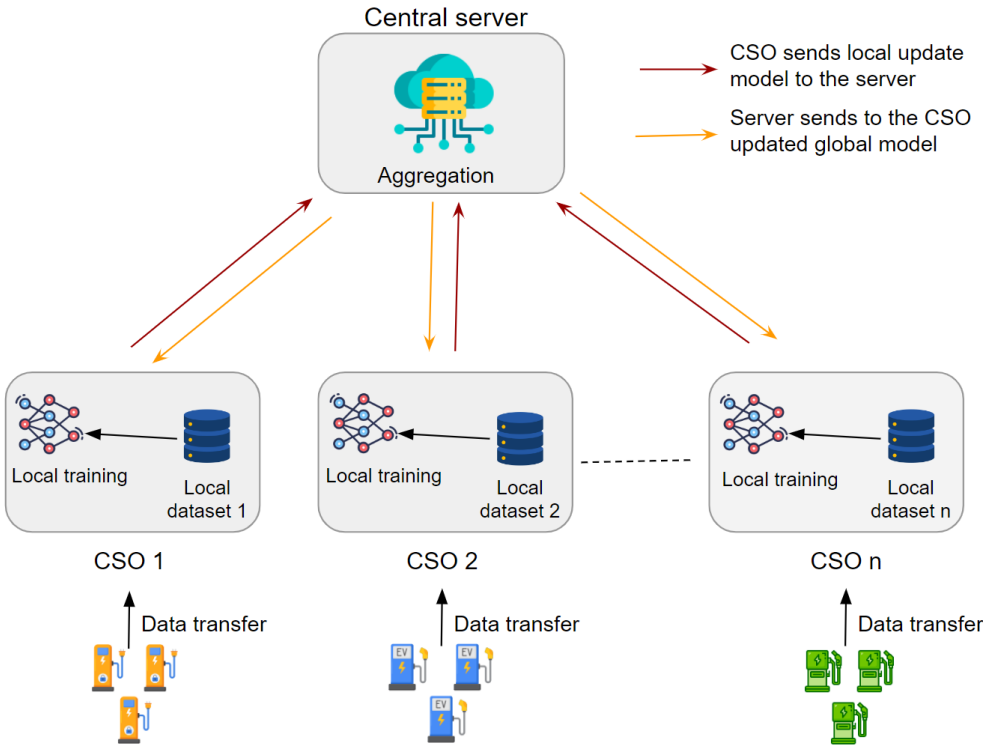


Fig. 1: A schematic representation of the proposed FL architecture to predict EVCS occupancy.

- *Step 2:* Each CSO C_i proceeds to train the model they received, utilizing their locally-stored data D^i. Subsequently, they transmit the locally updated model weights, denoted as w_i^t, back to the central server,
- *Step 3:* The server computes updated model weights by aggregating the weights provided by all clients through an aggregation algorithm (e.g., Fed-Avg), resulting in an updated global model called M^t with the new weights w^t,
- *Step 4:* The server sends the updated weights w^t to all CSOs for the next training iteration, and this process continues until model convergence is achieved.

The subsequent section delves into the details of the proposed DL models, individually trained by each CSO. Additionally, it outlines the FL aggregation techniques employed to construct the global prediction model.

3.2 FL-based EVCS occupancy prediction models

The adopted federated framework for EVCS occupancy prediction operates on two distinct levels: local and global. At the local level, DL models are individually trained at each CSO. In contrast, at the global level, these locally trained models are aggregated

on a centralized server, resulting in the creation of a comprehensive global model. The following sections provide a detailed description of each of these levels.

3.2.1 Local models

Figure 2 provides a visual representation of the workflow for constructing the local EVCS occupancy prediction models. The local model construction is performed in two phases: training and testing. After data preprocessing, we train the proposed DL models using the training local dataset. During this phase, we fine-tune the model parameters, optimizing them to minimize the loss function. Then, the trained models are employed to predict EVCS occupancy based on the testing dataset. Finally, we employ various evaluation metrics to assess the investigated models' predictive performance.

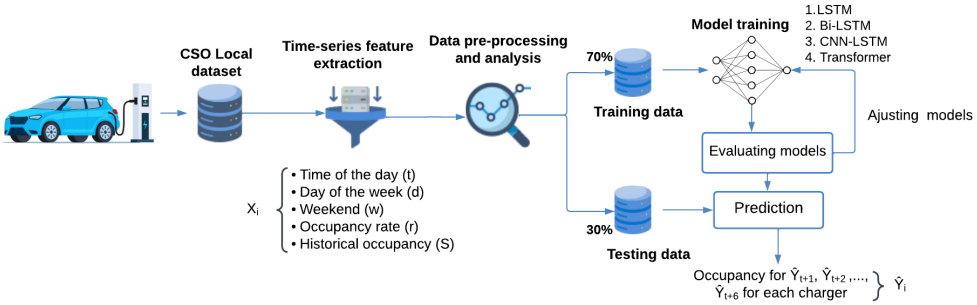


Fig. 2: Conceptual framework of the proposed local prediction models.

In the presented FL framework with N clients, each CSO is denoted as C_i with a local dataset $D^i = (x_i, y_i)$. We define a collection of charging stations as CS, where each CSO owns multiple CSs. For a given CSO, C_i, we denote their set of CSs as CS_i. The problem is defined as follows:

For each CS denoted as $cs_i \in CS$, a learning model takes input features $X_i =< t, d, w, or_t, s_t >$, representing time of day (t), day of the week (d), weekend indicator (w), occupancy rate (or_t^{cs}), and historical charging sequences ($s_t^{cs} = (y_{t-1}, y_{t-2}, \ldots)$, where y_{t-1} is the lagged feature representing the occupancy at the previous time step $t-1$). The model aims to predict the charging occupancy for each station cs_i at future time steps $t+1, t+2, \ldots, t+6$, as can be seen in Figure 2. Indeed, we represent the problem as a binary classification task where the decision function f maps the input X_i to a sequence of predicted values $\hat{Y}_i = \left[\hat{y_i}^{t+1}, \hat{y_i}^{t+2}, \ldots, \hat{y_i}^{t+6}\right]$, where $\hat{Y_i}^{t+k} \in \{0, 1\}$ indicates whether the station cs_i is predicted to be occupied (1) or unoccupied (0) at future time $t+k$.

The choice to predict up to 6 future time steps is motivated by an empirical evaluation of the model's predictive performance. Specifically, the analysis reveals a marked decline in prediction accuracy beyond the sixth time step, attributed primarily to the

increasing uncertainty associated with long-term occupancy prediction. Consequently, constraining the prediction horizon to 6 steps achieves an optimal balance between providing meaningful foresight and preserving predictive reliability over the next hour.

To evaluate the performance of the DL models, we employ the b inary cross-entropy (BCE) loss function [38]. This function quantifies the difference between the model's predicted binary outcomes and the actual values. It is defined as follows:

$$BCE = \frac{-1}{k} \sum_{j=1}^{k} y_j . \log \hat{y}_j + (1 - y_j) . \log(1 - \hat{y}_j), \tag{1}$$

where y_j is the real value at time j, and $\hat{y}_j$ is the predicted value.

In this study, we compare the performance of benchmark DL models for multi-step EVCS occupancy prediction. The models we consider are LSTM, BiLSTM, CNNLSTM, and a transformer. The overall architecture of the DL model is illustrated in Figure 3. These models were chosen due to their distinct architectural strengths and capabilities in addressing multi-step time series forecasting tasks.

1. *LSTM* [39]: are a type of Recurrent Neural Network (RNN) that can learn long-term dependencies in time-series data. LSTM networks achieve this by using gating mechanisms to control the flow of information through the network, making them a good choice for multi-step times series classification tasks.
2. *BiLSTM* [40]: is a type of LSTM that processes input sequences in both forward and backward directions, enabling it to capture contextual information from both past and future time steps. The model makes more accurate predictions than a standard RNN, which can only process sequences in the forward direction. However, BiLSTMs are computationally expensive.
3. *CNNLSTM* [41]: is a fusion model that combines the strengths of LSTM networks and CNNs for time series prediction. By combining these two architectures, CNNLSTM can learn spatial patterns and long-term dependencies in sequential data, leading to more accurate predictions. This model combines the strengths of CNNs in feature extraction and LSTMs in sequence modelling, but it requires more computational resources and data.
4. *Transformer* [42]: is a type of neural network that can capture long-term patterns and learn parallel relationships between different parts of the time series data. Transformers first encode the input time series into a sequence of vectors, then learn relationships between these vectors and predict the next value in the sequence. Transformers are well-suited for multi-step time series classification because they can capture long-range dependencies effectively through self-attention mechanisms.

