Flexible energy systems Leveraging the Optimal integration of EVs deployment Wave

Grant Agreement Nº: 101056730

Deliverable 4.2

Forecasting tool and algorithm advancement

Author(s): Ferran Pinsach, Roger Valdés, Lucía Igualada, Josh Eichman, Shahid Hussain, Fabiano Pallonetto, Ruisong Han, and Scott McDonald

Website FLOW

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Document information Table

Project Data			
Project Acronym	FLOW		
Project Title	Flexible energy system Leveraging the Optimal integration of EVs deployment Wave		
Grant Agreement n.	101056730		
Topic identifier	HORIZON-CL5-2021-D5-01-03		
Funding Scheme	RIA		
Project duration	48 months		
Coordinator	Catalonia Institute for Energy Research		
Website	https://www.theflowproject.eu/		
Deliverable Document Sheet			
Deliverable No.	D4.2		
Deliverable title	Forecasting tool and algorithm advancement		
Description	Describe and demonstrate the ability for advanced forecasting algorithms to improve management of e-mobility solutions focusing on three areas: 1) forecasting of electricity prices for e-mobility, 2) week-ahead EV energy consumption forecasts to support drivers and 3) charging demand forecasts at the station for a CSO/CPO.		
WP No.	4		
Related task	T4.2		
Lead beneficiary	IREC		
Author(s)	Ferran Pinsach [IREC], Roger Valdés [IREC], Lucía Igualada [IREC], Josh Eichman [IREC], Shahid Hussain [NUIM] and Fabiano Pallonetto [NUIM], Ruisong Han [EATON], and Scott McDonald [EATON]		
Contributor(s)	IREC, NUIM, EATON		
Type	Report		
Dissemination L.	Public		
Due Date	30/10/2023	Submission Date	22/12/2023

Funded by the European Union

Version	Date	Author(s)	Organisation(s)	Comments
V0.1	02/10/2023	Josh Eichman, Ferran Pinsach, Roger Valdés, and Lucía Igualada	IREC	Creation of the document, define document structure, optimization draft
V0.2	28/11/2023	Shahid Hussain and Fabiano Pallonetto	NUIM	Add electricity price for e-mobility forecast activities
V0.3	01/12/2023	Ruisong Han, and Scott McDonald	EATON	Add description of demand forecast activities
V0.4	04/12/2023	Roger Valdés, Ferran Pinsach, Lucía Igualada, and Josh Eichman	IREC	Add user location and energy consumption forecast and optimization model
V0.5	05/12/2023	Josh Eichman	IREC	Review
V0.6	19/12/2023	Alberto Danese, Ea onsmark, Torben Fog, Clara Marie Zacho Hansen, Sarah Cnockert	IREC, R2M, SPIRII	Internal FLOW Review
V0.7	21/12/2023	Ferran Pinsach, Roger Valdés, Shahid Hussain, Ruisong Han, and Scott McDonald	IREC, NUIM, EATON	Update report based on recommendations
V1.0	22/12/2023	Josh Eichman, Sarah Cnockert	IREC	Final review and submission on the EU portal

Funded by the European Union

DISCLAIMER

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Funded by the European Union

Table of contents

Funded by the European Union

Funded by the European Union

Funded by
the European Union

List of Acronyms

Acronym	Meaning
EV	Electric Vehicle
PHEV	Plug-in Hybrid Electric Vehicle
DC	Direct Current
AC	Alternate Current
WP	Work Package
ML	Machine Learning
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory
MILP	Mixed Integer Linear Programming
RF	Random Forest
SVR	Support Vector Regressor
SVM	Support Vector Machine

Table of Tables

Funded by the European Union

Table of Figures

Funded by the European Union

Funded by the European Union

Executive Summary

With a growing number of EVs, there is also an increase in the demand for electricity. If managed appropriately, EVs have the potential to enhance demand-side flexibility and improve grid operations while also reducing emissions. Methods and algorithms to optimize EV charging can rely heavily on the forecast of energy prices, travel patterns, and charging demand. This report implements new and improved methods for 1) forecasting energy prices using a deep-weighted ensemble model, 2) demand forecasting from a user perspective by providing a week-ahead forecast of travel patterns and energy consumption and 3) demand forecasting algorithms from the charge station perspective.

Funded by
the European Union

1. Introduction

The current transportation sector, primarily reliant on fossil fuels and responsible for approximately 15.00% of global energy-related emissions, where electric vehicles are poised to play a pivotal role in enabling the decarbonization of road transport [1]. In contrast to conventional fossil fuelled internal combustion engine vehicles (ICEVs), EVs offer various advantages, including but not limited to zero tailpipe emissions, independence from petroleum reliance, enhanced fuel efficiency, reduced maintenance requirements, and an enhanced driving experience characterized by improved acceleration, noise reduction, and the convenience of home and opportunity recharging [2]. Furthermore, considering the constrained availability of alternative options for liquid fossil fuels, EVs emerge as a viable avenue for mitigating overall greenhouse gas (GHG) emissions and facilitating the decarbonization of on-road transportation, particularly when charged with electricity from clean sources [3]. Consequently, the transportation landscape is rapidly evolving witnessing a growing acceptance of both plug-in hybrid electric vehicles (PHEVs) and battery electric vehicles (BEVs), heralding an anticipated widespread integration of electric vehicles (EVs) on roadways in the foreseeable future [4]. For instance, the global EV market has witnessed remarkable growth, surging from 0.72 million units in 2015 to 4.79 million units by 2019 [5]. Projections indicate a substantial future expansion, with the number of EVs on the road expected to surge by an impressive 36%, reaching an estimated 245 million by the year 2030 [6].

This trend underscores the accelerating adoption and promising trajectory of EVs worldwide is a pivotal shift towards sustainability and addressing various environmental concerns [7]. The trajectory of growing EV numbers directly contributes to mitigating challenges associated with traditional vehicles, notably reducing emissions that contribute to air pollution and climate change [8]. Furthermore, the quiet operation of EVs compared to traditional vehicles addresses the issue of noise pollution, particularly in urban environments [9]. The lower operational costs associated with EVs, including reduced fuel expenses and maintenance requirements, contribute to economic sustainability [10]. Moreover, recognized for their environmental friendliness, low fuel costs, safety, reliability, compact design, and lightweight construction, EVs also serve as distributed storage, supporting power grids and microgrids, particularly during peak demand, through innovative Vehicle-to-Grid (V2G) technology [11].

However, the rapid increase in EV numbers, coupled with their large-scale penetration, imposes a significant burden on the power grid due to additional power demand [12]. This influx may lead to transformer overloads, feeder congestion, circuit faults, and overall grid instability, posing challenges to the power supply infrastructure [13]. The rise in EV charging poses substantial challenges to electricity power infrastructure, exerting influence on overall power demand and altering its shape, particularly during peak demand periods [14]. Given the unanticipated increase in electric demand has profound impacts on electricity generation, transmission, and distribution infrastructures and there is a pressing need for coordinated control of EV charging loads at the distribution level to effectively manage and mitigate these impacts [15].

The methods and algorithms required to optimize EV charging for these systems are available, and while there is room for improvement, the outcome of the optimizations rely heavily on the forecast of energy prices, travel patterns, and charging demand. This report focuses on exploring new and improved methods for performing this forecast including 1) forecasting of electricity prices for e-

Funded by the European Union

mobility, 2) week-ahead EV energy consumption forecasts to support drivers and 3) charging demand forecasts at the station for a CSO/CPO. The report is separated into three sections to align with those three topics:

1. Section 2 examines methods for forecasting energy prices focusing on using a deep-weighted ensemble model. The goal of this activity is to improve the price forecasting for participation in arbitrage or ancillary service markets.
2. Section 3 focuses on demand forecasting from a user perspective by providing a week-ahead forecast of travel patterns and energy consumption to enable user-centric smart charging. The goal is to enable users to automatically select their desired SOC for use with smart charging based on their historical usage patterns and energy prices.
3. Section 4 develops demand forecasting algorithms from the charge station perspective. This work focuses on forecasts the day-ahead building EV charging demand for use in EATON's building energy management system.

2. Energy prices forecasting through deep-weighted ensemble model (NUIM)

The power grid incentivizes energy aggregators to engage in demand response, enabling them to manage and shift their charging loads from on-peak to off-peak periods, leveraging electricity prices and the dwell time of EVs while adhering to grid operational constraints [16]. In most cases, the utility companies offer diverse tariff structures, including peak, mid-peak, and off-peak rates within a time-of-use (TOU) tariff system, providing fixed prices for specific time intervals [17]. However, while TOU fixed rates and timing-based tariff systems are effective for controlling individual EV charging at residential premises, they pose challenges. The awareness of individual customers about low peak timings can lead to a herding problem, as they may rush to charge their EVs during those periods, making TOU fixed rates and timing-based tariff systems less suitable for aggregators [18].

The efficiency of charging control by aggregators is highly dependent on advanced knowledge of electricity prices [19]. Price forecasting serves as an essential tool, enabling aggregators to optimize their participation in demand response programs. This, in turn, allows them to effectively manage charging loads while meeting the requirements of EV users and maximizing overall benefits [20]. Considering the fluctuating nature of electricity prices, traditional statistical methods, which often rely on average cases, prove unsuitable for accurately forecasting the dynamic electricity price variations [21]. As a result, these methods may not provide the precise price knowledge essential for aggregators [22]. In contrast, machine learning algorithms excel in discerning intricate price patterns, offering energy aggregators nuanced pricing information [23]. This capability plays a pivotal role in efficiently managing charging loads at the aggregator level, contributing to enhanced system efficiency and adaptability [24]. Electricity prices are characterized as time series data and present challenges for accurate forecasting due to their intricate and fluctuating nature based on temporal dependencies [25]. Ensemble Long Short-Term Memory (LSTM) models emerge as critical tools in managing the complexities of such price variations and exhibit superior performance in accurately forecasting price patterns compared to simplified LSTMs, ML-based regression, and other deep learning models [26]. Ensemble LSTM models represent a specialized form of ensemble learning, utilizing multiple LSTM

Funded by the European Union

networks to collectively make predictions [27] where the LSTM, is a recurrent neural network (RNN) architecture designed to capture and learn long-term dependencies in sequential data [28]. The ensemble strategy involves training diverse LSTM models with varied initializations or architectures and then combining their predictions to enhance overall performance [29]. The challenge in constructing Ensemble LSTM models lies in determining optimal weights for each individual model within the ensemble, where the weights dictate the contribution of each model to the final prediction [30]. The process of finding optimal weights entails minimizing a loss function that quantifies the disparity between the ensemble's predictions and the actual outcomes. However, achieving precise and accurate optimal weights in ensemble LSTM models is challenging as it involves maintaining effective contribution factors from each model within the ensemble, striking a delicate balance for accurate predictions [31]. To address the existing knowledge gap, this paper introduces a Deep-Weighted Ensemble Model (DWEM) for wholesale electricity price forecasting and establishes a heuristic mechanism for determining optimal weights within the ensemble model. Our contributions can be summarized in three key aspects.

- We presented a novel Deep-Weighted Ensemble Model for Wholesale Electricity Price Forecasting, with the aim of optimizing EV charging at the aggregator level. Leveraging a carefully formulated representation of wholesale electricity prices, our approach employed both standard and stacked Long Short-Term Memory networks as building blocks of the proposed deep-weighted ensemble model. The ensemble paradigm was strategically used to harness the complementary strengths of diverse models, and the deep-weighting mechanism ensured an adaptive aggregation of predictions.
- In our ensemble construction process, we introduced a mechanism for determining optimal weights, utilizing a heuristic approach, which evaluated a diverse range of weight configurations, assessing each configuration's accuracy. The heuristic method employed a systematic exploration of the weight space, considering various combinations to identify those characterized by the highest levels of accuracy. This selection process employed ensures that only the most precise configurations are incorporated into the ultimate ensemble model. Through the implementation of this heuristic mechanism, our contribution not only bolsters the resilience of the ensemble but also establishes a systematic approach to determining weights and is pivotal for the precision and dependability of the Wholesale Electricity Price Forecasting model, especially in the context of optimizing Electric Vehicle charging at the aggregator level.
- We implemented our proposed model on a publicly available dataset sourced from the Houston electricity dataset. To enhance the dataset, we conducted a detailed data engineering process, incorporating a correlation matrix for feature selection and employing one-hot encoding to handle various label features. Subsequently, our developed model was applied to this refined dataset, and we conducted a series of comparative case studies, considering scenarios with and without outliers. Furthermore, we conducted benchmark analyses by comparing our results with those obtained from state-of-the-art models, such as XGBoost, Light Gradient Boosting, Linear Regression, Facebook's Prophet model, Fully Connected Neural Network, and LSTM models.

Funded by the European Union

2.1. State of the Art

With the proliferation of EVs and their consequential impact on the electric power generation, transmission, and distribution systems, researchers have increasingly directed their attention to the challenges posed by the substantial load induced by EV charging [31]. This has prompted a surge in scholarly exploration, leading to a comprehensive examination of the electric charging load predicament across private, semi-public, and public charging infrastructures in recent literature [32]. In addressing the divergent needs of the power grid, which seeks to minimize charging load, and EV users, who prioritize reducing both charging and waiting times, the authors in [33] devised a fuzzy inference-based mechanism. This approach is designed to optimize the charging and waiting times for a collective group of EVs within a parking lot while concurrently accommodating the constraints set by the power grid. The authors proposed a two-stage bi-layer game charging optimization model in [34] to address the non-coordination among a network operator, a distributed generation operator, and a charging agent. The first stage utilized a dynamic virtual price-based demand response model for pre-optimizing charging loads, leading to a significant reduction in energy abandonment and net load fluctuation. In the second stage, a bi-layer Stackelberg game model was introduced, allowing participants independent decision-making and achieving optimal comprehensive benefits in a multi-participant charging system. In our earlier study [35], we addressed challenges related to fixed-timing EV charging by developing a Charging Cost Optimization Algorithm (CCOA). This heuristic algorithm learns real-time price patterns and EV information to optimize charging loads and costs in residential settings. Simulation experiments were conducted to compare various charging scenarios, including both individual and aggregated charging models. These scenarios were contrasted against uncoordinated charging, fixed-rate charging, and coordinated time-of-use charging methodologies. The evaluation criteria centred on assessing the impact on the power grid in terms of potential overloading and analysing the associated charging costs. The study [36] introduced a cooperative energy management strategy that facilitates the sharing of energy among end-users, particularly focusing on intelligent charging and discharging of Electric Vehicles (EVs) for Vehicle-to-Anything (V2X) and Anything-to-Vehicle (X2V) modes. The proposed method employed a Mixed-Integer Programming (MIP) approach and utilized a robust Gurobi optimizer within a generic framework for Cooperative Power Management (CPM). The CPM ensured a target state of charge (SoC) at departure for all vehicles without causing a rebound peak in total grid power, even in the absence of photovoltaic power. The model includes two methods: the first involving one-way power flow and the second introducing two-way power flow, enabling vehicle-to-vehicle or vehicle-to-loads modes. Their analysis demonstrated the model's effectiveness in creating a robust and efficient charging and discharging schedule for multiple EVs, aligning with the sharing economy concept, reducing peak power demands, and enhancing user comfort. In another study [37], we introduced a Two-Layer Decentralized Charging Approach (TLDCA) using fuzzy data fusion to optimize the charging cost of residential EVs. The TLDCA addressed a fuzzy objective function through fuzzy integer linear programming. This approach considered multiple day-ahead price patterns and state-of-charge inputs, determining the optimal charging schedule to reduce costs and peak-to-average ratio. Simulations demonstrated the TLDCA's effectiveness in comparison to uncoordinated charging, standard-rate charging, and time-of-use charging schemes. The Two-Layer Decentralized Charging Approach (TLDCA) was improved with the development of a hybrid coordination scheme (HCS) [17] for EV charging in residential areas. This improvement addressed challenges associated with herding and user satisfaction in both centralized and decentralized charging approaches. The HCS incorporated a fuzzy inference mechanism to

Funded by the European Union

optimize peak load, mitigate herding issues, and reduce charging costs. Utilizing the IEEE 34 bus system for two case studies, the proposed hybrid coordination scheme demonstrated superior performance compared to alternative charging strategies, including uncoordinated charging, standard-rate charging, time-of-use charging, and two-layer decentralized approaches. Considering the impact of aggregated charging loads on the distribution network, the optimization of charging loads in a smart parking lot becomes crucial and the implementation of efficient charging strategies in smart parking lots is essential for maintaining grid stability and ensuring user satisfaction. It plays a significant role in reducing overall aggregated charging loads while satisfying the charging demands of EV users at the time of their departure. Considering the impact of aggregated charging loads on the distribution network, optimizing charging loads in a smart parking lot is crucial. Numerous studies have implemented effective charging strategies [38] that consider uncertain user behaviour, including arrival and departure patterns, battery capacities, and required energy for the next trip based on distance [39], as well as grid power availability [40].

These approaches ensured grid stability and user satisfaction by significantly contributing to reducing overall aggregated charging loads while meeting the charging demands of EV users upon departure In both residential and aggregated parking lots, these studies [17] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] have primarily considered day-ahead price patterns by assuming that the current day's prices follow the same pattern as the previous day. However, in real-world scenarios, prices are dependent on the electric load pattern and may differ from the previous day [41]. Consequently, these studies may lack robustness in optimizing charging costs and loads and necessities for predictive machine learning-based models that can adapt to dynamic price patterns and provide more accurate and responsive optimization in response to real-time variations in electricity prices and load patterns [42].

To fill-up the gap, in a recent study [43], a multi-bi-forecasting system is presented, incorporating multivariable and multi-input multi-output structures. The developed system adeptly manages high-frequency electricity price and load data, employing a multivariable arrangement for forecasting and a multi-input multi-output structure featuring three member models. The achieved results, obtained through a unified strategy leveraging the multi-objective Salp swarm algorithm, showcase superior forecasting capabilities for both point and interval forecasting. This is substantiated by quantitative assessments conducted in the Australian electricity market. The study in [23] introduced a data-driven demand-side management approach for a solar- powered EV charging station (CS) connected to a micro- grid. Their approach leveraged the station to addressed peak demand by compensating for energy requirements, reducing reliance on conventional sources. Real-time data from PV power stations, commercial and residential loads, and EVCSs were used for simulations. A deep learning approach was developed for energy supply control and off- peak hour charging, while two machine learning methods were compared for energy storage system state of charge estimation. The 24-hour case study demonstrated that the EVCS effectively compensated for peak demand. Amid the growing adoption of EVs, the research outlined in [44] addresses the need for effective charge management systems to anticipate peak loads in charging infrastructure. The study evaluates multiple machines learning models, emphasizing the superior performance of LSTM in optimizing peak voltage, reducing power losses, and improving voltage stability by compressing the load curve. These outcomes contribute to minimized billing costs, showcasing the effectiveness of the proposed machine-learning-based approach. In response to the challenges posed by global economic trends and sharp fluctuations, the study [45] focused on predicting energy futures prices. The proposed multiscale model integrates a decomposition-ensemble approach with a subcomponent clustering method, allowing the derivation

Funded by the European Union

of subseries with different frequencies from the decomposed energy futures price series. This integration aims to enhance the feasibility of energy futures prediction. The ensemble model incorporates both linear model forecasts for linear component trends and machine learning methods for predicting nonlinearity. The study conducted in [46] introduced a hybrid model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for daily electricity price forecasting in the Iranian electricity market. The primary objective was to provide an accurate estimation of energy prices during peak hours, enabling precise planning and revenue maximization for hydropower generation. The model underwent testing using hourly data spanning the period from 2020 to 2021 and was compared against a multivariate linear regression model. The results indicated that the proposed hybrid model exhibited superior accuracy in electricity price forecasting compared to the multivariate linear regression model. The research [47] cantered on utilizing the XG Boost (XGb) and Light Gradient Boosting Model (LGBM) models to predict electricity prices in the Integrated Single Electricity Market~(ISEM) for energy market trading in Ireland. Eight novel technical indicators were derived from hourly electricity price data collected between February 2019 and November 2019. The study sought to evaluate whether incorporating these technical indicators as inputs could improve the performance of the XG Boost model. The outcomes demonstrated that the proposed technical indicators effectively contribute to accurate predictions of electricity prices, highlighting their efficacy in forecasting. The study in [48] investigated the efficacy of Multivariate (MRV)-LSTM in forecasting electricity prices, underscoring the significance of considering seasonality. The research challenges the belief that intricate architectures like MRV-LSTM are indispensable for incorporating seasonal behaviour, demonstrating competitive performance with simpler models. In a multi-year examination of the German electricity market, the proposed neural networks with an embedding layer surpassed MRV-LSTM and time-series benchmark models in short-term price forecasting, showcasing their practical utility and offering potential economic insights. The study in [49] introduced a Multiple Linear Regression (MLR) method for electricity price forecasting, emphasizing the consideration of various predictors to minimize the mean absolute percentage error. Conducted on training data from September 2018 to September 2019 in the day-ahead electricity market in Turkey, the research highlighted the crucial role of lagged electricity prices (previous day, one week, and lagged moving average prices) in achieving precise price estimation. Additionally, the inclusion of natural gas, oil, and coal prices, among other coefficients, contributed to enhanced result accuracy. The study emphasized the importance of training data length in reducing error proportions and noted comparable error rates to regular regression methods and dynamic regression models in electricity price forecasting.

Nonetheless, all these studies employ either single models [23] [43] [44] or hybrid models [45] [46] [47] [48] [49]. The single models often struggle to capture the temporal fluctuations in electricity prices, while the hybrid models lack exploration of coupling mechanisms, rendering them un- suitable for aggregator-level price prediction. Consequently, their performance is questionable, exposing a notable research gap in providing adequate foresight into electricity prices for effective management and control of EVs at both low-voltage and aggregator levels.

2.2. Deep-Weighted Ensemble Model

In this section, we explore the development of the proposed DWEM, leveraging the concept of weights to integrate standard and stacked LSTMs, aiming to enhance the accuracy of predicting temporal fluctuated prices. The DWEM is crafted to capture intricate temporal price patterns within the input time series data, effectively leveraging the strengths of both standardized and stacked LSTM

Funded by the European Union

architectures to mitigate individual model biases and improve overall forecasting accuracy. We begin by elucidating the problem formulation and the architecture of the proposed model, emphasizing the application of DWEM in managing EVs at the LV-aggregator level. Subsequently, we delve into the details of the standardized and stacked LSTM models, followed by an exploration of the DWEM formation, elucidating the heuristic mechanism for determining optimal weights. The power system network primarily comprises three major functional entities: power generation companies, utility companies acting as both buyers and sellers of energy, and consumers who are the end-user customers purchasing energy, as shown in Figure 1.

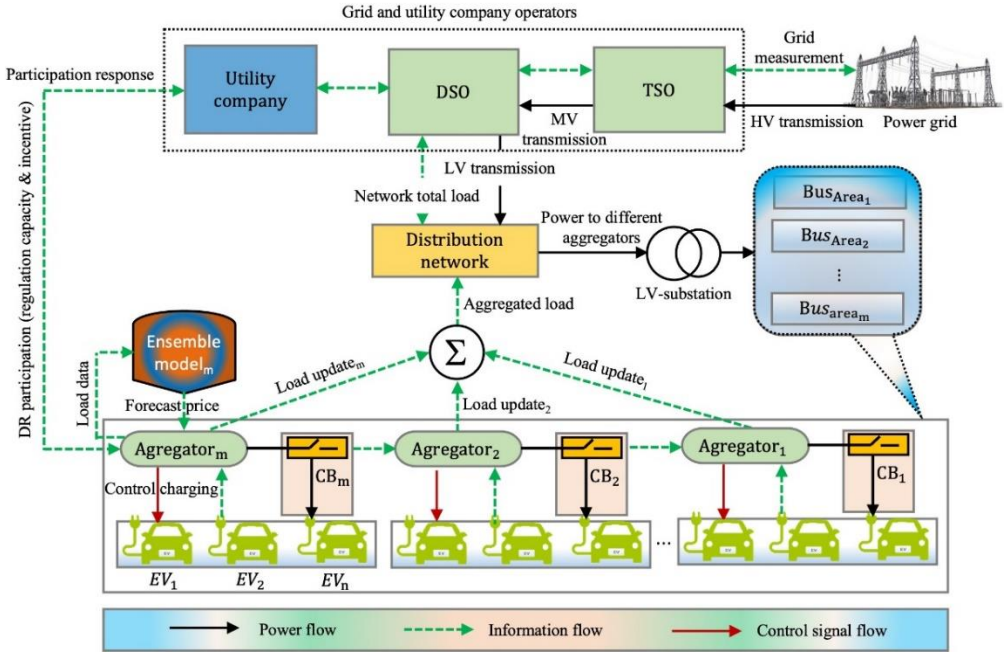


Figure 1. System model of the proposed deep-weighted ensemble model for wholesale electricity price forecasting to manage the charging EVs at the aggregator level.

The network spans three sub-transmission systems, each operating at different standard voltage levels: high voltage (HV) transmission at 110kV, medium voltage (MV) transmission at 38kV, and low voltage (LV) transmission at 230V [50]. The LV distribution system connects energy aggregators and end-users, with aggregators mainly procuring energy from utility companies and providing it to end-users to facilitate their needs. Transmission System Operators (TSOs) oversee the operations of both the HV and LV transmission systems. Meanwhile, Distribution System Operators (DSOs), in collaboration with utility companies, bear the responsibility for ensuring the seamless operations of the LV transmission system [51]. Aggregators play a crucial role in optimizing the aggregated load of Electric Vehicles (EVs), prompting utility companies to provide incentives for aggregator participation in Demand Response (DR). This support aids the power grid in managing aggregated charging loads, as illustrated in the architecture of the proposed Deep Ensemble Model (DWEM) in **Figure 1**. Each *i*-th EV is characterized by its specific arrival and departure sequence denoted as battery capacity (CB_i) current state-of-charge (SoC_i) and departure state-of-charge$\left(SoC_i^{dep}\right)$. Given the arrival and departure sequence at time t , we define the dwell time(DT_i) and the required state-of-charge(SoC_i^r) for the i EV, as outlined in Equations (1) and (2).

Funded by the European Union

$$DT_i = t_i^{dep} - t_i^{arr} \quad (1)$$

$$SoC_i^r(t) = \begin{cases} 1 - SoC_i(t) & \text{if } SoC_i^r = 1 \\ SoC_i^{dep} - SoC_i(t) & \text{if } SoC_i < SoC_i^r < 1 \end{cases} \quad (2)$$

In the current time step (t), the required charging time(T_i^r) and the energy (E_i) delivered to the i EV battery can be computed by accumulating charging rate (C_i) and the SoC_i in the previous time step $(t-1)$, considering the BC_i and charging efficiency η , as presented in Equations (3) and (4). The total energy (E_{Total}) consumption at the aggregator level at time step t is computed by summing the overall energies times the price(P) of the connected EVs, as presented in Equation (5).

$$T_i^r = \frac{SoC_i^r \times BC_i}{C_r \times \eta} \quad (3)$$

$$E_i(t) = \left(SoC_i(t-1) \times BC_i\right) + \left(\eta \times C_r\right) \quad (4)$$

$$E_{Total}(t) = \sum_{i=1}^{N} \left(E_i(t) \times P_i(t)\right) \quad (5)$$

The aggregator garners revenue from EV customers through the provision of charging services for their electric vehicles. The charging cost(C_i) for the i customer is determined by considering the markup price and the wholesale price, as elucidated in Equation (6). While, in the DR program (**Figure 1**), the utility company provides the regulation capacity(E_r), representing the amount of load reduction in kilowatts to the aggregators and the incentive $(P_{utility})$from the utility grid based on the regulation capacities offered. This dual revenue model implies that, on one hand, an aggregator generates income from customers through the price difference between retail and wholesale rates, while, on the other hand, it receives revenue [52] from the utility company for providing regulation services and we compute the overall revenue $(R)$of the aggregator as presented in Eq. (7).

$$C_i(t) = M + G(t) \quad (6)$$

$$R = \sum_{t=1}^{T} \sum_{i=1}^{N} C_i(t) + \left(P_{utility}(t) \times \frac{\Delta E_r(t)}{E_r}\right) \quad \text{for } \Delta E_r(t) \le E_r \quad (7)$$

In Equation (6), M represents the markup price, defined as an additional margin over the wholesale price G , contributing to the aggregator's revenue. In Equation (7), $P_{utility}$ signifies the incentive from the utility company, E_r denotes the total amount of regulated energy, and ΔE_r represents the amount of regulated energy reduced by the aggregator in the time horizon T such that $t = \{1, 2, 3, \cdots, T\}$. Considering the fixed amount the EV customer pays for their consumed energy, which remains constant in nature, the aggregator's revenue exhibits a linear dependency on their regulation services. Therefore, to maximize revenue, the aggregator must optimize energy consumption within the DR program, providing increased regulation services. As a result, the aggregator needs to manage charging EVs' following the wholesale price pattern, thus Equation (5) can be reformulated as an objective function, incorporating wholesale prices, to minimize total energy consumption, as defined in Equation (8).

$$\min\left(E_{total}\right) = \sum_{i=1}^{N} \left(E_i(t) \times G_i(t)\right) \quad (8)$$

Funded by the European Union

The effective management of charging EVs relies heavily on the accuracy of wholesale energy price forecasting. A more precise forecast of wholesale prices has the potential to significantly enhance the aggregator's efficiency, thereby contributing to increased revenue. In the following sections, we delve into the discussion of the proposed weighted-ensemble model for wholesale energy price forecasting. This model aims to assist energy aggregators in efficiently managing the charging of EVs, thereby supporting both the utility company and the power grid with energy regulation services.

2.3. The Mechanism of the Proposed Deep-Weight Ensemble Mode

The energy prices follow a time series sequential data format, recording observations at half-hourly or hourly intervals, which poses challenges during training with deep neural networks. In such cases, gradients can become very small, impeding the model's learning process [53]. The LSTM, a specialized type of recurrent neural network (RNN), is well-suited for mitigating these issues. Specifically designed to address the vanishing gradient problem, LSTMs excel at capturing long-term dependencies in sequential data, making them particularly effective for modelling and forecasting energy prices over time [54]. The LSTM architecture incorporates distinct functional elements, each designed for specific purposes in capturing and handling sequential dependencies. These components are associated with the two primary layers: the input layer and the LSTM layer [55], and are outlined below.

- The Input Layer: The input layer in an LSTM architecture serves as the initial stage for introducing external information into the network. Its primary responsibility is to process and prepare the input data before engaging with the LSTM cell. The key components of the input layer include input gate, forget gate, and the cell state update [56].
- The LSTM Layer: The LSTM layer is a crucial component of the LSTM architecture, explicitly crafted to overcome challenges linked to lengthy sequences in neural networks. It effectively addresses the vanishing gradient issue frequently encountered in traditional RNNs. The key constituents associated with the LSTM layer are the hidden state and output gate [57].

These linked components collectively form the foundation of the LSTM architecture, where the input layer manages the flow of information, and the LSTM layer orchestrates the cell state and hidden state dynamics [58]. To enhance the standard LSTM model, we replaced the tanh activation function in the LSTM layer with the Rectified Linear Unit (ReLU) activation. This modification aims to reduce the model's resource consumption while maintaining higher accuracy. The ReLU function introduces non-linearity, which can mitigate the vanishing gradient problem and enable the model to learn more intricate representations of the data, resulting in improved performance. Furthermore, recognizing the crucial role of hyperparameter tuning in optimizing the model's architecture, we utilized the Random Search technique. This approach yielded optimal values of 128 units for the input layer and 64 units for the LSTM layer. These optimized hyperparameters strike a balance between capturing complex patterns in the temporal fluctuated energy price data and avoiding overfitting as shown in Algorithm 1.

Funded by the European Union

Algorithm 1 Standard LSTM Model with Random Search for Wholesale Energy Price Forecast

Input: Energy price sequential data
Output: Forecasting results
Initialize Parameters:
forget_activation ← sigmoid, output_activation ← relu
input_units[a, b] ← RandomSearch(256, 224) ▷ Random search for input units between 256 and 224
Build the LSTM model:
$h_t = 0, c_t = 0$ ▷ Initialize hidden state and cell state
for each time step t in X_{train} **do**
    input_data = $X_{\text{train}}[t, :, :]$
    concat_input = [input_data, h_{t-1}] ▷ Concatenate input and previous hidden state
    forget_gate ← $\sigma(W_f \cdot \text{concat\_input} + b_f)$
    input_gate ← $\sigma(W_i \cdot \text{concat\_input} + b_i)$
    candidate_cell ← $\tanh(W_c \cdot \text{concat\_input} + b_c)$
    output_gate ← $\sigma(W_o \cdot \text{concat\_input} + b_o)$
    $c_t \leftarrow (\text{forget\_gate} \odot c_{t-1}) + (\text{input\_gate} \odot \text{candidate\_cell})$ ▷ Update cell state
    $h_t \leftarrow \text{output\_gate} \odot \tanh(c_t)$ ▷ Update hidden state
end for
Build the Sequential Model:
LSTM(input_units, forget_activation, output_activation) ▷ Add the first LSTM layer
LSTM(output_units, forget_activation, output_activation) ▷ Add the second LSTM layer
Flatten ▷ Add the Flatten layer
Dense(*input_units*, activation ← sigmoid) ▷ Add the first Dense layer
Dense(64, activation ← relu) ▷ Add the second Dense layer
Dropout(0.1) ▷ Add the Dropout layer with dropout rate of 0.1
Dense(32, activation ← relu) ▷ Add the third Dense layer
Dropout(0.1) ▷ Add another Dropout layer with dropout rate of 0.1
Dense(*next_steps*, activation ← linear) ▷ Add the final Dense layer
Compile the Model:
Compile(optimizer ← 'adam', loss ←'mse')
return Forecasting results

The Stacked LSTM model is an advanced variant of RNNs designed to address the challenges of capturing long-term dependencies in sequential data. The stacked LSTM architecture proves especially effective in tasks related to time series prediction. The hierarchical structure of stacked LSTMs enables them to adeptly learn and leverage complex representations, making these models a powerful choice for effectively modelling sequential data with inherent temporal dependencies. The pseudocode for the Stacked LSTM is presented in Algorithm 2, with a detailed explanation of the functional components of the model provided as follows:

- The Input Layer: The input layer serves as the entry point for processing the temporal patterns inherent in the time series data within the stacked LSTM architecture. In the case of a time series electricity prediction model, the Input Layer plays a crucial role by transforming energy price data into a three-dimensional format. This adaptation is essential since LSTM requires input data in a three-dimensional structure for effective computation and ensures that the subsequent LSTM layers can appropriately process the temporal dynamics and patterns inherent in the energy price data.
- The Stacked Layer: Stacked LSTMs are comprised of several LSTM layers organized sequentially. As the input sequence traverses through each LSTM layer, it undergoes processing, and the resulting hidden state is transmitted to the subsequent layer. This processing stage includes the transformation of the 3D-data back to 2D-data using the Flatten method. Subsequently, the flattened data is forwarded to the convolutional neural network

Funded by the European Union

(CNN) layers through the Dense layer to seamlessly integrate of CNN components, facilitating the extraction of spatial features from the transformed data. This stacking mechanism is pivotal, as it empowers the model to discern and encapsulate hierarchical features and temporal dependencies present within the data. The sequential arrangement of LSTM layers facilitates the extraction of intricate patterns at varying levels of abstraction, contributing to the model's capability to understand the complexities embedded in the input data.

- The Parameters and Configuration Layer: The tuning of hyperparameters, including the quantity of LSTM units in each layer and the total number of layers in the stack, is a critical aspect that can be adjusted based on the intricacy of the task. In our model, the initial LSTM layer comprises 128 hidden units, receiving the historical energy prices data as input. The subsequent LSTM layer is configured with 256 hidden units, and both layers employ the ReLU activation function. The hierarchical stacking of LSTM layers builds upon the representation learned by the preceding layer, progressively refining the model's understanding of temporal patterns within the data.
- The Output Layer: To prepare the output from the LSTM layers for the final prediction step, we utilize the 'Flatten()' operation. This operation reshapes the 3D tensor into a 2D format, enhancing the model's ability to capture correlations between temporal features and the target variable. Following the flattening process, we introduce two fully connected Dense layers, featuring 128 and 64 neurons, respectively, both activated by the ReLU activation function. These dense layers play a crucial role in learning complex feature interactions derived from the LSTM layers, allowing the model to develop a more profound understanding of the input sequences. To mitigate overfitting and enhance generalization, we incorporate dropout regularization with a rate of 0.1 after the first Dense layer. Dropout randomly deactivates a proportion of neurons during training, promoting reliance on multiple paths for information flow and reducing dependence on specific features. The subsequent Dense layer with 32 neurons continues to extract relevant features from the learned representation. Another dropout layer with a rate of 0.1 follows, further enhancing robustness and preventing overfitting. This comprehensive architecture ensures the model's capacity to capture intricate patterns while promoting generalization and preventing overfitting.

After training and testing both the standard LSTM and stacked LSTM models on energy price data, we proceeded to develop a Deep Ensemble Model. This ensemble leverages the weighted ensemble mechanism, combining predictions from the base models by assigning weights to each model's output. The process involves training individual models and aggregating their predictions in a weighted manner, as presented in Eq. (9).

$$E(x) = \left(w_{\text{std}} \times P_{\text{std}}(x)\right) + \left(w_{\text{stk}} \times P_{\text{stk}}(x)\right) \tag{9}$$

Where $E(x)$ represents the ensemble prediction, while P_{std}, P_{stk} denote the predictions of the standard and stacked LSTM models, respectively. Moreover, P_{std}, and W_{stk} refer to the respective weights assigned to these models. These weights play a crucial role in determining the influence of each model on the final ensemble prediction, contributing to a more robust and accurate overall prediction.