3.2.2 FL global aggregation models

In each FL round r, each CSO C_i trains a local model denoted as M_i^r. As illustrated in Figure 1, the central server utilizes an aggregation algorithm to merge the weights of all local models received from CSOs, creating the aggregated global model denoted as M^r for round r.

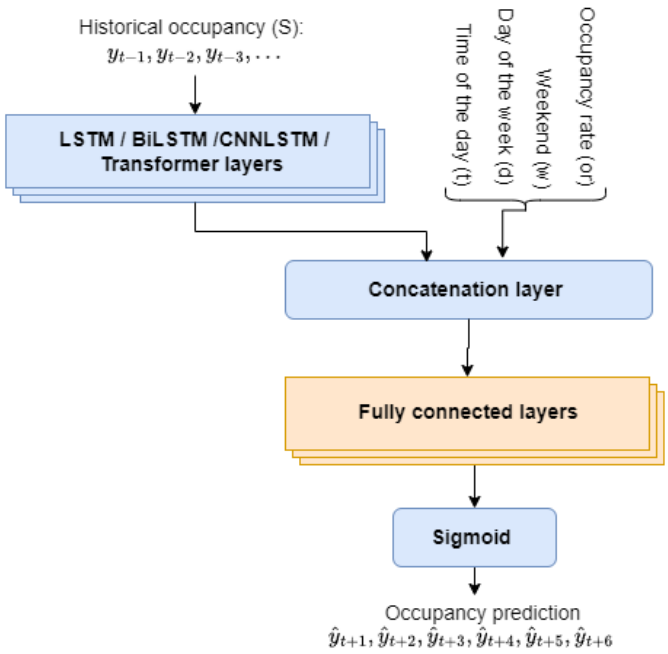


Fig. 3: Architecture of the proposed DL models.

One of the key challenges in the aggregation process within FL is managing the heterogeneity of local data, also known as non-IID (non-independent and identically distributed) data [43]. This occurs because each participating CSO's local dataset has a distinct distribution, making the local training objectives misaligned with the global optimum. The variability in data distributions can significantly hinder the performance of a global model. Several studies have aimed to mitigate this challenge by optimizing the loss function across the entire training dataset within the non-IID data context. Below, we summarize some state-of-the-art aggregation algorithms we implemented to tackle this issue, ensuring robust performance in a federated environment.

1. **Federated Averaging (FedAvg):** The FedAvg algorithm [29] is introduced as the aggregation method in Google's implementation of an FL system. A central server initializes a neural model and sends it to all selected clients for local training. Once the local training is completed, each client transmits their model's weights to the server for aggregation. The server uses federated averaging defined in (2) to combine all the weights received from the clients, giving more weight to clients with larger datasets, thus significantly influencing the aggregated model.

$$\min_{\theta} F(\theta) = \sum_{i=1}^{N} p_i F_i(\theta), \tag{2}$$

where p_i is the weight of the i-th client such that $p_i \geq 0$ and $\sum_{i=1}^{N} p_i = 1$. $F_i(\cdot)$ represents the loss.

2. **Federated Proximal (FedProx):** FedProx algorithm [16] is a generalization of FedAvg with a proximal term into the objective function during local training to address the heterogeneity of data and systems. To limit the distance between the local models and the global model, FedProx introduces L_2 regularization, as shown in the equation 3:

$$\min_{\theta} F(\theta) = \sum_{i=1}^{N} p_i F_i(\theta) + \frac{\mu}{2} \|\omega - \omega^t\|^2, \tag{3}$$

where ω is the local parameter to optimize, and ω^t is the global parameters at time t, μ is the fedprox hyper-parameter to control the weight of the L_2 regularization.

3. **Stochastic Controlled Averaging for FL (SCAFFOLD):** SCAFFOLD [17] addresses the challenge of non-IID data in FL by introducing variance among clients and employing variance reduction techniques to correct local updates. This is achieved through the use of control variates, which each client updates during local training. These control variates are adjusted either by computing gradients from the client's local data relative to the global model or by reusing previously computed gradients. SCAFFOLD significantly enhances convergence speed, especially in heterogeneous environments, but this improvement comes with a trade-off: it increases the communication size per round compared to simpler methods like FedAvg. The extra communication overhead results from the transmission of both model parameters and control variates, but this cost is often outweighed by the enhanced model accuracy and efficiency in non-IID settings.
4. **FL with Personalization Layers (FedPer):** The idea behind the FedPer approach [18] is to split the model into two distinct components: base layers and personalized layers. Only the base layers are communicated to and aggregated by the central server, while the personalized layers remain exclusive to each client. The base layers focus on learning general representations that are useful across all clients and can be shared through the aggregation process, fostering collaboration. In contrast, the personalized layers are tailored to each client's unique data and handle decision-making tasks, allowing for specialization based on local input. This dual-layer structure enables FedPer to accommodate the heterogeneity of client data, ensuring robust, individualized performance while still benefiting from shared global knowledge.

4 Data description and pre-processing

In this section, we provide a comprehensive analysis of the dataset, outlining key derived features and conducting a thorough data examination to capture its underlying characteristics and trends.

This study leverages the EVCS Dundee City dataset, which encompasses 16,659 charging sessions between April 04, 2018, and May 31, 2018. Figure 4 illustrates the geographical distribution of EVCS in the Dundee city, UK. The dataset categorizes

CSs into three types based on their power capacity: slow stations (7kW), fast stations (22kW), and rapid stations ($\geq$ 43kW). Notably, the distribution of sessions across these station types was 53.9% for slow chargers, 28.4% for rapid chargers, and 17.7% for fast chargers. Outliers displaying unusual charging duration patterns, such as sessions lasting over 20 hours or shorter than 10 minutes, were were systematically removed from the dataset to ensure data quality.

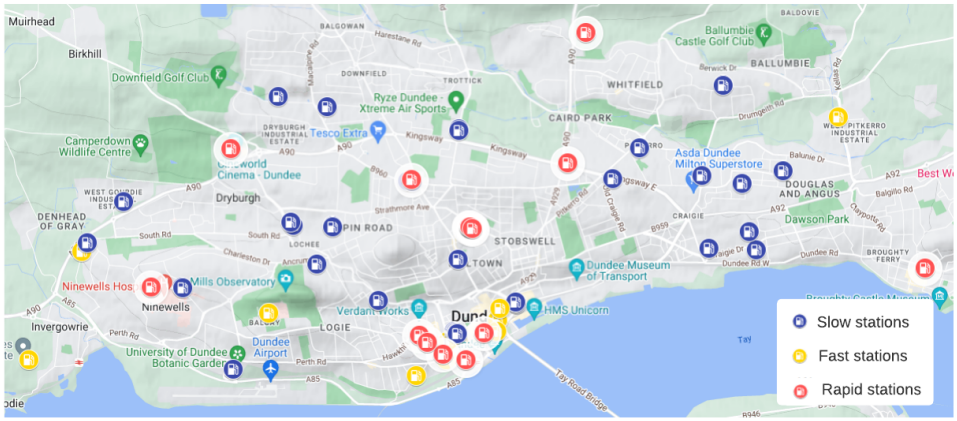
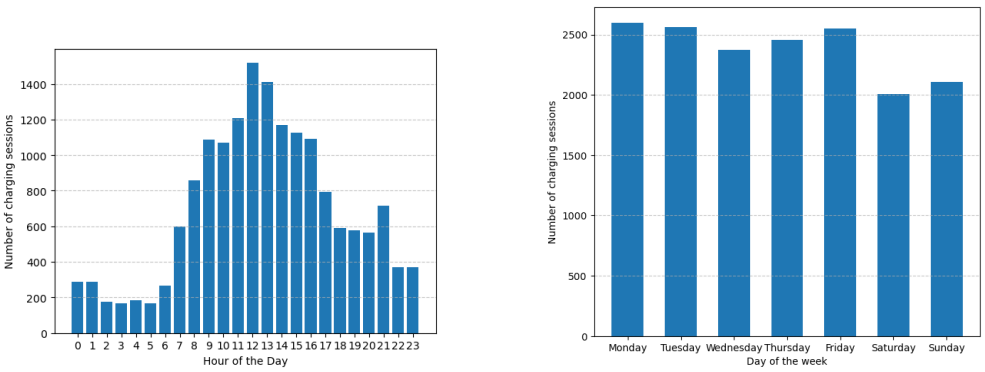


Fig. 4: Spatial distribution of EVCS in the city of Dundee, UK.