Funded by the European Union

Algorithm 2 Stacked LSTM with Random Search Hyperparameter Tuning

Input: Historical data sequences, Target variable, nextSteps, num_trials
Output: Stacked LSTM model with tuned hyperparameters
function STACKED_LSTM_RANDOM_SEARCH(historical_data, target_variable, nextSteps, num_trials)
    Initialize hyperparameter search space:
    $lstm\_units\_layer1 \leftarrow [64, 128, 256]$
    $lstm\_units\_layer2 \leftarrow [128, 256, 512]$
    $dense\_units\_layer1 \leftarrow [64, 128, 256]$
    $dense\_units\_layer2 \leftarrow [32, 64, 128]$
    $dropout\_rate \leftarrow [0.1, 0.2, 0.3]$
    Initialize best_mse and best_model:
    $best\_mse \leftarrow \infty$
    $best\_model \leftarrow$ None
    for $i = 1$ to num_trials **do**
        Sample hyperparameters randomly:
        $lstm\_units1 \leftarrow$ random_choice($lstm\_units\_layer1$)
        $lstm\_units2 \leftarrow$ random_choice($lstm\_units\_layer2$)
        $dense\_units1 \leftarrow$ random_choice($dense\_units\_layer1$)
        $dense\_units2 \leftarrow$ random_choice($dense\_units\_layer2$)
        $dropout\_rate \leftarrow$ random_choice($dropout\_rate$)
        Build the Stacked LSTM model with current hyperparameters:
        $model \leftarrow$ Sequential()
        $model$.add(LSTM($lstm\_units1$, activation $\leftarrow' relu'$, input_shape $\leftarrow (timesteps, features)$))
        $model$.add(LSTM($lstm\_units2$, activation $\leftarrow' relu'$, return_sequences $\leftarrow$True))
        $model$.add(Flatten())
        $model$.add(Dense($dense\_units1$, activation $\leftarrow' relu'$))
        $model$.add(Dropout($dropout\_rate$))
        $model$.add(Dense($dense\_units2$, activation $\leftarrow' relu'$))
        $model$.add(Dropout($dropout\_rate$))
        $model$.add(Dense($nextSteps$, activation $\leftarrow' linear'$))
        Compile the model:
        $model$.compile(optimizer $\leftarrow' adam'$, loss $\leftarrow' mse'$)
        Train the model on historical_data:
        $model$.fit($historical\_data, target\_variable$, epochs $\leftarrow$ 100, batch_size $\leftarrow$ 32, verbose $\leftarrow$ 0)
        Evaluate the model on validation data:
        $val\_mse \leftarrow model$.evaluate($validation\_data, validation\_target$, verbose $\leftarrow$ 0)
        if $val\_mse < best\_mse$ **then**
            $best\_mse \leftarrow val\_mse$
            $best\_model \leftarrow model$
        end if
    end for
    return $best\_model$

However, determining the optimal weights for an ensemble model is challenging due to the non-convex nature of the weight space, introducing complexities with multiple minima and maxima. To ascertain the optimal weights and enhance the forecasting performance of the proposed DWEM, we introduce a heuristic algorithm. This algorithm systematically tracks the weights, predictions, and their corresponding results, adaptively updating these weights to iteratively seek the most optimal configuration, thereby enhancing forecasting performance. The flowchart of the algorithm is presented in **Figure 2** and **Figure 3**, while the main steps are outlined below:

- Run the Algorithm 3 (refer to **Figure 2**) and set the flag to 0 to monitor the weights of the standard LSTM model, as discussed in Algorithm 1.
- Examine the flag value to dynamically adjust the weight of the standard LSTM model. If the flag is set to 0, indicating a specific condition, set the weight W_1 to 1.0. Conversely, if the flag is set to 1, implying an alternative scenario, decrement W_1 by 0.1. Simultaneously, establish the weight W_2 at 0.1 and proceed to execute the DWEM to acquire the updated forecasting results with these updated weights.

Funded by the European Union

- Call the function (refer to **Figure 3**) and assess the accuracy of the DWEM. If the accuracy is non-zero, compare it with the previous accuracy. If the newly obtained accuracy surpasses the previous one, update it with the newly obtained accuracy and save the corresponding weights. However, if there is no improvement in the accuracy, retain the previous accuracy and its corresponding weights. Subsequently, return the results to the calling algorithm (**Figure 2**)
- Check if the weight W_2 of stacked LSTM is less than or equal to the predefined criteria value of 1.0. If this condition is met, increment W_2 by 0.1 and return to step 1 to once again collect the ensemble model results with the updated value of the weight obtained from the stacked LSTM. Invoke the function (refer to **Figure 3**) to update the corresponding weights and results iteratively. Continue this process until the inner loop criterion is satisfied. This iterative approach ensures a thorough exploration of weight adjustments until the specified criteria are met.
- Update the flag value by setting it to 1. Examine the weight W_1 against the predefined value of 0.1. If W_1 is greater than or equal to 0.1, proceed to step 2. In this iteration, decrement the W_1 to 0.1 while resetting W_2 to 0.1. Pass the updated W_1 and W_2values to the DWEM to record the updated results. Adjust the weights of W_2 by calling the function (refer to **Figure 3**) for the second iteration of the outer loop, handling the standard LSTM weight W_1. Repeat the overall process from step 2 to step 4. However, if the value of W_1fails to meet the predefined criteria, print out the optimal results and conclude the algorithm (refer to **Figure 2**). This approach ensures a systematic exploration of weight adjustments, maximizing the adaptability and performance of the algorithm until the optimal weights are identified with highest accuracy.

Funded by the European Union

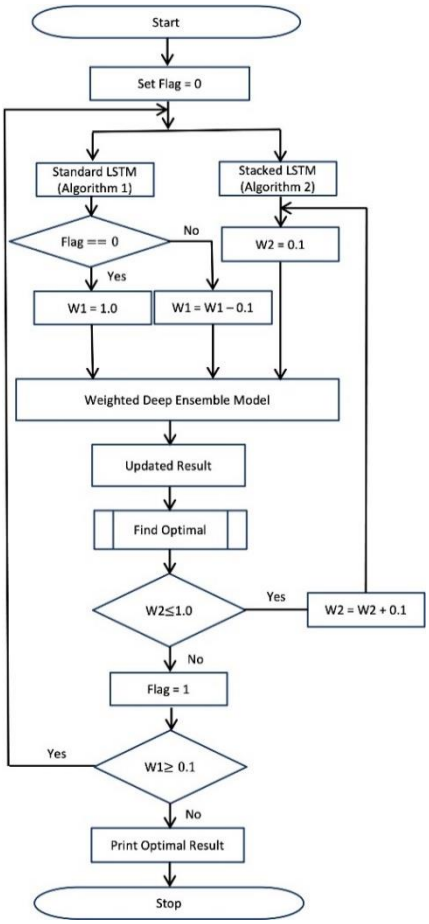


Figure 2. Flowchart of the proposed DWEM for determining the optimal weights through the heuristic approach.

Funded by the European Union

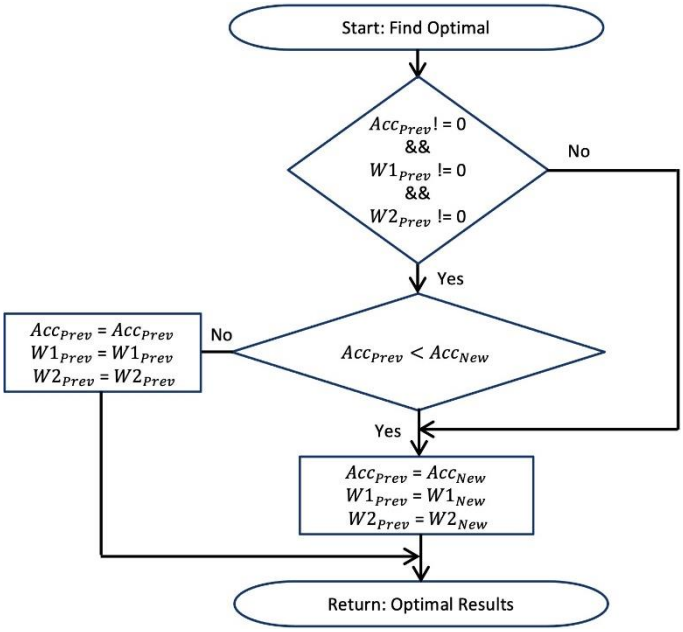


Figure 3. Flowchart of the subroutine of updating the weights for each of the iteration in the heuristic approach.

The foundational mechanism for attaining optimal weights through heuristic algorithms is elucidated in **Figure 4**. In this figure, the horizontal axis $(x - axis)$ signifies the W_1 of the standard LSTM, while the vertical axis $(y - axis)$ denotes the W_2 of the stacked LSTM model. The output of their combined contribution in the DWEM is gauged by accuracies $(Acc_1, Acc_2, \cdots, Acc_{10})$. In the initial iteration, these weights are set to their respective initial values, and the results are stored in the variable Acc_1. Following these results, the weights are updated by decrementing the standard LSTM weight (W_1) by 0.1 while incrementing the stacked LSTM weight (W_2). This process iteratively unfolds while maintaining W_1 constant and updating W_2, thereby collecting corresponding results. Once all W_2 values are tested against the initial W_1 value is decremented by 0.1, and W_2 is reset to its initial value. This iterative updating continues, and the results are recorded. Consequently, after testing all weights $(W_1 \; and \; W_2)$ and their corresponding performance (accuracy), the algorithm selects the weights resulting in the highest accuracy. This heuristic approach significantly enhances the performance of the proposed DWEM by determining optimal weights for both the standard and stacked LSTMs, contributing to a substantial boost in model performance.

Funded by the European Union

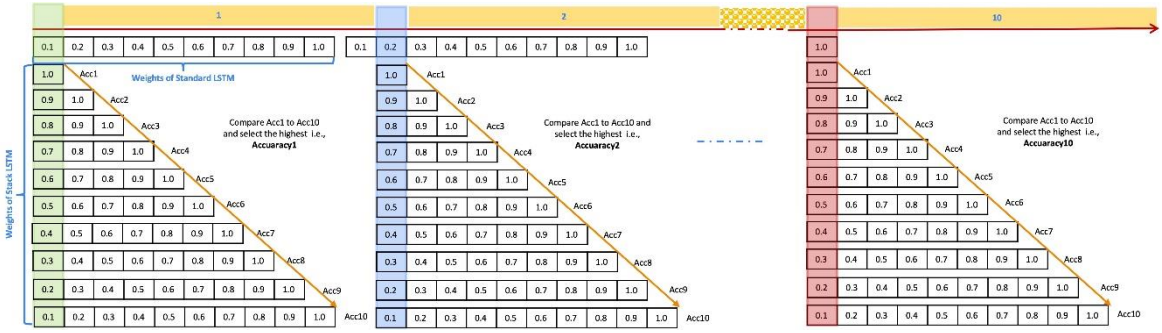


Figure 4. An illustration of determining the optimal weights for enhancing the efficiency of the proposed DWEM through the heuristic approach.

2.4. Experimental Setup and Results Discussion

In this section, we demonstrate the effectiveness of the proposed DWEM through experimental validation and a comparative analysis of results. We commence with data engineering, emphasizing the dataset's characteristics and preprocessing steps, including the criteria for feature selection. Subsequently, we delve into the discussion of performance metrics. Finally, we conduct a comparative study, considering these performance metrics to thoroughly evaluate and compare the effectiveness of the proposed DWEM. To simulate the proposed DWEM for forecasting energy prices to manage EVs at the aggregator level, we employ a dataset sourced from the Texas electricity market, specifically the ERCOT (Electric Reliability Council of Texas) market. ERCOT divides the Texas region into four congestion management zones (CMZs), namely West, North, South, and Houston. This study focuses on the dataset representing wholesale electricity prices in the Houston zone. The original data was collected at 15-minute intervals; however, for this study, we computed hourly averages by aggregating prices over four consecutive 15-minute periods. The detailed presentation of this processed data is provided in the subsequent sub-sections. The experimental dataset spans from January 2015 to December 2018, encompassing a total of 34,542 samples and featuring nine distinct attributes: Delivery Date, Delivery Hour, Repeated Hour Flag, Settlement Point Name, Settlement Point Type, DayStatus, Temperature in F, Load in Houston, and Settlement Point Price. To facilitate model training and evaluation, we partitioned the dataset into two subsets: 90 % (31,087 samples) for training and 10 % (3,455 samples) for validation. Throughout the training process, we initialized the learning rate at 0.01 and implemented a dynamic learning rate schedule, adjusting the rate as necessary to facilitate model convergence. This adjustment involved incorporating a factor of 0.1 to decrease the learning rate, and we set the patience value to 10 epochs, determining the duration the model waits for improvement before further adjusting the learning rate. Considering the number of features, the target variable is the settlement point price. Among the remaining eight variables, we selected the input variables based on the correlation matrix in **Figure 5**.

Funded by the European Union

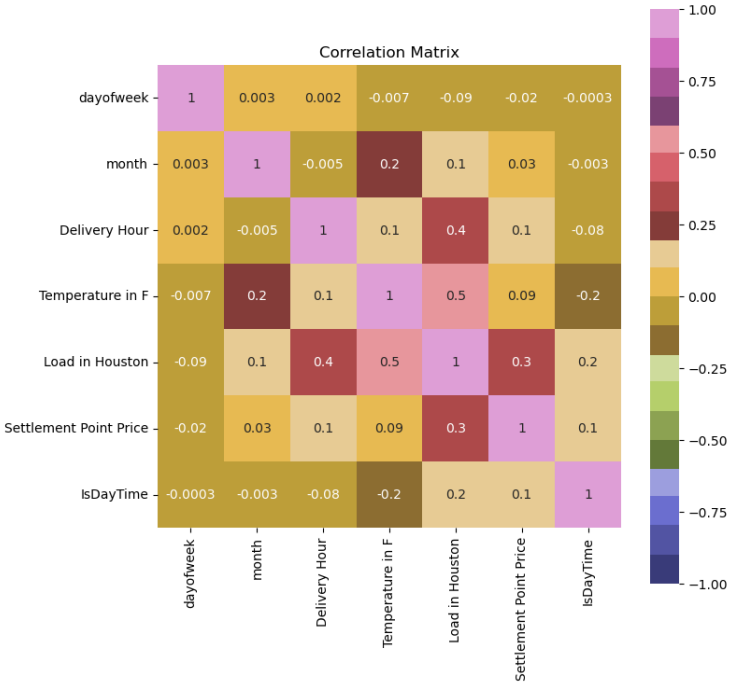


Figure 5. Correlation matrix highlighting a relationship between the target variable and the input variables.

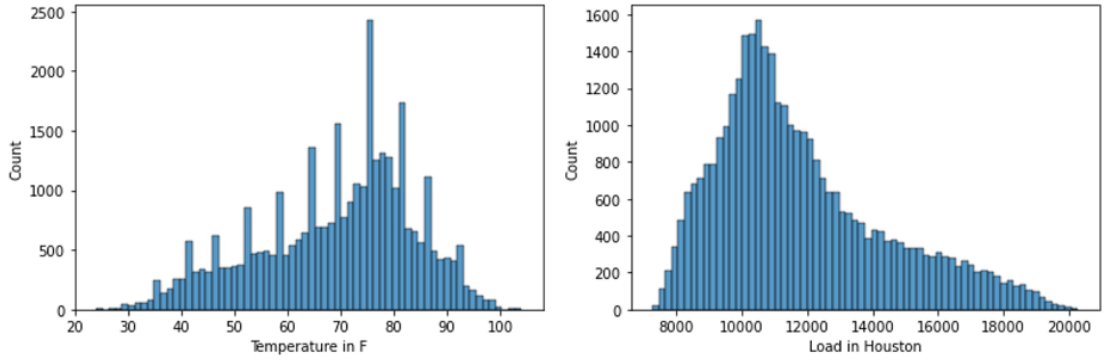


Figure 6. A representation of the temperature and load count in Houston region.

Funded by the European Union

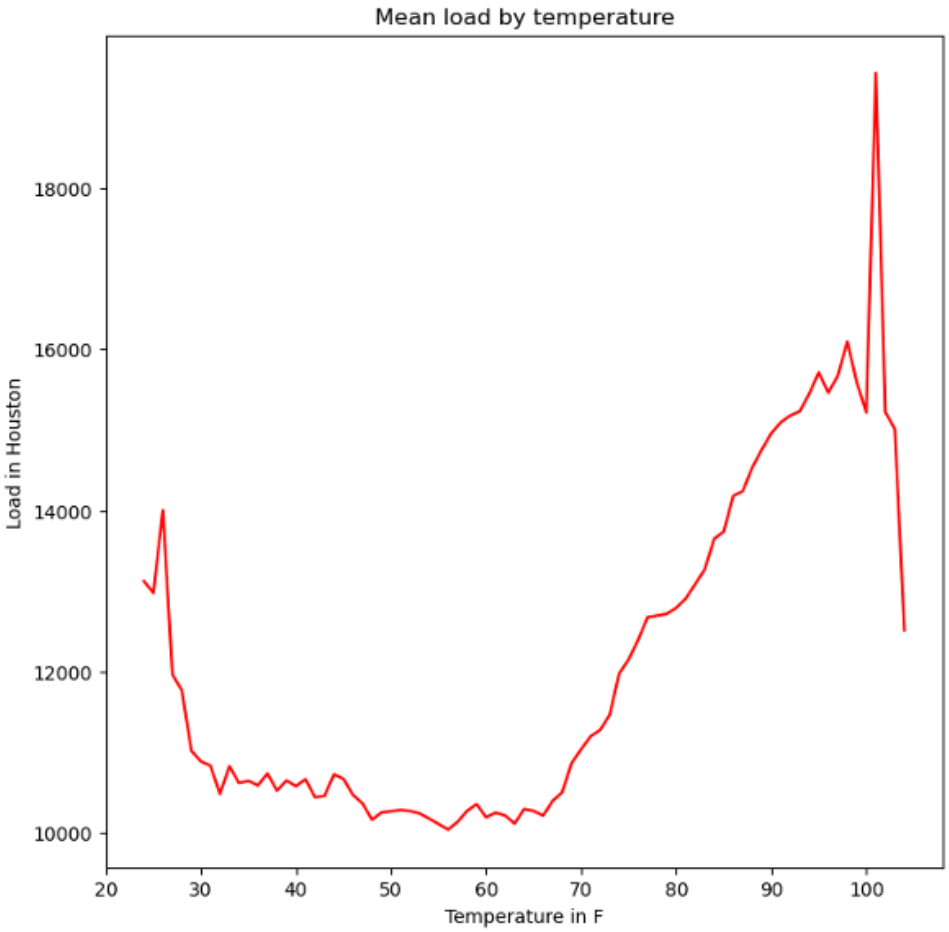


Figure 7. The relationship between load and the temperature highlighting the increasing trend of load with the increasing temperature.

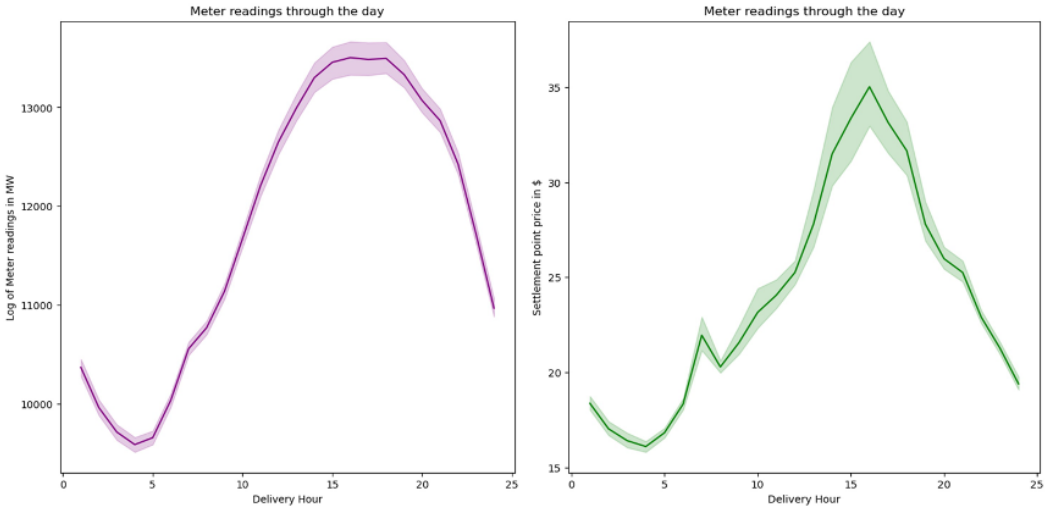


Figure 8. The relationship between load and the settlement price highlighting the increasing trend of price pattern with the increasing load.

Funded by the European Union

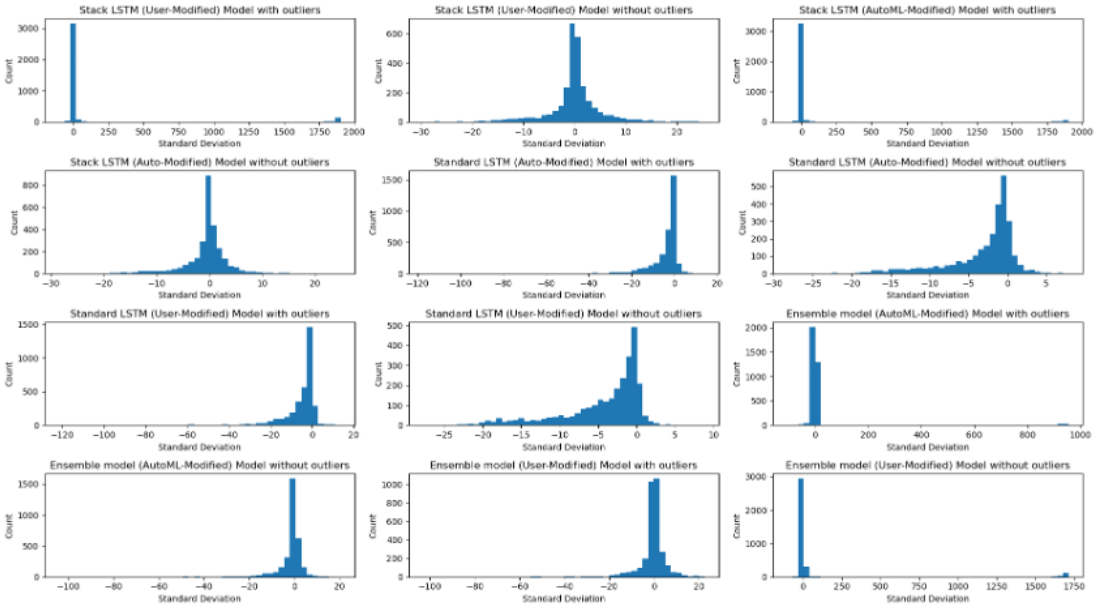


Figure 9. Representation of standard deviation with different models, including user-modified and auto-modified parameter tuning, in scenarios with and without outliers.

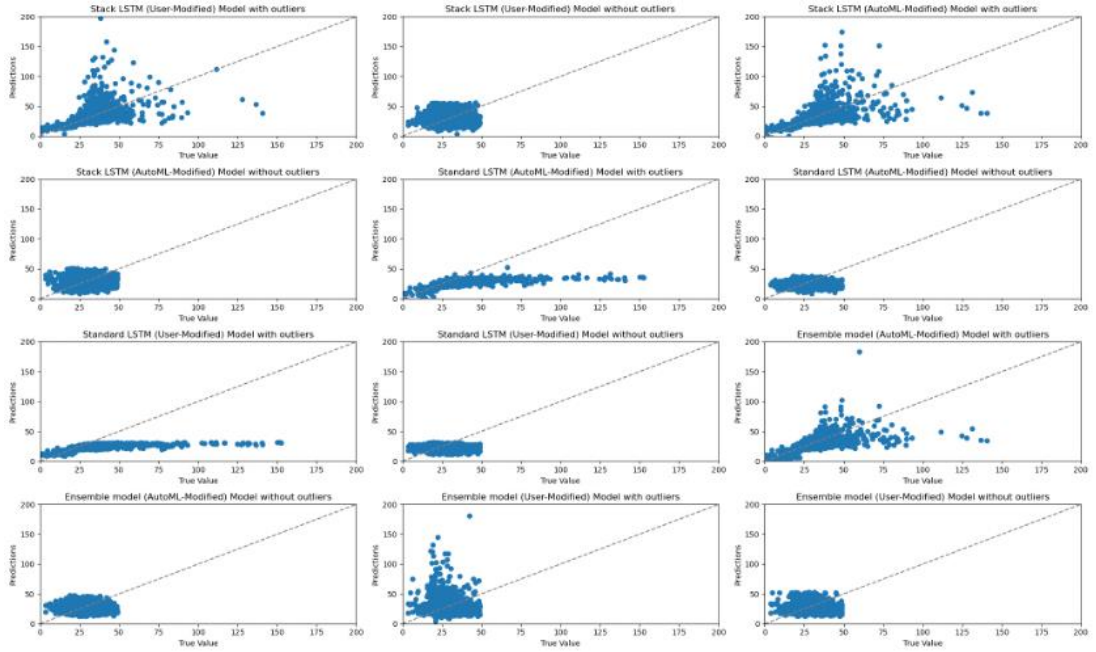


Figure 10. Fitting of different models corresponding to various standard deviation values.

Table 1. An evaluation of the accuracy of the proposed DWEM model according to the different standard deviations.

Standard deviation ($/MWh)	Accuracy [%]
1	14.09

Funded by the European Union

2	41.05
3	64.08
4	69.60
5	75.56
6	78.35
7	82.49
8	84.09
9	85.91
10	87.41

Table 2. The accuracy comparison of the proposed DWEM (Auto-Modified Hyperparameters) against state-of-the-art models (Auto-Modified Hyperparameters) concerning to the various values of the standard deviations in the without outlier's scenarios.

Standard deviation ($/MWh)	Accuracy [%]				
	DWEM	XGboost [47]	LGBM [47]	LR model [57]	MVR model [58]
1	14.09	2.09	17.86	13.97	18.93
3	64.08	8.47	39.88	40.55	53.35
5	75.56	21.86	64.29	61.62	67.32
7	82.49	44.54	76.19	76.36	80.09
10	87.41	67.70	79.32	80.86	83.82

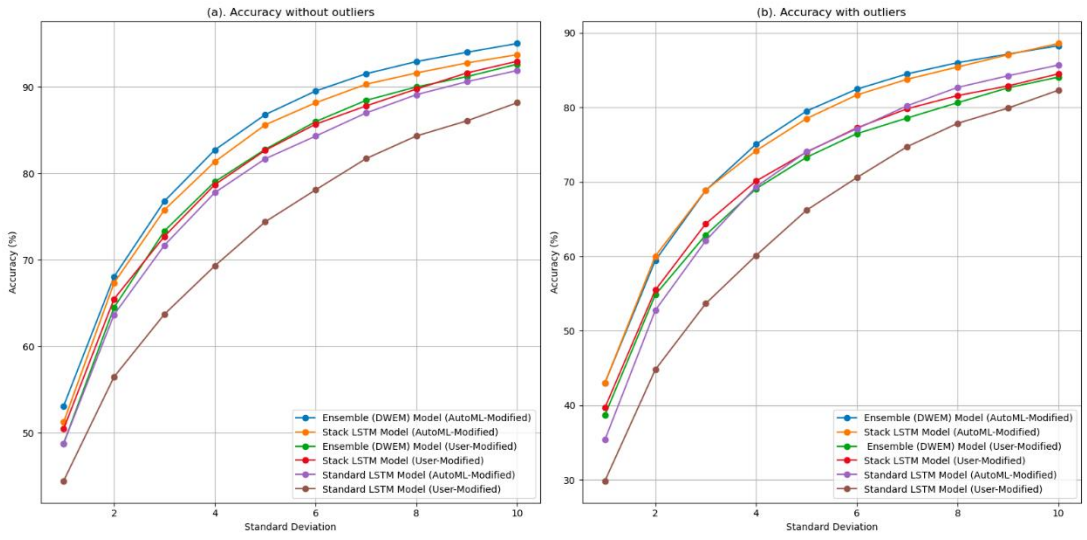


Figure 11. A comparison of the accuracy of the different models considering the User and Auto Modified hyperparameters in both without and with outlier's scenarios.

Funded by the European Union

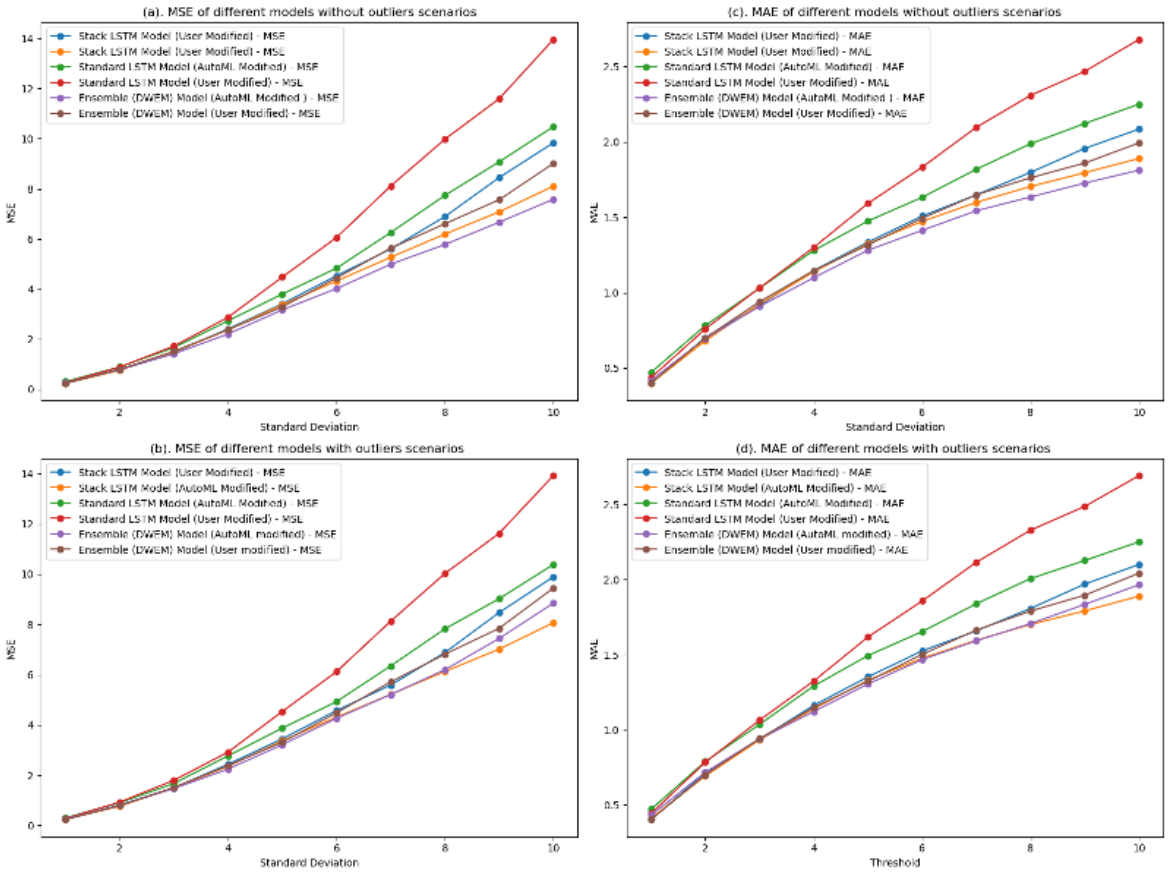


Figure 12. A compression of the MSE and MAE of the different models considering the User and Auto Modified hyperparameters in both without and with outlier's scenarios.

3. EV Demand Forecasting algorithm to optimally manage a week of EV usage (IREC)

3.1. Introduction

This activity aims to produce a smart tool that determines the optimal management for the charging session of an EV, while knowing the future medium-term mobility and consumption of the vehicle. The forecast developed can provide economical savings, both directly to the user, and indirectly through prolonging the lifetime of the battery.

For an electric vehicle (EV) to begin smart charging or bidirectional, the EV driver must provide some information to the charger. This typically includes their departure time, desired state-of-charge, minimum state-of-charge and for bidirectional charging the amount of cycling allowed. Requiring this information complicates the EV charging experience, and while a driver may have some sense for this information they may incorrectly estimate their departure time or the state-of-charge that they need. These conditions can cause an increase in the cost of charging for the driver. Additionally, when charging at a charging station managed by a charge station operator (CSO), the incentives for a CSO and the EV driver are not always aligned. The CSO wants to maximize their revenue so, while they may

Funded by the European Union

want to implement smart charging to reduce their energy cost, they are incentivized to charge the vehicle as much as possible and do not have a direct incentive to reduce degradation of the battery.

The goal of this work is to deliver a tool that will optimize smart and bidirectional charging processes by estimating vehicle trips for the following week, trip energy consumption, desired and minimum state-of-charge to minimize the cost of charging, while also considering the impacts of degradation (for both unidirectional and bidirectional charging). Looking one week ahead into the future will allow to consider energy prices fluctuations during the weekend or caused by renewables that might affect charging behavior. For this, a combined approach of forecasting and optimization techniques are proposed. While mobility data of drivers is still not easy to access, the authors used the public Oporto's taxi mobility dataset [59] [60] to develop a forecasting algorithm. The forecasting algorithm aims to determine the location and energy trip consumption of the vehicle for the following week. Under some assumptions for electricity prices, charging power, vehicle battery capacity, etc. a week-long forecast with five minute granularity is created then that forecast is inputted into a Mixed Integer Linear Programming (MILP) optimization model to determine the optimal charging profile from the perspective of the vehicle owner to minimize their costs. This information could be automatically provided to the charge controller to streamline the implementation of smart and bidirectional charging for EV drivers.

Below, see the structure of the document. The following sections describe the state-of-the-art in forecasting algorithms, smart charging optimization, use case dataset specifics, and results interpretation.

Forecasting Algorithms

This section provides an overview of state-of-the-art forecasting algorithms is presented. The chosen algorithms are introduced along with an explanation of their relevance to the specific scenario. A selected use case is outlined to demonstrate how these algorithms are practically applied.

Optimal charging session schedule

This section delves into the optimization of EV charging schedules, focusing on weekly trip patterns. It provides insights into the parameters and variables considered in the optimization process, the objective function guiding the optimization and the relevant constraints.

Use Case and Dataset Description

Here, we specify the use case employed for testing forecasting and optimization algorithms. A detailed description of the dataset used in these evaluations is provided.

Results Interpretation

This section is dedicated to presenting and interpreting the results derived from the application of forecasting and optimization models. It includes a comprehensive analysis of the outcomes, emphasizing their implications and relevance to the study.

Funded by the European Union

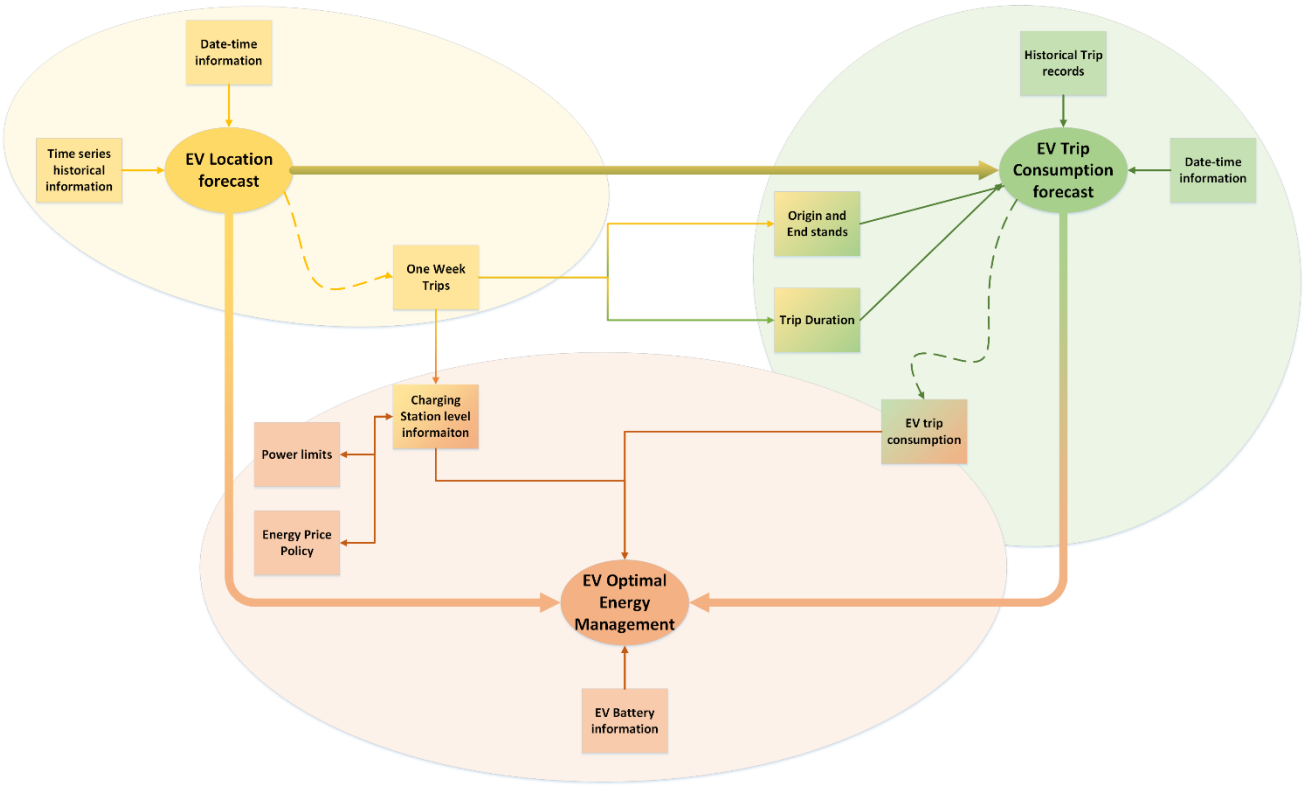


Figure 13. Forecasting model framework: Wider arrows represent the general modules outputs flow. Inside each box, dashed lines represent the forecasting models inputs and outputs.

3.2. Forecasting algorithms

In forecasting section, we have developed two different forecasting models: one for forecasting vehicle location and one for forecasting EV energy consumption over a week.

3.1.1. State of the Art

In the recent years, similar studies, either on input dataset or in the horizon of prediction challenge nature, have been performed.