Before describing the used features, let us define the set of CSs as CS and introduce $T = \{1, 2, \ldots, 144\}$ as the set of discrete time indices, which correspond to a day of 24 hours, segmented into 144 intervals of 10 minutes, as previously defined by [13]. Specifically, the considered features in this study are generated using the start date and time of each session and the corresponding end date and time. Five features are extracted: time of the day, day of the week, weekend indicator, charging occupation and occupancy rate.

- **Charging occupation (y_t^{cs}):** A charging occupation is a binary flag that indicates a charging record of the CS at the time $t \in T$. A binary flag called 'charging occupation' indicates whether a $cs \in CS$ is used at a specific time $t \in T$.
- **Occupancy rate (or_t^{cs}):** It's the average proportion of time, within a defined time window, during which the CS is occupied. It is calculated by taking the mean (average) of the occupancy values recorded for a station cs_i within the specified time t.
- **Charging sequence (s_t^{cs}):** Sequential charging records for a particular $cs \in CS$, occurring in chronological order, denoted as $s \in S$. This sequence comprises a series of charging occupation $s =< y_t^{cs}, y_{t+1}^{cs}, y_{t+2}^{cs} \ldots y_n^{cs} >$. Where $t, t+1, \ldots, n$ represent a sequence of temporally consecutive time points, S encompasses the complete set of charging sequences.

Figure 5a illustrates the distribution of hourly charging sessions. As expected, a bell-shaped distribution is observed, with a peak around noon. Notably, most charging sessions occur between 07:00 am and 09:00 pm. This midday peak could be attributed to users optimizing their lunch breaks for charging. Indeed, a notable observation is the reduced frequency of charging sessions during nighttime hours. This phenomenon is particularly prominent in public CSs, as users may prefer to charge at home at night.



(a) Hour of the Day

(b) Day of the Week

Fig. 5: Distribution of charging sessions by time in the Dundee City EVCS dataset.

Following a comprehensive data analysis, Figure 5b depicts the distribution of daily charging sessions. Notably, we observe high demand on Mondays, Tuesdays, and Fridays, likely aligned with workweek commuting patterns. Conversely, weekends significantly drop, with Saturday having the lowest demand. Understanding user behaviors is crucial for optimizing resource allocation within the EV charging infrastructure.

Figure 6 illustrates the data's Pearson correlation matrix [44]. Each cell's value indicates the degree and direction of linear association between two variables in this matrix. Notably, we observe a strong positive correlation of 0.71 between the 'lagged feature' and 'occupancy,' indicating that past occupancy strongly influences current occupancy levels. Additionally, we identify a moderate correlation of 0.55 between 'occupancy rate' and 'occupancy', suggesting that the CSs' occupancy rate is moderately related to their current occupancy status.

While the correlation matrix is valuable for assessing linear relationships between features, it may not capture more complex or non-linear connections. We conducted a comprehensive feature analysis using the Random Forest (RF) algorithm [45] to understand feature importance better. The results of this analysis are depicted in Figure 7. Results indicate that 'lagged features' are the most influential, contributing to over 70% of the overall feature importance (Figure 7). These results highlight the significance of historical charging patterns in predicting CS occupancy. The 'occupancy

Fig. 6: Correlation matrix of the used data.

rate' feature follows, contributing approximately 12% to the model's predictive power. In contrast, 'day of the week' and 'weekend' features show lower importance, at 4% and 1%, respectively. This suggests that they have a more limited impact on the model's predictions than the 'lagged features' and 'occupancy rate.'

To understand how much each feature contributes to a model's output using the Shapley Additive exPlanation (SHAP) [46] with an XGBoost model. Figure 8 illustrates SHAP plots generated using the XGBoost model. In the x-axis of the plot, we find the SHAP values associated with each input feature. These values represent the average contribution of each variable to the model output. Positive SHAP values indicate a positive impact on the prediction, while negative values suggest a negative impact. The y-axis lists the input features, and their positions on the plot indicate the magnitude and direction of their influence. The color of each point corresponds to the feature value: red represents high values, and blue represents low values. The vertical spread of the points for each variable illustrates the distribution of SHAP values, indicating the variability in the impact of that variable. The SHAP plot in Figure 8 indicates that the feature y_{t-1} representing occupancy at time $t-1$ emerges as the most significantly influential factor, followed by 'occupancy rate'. This suggests that the occupancy information from the immediate past has a pronounced impact on the predictive outcome. Furthermore, the 'occupancy rate' feature closely aligns in terms of influence, indicating that the average proportion of time during which the CS is occupied significantly shapes the model's predictions.

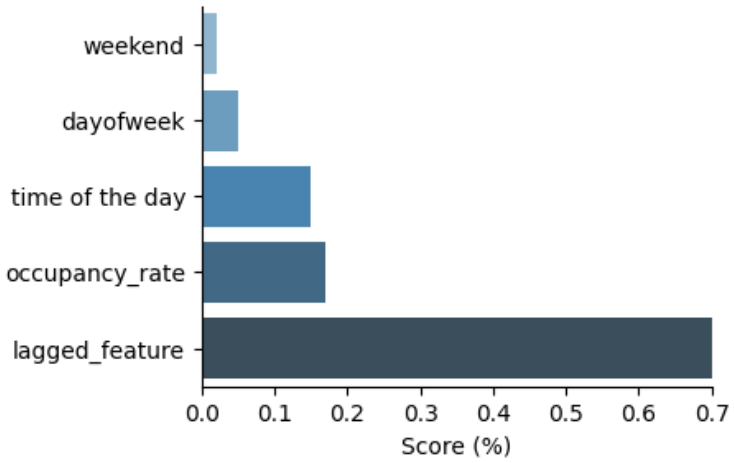


Fig. 7: Features importance identification using RF.

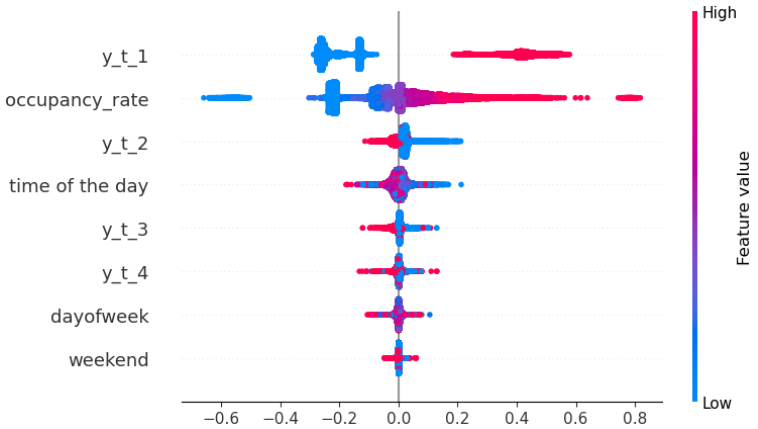


Fig. 8: SHAP values visualization for feature contributions in an XGBoost model.

From Figure 8, we also observe that features such as time of the day, day of the week, and weekend, along with other lagged data, show a relatively weaker impact on the model's predictions.

The data analysis highlights the importance of capturing temporal patterns for accurate predictions and helped us better understand our dataset. Based on these insights, we proceed to train LSTM-based models, which are specifically designed to handle time series data and effectively capture these temporal dependencies to predict multi-step occupancy of CSs.