Regarding the location prediction, in 2019 [61], a LSTM-based short term forecasting model for vehicle trip prediction was developed. The LSTM model results were compared against Markov chain models, obtaining notable improvements in terms of precision of the forecast. The nature of the dataset is similar to the dataset in our problem, using seven months of hourly records of locations records of roamers in Italy. In 2020 [62], a hierarchical temporal attention-based LSTM encoder-decoder model for individual location sequence prediction was proposed. The algorithm forecasted both daily and weekly behaviour obtaining accurate results in a private car mobility dataset from China. The dataset recorded 37,854 trips, in total, for 49 individuals from March, 2017 to October, 2018 using onboard GPS equipment. Each trip was represented by a sequence of time-stamped points with the location of the vehicle. In 2020 [63], a LSTM based model which encodes the locations as semantic words, was tested against the same taxi dataset used in our study. The goal of their algorithm was different, as it

Funded by the European Union

was used to predict the next taxi destination, and it is unclear how well this model would perform on a long-term prediction. Based on this review, the use of an LSTM model to predict vehicle location over a 1 week period. Finally, a systematic review of how is the scientific community working in mobility forecasting was performed in [4], indicating the key points that are lacking in the studies and their methodology.

Regarding EV demand, we have not found any paper focusing on the goal proposed in this paper, which is to predict energy consumption over a week. However, there are studies focused on charging demand of the EV. In 2023 [64] forecasted the aggregated EV charging demand. They concluded that both Seq2Seq and LSTM models are the best options for one step and multiple step forecasting, with Seq2Seq being the best option for the latter. In 2023 [65], a medium-term forecast methodology of EV charging demand is proposed. The goal of that paper is to predict the demand for charging station management, and not the consumption of a specific EV. Regarding energy consumption and demand, models such as LSTM, SVM, k-NN, Random Forest, have been widely used when forecasting other energy profiles such as electric building consumption [66], photovoltaic generation [67], and others. As a result, we have tested LSTM, SVM and Random Forest models in order to determine which is the best model choice for this particular problem.

3.1.2. Forecasting models

In this section, the forecasting algorithms chosen for testing in our use case have been described. In terms of the prediction of the location of the vehicle, a widely used model in the literature has been implemented: Long Short Term Memory recurrent neural network (LSTM RNN). For the prediction of energy consumption, we have performed a comparison between the Support Vector Machine (SVM), Random Forest (RF) and LSTM models.

3.1.2.1. Long Short Term Memory Recurrent Neural Network (LSTM RNN)

A LSTM model belongs to the so called family of Recurrent Neural Network (RNN). A Recurrent Neural Network (RNN) is a type of neural network architecture designed for processing sequences of data. The basic idea of an RNN is to use the information from previous time steps to influence the processing of the current time step. This makes RNNs particularly effective for tasks where context or temporal dependencies are important, such as natural language processing, speech recognition, and **time series prediction**. While RNNs are powerful for capturing sequential dependencies, they have some limitations, particularly in handling long-term dependencies. RNNs can be challenging to train on tasks that involve long sequences due to the **vanishing gradient problem**. Training may become slow or even fail to converge. The vanishing gradient problem is a challenge that occurs during the training of deep neural networks. The weights of the cells are computed through backpropagation, and the gradients which modify these weights are calculated with respect to a loss function. The vanishing gradient problem arises when these gradients become extremely small as they are propagated backward through the layers of the network. As a result, the weights of the earlier layers receive very small updates, and their learning process slows down. Long Short-Term Memory (LSTM) networks, are an extension of RNNs designed to address these limitations. LSTMs address the vanishing gradient

Funded by the European Union

problem by introducing a more sophisticated memory cell that can store information for long durations. These networks have separate input, **forget**, and output gates that control the flow of information into, out of, and within the memory cell, allowing them to selectively update and forget information. RNN and LSTM mathematical formulations have been explained extensively in the literature including in many of the references provided previously [61] [62]. Check the publication from Alex Shertinsky covering both general description of RNNs and LSTMs [68]. Also, see the original manuscript from Sepp Hochreiter and Jurgen Schmidhuber [69].

3.1.2.2. Random Forest

A Decision Tree is a supervised machine learning model used for both classification and regression tasks. It is a tree-like structure where each internal node represents a database-breaking decision based on the value of a particular feature in the dataset, each branch represents the outcome of the decision, and each leaf node represents the final decision or the predicted outcome. Decision trees are popular due to their simplicity, interpretability, and ease of visualization. The decision tree algorithm determines the optimal splitting criteria at each node based on the training data, so the users do not need to specify the rules or decisions at each node; rather, the algorithm learn these patterns from the provided data.

A Random Forest (RF) is a classifier and regression model that operates by constructing a multitude of decision trees, and gives as an output the modal (classification) or mean (regression) value of the individual decision trees output. In the case of a Random Forest, the process involves building an ensemble of decision trees, where each tree is trained on a random subset of the data and random subset of the features. Amongst the benefits of using a Random Forest over individual Decision Trees, (1) the random forest excels in **robustness** (it is less prone to overfit), and (2) the RF model returns a measure of feature importance in the dataset.

The concept of decision trees has roots in the field of statistics. One notable milestone was the development of the Iterative Dichotomiser 3 algorithm by Ross Quinlan [70], which popularized the use of decision trees for classification tasks. As for Random Forest, the algorithm was introduced by Leo Breiman in 2001 [71].

3.1.2.3. Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm primarily known for their use in classification tasks, but they can also be applied to time series forecasting problems. As Corinna Cortes and Vladimir Vapnik explain in the original manuscript [72], "the SVM maps the input vectors into some high dimensional feature space Z through some non-linear mapping chosen a priori" (see also the full manuscript for the mathematical formulation).

In a classification problem, the main idea behind SVM is to find the optimal hyperplanes that maximize the margin between the output classes, the classes being the possible values. There are different strategies in order to determine these hyperplanes, primarily, One-vs-One (OvO), or One-vs-All (OvA). The latter is the one preferred in most cases due to computational complexity and overall performance.

Funded by the European Union

In order to use a SVM for time-series forecasting, the formulation needs to treat the time series forecasting problem as a regression problem, where the goal is to predict a continuous output (e.g., the future values of a time series). The SVM regression model aims to find a hyperplane in the feature space that best fits the relationship between the input features and the continuous target variable. This hyperplane is used to make predictions for new, unseen data points, providing an estimate of the numeric output.

3.1.3. KPIs for the Forecasting Models

In this section, we are going to introduce the metrics used to evaluate the forecasting algorithms. Balanced accuracy has been used to evaluate the Location Forecasting model, while Root Mean Squared Error (RMSE) and Mean Average Percentage Value (MAPE) have been used to evaluate the energy consumption forecasting model.

3.1.3.1. Balanced accuracy

Balanced Accuracy is a performance metric used to evaluate the classification accuracy of a model. It is an extension of the traditional accuracy metric and takes into account the distribution of classes in the dataset. Let $\boldsymbol{TP}$ be the number of true positives, and $\boldsymbol{FN}$ the number of false negatives. The sensitivity is the proportion of actual positive instances correctly predicted by the model.

$$Sensitivity = \frac{TP}{TP + FN},$$

Specificity is the proportion of actual negative instances correctly predicted by the model. It is calculated as

$$Specifity = \frac{TN}{TN + FP},$$

where TN is the number of true negatives, and FP is the number of false positives. Finally, the balanced accuracy metric is given by:

$$BA = \frac{Sensitivity + Specifity}{2}.$$

The key idea behind balanced accuracy is to give equal importance to both positive and negative classes, preventing the metric from being overly influenced by the class with more instances.

In our case, we are dealing with the non-binary variable $location$. The average value of the Balanced Accuracy metric for every possible output value is considered:

$$BA = ' = \sum_{l \in \mathcal{L}} \frac{BA_l}{n}$$

Where BA_l is the balanced accuracy for the location l, and n is the number of locations.

Funded by the European Union

3.1.3.2. Root mean squared error

The Root Mean Squared Error (RMSE) is a commonly used metric to measure the accuracy of a regression model. It provides a measure of the average magnitude of the errors between predicted and actual values, and it's particularly useful when the errors are expected to be normally distributed

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2},$$

where T is the number of observations of the observed variable y_t, and $\hat{y}_t$ is the predicted value.

The RMSE is expressed in the same units as the variable being measured, which makes it interpretable and easy to compare to the scale of the original data. Lower RMSE values indicate better model performance, as they represent smaller average errors between predicted and actual values.

3.1.3.3. Mean absolute percentage value

The Mean Absolute Percentage Value (MAPE) is an accuracy metric used to assess the quality in regression problems. Usually it is defined as follows:

$$MAPE = \frac{1}{n}\sum_{t=1}^{T}\left|\frac{y_t - \hat{y}_t}{y_t}\right|,$$

where, as before, T is the number of observations of the observed variable y_t, and $\hat{y}_t$ is the predicted value.

MAPE is not suitable for observed variables y that includes the zero value in its domain. However, as the aim of our proposed forecasting method is to predict trip energy consumption, the zero division is not real problem.

3.1.4. Prediction of weekly trip profile through EV location

This model solves a classification model for knowing the location of the vehicle in the following timesteps.

The input data for the EV location forecasting algorithm is summarised in the following **Table 3**. Time (hour and minutes) information it has been transformed through the cosine and sine functions: each pair (t_{sin}, t_{cos}) represents a bijective relation with time, this gives the opportunity to create a continuous and cyclic transformation of time. On the other hand, we dichotomized Month and Weekday variables, i.e. we created dummy variables to indicate each class. For instance, from Weekday variables we created 6 different variables defined as follows:

$$Weekday_i = \begin{cases} 1 & if\ Weekday = i, \\ 0 & otherwise. \end{cases}$$

The remaining day is represented by the absence of the first 6 days, $Weekday_i = 0\ \forall i \in \{0, \dots, 5\}$.

Funded by the European Union

Moreover, we selected temporal information of the independent variable to capture a possible time dependency between the observed information one week and one day before.

The model input is a matrix with 2016 rows, equivalent to one week of information with its respective columns, and the output is a vector containing 288 values, which corresponds to one day of information. In order to obtain the whole week of forecasting information, the day ahead predictions are repeated seven times adding predicted labels as inputs in the following iteration. This procedure is adopted due to computational limitations: when predicting a whole week, the input size usually increases and the Tensorflow dataset could grow too large to be handled with the available computer memory.

Table 3. Weekly EV location prediction summary table.

Metadata		
Granularity	This dataset consists of trip events, and so, they are not found at any given frequency. The precision with which we can find separate events is seconds.	
Historicity	1 year (from July 2013 to July 2014)	
Input width*	2016	
Output width*	288	
*This variable indicates that every instance of the algorithm uses one week (2016 timesteps) to predict one day (288 timesteps). The algorithm is called 7 times in order to predict one entire week.		
Training dataset		
Training time window	9 months	
Training loss indicator	Sparse Categorical Crossentropy	
Features	***Description***	***Domain***
$month$	Month.	$month \in [1, \dots, 12]$
$weekday$	Day of the week. Holiday days are labelled as Sunday.	$weekday \in [0, \dots, 6]$
t_{sin}	Transformation of the hour through the sin function: $t_{cos} = \sin\left(\pi \cdot \frac{\left(hour + \frac{minute}{60}\right)}{24}\right)$ Where $hour$ and $minute$ are the hour and minute of the respective timestep t.	$t_{sin} \in [-1,1]$
t_{cos}	Transformation of the hour through the cos function: $t_{cos} = \cos\left(\pi \cdot \frac{\left(hour + \frac{minute}{60}\right)}{24}\right)$ Where $hour$ and $minute$ are the hour and minute of the respective timestep t.	$t_{cos} \in [-1,1]$
l_{t-1}	Location in previous timestep	$l_{t-1} \in \mathcal{L}$
$\forall l \in \mathcal{L}, I^l_{t-288}$	Indicates if the vehicle was in location l in the same timestep 1 day ago.	$\forall l \in \mathcal{L}, I^l_{t-288} \in \{0,1\}$
$\forall l \in \mathcal{L}, I^l_{t-2016}$	Indicates if the vehicle was in location l in the same timestep 7 days ago.	$\forall l \in \mathcal{L}, I^l_{t-2016} \in \{0,1\}$

Funded by the European Union

The loss function used for the training is Sparse Categorical Crossentropy, which is commonly used in the training of neural networks for multi-class classification tasks. It takes two sets of probabilities as input: the predicted probabilities (output of the neural network) and the true class labels represented as integers (not one-hot encoded). It is commonly used when dealing with problems where there are more than two classes, and the classes are mutually exclusive (each instance belongs to exactly one class).

For each instance in the dataset, it computes the cross-entropy loss between the predicted probabilities and the true class labels. The cross-entropy loss measures the dissimilarity between the predicted probability distribution and the true distribution.

Let $y_{i,t}$, represent the predicted probability distribution for class i in timestep t, and $y_{true,t}$ represent the true class labels (integers), the formula for Sparse Categorical Crossentropy (SPC) is:

$$SPC = -\frac{1}{T}\sum_{t}\left(\sum_{i}\log\left(\frac{e^{y_{i,t}}}{\sum_j e^{y_{j,t}}}\right)\cdot \chi_{i,t}\right)$$

Where $\chi_{i,t}$ is given by:

$$\chi_{i,t} = \begin{cases} 1, & if\ i\ is\ the\ true\ class\ in\ timestep\ t \\ 0, otherwise \end{cases}$$

3.1.5. Prediction of EV trip energy consumption

This algorithm aims to determine the energy consumption for all the trips detected by the location forecasting algorithm.

Given an speficic expected EV trip called $trip$, it is determined by its origin and final stand location (o^{stand}, f^{stand}, respectively) and the trip duration $trip^{duration}$. The goal of this algorithm is to obtain a forecast of the energy consumption of the EV in kWh. Due to data limitations, these are the only features included, but it is expected that adding weather conditions such as precipitation, temperature, and also traffic conditions would be a nice addition to the dataset. In an exhaustive implementation of the proposed algorithms, the input trip duration, $trip^{duration}$, should be provided by the location forecasting proposed in Section 3.1.4.

As previously mentioned, Month and Weekday information are dichotomized. In addition, continuous variables are standardized, Equation (1), in order to obtain inputs within similar domain.

$$x_t^* = \frac{x_t - \bar{x}}{std(x)}, \quad (1)$$

where x_t^* is an observation of variable x. $\bar{x}$ represents the mean value of variable x and $std(x)$ refers to the standard deviation of the variable.

Table 4. EV trip energy consumption prediction summary table.

Metadata	
Granularity	This dataset consists of trip events, and so, they are not found at any given frequency. The precision with which we can find separate events is seconds.
Historicity	1 year (from July 2013 to July 2014)

Funded by the European Union

Input width*	50	
Output width*	1	
*This variable indicates that every instance of the algorithm uses the 50 previous trips as input, to predict the trip consumption of the next trip. The algorithm is launched N times, where N is the number of trips predicted for the following week.		
Training dataset		
Training time window	9 months	
Training loss function	Root mean squared error	
Features	***Description***	***Domain***
$month$	Month.	$month \in [1, \dots, 12]$
$weekday$	Day of the week. Holiday days are labelled as Sunday.	$weekday \in [0, \dots, 6]$
$trip^{duration}$	Trip duration in seconds. [s]	$trip^{duration} \geq 0$
o^{stand}	Origin stand.	$o^{stand} \in \mathcal{L}$
f^{stand}	Final stand.	$f^{stand} \in \mathcal{L}$
t_{sin}	Transformation of the hour through the sin function: $t_{cos} = \sin\left(\pi \cdot \frac{\left(hour + \frac{minute}{60}\right)}{24}\right)$ Where $hour$ and $minute$ are the hour and minute of the respective timestep t.	$t_{sin} \in [-1,1]$
t_{cos}	Transformation of the hour through the cos function: $t_{cos} = \cos\left(\pi \cdot \frac{\left(hour + \frac{minute}{60}\right)}{24}\right)$ Where $hour$ and $minute$ are the hour and minute of the respective timestep t.	$t_{cos} \in [-1,1]$
Target output	***Description***	***Domain***
$trip^{cons}$	Energy consumption for the trip. [kWh]	$trip^{cons} \geq 0$

3.2. User Smart Model: Medium-term optimization approach for scheduling EV charging sessions

We aim to develop and test an optimal charging scheduler for Electric Vehicle (EV) users. The model is designed to consider the weekly trips with the energy consumption associated with each trip and the EV's location forecasted in Section 3.1. The model time horizon will be extended to one week. In this context, the charging sessions schedule will be determined from the point of view of EV owner, for this reason we are also going to refer to it by the name of **User Smart Model**. The algorithm will be designed to make decisions such as to partially charge an EV if the vehicle has enough energy to accommodate upcoming trips without impacting driver needs. The model will consider both economic and battery health trade off among weekly charging sessions, and equally important the following theoretical proposal will be able to be applied to an arbitrary time horizon.

Funded by the European Union

3.2.1. State of the Art

In recent years the number of Electric Vehicles (EV), either fully electric (BEV) or Plug-in Hybrid (PHEV), has increased significantly. China, the United States and European countries are the ones where the EVs have the greatest presence [73].

Different articles have been published to cover different challenges that these vehicles represent: power system from infrastructure and energy generation point of view [74] [75] [76], battery degradation [77] [78], flexibility [79] [80] or environmental impact [81].

Another widely studied concern involves managing the energy needs of EVs, however most approaches have focused either on the Charging System Operator (CSO) perspective or the integration of the EV at house grid level. In this case, the CSO focuses on reducing energy costs related on the charging station and the energy demands of its customers. Common strategies involve shifting charges to minimize the total energy bill [82] [83], reducing peak power demand [84] [85], and mitigating the environmental impacts or energy generation [86] [87]. However, in the proposed approach a set of locations where the EV could be parked is considered, each of them with their own (dis)charging characteristics and energy price policies. These considerations should help the EV to find the most profitable strategy, deciding what to do at each all possible charging stations where the EV will be parked in the future time horizon.

On the other hand, mid-term Battery Electric Vehicle energy management has limited studied, whereas research focused on CSO profit [88]. Exploring long-term optimization from the Electric Vehicle owner perspective opens new opportunities:

- **Selective vehicle charging**. If the trip pattern of the vehicle user is known, the model can make informed decisions, not only about partial charges but also about abstaining from charging when it is neither economically nor technically optimal, thereby optimizing the overall charging strategy through the optimization horizon.
- **Battery degradation reduction.** Battery degradation is influenced by various factors such as driving behaviour, State of Charge (SOC), charging power, among others. A medium-term horizon and the selective vehicle charging sessions offer an opportunity to optimize the charging power levels and maintain the EV battery within the most beneficial SOC and power ranges for battery health.
- **Greater economic savings potential.** A charging algorithm capable of determining the minimum energy required for next trips during expensive energy price periods can save energy costs by waiting for cheaper periods to charge.

Funded by the European Union

3.2.2. List of Parameters and Variables

In this section, we are presenting the parameters and variables used to describe the optimisation model. See the detailed description in **Table 5** and **Table 6**.

Table 5. List of parameters.