It is important to note that the Dundee city charging data follows an IID distribution. However, in practical scenarios, charging data exhibits a non-IID pattern. For example, a CS near a commercial center might encounter higher usage during weekends

than on weekdays. In the absence of publicly available non-IID EVCS datasets, we have created non-IID datasets derived from the original Dundee city dataset, including two non-IID scenarios:

1. **Feature distribution skew:** In this scenario, the data is split into four distinct subsets, each representing a unique CSO. These subsets capture distinct charging patterns based on both the time of day and day of the week, reflecting real-world behavioral differences in EV usage. The subsets are designed to reflect diverse charging needs across different user profiles, as follows:

   - Subset 1 (Urban): represents urban commuters, with peak utilization during weekday mornings (6 AM to 9 AM), reflecting high demand from workers commuting to city centers, and significantly reduced activity on weekends.
   - Subset 2 (Suburban): reflects suburban users, exhibiting high activity during evening hours (6 PM to 9 PM) and increased usage on weekends, as these users typically charge their vehicles after work and during leisure periods.
   - Subset 3 (Commercial): reflects commercial areas, where charging demand spikes during lunch hours (12 PM to 3 PM), aligning with business operations, and shows heightened usage over the weekends when commercial traffic increases.
   - Subset 4 (Residential): represents residential areas, with moderate evening peaks (5 PM to 7 PM) as residents return home from work, and steady usage throughout the week, reflecting consistent residential charging behavior.

   This division simulates diverse utilization patterns at different locations, mimicking real-world conditions where CSs exhibit varying charging behaviors. This ensures that each CSO dataset captures unique and realistic usage patterns.
2. **Imbalanced datasets:** In this scenario, the data is again divided into four distinct subsets, each assigned to a different CSO. To emulate real-world disparities in station occupancy, we deliberately introduced imbalances in the class distribution of the target variable. These imbalances capture variations in the usage of CSs, highlighting different occupancy patterns:

   - Subset 1 (Low demand): primarily consists of underutilized or mostly unoccupied stations, where class 0 (unoccupied) significantly dominates class 1 (occupied), representing stations with low demand.
   - Subset 2 (Severe underutilization): exhibits an even stronger class 0 imbalance, with extreme underutilization and very few instances of occupied stations, modeling severely underused CSs.
   - Subset 3 (High demand): mainly consists of highly occupied stations, where class 1 (occupied) overwhelmingly surpasses class 0, reflecting locations with consistently high demand and usage.
   - Subset 4 (Moderate demand): a mixed or more balanced subset, where the distribution between class 0 and class 1 is closer to even, representing moderate utilization.

   CSs often experience unequal demand based on their location and the surrounding area's usage patterns. By introducing these imbalances we create a dataset that better mirrors the diverse conditions seen in practice. This allows us to evaluate

how FL algorithms perform under imbalanced class distributions, particularly when certain classes, like station occupancy, are underrepresented.

5 Results and discussions

In this section, we explain the methodology used to train the models and provide an overview of the metrics employed to evaluate the effectiveness of these proposed DL models. We then present the results of our analysis, focusing on two distinct scenarios: the impact of feature distribution skewness and the challenge of imbalanced datasets. Finally, we engage in a comprehensive discussion of our findings.

5.1 Dataset partitioning and model hyperparameters

In our experiments, we prepare the datasets of the CSOs by carefully partitioning them into a training set and a test set, maintaining a balanced 7:3 ratio. The training process used 5-fold cross-validation to prevent overfitting and improve generalization. The dataset was split into five parts, with each part taking turns as the validation set while the others were used for training, ensuring robust performance evaluation. To achieve optimal performance, we conduct hyperparameter fine-tuning. The number of federated clients (i.e., CSOs) is set to $N = 4$. We settle on a learning rate of $\eta = 0.001$, training epochs $E = 30$, batch size $B = 256$, and training rounds $R = 20$. The LSTM model comprises two layers, the BiLSTM model has two layers of LSTM, the CNNL-STM model has one convolutional layer and 2 LSTM layers, and the Transformer model has three transformer layers. All models incorporate three fully connected layers. As for the activation functions, we select a Rectified Linear Unit (ReLU) for LSTM and fully connected layers, while the output layer uses Sigmoid. Regarding aggregation methods employed at the central server, in the FedProx approach, a proximal term with a parameter value of $\mu = 0.01$ is applied, while FedAvg utilizes $\mu = 0$. The FedPer approach incorporates two personalized layers into each proposed model. For the SCAFFOLD, a weight decay of 0.01 is defined.

The models evaluated in this paper are developed in Python and simulated on a Windows workstation equipped with an NVIDIA GeForce RTX3060 GPU, an Intel 12th Gen Core i7-12700 Processor CPU, and 32 GB of RAM. The simulations are performed using PyTorch 1.11.0 on the Visual Studio Code platform.

5.2 Evaluation metrics

The performance of the prediction models is assessed using the following key metrics [47]:

- **Accuracy :** The proportion of correct predictions out of the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- **Precision :** The proportion of correct positive predictions, reducing false positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

- **Recall (Sensitivity):** The proportion of actual positives correctly predicted, reducing false negatives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

- **F1 score:** The harmonic mean of precision and recall, balancing false positives and negatives.

$$\text{F1-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

- **AUC (Area Under ROC Curve):** Measures a model's ability to distinguish between classes. A higher AUC indicates better performance, with 1 representing perfect classification.

5.3 Case 1: Features distribution skew

In this section, the results of multistep predictions are presented, evaluating the performance of the different aggregation algorithms (FedAvg, FedProx, SCAFFOLD, FedPer) across DL models (LSTM, BiLSTM, CNNLSTM, Transformer) in the case of features distribution skew. The prediction time window extends from 1 (10 minutes) to 6 (60 minutes) time steps ahead.

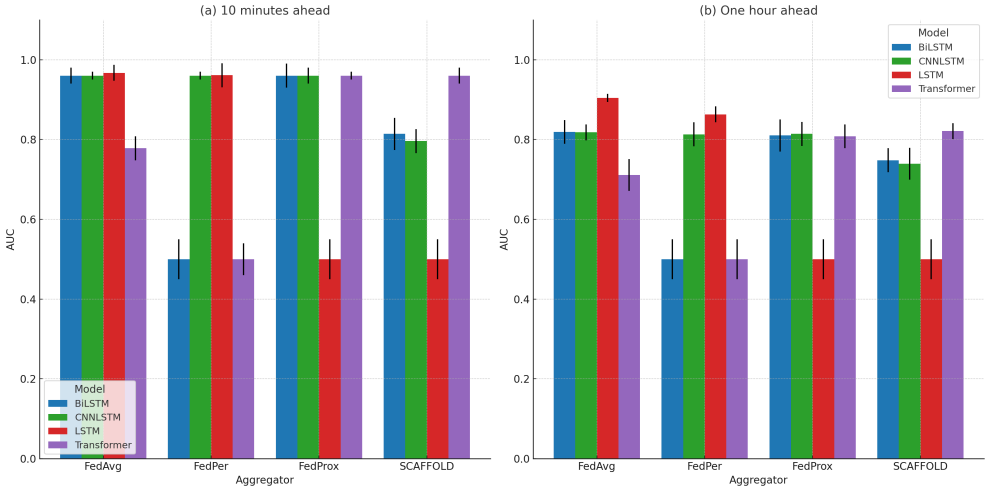


Fig. 9: AUC values with 95% confidence intervals for different Models and Aggregators across varying time intervals within the context of features distribution skew.

In Figure 9, the AUC values, along with their 95% confidence intervals, demonstrate diverse model performance across various aggregation methods, ranging from

0.5 to 0.967 for one time step ahead with a 10-minute interval. The error bars in each plot represent the confidence intervals, indicating the level of uncertainty associated with each model's performance. Notably, smaller error bars suggest more consistent and reliable model predictions, whereas larger error bars imply greater variability and less stability in the results.

The FedAvg aggregator demonstrates robust performance across all models, achieving high AUC values, such as 0.97 for LSTM and 0.96 for both BiLSTM and CNNLSTM, while maintaining relatively narrow error bars, indicating reliability in its predictions. Additionally, under FedProx, BiLSTM, CNNLSTM, and Transformer, they achieve AUC values of 0.96, but with varying confidence intervals. FedPer produces notable results, with LSTM and CNNLSTM reaching AUC values of 0.96. In the case of SCAFFOLD, the Transformer model stands out with an AUC of 0.96, although the error bars suggest slightly higher uncertainty in its predictive performance.

Table 2: Performance metrics for FedAvg, FedProx, SCAFFOLD, and FedPer using different models in the feature distribution skew scenario for 10 minutes ahead.

Aggregators	FL Model	Metrics			
		Accuracy	Precision	Recall	F1-Score
FedAvg	LSTM	**0.972**	**0.972**	**0.972**	**0.972**
	BiLSTM	0.967	0.967	0.967	0.967
	CNNLSTM	0.967	0.967	0.967	0.967
	Transformer	0.697	0.840	0.697	0.707
FedProx	LSTM	0.700	0.490	0.700	0.577
	BiLSTM	0.967	0.967	0.967	0.967
	CNNLSTM	0.967	0.967	0.967	0.967
	Transformer	0.967	0.967	0.967	0.967
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577
	BiLSTM	0.749	0.853	0.749	0.759
	CNNLSTM	0.842	0.839	0.842	0.840
	Transformer	0.966	0.966	0.966	0.966
FedPer	LSTM	0.967	0.967	0.967	0.967
	BiLSTM	0.700	0.490	0.700	0.577
	CNNLSTM	0.967	0.967	0.967	0.967
	Transformer	0.700	0.490	0.700	0.577

Table 2 and Table 3 demonstrate that FedAvg achieves strong performance with high accuracy, precision, recall, F1-Score, and AUC across LSTM, BiLSTM, and CNNLSTM models. Specifically, it achieves an accuracy of 0.972 for the LSTM model and 0.967 for both BiLSTM and CNNLSTM while showing a comparatively lower accuracy of 0.697 for the Transformer model. In contrast, FedProx and SCAFFOLD showed more mixed results. While the LSTM models within these approaches faced challenges with an accuracy of 0.7, other models like BiLSTM, CNNLSTM, and Transformer excelled with 0.967 accuracy. FedPer also presented various results, with some models achieving an accuracy of 0.967 while others faced challenges with an accuracy of 0.7.

Table 3: Performance metrics for FedAvg, FedProx, SCAFFOLD, and FedPer using different models in the feature distribution skew scenario for one-hour ahead.

Aggregators	FL Model	Metrics			
		Accuracy	Precision	Recall	F1-Score
FedAvg	LSTM	**0.918**	**0.919**	**0.918**	**0.919**
	BiLSTM	0.845	0.846	0.845	0.845
	CNNLSTM	0.846	0.847	0.846	0.847
	Transformer	0.639	0.778	0.639	0.650
FedProx	LSTM	0.700	0.490	0.700	0.577
	BiLSTM	0.844	0.843	0.844	0.843
	CNNLSTM	0.844	0.844	0.844	0.844
	Transformer	0.839	0.839	0.839	0.839
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577
	BiLSTM	0.699	0.792	0.699	0.712
	CNNLSTM	0.799	0.792	0.799	0.794
	Transformer	0.835	0.842	0.835	0.838
FedPer	LSTM	0.886	0.886	0.886	0.886
	BiLSTM	0.700	0.490	0.700	0.577
	CNNLSTM	0.812	0.832	0.812	0.817
	Transformer	0.700	0.490	0.700	0.577

These findings show that, for the given prediction task, the choice of aggregation algorithm significantly influences model performance. LSTM-FedAvg stands out as a robust option, particularly for improving prediction accuracy in multistep ranging from 10 minutes to one hour. Additional details regarding the results for all prediction time steps ahead can be found in Appendix 6.

5.4 Case 2: Imbalanced dataset

Table 4 and Table 5 present the performance metrics for various FL aggregation algorithms, including FedAvg, FedProx, SCAFFOLD, and FedPer, across distinct DL models in the scenario of an imbalanced dataset (e.g., high, moderate, low, and variable utilization stations), across the 10 minutes, FedAvg achieves a moderate accuracy range of 0.7 to 0.82. It demonstrates a balanced precision, recall, and F1-Score for LSTM and BiLSTM models. The CNNLSTM model stands out with a higher accuracy range of 0.82 to 0.82, showcasing a well-balanced performance. However, the Transformer model consistently lags with an accuracy of 0.7. FedProx significantly enhances performance, particularly for LSTM and BiLSTM models, achieving accuracy above 0.92. SCAFFOLD and FedPer outperform other aggregators across all models, displaying high accuracy 0.97, precision, recall, F1-Score, and AUC. Moving to the 60-minute prediction horizon, FedProx and FedPer achieve an accuracy ranging from 0.82 to 0.96. SCAFFOLD outperforms FedAvg in most cases. The results underscore the effectiveness of SCAFFOLD and FedPer in handling imbalanced datasets and long-term predictions.

From Figure 10, for a one-time step prediction ahead, AUC achieved with FedPer ranges from 0.92 to 0.96 across the four DL models. The error bars for FedPer, although

Table 4: Performance metrics for FedAvg, FedProx, SCAFFOLD, and FedPer using different models in the imbalanced dataset scenario for 10 minutes ahead.

Aggregators	FL Model	Metrics			
		Accuracy	Precision	Recall	F1-Score
FedAvg	LSTM	0.789	0.824	0.789	0.749
	BiLSTM	0.705	0.774	0.705	0.587
	CNNLSTM	0.820	0.819	0.820	0.820
	Transformer	0.700	0.490	0.700	0.577
FedProx	LSTM	0.924	0.926	0.924	0.922
	BiLSTM	0.961	0.961	0.961	0.961
	CNNLSTM	0.700	0.490	0.700	0.577
	Transformer	0.700	0.490	0.700	0.577
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577
	BiLSTM	**0.967**	**0.967**	**0.967**	**0.967**
	CNNLSTM	0.966	0.966	0.966	0.966
	Transformer	**0.967**	**0.967**	**0.967**	**0.967**
FedPer	LSTM	0.945	0.945	0.945	0.944
	BiLSTM	0.966	0.966	0.966	0.966
	CNNLSTM	0.966	0.966	0.966	0.966
	Transformer	**0.967**	**0.967**	**0.967**	**0.967**

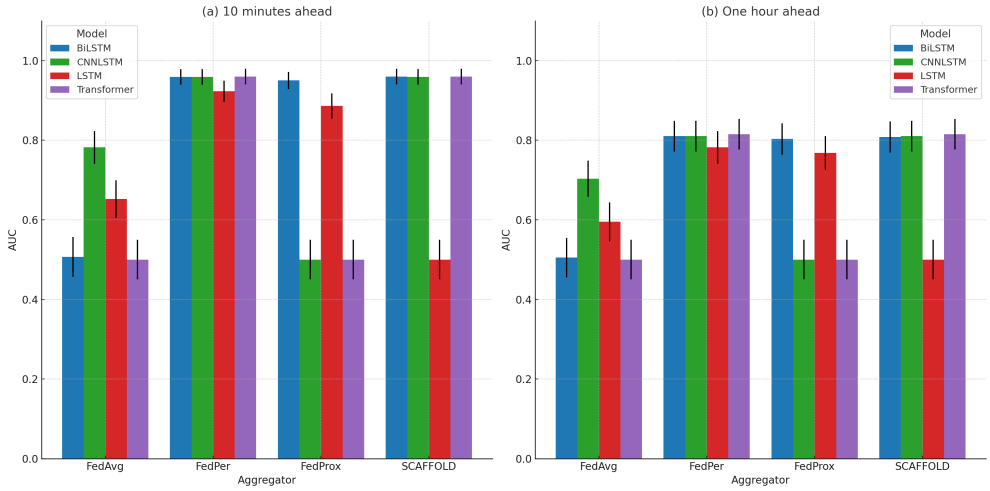


Fig. 10: AUC values with 95% confidence intervals for different Models and Aggregators across varying time intervals within the context of imbalanced dataset.

present, are relatively small, indicating moderate variability in model performance, with a consistent reliability of predictions.