Symbol	Description [Units]	Domain
	Sets	
$\mathcal{T}$	Set of time steps	$\mathcal{T} = \{1, \dots, T\}$
$\mathcal{L}$	Set of charging station locations	$\mathcal{L} = \{1, \dots, L\}$
$\underline{\mathscr{L}}$	Subset of forecasted locations as parking point for the EV	$\underline{\mathscr{L}} \subseteq \mathscr{L}$
	Matrices	
R	Vector T of positions in $\underline{\mathscr{L}}$, 0 if the EV is driving (Temporary list of stop locations)	$\{R_t\} \in \mathcal{L} \cup \{0\}$
Λ^{buy}	matrix T×$\lvert\underline{\mathscr{L}}\rvert$ expressing the energy cost for each location l in $\underline{\mathscr{L}}$ and time *t*. [€/kWh]	$\{\lambda_{t.l}^{buy}\} \in \mathbb{R}^+$
Λ^{sell}	Matrix T×$\lvert\underline{\mathscr{L}}\rvert$ expressing the energy compensation/sell price for each location l and time *t*. [€/kWh]	$\{\lambda_{t.l}^{sell}\} \in \mathbb{R}^+$
P^{Cmax}	Vector of size L expressing the maximum charging power at location l in $\underline{\mathscr{L}}$. [kW]	$\{P_l^{Cmax}\} \in \mathbb{R}^+ \cup \{0\}$
P^{Cmin}	Vector of size L expressing the minimum charging power at location l in $\underline{\mathscr{L}}$ [kW]	$\{P_l^{Cmin}\} \in \mathbb{R}^+ \cup \{0\}$
P^{Dmax}	Vector of size L expressing the maximum discharging power at location l in $\underline{\mathscr{L}}$. [kW]	$\{P_l^{Dmax}\} \in \mathbb{R}^+ \cup \{0\}$
P^{Dmin}	Vector of size L expressing the minimum discharging power at location l in $\underline{\mathscr{L}}$. [kW]	$\{P_l^{Dmin}\} \in \mathbb{R}^+ \cup \{0\}$
	Vectors	
D_t^{ev}	EV energy consumption [kWh]	$\mathbb{R}^+ \cup \{0\}$
	Scalars	
Δ_t	Time step duration [hours]	$\mathbb{R}^+$
M	An arbitrary large constant	$M \gg 0$
β	EV energy demand security parameter	[1,2)
α^C	SOC threshold where EV maximum charge power decreases [%]	[0,1]
α^D	SOC threshold where EV maximum discharge power decreases [%]	[0,1]
E^B	Battery capacity [kWh]	$\mathbb{R}^+$
η^{ch}	Battery charging efficiency [%]	(0,1)
η^{dch}	Battery discharging efficiency [%]	(0,1)
C^{health}	Penalization cost for an extra battery health reduction during V2G discharging [€/kWh]	$\mathbb{R}^+$
$\overline{SOC}$	Maximum SOC value [%]	[0,1]
$\underline{SOC}$	Minimum SOC value [%]	[0,1]
SOC_0	Initial SOC [%]	[0,1]
SOC_T	Minimum SOC at the time window end [%]	[0,1]

Funded by the European Union

It is important to note that in the position vector $R(t)$, a value of 0 means the EV's active state, i.e., the vehicle is in motion or driving. Consequently, the initial position (0) within the vectors, defining charging point characteristics, always holds the value 0. This location (0) acts as a constructed charging point, deliberately set with all its parameters, such as boundaries and prices, at zero.

Table 6. List of variables.

Symbol	Description [Units]	Domain
p_t^c	Charging power to the EV [kW]	$\mathbb{R}^+ \cup \{0\}$
p_t^d	Discharging power from the EV [kW]	$\mathbb{R}^+ \cup \{0\}$
u_t^c	Active charge indicator	$\{0,1\}$
u_t^d	Active discharge indicator	$\{0,1\}$
soc_t	State Of Charge of the EV	[0,1]
soc_t^{slack}	Slack variable to fulfil SOC's lower bound	[0,1]

3.2.3. Objective function

The objective function is,

$$F(p^c, p^d, s^{slack}) = \sum_{t \in \mathcal{T}} \left(\underbrace{\left(\Delta_t \cdot \left(p_t^c \cdot \lambda_{t,R(t)}^{buy} - p_t^d (\lambda_{t,R(t)}^{sell} - C^{health}) \right) \right)}_{A} + \overbrace{\mathrm{M} \cdot \mathrm{s}_\mathrm{t}^{\mathrm{slack}}}^{B} \right). \quad (2)$$

Where, expression *A* in $F(p^c, p^d, s^{slack})$ refers to the energy bill resulting from purchasing (p_t^c) and selling/injecting (p_t^d) energy to charging station $R(t)$, and a discharging penalization cost, C^{health} $[€/kWh]$, which is calculated as:

$$C^{health} = \frac{C^B}{L^B},$$

where C^B refers to battery cost in €/kWh, and L^B is the expected lifetime of the battery in terms of cycles. C^{health} cost is introduced to mitigate the extra battery degradation impact when discharging the EV due to V2G activities. Expression B in $F(p^c, p^d, s^{slack})$ sets the behaviour of $\mathrm{s}_\mathrm{t}^{\mathrm{slack}}$ variables that we will discuss in Section 3.2.4.1.

3.2.4. Constraints

3.2.4.1. State of Charge constraints

We defined the State of Charge (SOC) balance between time steps in equation (3), and the different SOC limitations in equations (4) to (7).

$$t = 1 \quad (3)$$

Funded by the European Union

$$soc_t = \begin{cases} SOC_0 + \dfrac{\Delta_t \left(p_t^c \eta^{ch} - \dfrac{p_t^d}{\eta^d} \right) - D_t^{ev} \cdot \beta}{E^B} \\ soc_{t-1} + \dfrac{\Delta_t \left(p_t^c \eta^{ch} - \dfrac{p_t^d}{\eta^d} \right) - D_t^{ev} \cdot \beta}{E^B} \end{cases} \qquad \forall t \in \mathcal{T} \setminus \{1\}$$

$$soc_t \leq \overline{SOC} \qquad \forall t \in \mathcal{T} \qquad (4)$$

$$soc_t + s^{slack} \geq \underline{SOC} \qquad \forall t \in \mathcal{T} \qquad (5)$$

$$soc_t > 0 \qquad \forall t \in \mathcal{T} \qquad (6)$$

$$soc_T \geq SOC_T \qquad \forall t \in \mathcal{T} \qquad (7)$$

Variable s_t^{slack}, in Constraint (5), prevents infeasibilities in the Optimization problem when the EV is plugged into a charger with a SOC lower than the minimum allowed ($\underline{SOC}$). As we have seen in the Objective function, because we are weighting the variable by M, when an EV arrives below $\underline{SOC}$ the first action will be to charge the EV until this value.

3.2.4.2. Power constraints

The following set of constraints express the bounds for (dis)charging according to both charger and EV limitations.

$$p_t^c \leq u_t^c \cdot P_{t,R(t)}^{Cmax} \qquad \forall t \in \mathcal{T} \qquad (8)$$

$$p_t^c \geq u_t^c \cdot P_{t,R(t)}^{Cmin} \qquad \forall t \in \mathcal{T} \qquad (9)$$

$$p_t^d \leq u_t^d \cdot P_{t,R(t)}^{Dmax} \qquad \forall t \in \mathcal{T} \qquad (10)$$

$$p_t^d \geq u_t^d \cdot P_{t,R(t)}^{Dmin} \qquad \forall t \in \mathcal{T} \qquad (11)$$

$$u_t^d + u_t^c \leq 1 \qquad \forall t \in \mathcal{T} \qquad (12)$$

Moreover, it is well known that EV cannot always charge or discharge at maximum speed because of SOC values, represented in **Figure 14**–other factors can also affect this behaviour, such us battery temperature, however they are not considered on this approach. For this reason, the following constraints are considered:

$$p_t^c \leq \begin{cases} P_t^{Cmax} + \dfrac{P_{t,R(t)}^{Cmin} - P_{t,R(t)}^{Cmax}}{1-\alpha} \left(\dfrac{soc_t + SOC_0}{2} - \alpha \right) & t = 1 \\ P_t^{Cmax} + \dfrac{P_{t,R(t)}^{Cmin} - P_{t,R(t)}^{Cmax}}{1-\alpha} \left(\dfrac{soc_t + soc_{t-1}}{2} - \alpha \right) & \forall t \in \mathcal{T} \setminus \{1\} \end{cases} \qquad (13)$$

Funded by the European Union

$$p_t^d \leq \begin{cases} P_{t,R(t)}^{Dmax} \dfrac{SOC_0}{\alpha^D} & t = 1 \\ P_{t,R(t)}^{Dmax} \dfrac{soc_{t-1}}{\alpha^D} & \forall t \in \mathcal{T} \setminus \{1\} \end{cases} \quad (14)$$

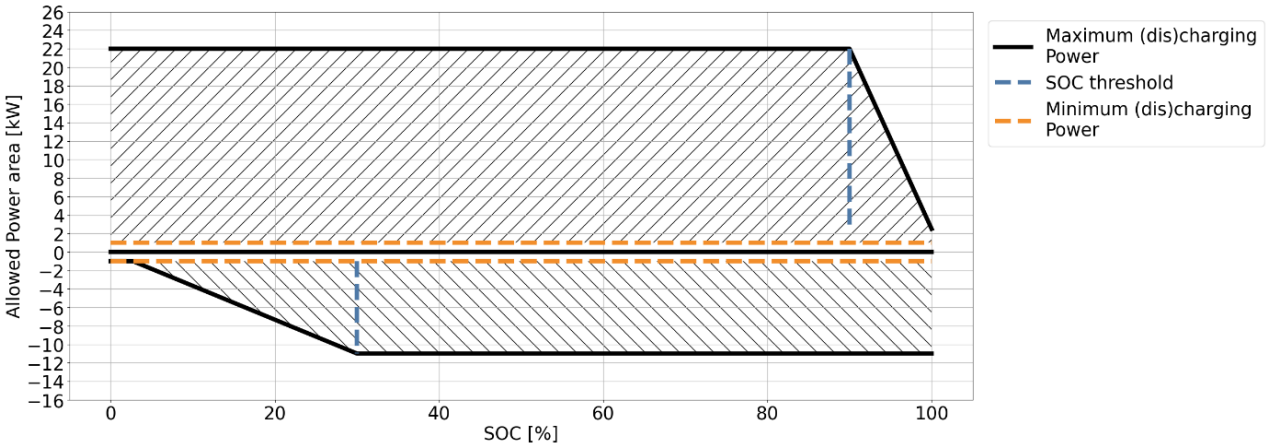


Figure 14. Charge and discharge maximum power profiles. Scratched zones show feasible areas of both, charging and discharging power variables, p_t^c and p_t^d respectively.

3.2.5. Optimization problem

After introducing the objective function and the set of constraints, the proposed User Smart Model will be set as the optimization problem and defined as follows:

Min $F(p^c, p^d, s^{slack})$

Subject to,

Equations (3), (4), (5), (6),(7), (8), (9), (10), (11), (12), (13), (14)

Which corresponds to a Mixed Integer Linear Optimization Problem (MILP).

3.3. Use case: Oporto Taxi Dataset

In this section, a general description of the use case is provided.

The Oporto Taxi dataset [59] [60] was released in context of the Porto European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2015 edition) and the Kaggle competition Taxi Trajectory Prediction (I).

This dataset provides a whole year of information, from July 1st 2013 to June 30th 2014, of 442 taxis and their trips. Moreover, the timestamp is registered, the call type data --differentiating between central dispatched, specific taxi station or other kind of calls--, and the origin stand identifier. The original dataset includes the GPS coordinates (WGS84 format) recorded every 15 seconds. This GPS track will help to obtain trip distance, duration, mean speed, and to estimate the energy demand.

Funded by the European Union

Due to the high number of possible starting and ending trip locations –any street is suitable for being a beginning or ending point of a taxi trip–, it is assumed that all trips start and finish in taxi stations. Another assumption is that all taxi stations have a charging point per parking slot. Under these considerations, the data is prepared as it is showed in **Figure 17**. All sub-processes appearing in **Figure 17** are described below:

- **Process 1:** The GPS location is recoded every 15 seconds, this provides an approximation of the distance –considering geodetic distance– and travelled time. These two measurements help us to obtain the taxi mean speed. Even though the shortest path between two points do not exactly fit with the idea of EV trip, as records are registered every 15 seconds, final distance should approximately match with the real trip distance.

  Process 2: During data processing procedure, the vehicle model is assumed to be the Nissan Leaf e+ N-connecta model[1] (Spanish version), which features a battery size is 62 kWh. The 62kWh battery size is assumed to be reasonable because it aligns closely with the mean value of sampled EV battery capacities available today, according to the European Alternative Fuels Observatory (EAFO) [89]. External temperature is assumed to be 25°C. Energy consumption calculations are also based on Nissan Leaf e+ autonomy[2], utilizing distance travelled and mean speed obtained from process 1.

- **Process 3**: The original dataset is processed such that a trip starts and finishes at specific taxi stands and each trip can include several travel to different locations. However, the original dataset lacks identification of whether a taxi ends its trip at a stand or not. To solve this ambiguity, we established a criterion: if the j-th trip starts at stand A and the k-th trip starts at stand B, where j<k, and there are no intermediate trips starting at any other stand, then we define trip (j,k) as starting at stand A and concluding at stand B. **Figure 15** depicts this concept, denoting Stands A and B as potential starting and ending points for merged trips.

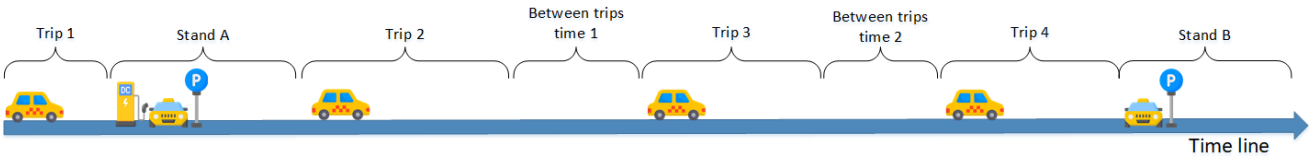


Figure 15. Graphical representation of Oporto's raw dataset. Car pics were obtained from Flaticon.

It is also assumed that if the time gap between trips i and $i+1$ (where $j < i < i+1 < k$) exceeds 45 minutes, we consider that trip i-th has finished at an unidentified position. This unidentified position is presumed to lack chargers. Following the example in **Figure 15**, if the time interval between Trip 2 and Trip 3, is under 45 minutes, they will be combined. However, if this time interval exceeds the 45-minute threshold, it is assumed that the car stopped at a

[1] https://configurador.nissan.es/leaf
[2] https://leaf-range-calculator.nissan.es/es/spain/autonomy

Funded by the European Union

location without being possible to charge the vehicle. **Figure 16** represents the application of this assumption, building on the initial situation shown in **Figure 15**.

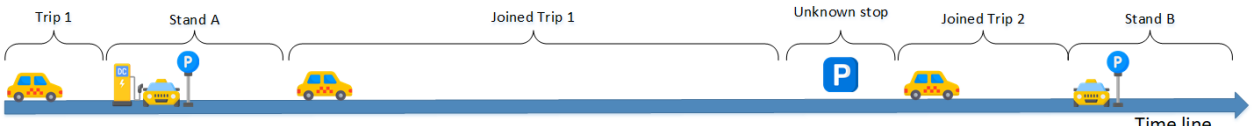


Figure 16. Trip Merging at Taxi Stands and Identifying Unidentified Stops: Using Figure 15 Scenario with the Assumption of 'Between Trips Time 1' < 45 and 'Between Trips Time 2' ≥ 45. Car pics were obtained from Flaticon.

After processing the data, 84 taxis were selected to train and test the algorithms. The dataset used to forecast the next week trajectories was formatted to time series. **Table 7** shows the time series data that the forecast tool expects, where location label "-1" stands for on the road situation. This final database includes time, one previous week of Location and Energy Consumption variables.

Table 7. From trips register to time series profile of a single taxi.

Starting time	Starting Location	Ending Time	Ending Location	Energy Consumed
20/08/2023 07:45	A	20/08/2023 10:15	B	15 kWh

Date	Location	Energy Consumed
20/08/2023 07:30	A	1.5
20/08/2023 07:45	-1	1.5
20/08/2023 08:00	-1	1.5
…	…	
20/08/2023 10:00	-1	1.5
20/08/2023 10:15	B	1.5

Funded by the European Union

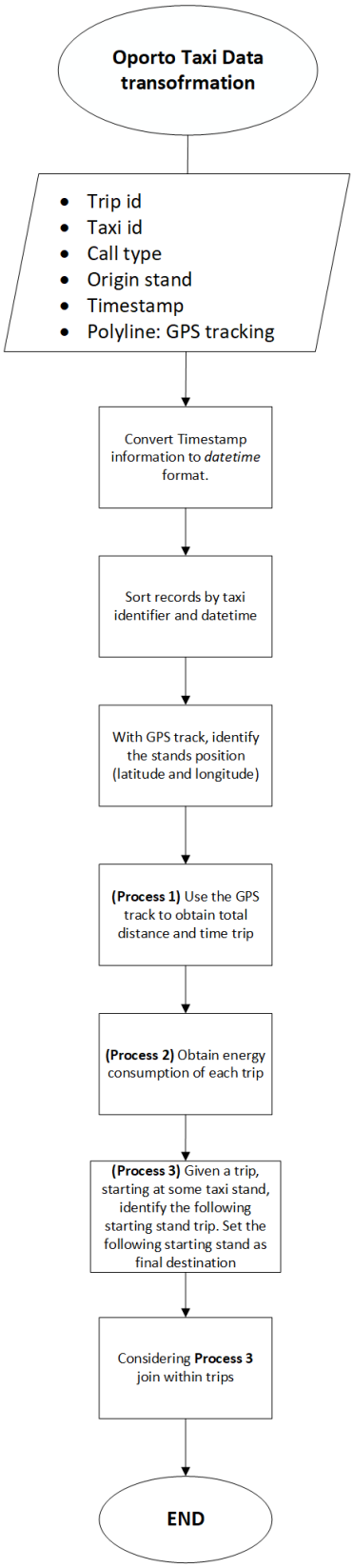


Figure 17. Oporto's Data set pre-processing schematic.

Funded by the European Union

3.4. Results

In this section results are presented for each of the three different proposals explained previously. To code the Forecasting models, we used Python and, more precisely, libraries TensorFlow and Scikit-Learn. Meanwhile, we used GAMS and solver CPLEX to obtain the optimization results. The forecasting models have been launched on a an Intel(R) Core(TM) i7-7800X CPU with 32GB of RAM, and the optimization has been done using an Intel(R) Core(TM) i5-8265U with 8GB of RAM.

3.4.1. Forecasting Performance

The results of our study have revealed outcomes that, while falling short of initial expectations, provide valuable insights into the complexities of EV mobility forecasting. Recognizing this, we present both the strengths and limitations inherent in our methodology. Most notably, given the nature of the data set (i.e., predicting taxi travel), which can have significantly more locations that need to be considered than for private vehicle transport, predicting the location and energy consumption during that travel is more challenging than for typical private vehicle transport. Without having access to private vehicle transport data we had to use the taxi data. As will be described, we envision a marked performance improvement when applying this method to personal private vehicle data.