Additionally, SCAFFOLD achieved a high AUC of 0.96 using BiLSTM and Transformer, and the error bars for this algorithm are narrow, suggesting stable performance with low variability across these models. In contrast, FedAvg shows a lower maximum AUC of 0.78 with CNNLSTM, and the larger error bars associated with this

combination reflect a higher level of uncertainty in its predictions, highlighting the inconsistency of FedAvg under this configuration.

Results revealed that FL based on SCAFFOLD and FedPer performs well in handling imbalanced datasets. This could be attributed to SCAFFOLD's utilization of control variates, allowing it to converge to the global model for the union of all CSOs' datasets. Specifically, the control variates in SCAFFOLD will enable the model to incorporate information from the combined datasets of all CSOs and ensure that the global model is not biased towards the dataset of any specific CSO. On the other hand, FedPer's effectiveness can be attributed to its incorporation of personalized layers designed to capture the particular characteristics and patterns within each federated client (i.e., CSO). Incorporating personalized layers allows FedPer to adapt its model parameters to better align with the nuances of individual CSO datasets. These aggregation methods enhance performance in predicting EVCS availability, mainly when dealing with imbalanced data distributions. This is crucial in scenarios where certain classes, such as occupied or unoccupied CSs, may be underrepresented in specific CSOs.

Table 5: Performance metrics for FedAvg, FedProx, SCAFFOLD, and FedPer using different models in the imbalanced dataset scenario for one-hour ahead.

Aggregators	FL Model	Metrics			
		Accuracy	Precision	Recall	F1-Score
FedAvg	LSTM	0.746	0.750	0.746	0.692
	BiLSTM	0.702	0.722	0.702	0.584
	CNNLSTM	0.755	0.753	0.755	0.754
	Transformer	0.700	0.490	0.700	0.577
FedProx	LSTM	0.825	0.820	0.825	0.820
	BiLSTM	0.837	0.836	0.837	0.837
	CNNLSTM	0.700	0.490	0.700	0.577
	Transformer	0.700	0.490	0.700	0.577
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577
	BiLSTM	0.839	0.839	0.839	0.839
	CNNLSTM	**0.842**	**0.842**	**0.842**	**0.842**
	Transformer	0.839	0.841	0.839	0.840
FedPer	LSTM	0.827	0.824	0.827	0.825
	BiLSTM	**0.842**	**0.842**	**0.842**	**0.842**
	CNNLSTM	**0.842**	**0.842**	0.**842**	**0.842**
	Transformer	0.839	0.841	0.839	0.840

Table 6 summarizes the best-performing methods for predicting EVCS occupancy 10 minutes ahead. It highlights the most suitable combinations of DL and FL aggregation strategies tailored to each scenario. The performance metrics—accuracy, F1-Score, and AUC—demonstrate the effectiveness of these methods, achieving values as high as 97% in both cases.

Table 6: Summary of the best FL aggregation methods and DL models across both scenarios.

Scenario	Best DL Model	Best Aggregator	Key Performance
Case 1: Feature distribution skew	LSTM	FedAvg	Accuracy: 97% F1-Score: 97% AUC: 97%
Case 2: Imbalanced dataset	BiLSTM/CNNLSTM	SCAFFOLD/FedPer	Accuracy: 97% F1-Score: 97% AUC: 97%

6 Conclusion

As the adoption of EVs continues to rise, the need for effective management of CS resources becomes paramount. This study addresses the growing demand for public CSs, emphasizing the vital role of occupancy forecasting in enhancing the accessibility of EV charging infrastructure. Our goal is to optimize EVCS utilization through precise availability prediction, ultimately fostering greater convenience for EV users. The paper introduces an efficient Federated Deep Learning framework, focusing primarily on the advantages of preserving data privacy and reducing data transfer. Within the context of FL, multiple heterogeneous CSOs operate as federated clients, collaborating under the supervision of a central server responsible for aggregating their local models. The study leverages a diverse set of aggregation algorithms, including FedAvg, FedProx, SCAFFOLD, and FedPer, to address the significant challenge posed by the real-world scenario of non-IID CS data. Four distinct local DL models, encompassing LSTM, BiLSTM, CNNLSTM, and Transformer, are proposed for EVCS occupancy prediction. The study's evaluation utilises a non-IID dataset derived from the real-world Dundee city EVCS dataset. In the initial scenario, a feature distribution skew is introduced, with each CSO displaying a distinct feature distribution, setting them apart from the other CSOs. The second scenario is characterised by imbalanced class distributions among the CSOs, with some having significantly more or fewer instances for the predicted class, whether 'free' or 'occupied'. These scenarios provide a robust evaluation framework, allowing for a comprehensive assessment of the models' performance in addressing real-world challenges associated with EVCS availability prediction.

The results underscore the effectiveness of FedPer and SCAFFOLD-based FL in handling imbalanced datasets. In contrast, FedAvg excels when dealing with data characterized by skewed feature distributions, particularly for forecasts one hour ahead based on data sampled at 10-minute intervals. Additionally, the analysis using Random Forest and SHAP values indicates that including lagged data significantly improves multistep predictions. However, features such as time of the day, day of the week, and weekend showed a weaker impact on the model's predictions. As these features contribute less substantially to the model's predictions in comparison to the more

influential features, this could justify the satisfactory results of using FedAvg in the context of skewed datasets, as described in scenario one.

Future work in this domain can explore several research directions. Firstly, further research could focus on refining the FL process by optimizing communication overhead, a critical aspect of real-world applications. Secondly, extending the study to include more diverse and larger datasets from various cities or regions could provide a more comprehensive evaluation of the proposed methods' generalizability and robustness. Additionally, integrating predictive occupancy models with an EVCS recommender system could enhance the overall user experience by providing drivers with personalized recommendations for available CSs. Finally, the predictive models developed in this study could be leveraged to improve the management and maintenance planning of CSs, enabling CSOs to make data-driven decisions for optimizing CS utilization.

Another important direction for improvement lies in incorporating uncertainty quantification into the FL process. In future work, we plan to explore Bayesian Federated Deep Learning, which incorporates uncertainty quantification. This approach will enable not only a more comprehensive evaluation of model performance but also a deeper understanding of the uncertainty in predictions. By capturing uncertainties, this method will offer more robust insights into the accuracy, precision, recall, and F1 scores, including their variances. Compared to conventional Federated Deep Learning, Bayesian FL provides the added advantage of quantifying uncertainty, which is particularly valuable in applications like EVCS occupancy prediction where the reliability of model predictions directly influences operational decisions. By offering more precise information about model confidence, this approach can significantly enhance decision-making processes, improving the efficiency of grid management and optimizing CS availability in real-world, dynamic environments.

Acknowledgment

This paper is supported by the OPEVA project that has received funding within the Chips Joint Undertaking (Chips JU) from the European Union's Horizon Europe Programme and the National Authorities (France, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Chips JU.

References

[1] Koengkan, M., Fuinhas, J.A., Belucio, M., Alavijeh, N.K., Salehnia, N., Machado, D., Silva, V., Dehdar, F.: The impact of battery-electric vehicles on energy consumption: A macroeconomic evidence from 29 european countries. World Electric Vehicle Journal **13**(2), 36 (2022)

[2] Xu, X., Niu, D., Li, Y., Sun, L.: Optimal pricing strategy of electric vehicle charging station for promoting green behavior based on time and space dimensions. Journal of Advanced Transportation **2020**, 1–16 (2020)

[3] Saputra, Y.M., Hoang, D.T., Nguyen, D.N., Dutkiewicz, E., Mueck, M.D., Srikanteswara, S.: Energy demand prediction with federated learning for electric vehicle networks. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2019). IEEE

[4] Li, Y., Wang, J., Wang, W., Liu, C., Li, Y.: Dynamic pricing based electric vehicle charging station location strategy using reinforcement learning. Energy **281**, 128284 (2023)

[5] Xia, F., Chen, H., Chen, L., Qin, X.: A hierarchical navigation strategy of ev fast charging based on dynamic scene. IEEE Access **7**, 29173–29184 (2019)

[6] Shi, X., Xu, Y., Guo, Q., Sun, H., Gu, W.: A distributed ev navigation strategy considering the interaction between power system and traffic network. IEEE Transactions on Smart Grid **11**(4), 3545–3557 (2020)

[7] Saha, P.K., Chakraborty, N., Mondal, A., Mondal, S.: Optimal sizing and efficient routing of electric vehicles for a vehicle-on-demand system. IEEE Transactions on Industrial Informatics **18**(3), 1489–1499 (2021)

[8] Charly, A., Thomas, N.J., Foley, A., Caulfield, B.: Identifying optimal locations for community electric vehicle charging. Sustainable Cities and Society **94**, 104573 (2023)

[9] Xie, M., Wang, H., Gao, Y., Wang, Y.: A rolling-horizon framework for managing shared parking and electric vehicle charging. Sustainable Cities and Society **98**, 104810 (2023)

[10] Diaz-Cachinero, P., Muñoz-Hernandez, J.I., Contreras, J.: Integrated operational planning model, considering optimal delivery routing, incentives and electric vehicle aggregated demand management. Applied Energy **304**, 117698 (2021)

[11] Kaya, Ö., Alemdar, K.D., Atalay, A., Çodur, M.Y., Tortum, A.: Electric car sharing stations site selection from the perspective of sustainability: A gis-based multi-criteria decision making approach. Sustainable Energy Technologies and Assessments **52**, 102026 (2022)

[12] Luo, R., Zhang, Y., Zhou, Y., Chen, H., Yang, L., Yang, J., Su, R.: Deep learning approach for long-term prediction of electric vehicle (ev) charging station availability. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 3334–3339 (2021). IEEE

[13] Ma, T.-Y., Faye, S.: Multistep electric vehicle charging station occupancy prediction using hybrid lstm neural networks. Energy **244**, 123217 (2022)

[14] Sao, A., Tempelmeier, N., Demidova, E.: Deep information fusion for electric vehicle charging station occupancy forecasting. In: 2021 IEEE International Intelligent

Transportation Systems Conference (ITSC), pp. 3328–3333 (2021). IEEE

[15] Douaidi, L., Senouci, S.-M., El Korbi, I., Harrou, F.: Predicting electric vehicle charging stations occupancy: A federated deep learning framework. In: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), pp. 1–5 (2023). IEEE

[16] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)

[17] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143 (2020). PMLR

[18] Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)

[19] Luo, Z., Song, Y., Hu, Z., Xu, Z., Yang, X., Zhan, K.: Forecasting charging load of plug-in electric vehicles in china. In: 2011 IEEE Power and Energy Society General Meeting, pp. 1–8 (2011). IEEE

[20] Lee, Z.J., Li, T., Low, S.H.: Acn-data: Analysis and applications of an open ev charging dataset. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems, pp. 139–149 (2019)

[21] Gruosso, G., Mion, A., Gajani, G.S.: Forecasting of electrical vehicle impact on infrastructure: Markov chains model of charging stations occupation. ETransportation **6**, 100083 (2020)

[22] Dastpak, M., Errico, F., Jabali, O., Malucelli, F.: Dynamic routing for the electric vehicle shortest path problem with charging station occupancy information. arXiv preprint arXiv:2305.11773 (2023)

[23] Masum, S., Liu, Y., Chiverton, J.: Multi-step time series forecasting of electric load using machine learning models. In: Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, 2018, Proceedings, Part I 17, pp. 148–159 (2018). Springer

[24] Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparison of arima and lstm in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394–1401 (2018). Ieee

[25] Straka, M., De Falco, P., Ferruzzi, G., Proto, D., Van Der Poel, G., Khormali, S., Buzna, L.: Predicting popularity of electric vehicle charging infrastructure in urban context. IEEE Access **8**, 11315–11327 (2020)

[26] Motz, M., Huber, J., Weinhardt, C.: Forecasting bev charging station occupancy

at work places. In: GI-Jahrestagung, pp. 771–781 (2020)

[27] Soldan, F., Bionda, E., Mauri, G., Celaschi, S.: Short-term forecast of ev charging stations occupancy probability using big data streaming analysis. arXiv preprint arXiv:2104.12503 (2021)

[28] Luo, R., Song, Y., Huang, L., Zhang, Y., Su, R.: Ast-gin: Attribute-augmented spatiotemporal graph informer network for electric vehicle charging station availability forecasting. Sensors **23**(4), 1975 (2023)

[29] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR

[30] Wang, X., Zheng, X., Liang, X.: Charging station recommendation for electric vehicle based on federated learning. In: Journal of Physics: Conference Series, vol. 1792, p. 012055 (2021). IOP Publishing

[31] Savi, M., Olivadese, F.: Short-term energy consumption forecasting at the edge: A federated learning approach. IEEE Access **9**, 95949–95969 (2021)

[32] Shanmuganathan, J., Victoire, A.A., Balraj, G., Victoire, A.: Deep learning lstm recurrent neural network model for prediction of electric vehicle charging demand. Sustainability **14**(16), 10207 (2022)

[33] Tun, Y.L., Thar, K., Thwal, C.M., Hong, C.S.: Federated learning based energy demand prediction with clustered aggregation. In: 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 164–167 (2021). IEEE

[34] Liu, L., Xi, Z., Zhu, K., Wang, R., Hossain, E.: Mobile charging station placements in internet of electric vehicles: A federated learning approach. IEEE Transactions on Intelligent Transportation Systems **23**(12), 24561–24577 (2022)

[35] Sun, C., Huang, C., Shou, B., Huang, J.: Federated learning in competitive ev charging market. arXiv preprint arXiv:2310.08794 (2023)

[36] Basmadjian, R.: Communication vulnerabilities in electric mobility hcp systems: a semi-quantitative analysis. Smart Cities **4**(1), 405–428 (2021)

[37] Liu, J., Huang, J., Zhou, Y., Li, X., Ji, S., Xiong, H., Dou, D.: From distributed machine learning to federated learning: A survey. Knowledge and Information Systems **64**(4), 885–917 (2022)

[38] Ho, Y., Wookey, S.: The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. IEEE access **8**, 4806–4813 (2019)

[39] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation

9(8), 1735–1780 (1997)

[40] Kiperwasser, E., Goldberg, Y.: Simple and accurate dependency parsing using bidirectional lstm feature representations. Transactions of the Association for Computational Linguistics **4**, 313–327 (2016)

[41] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems **28** (2015)

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[43] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv preprint arXiv:1806.00582 (2018)

[44] Zhu, H., You, X., Liu, S.: Multiple ant colony optimization based on pearson correlation coefficient. Ieee Access **7**, 61628–61638 (2019)

[45] Disha, R.A., Waheed, S.: Performance analysis of machine learning models for intrusion detection system using gini impurity-based weighted random forest (giwrf) feature selection technique. Cybersecurity **5**(1), 1 (2022)

[46] Nohara, Y., Matsumoto, K., Soejima, H., Nakashima, N.: Explanation of machine learning models using improved shapley additive explanation. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 546–546 (2019)

[47] Murphy, K.P.: Machine Learning: a Probabilistic Perspective. MIT press, Cambridge, MA (2012)

Appendix

This section provides detailed insights into the results obtained using various metrics across both prediction scenarios and for all time intervals.

Table (6) Performance metrics in the feature distribution skew scenario for (a) 20 min and (b) 30 min (c) 40 min (d) 50 min ahead.