The location forecasting algorithm has been tested against the historic mobility dataset of several different taxis (two of which are explored in detail below). The high computational time required for the training of the LSTM RNN has been inconvenient when trying to obtain the model for each taxi. This high computational requirement is explained by the fact that the LSTM Network is trained with 9 months of 5-minute frequency data. Moreover, the LSTM Network is trained in such a manner that 14 days is the input data to predict 1 day. The number of features is linearly dependant on the number of locations,

$$N_{features} = 21 + 2 \cdot N_{locations},$$

The first 21 fixed features are:

- 7 binary variables indicating day of the week,
- 12 binary variables indicating month,
- 2 continuous variables indicating hour of the day t_{sin}, t_{cos}.

Each location has two features, l_{t-288} and l_{t-2016}, which indicate the location of the car in the same timestep of the previous day in the previous week, respectively. The training of the NN was limited to 10 epochs in order to set a balanced value between computational time and performance of the model.

Funded by the European Union

Table 8. Results of the location forecasting model.

Training	***Taxi A***	***Taxi B***
Training accuracy	0.75	0.50
Training loss (Sparse Categorical Crossentropy)	1.99	1.93
Training time per epoch [minutes]	59	45
Test	***Taxi A***	***Taxi B***
Balanced Accuracy	0.53	0.17
Number of real trips / Number of predicted trips	77 / 34	70 / 14

The two taxis showed similar training time, with the accuracy of the model stabilizing rapidly (epoch 3 already showed almost identical accuracy and loss values as epoch 10). The model have shown difficulties to adapt to the nature of the taxi data. This is confirmed when using the model against the test dataset. The total number of predicted trips and the accuracy values for the locations were very low, which we consider unacceptable for commercial implementation. Our study employed a one-hot encoding for the GPS coordinate of the vehicles, along with a series of assumptions regarding the trips of the taxis that introduced limitations to the time series.

Regarding the trip energy consumption forecasting model, the three algorithms developed all had relatively low accuracy as shown in Table 9. The main reason is that the method for combining trips in order to limit locations to only taxi stands means that the energy consumed during trips is highly unpredictable since it could involve a single sub-trip or many sub-trips. As a result, simplifications to improve results for the location forecasting have negatively affected the energy consumption forecasting. Again, this is unlikely to be an issue with private transport thus the methods developed here still have merit.

Funded by the European Union

Table 9. Results on the trip consumption model.

Training		
SVR	**Taxi A**	**Taxi B**
Training accuracy (RMSE)	0.58	0.56
Training time per epoch	0.76s	0.93s
Random Forest	**Taxi A**	**Taxi B**
Training accuracy (RMSE)	0.49	0.53
Training time per epoch	14 s	17 s
LSTM	**Taxi A**	**Taxi B**
Training accuracy (RMSE)	0.62	0.46
Training time per epoch	3s	3s
Test		
SVR	**Taxi A**	**Taxi B**
RMSE	0.63	0.59
MAPE	2.18	0.91
Random Forest	**Taxi A**	**Taxi B**
RMSE	0.56	0.58
MAPE	2.11	2.62
LSTM	**Taxi A**	**Taxi B**
RMSE	0.89	0.76
MAPE	1.70	1.28

The training time is greatly reduced in this model given that the data structure is far more simple, the model is simply obtaining a target variable from the previous 50 records, whereas in the location model the algorithm is trained to use a full week of data to predict one day of target locations.

3.4.2. Optimization Performance

In this section the performance of the proposed Medium-term optimization approach for scheduling EV charging sessions (called User Smart Model from now on), described in Section 3.2, is evaluated. To facilitate a comparative analysis, we introduce two additional charging procedure:

- **Immediate charging Model**: Vehicles are promptly charged upon connection to any charging point, aiming for full battery charge as soon as possible.
- **CSO Smart charging Model**: Upon plugging in, the algorithm ensures a minimum SOC of 85% by the end of the charging session. This allows for charge shifting and energy price optimization.

The CSO Smart Model does not consider the discharging energy penalization cost, C^{health}, in the objective function, because of the Charging System Operator is not incentivized to care about a user's battery degradation. Additionally, it is interesting to remark that User Smart Model is neither obligated to charge the vehicle every time it is parked at a taxi station nor to fully charge the vehicle, or to guarantee a minimum SOC, at the end of the charging session.

Funded by the European Union

For testing purposes, a sample of 84 different taxis are tested using these three models. Moreover, each vehicle and model initialization includes four different state of charge values: 30%, 50%, 70% and 90%. Finally, given the number of different usage curves and the considered initial State of Charge values, a total of 366 tests were performed.

To facilitate comparison, we defined four Key Performance Indicators (KPIs) listed below:

- **Total Energy Purchase Cost [€]:** How much the user paid for the energy they used to charge their vehicle, also considering compensation for discharging the battery (i.e., bidirectional charging).

$$TEC_{taxi} = \sum_{t \in \mathcal{T}} \Delta_t \cdot p_t^c \cdot \lambda_{t,R(t)}^{buy} \quad [€] \tag{15}$$

- **Mean Energy Purchase Cost [€/kWh]:** This indicator provides a measure of the average cost of energy, taking into account both the price and the quantity of energy consumed. The KPI is defined according to Expression (16).

$$\bar{\lambda}_{taxi} = \begin{cases} 0 & TEB_{taxi} = 0, \\ \dfrac{\sum_{t \in \mathcal{T}} \Delta_t \cdot p_t^c \cdot \lambda_{t,R(t)}^{buy}}{\sum_{t \in \mathcal{T}} \Delta_t \cdot p_t^c} & TEB_{taxi} > 0 \end{cases} \tag{16}$$

- **Battery Degradation associated cost [€]:** The calculation methodology described in FLOW deliverable 4.3, Section 4.2 is applied to obtain the Battery Degradation Cost.

- **Total Cost [€]:** Sum of the energy bill and the associated battery degradation cost. It is calculated as defined in Equation 16, where BD_{taxi} represents the battery degradation cost.

$$TC_{taxi} = BD_{taxi} + \sum_{t \in \mathcal{T}} \Delta_t \cdot \left(p_t^c \cdot \lambda_{t,R(t)}^{buy} - p_t^d \cdot \lambda_{t,R(t)}^{sell} \right) \quad [€] \tag{176}$$

Differences between the smart charging model are also used to compare the results from the three models, as illustrate the expression (187):

$$\begin{aligned} \Delta_{IM}(KPI) &= KPI^{Smart} - KPI^{Immediate} \\ \Delta_{CSO}(KPI) &= KPI^{Smart} - KPI^{CSO} \end{aligned} \tag{187}$$

and the percentage of change (PC)

$$PC_{IM}(KPI) = \frac{\Delta_{IM}(KPI)}{|KPI^{IM}|} \cdot 100 \tag{198}$$

Funded by the European Union

$$PC_{CSO}(KPI) = \frac{\Delta_{CSO}(KPI)}{|KPI^{CSO}|} \cdot 100$$

The following list collects important assumptions that were used in the analysis:

- All taxi stations have available chargers, i.e. a taxi can always plug-in when it arrives.
- The charger characteristics are the same across taxi stations (i.e., charging power limits).
- The vehicles have a battery capacity of 62 kWh.
- The initial datetime for each test set is randomly selected to recreate different scenarios, with the condition that a minimum of 15% of the sampled data must show an EV usage, i.e. driving. That is done considering values from EV Consumption dataset variable – blue line in **Figure 18**.
- The energy prices data correspond to Spain's Voluntary price for the small consumer (PVPC[3]) from 2022 to 2023, while the Oporto data span July 2013 to June 2014. To match these, we adjusted the date range for the prices to match the Oporto data (i.e., maintaining weekdays and weekend periods). For instance, if the historical data is from 2014-06-24 01:35 to 2014-06-31 01:30, the selected energy price date indices are in the interval 2023-06-20 01:00 - 2023-06-27 01:00, maintaining the same weekdays (from Thursday to Thursday).

Due to the impact of the initial State of Charge on Immediate Charge and CSO Smart models, it is important to quantify and compare the performance of the User Smart Model and the charging limited models. For this reason, when comparing Immediate Charging and CSO Smart models against the User Smart Model, the first twenty-four hours horizon period is also analysed. Including this period guarantees that more than 94% of experiments remained in a taxi station, i.e. charge availability, for at least 1 hour.

Table 10. Parameter values in optimisation tests.

Symbol	Description [Units]	Value
	Sets	
$\mathcal{T}$	Set of time steps	$\mathcal{T} = \{1, \dots, 2016\}$
$\mathcal{L}$	Set of locations that the EV could be parked	$\mathcal{L} = \{1, \dots, 68\}$
$\underline{\mathcal{L}}$[4]	Subset of forecasted locations as parking point for the EV	$\{0,1, \dots, 68\}$
	Scalars	
Δ_t	Time step duration [hours]	5/60
M	An arbitrary large constant	10^9
β	EV energy demand security parameter	1
α^C	SOC threshold where EV maximum charge power decreases [%]	0.9
α^D	SOC threshold where EV maximum discharge power decreases [%]	0.3
E^B	Battery capacity [kWh]	62

[3] https://www.esios.ree.es/en/analysis/1739?compare_indicators=1001

[4] Each of the 84 taxis have a different subset of possible locations, $\underline{\mathcal{L}} \subseteq \mathcal{L}$, for this reason, in Table 4 this set includes all the stations visited by any of the 84 taxis.

Funded by the European Union

SOH	Actual State of Health [%]. Parameter used to obtain degradation.	95
C^{sell}	Penalization cost when discharging [€/kWh]	0.053
η^{ch}	Battery charging efficiency [%]	0.9
η^{dch}	Battery discharging efficiency [%]	0.9
$\overline{SOC}$	Maximum SOC value [%]	1
$\underline{SOC}$	Minimum SOC value [%]	0.2
SOC_0	Initial SOC [%]	0.3, 0.5, 0.7, 0.9
SOC_T	Minimum SOC at the time window end [%]	0.7
P^{Cmax}	Maximum charging power when the EV is plugged in [kW]	$22 \quad \forall t \in \mathcal{T}$
P^{Cmin}	Minimum charging power when the EV is plugged in [kW]	$1 \quad \forall t \in \mathcal{T}$
P^{Dmax}	Maximum discharging power when the EV is plugged in [kW]	$11 \quad \forall t \in \mathcal{T}$
P^{Dmin}	Minimum discharging power when the EV is plugged in [kW]	$0.3 \quad \forall t \in \mathcal{T}$

Figure 18 presents the main time resolved inputs of the optimisation problems (provided in table form in **Table 10**). The orange zones represent time periods when the vehicle can be charged (positive) or discharged (negative). Energy values are obtained considering the relation $Energy^{Cmax} = \Delta_t P^{Cmax}$. In addition, the blue line represents the energy consumed by the EV when driving.

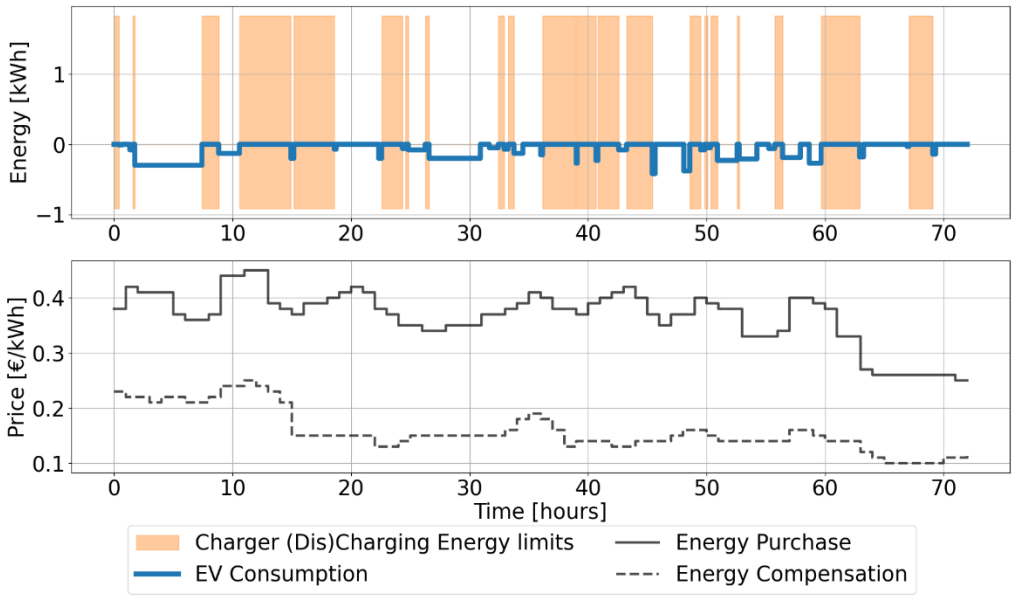


Figure 18. First three days EV trip energy consumption (blue line), the charging energy (positive) or discharging (negative) limits (orange), the purchase energy cost (solid black) and injection compensation (dashed black) for a single taxi.

Funded by the European Union

3.4.2.1. User Smart Model results analysis

Testing the User Smart Model across 336 scenarios with various initial SOC values and EV usage patterns revealed interesting outcomes. Table 11 presents the distribution of KPIs throughout a weekly timeframe and Figure 19 displays a graphical representation of weekly KPIs obtained in each studied model discharging, and being compensated.

Table 11. User Smart Model KPIs across all simulations.

	Energy Purchase Cost [€]	Mean Energy Purchase Cost [€/kWh]	Degradation [€]	Total Cost [€]
Mean	12.15	0.14	8.35	19.61
Std. Dev.	10.04	0.09	0.49	10.79
Minimum	0.79	0.03	6.87	4.12
25%	4.66	0.06	8.11	11.69
Median	8.80	0.11	8.41	16.30
75%	16.94	0.19	8.67	25.00
Maximum	54.88	0.38	9.49	63.76

Considering the 366 total scenarios for each implementation model, a box and whisker plot is used to develop an understanding for the performance of the three different implementation models (Figure 19). To answer this, we conducted a paired Wilcoxon Signed Rank Test, with the null hypothesis states no mean value difference.

To analyse statistical differences between the weekly distributions of the three implementation models, a paired Wilcoxon Signed Rank Test was conducted assuming the null hypothesis that there is no mean value difference. All User Smart Model possible comparisons presented p-values less than 0.0001, leading to the rejection of the null hypothesis. In other words, we can say that the User Smart Model significantly reduces all the KPIs considered, when it is compared with Immediate Charge Model and CSO Smart Model.

Funded by the European Union

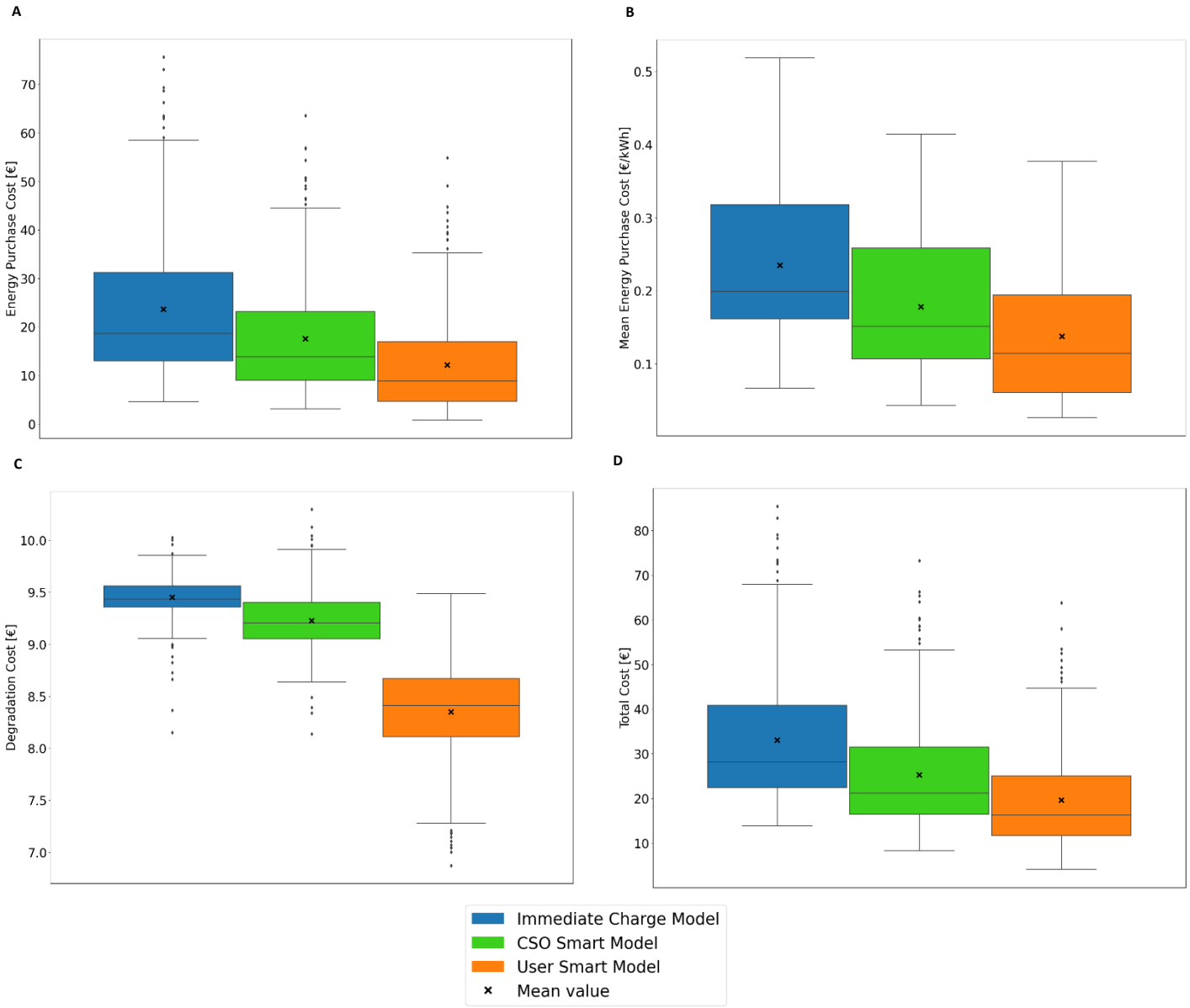


Figure 19. KPI distribution through the sample of the 336 simulations for each model considered. Panel A: Energy Bill, Panel B: Mean energy purchase cost, Panel C: Associated degradation cost, Panel D: Total Cost.