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	**0.954**	**0.954**	**0.954**	**0.954**	**0.946**
	BiLSTM	0.936	0.936	0.936	0.936	0.923
	CNNLSTM	0.936	0.936	0.936	0.936	0.923
	Transformer	0.685	0.827	0.685	0.695	0.764
FedProx	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.936	0.936	0.936	0.936	0.923
	CNNLSTM	0.936	0.936	0.936	0.936	0.923
	Transformer	0.936	0.936	0.936	0.936	0.923
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.738	0.840	0.738	0.748	0.800
	CNNLSTM	0.833	0.830	0.833	0.830	0.785
	Transformer	0.936	0.936	0.936	0.936	0.923
FedPer	LSTM	0.941	0.941	0.941	0.941	0.933
	BiLSTM	0.700	0.490	0.700	0.577	0.500
	CNNLSTM	0.936	0.936	0.936	0.936	0.923
	Transformer	0.700	0.490	0.700	0.577	0.500

(a)

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	**0.943**	**0.943**	**0.943**	**0.943**	**0.933**
	BiLSTM	0.908	0.908	0.908	0.908	0.890
	CNNLSTM	0.908	0.908	0.908	0.908	0.890
	Transformer	0.673	0.815	0.673	0.683	0.751
FedProx	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.909	0.909	0.909	0.909	0.890
	CNNLSTM	0.908	0.908	0.908	0.908	0.888
	Transformer	0.908	0.908	0.908	0.908	0.890
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.726	0.828	0.726	0.737	0.786
	CNNLSTM	0.824	0.820	0.824	0.821	0.774
	Transformer	0.908	0.908	0.908	0.908	0.890
FedPer	LSTM	0.921	0.922	0.921	0.922	0.911
	BiLSTM	0.700	0.490	0.700	0.577	0.500
	CNNLSTM	0.908	0.909	0.908	0.908	0.893
	Transformer	0.700	0.490	0.700	0.577	0.500

(b)

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	**0.936**	**0.936**	**0.936**	**0.936**	**0.923**
	BiLSTM	0.884	0.884	0.884	0.884	0.860
	CNNLSTM	0.884	0.883	0.884	0.884	0.860
	Transformer	0.661	0.802	0.661	0.672	0.737
FedProx	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.885	0.885	0.885	0.885	0.861
	CNNLSTM	0.883	0.882	0.883	0.882	0.856
	Transformer	0.883	0.882	0.883	0.882	0.860
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.717	0.815	0.717	0.729	0.773
	CNNLSTM	0.815	0.810	0.815	0.811	0.761
	Transformer	0.883	0.883	0.883	0.883	0.860
FedPer	LSTM	0.906	0.907	0.906	0.906	0.892
	BiLSTM	0.700	0.490	0.700	0.577	0.500
	CNNLSTM	0.869	0.875	0.869	0.871	0.859
	Transformer	0.700	0.490	0.700	0.577	0.500

(c)

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	**0.929**	**0.929**	**0.929**	**0.929**	**0.915**
	BiLSTM	0.862	0.860	0.862	0.861	0.829
	CNNLSTM	0.863	0.862	0.863	0.863	0.835
	Transformer	0.650	0.790	0.650	0.661	0.724
FedProx	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.864	0.863	0.864	0.863	0.834
	CNNLSTM	0.862	0.862	0.862	0.862	0.835
	Transformer	0.860	0.860	0.860	0.860	0.832
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.708	0.804	0.708	0.720	0.760
	CNNLSTM	0.806	0.800	0.806	0.801	0.748
	Transformer	0.858	0.860	0.858	0.859	0.838
FedPer	LSTM	0.895	0.895	0.895	0.895	0.877
	BiLSTM	0.700	0.490	0.700	0.577	0.500
	CNNLSTM	0.839	0.850	0.839	0.842	0.833
	Transformer	0.700	0.490	0.700	0.577	0.500

Table (7) Performance metrics in the imbalanced dataset scenario for (a) 20 min and (b) 30 min (c) 40 min (d) 50 min ahead.

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	0.776	0.805	0.776	0.732	0.636
	BiLSTM	0.704	0.757	0.704	0.586	0.506
	CNNLSTM	0.805	0.804	0.805	0.805	0.765
	Transformer	0.700	0.490	0.700	0.577	0.500
FedProx	LSTM	0.898	0.898	0.898	0.895	0.853
	BiLSTM	0.931	0.931	0.931	0.931	0.914
	CNNLSTM	0.700	0.490	0.700	0.577	0.500
	Transformer	0.700	0.490	0.700	0.577	0.500
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	**0.936**	**0.936**	**0.936**	**0.936**	**0.923**
	CNNLSTM	**0.936**	0.935	**0.936**	**0.936**	0.922
	Transformer	**0.936**	**0.936**	**0.936**	**0.936**	**0.923**
FedPer	LSTM	0.916	0.916	0.916	0.915	0.889
	BiLSTM	**0.936**	0.935	**0.936**	**0.936**	0.922
	CNNLSTM	**0.936**	0.935	**0.936**	**0.936**	0.922
	Transformer	**0.936**	**0.936**	**0.936**	**0.936**	**0.923**

(a)

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	0.766	0.789	0.766	0.719	0.622
	BiLSTM	0.703	0.745	0.703	0.585	0.506
	CNNLSTM	0.790	0.789	0.790	0.790	0.746
	Transformer	0.700	0.490	0.700	0.577	0.500
FedProx	LSTM	0.878	0.877	0.878	0.875	0.831
	BiLSTM	0.904	0.903	0.904	0.903	0.882
	CNNLSTM	0.700	0.490	0.700	0.577	0.500
	Transformer	0.700	0.490	0.700	0.577	0.500
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	**0.908**	**0.908**	**0.908**	**0.908**	**0.890**
	CNNLSTM	**0.908**	**0.908**	**0.908**	**0.908**	0.889
	Transformer	**0.908**	**0.908**	**0.908**	**0.908**	**0.890**
FedPer	LSTM	0.890	0.889	0.890	0.889	0.857
	BiLSTM	**0.908**	**0.908**	**0.908**	**0.908**	0.889
	CNNLSTM	**0.908**	**0.908**	**0.908**	**0.908**	0.889
	Transformer	**0.908**	**0.908**	**0.908**	**0.908**	**0.890**

(b)

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	0.760	0.775	0.760	0.712	0.615
	BiLSTM	0.703	0.740	0.703	0.584	0.505
	CNNLSTM	0.778	0.776	0.778	0.777	0.732
	Transformer	0.700	0.490	0.700	0.577	0.500
FedProx	LSTM	0.860	0.857	0.860	0.856	0.811
	BiLSTM	0.880	0.879	0.880	0.879	0.853
	CNNLSTM	0.700	0.490	0.700	0.577	0.500
	Transformer	0.700	0.490	0.700	0.577	0.500
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	**0.883**	0.882	**0.883**	0.882	0.860
	CNNLSTM	**0.883**	**0.883**	**0.883**	**0.883**	0.860
	Transformer	**0.883**	**0.883**	**0.883**	**0.883**	**0.861**
FedPer	LSTM	0.867	0.865	0.867	0.865	0.829
	BiLSTM	**0.883**	**0.883**	**0.883**	**0.883**	0.860
	CNNLSTM	**0.883**	**0.883**	**0.883**	**0.883**	0.860
	Transformer	**0.883**	**0.883**	**0.883**	**0.883**	**0.861**

(c)

Aggregators	FL Model	Metrics				
		Accuracy	Precision	Recall	F1-Score	AUC
FedAvg	LSTM	0.750	0.761	0.750	0.697	0.600
	BiLSTM	0.703	0.725	0.703	0.584	0.505
	CNNLSTM	0.766	0.764	0.766	0.765	0.717
	Transformer	0.700	0.490	0.700	0.577	0.500
FedProx	LSTM	0.841	0.837	0.841	0.837	0.787
	BiLSTM	0.857	0.856	0.857	0.856	0.826
	CNNLSTM	0.700	0.490	0.700	0.577	0.500
	Transformer	0.700	0.490	0.700	0.577	0.500
SCAFFOLD	LSTM	0.700	0.490	0.700	0.577	0.500
	BiLSTM	0.860	0.860	0.860	0.860	0.832
	CNNLSTM	**0.861**	**0.861**	**0.861**	**0.861**	0.833
	Transformer	0.859	0.860	0.859	0.860	**0.836**
FedPer	LSTM	0.846	0.843	0.846	0.844	0.805
	BiLSTM	**0.861**	**0.861**	**0.861**	**0.861**	0.833
	CNNLSTM	**0.861**	**0.861**	**0.861**	**0.861**	0.833
	Transformer	0.859	0.860	0.859	0.860	**0.836**