3.4.2.2. Individual taxi analysis

Figure 20 shows the performance of a vehicle and the related energy prices considering an initial State of Charge of 30% over a week. In these curves, we observe prolonged periods where the User Smart Model either refrains from charging or just charges a minimum amount of energy for subsequent trips. Equation (5) permits a SOC value below the minimum required, but incurs an economic penalty denoted by M. As a result, the algorithm consistently aims to charge the vehicle with enough energy to guarantee a SOC level above or equal to the minimum SOC ($\underline{SOC}$) at the beginning of next charging session. Compared with the Immediate Charging Model, smart models rarely fully charge the battery

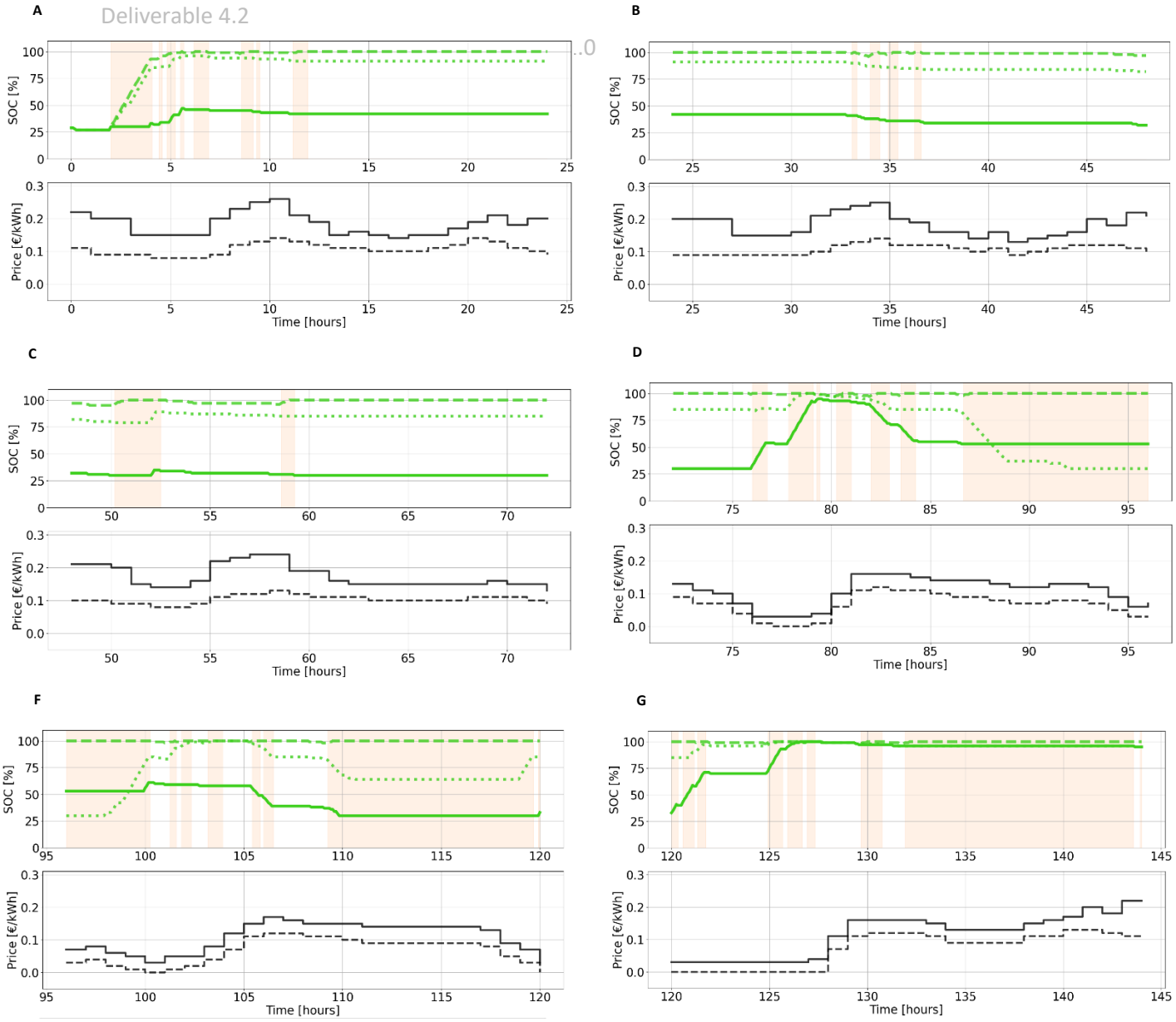
Funded by the European Union

except in specific situations —Subfigures D, G, and H—, and all excess energy is injected into the distribution network. It is interesting to compare the behaviour of the CSO Smart Model and the User Smart Model (panels D and F). The User Smart Model provides users with information that they need to make more strategic multi-day decisions. While the CSO Smart Model may allow for bidirectional strategies to generate additional revenue, the value of that service is severely limited if the vehicle must achieve a specific SOC value by the end of the charging sessions (i.e., 85% for the CSO Smart Model). Notably, the average SOC for the User Smart Model in the example below is 54% while for the CSO Smart Model it is 83% and for immediate charge is 98%. While allowing dynamic SOC levels at the end of the charging session can provide value, as in the User Smart Model example, this also creates a situation where in an operational environment, a buffer needs to be built into the estimation process to ensure that the driver's mobility needs are met in the case of variability in their driving patterns.

Again, following the example in Figure 20, in terms of total energy cost, the User Smart Model resulted in 5.75€ whereas the other models obtained 15.08€ (Immediate Charge Model) and 13.24€ (CSO Smart Model). Additionally, the CSO Smart Model discharged more energy from the EV battery and hence the one who charged more energy –User Smart Model has charged 119.53kWh and injected 41.1kWh, the CSO Smart Model 144.32kWh and 52.87kWh respectively, and Immediate Charge Model charged 89.4kWh. The more conservative behaviour when discharging energy and the no obligation of reaching a specific desired SOC by the end of each charging sessions permitted the User Smart Model to perform better in Degradation associated cost, reaching a value of 8.23€. In contrast, the CSO Smart Model obtained the worst result with 9.75€ (Immediate Charge Model: 9.69€). Considering the charged/discharged energy and battery degradation, the User Smart Model obtained a Total Cost value of 9€, whereas CSO Smart and Immediate models scored 17.82€ and 24.77€, respectively. Finally, there was a remarkable reduction in the mean energy purchase cost, from 0.17€/kWh in the Immediate Charge Model to 0.048€/kWh in the User Smart Model (CSO: 0.092 €/kWh).

Figure 21 displays a second set of charging profiles, this time with an initial State of Charge of 70%. The User Smart Model and CSO Smart Model exhibit similar behaviour from panel A to panel D, but just as with the results in Figure 20, the User Smart Model algorithm limits its SOC – only charging the vehicle once from panel F to panel H – while the CSO Smart Model is forced to charge three times to maintain the rule of 85% of SOC at the end of each charging session. Specifically, the full week energy purchased cost for the User Smart Model is 5.22€, whereas for CSO Smart Model it is 7.47€. The Degradation associated Cost in CSO Smart Model is higher than the User Smart Model (CSO: 9.13€, User: 8.85€), because the CSO Smart Model charged and injected more energy. In general terms, the Immediate Charge Model offered poor results, resulting in 22.74€ for the Total Cost KPI while the others resulted in 12.98€ and 10.88€ for CSO and User Smart Model, respectively. Finally, the mean energy cost was 0.21€/kWh, 0.09€/kWh and 0.074€/kWh for Immediate Charge, CSO Smart Model and User Smart Model, respectively.

Funded by the European Union



Funded by the European Union

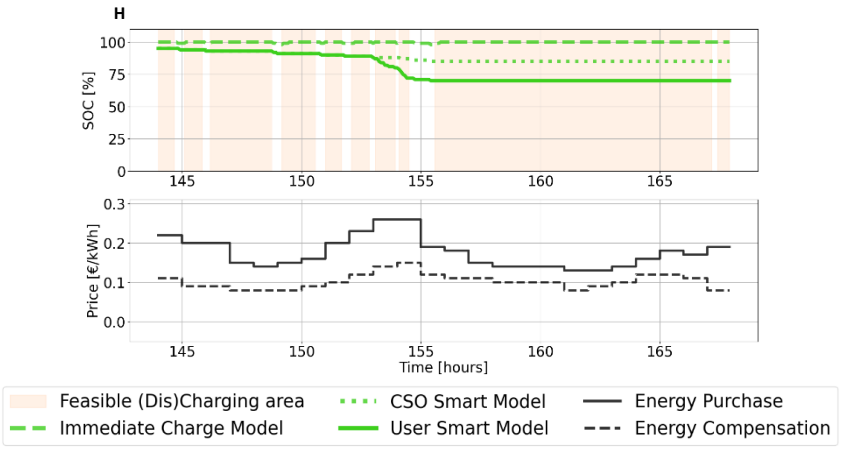
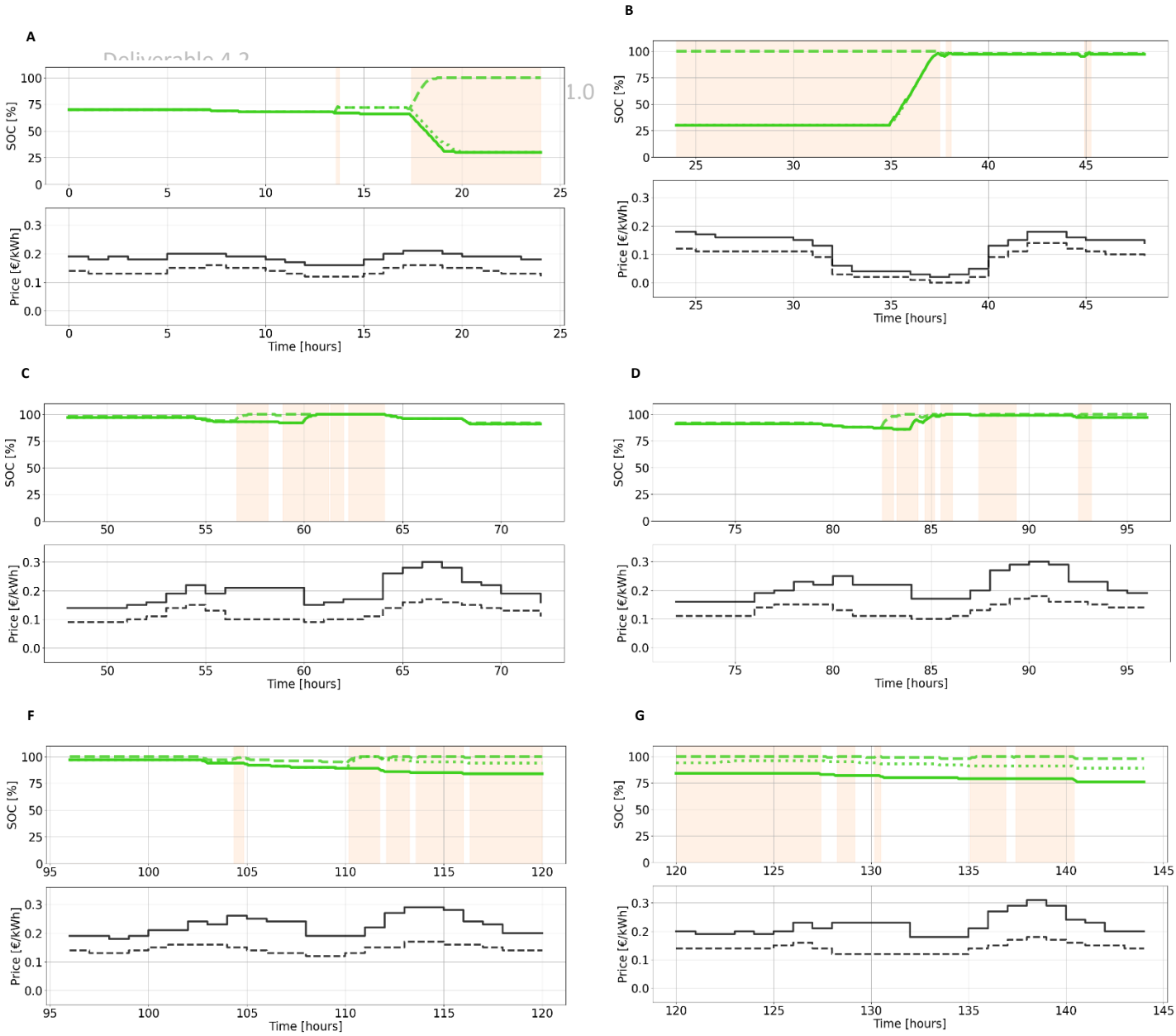


Figure 20. Immediate Charge, CSO and User Smart models comparison over a week-ahead optimization grouped by periods of twenty-four hours. Taxi 20000267 between 12-11-2013 06:00 and 19-11-2013 05:55, whereas energy prices correspond to the period 07-12-20

Funded by the European Union



Funded by the European Union

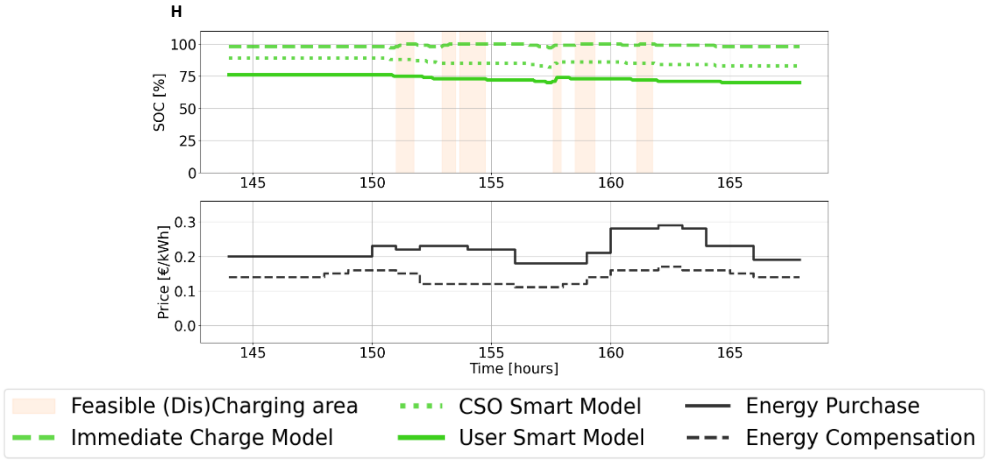


Figure 21. Immediate Charge, CSO and User Smart models comparison over a week-ahead optimisation grouped by periods of twenty-four hours. Taxi 20000596 between 22-02-2014 02:00 and 01-03-2014 01:55, whereas energy prices correspond to the period 25-02-2023 02:00

3.4.2.3. User Smart Model versus Immediate Charge Model

The results, summarized in Table 12, showcase the mean paired differences (User Smart Model – Immediate Charge Model) for KPIs over both a week and a 24 hour period, in brackets the standard deviation. More precisely, negative values point out the advantage of using the User Smart Model. These values consistently depict reductions when considering the User Smart optimization Model. The large standard deviation can be explained be the presence of extreme values.

Table 12. One-week and first twenty-four hours mean KPIs difference (User Smart Model vs Immediate Charge Model) obtained over the whole set of simulations.

	Energy Purchase Cost [€]	Mean Energy Purchase Cost [€/kWh]	Degradation [€]	Total Cost [€]
1 Week mean Difference (std. dev.)	-11.48 (5.59)	-0.1 (0.04)	-1.1 (0.48)	-13.46 (5.29)
1st Day mean Difference (std. dev.)	-6.31 (4.95)	-0.06 (0.05)	-0.26 (0.21)	-6.79 (5.1)

Combining results from Table 12 and Table 13, we can conclude the advantages of using the User Smart Model against the Immediate charging procedure. The greater reduction in first-twenty four hours can be explained by the initial State of Charge values, which are coercing the Immediate Model to charge a large amount of energy (iSOC of 30% implies charging 43kWh as soon as possible). These results are consistent with findings from Section 3.4.2.2 (User Smart Model analysis).

Funded by the European Union

Table 13. One-week and first twenty-four hours mean KPIs percent change (User Smart Model vs Immediate Charge Model) obtained over the whole set of simulations.

	Energy Purchase Cost [€]	Mean Energy Purchase Cost [€/kWh]	Degradation [€]	Total Cost [€]
1 Week mean Percentage Reduction (std. dev.)	-53.27 (14.66)	-45.16 (17.80)	-11.62 (5.03)	-42.52 (10.3)
1st Day mean Percentage Reduction (std. dev.)	-78.08 (25.69)	-56.36 (39.21)	-17.7 (13.45)	-66.17 (32.3)

3.4.2.4. User Smart Model versus CSO Smart Model

The CSO Smart Model presents an advantage against the CSO Smart Model when considering energy price arbitrage within each charging session, but the EV should finish the connected time period with, if it is possible, a minimum SOC of 85% for every stop location.

Table 14 shows the mean difference between User Smart Model and CSO Smart Model, in both weekly and daily periods. The User Smart Model presented better results and the findings indicate that the majority of energy bill savings occurs during the first twenty-four hours (likely attributed to equilibration of the initial SOC conditions). The increase of the Degradation cost can be explained by the calendar ageing effect on battery health. Holding batteries at a lower SOC will reduce the impact of the calendar ageing and the User Smart Model allows for the SOC to be lower than the limit imposed for the CSO Smart Model – 85% at the end of each charging session – thus resulting in a greater reduction for the User Smart Model.

Table 14. One-week and first twenty-four hours mean KPIs difference (User Smart Model vs CSO Smart Model) obtained over the whole set of simulations.

	Energy Purchase Cost [€]	Mean Energy Purchase Cost [€/kWh]	Degradation [€]	Total Cost [€]
1 Week mean Difference (std. dev.)	-5.35 (2.98)	-0.04 (0.02)	-0.88 (0.49)	-5.66 (2.9)
1st Day mean Difference (std. dev.)	-4.21 (3.91)	-0.03 (0.04)	-0.22 (0.19)	-4.44 (4.06)

In terms of relatives changes, Table 15 shows the percent change when moving from CSO Smart Model to User Smart Model. It is remarkable the standard deviation values when considering the first twenty-four hours window, this could be explained by: (1) occasionally the CSO Smart Model could take advantage of the energy stored in the battery by discharging it and charge it afterwards, however the User Smart Model is not encouraged to do that due to the degradation factor included in the objective function; (2) The CSO Smart Model is compelled to charge the battery up to the 85% of SOC, meanwhile the User Smart Model could decide to limit charging.

At the same time, the relative difference of Degradation decreased from one day analysis to one week analysis**.** This could be explained, as before, by the rules of CSO Smart Model. Moreover, when the

Funded by the European Union

initial State of Charge is low, the Immediate Charge and the CSO Smart models must make an effort to charge the battery and, hence, a higher degradation is obtained.

On average, the Mean Energy Purchase Cost is reduced around 26% for the whole week. Although the standard deviation is still large, the sample distribution does not present a remarkable extreme value. The first twenty-four hours presents almost a 50% percent reduction; however, the deviation value is considerably larger, but explained by the 122 samples from User Smart Model that did not charge the vehicle.

Table 15. One-week and first twenty-four hours mean KPIs percent change (User Smart Model vs CSO Smart Model) obtained over the whole set of simulations.

	Energy Purchase Cost [€]	Mean Energy Purchase Cost [€/kWh]	Degradation [€]	Total Cost [€]
1 Week mean Percentage Reduction (std. dev.)	-35.39 (18.02)	-26.49 (16.56)	-9.5 (5.2)	-24 (11)
1st Day mean Percentage Reduction (std. dev.)	-70.31 (33.64)	-45.40 (42.38)	-15 (12.4)	-54.48 (50.3)

3.5. Summary and future work

This report describes the tools and methodologies that have been implemented in order to obtain the optimal management for the charging session of an EV, considering the future usage of the vehicle. For this purpose, we have developed a *forecasting model* aimed at predicting EV locations for the upcoming week and an *optimization model* designed to optimize battery management based on these forecasts. A user can use this tool when they arrive at a charging station to automatically select a desired SOC and even predict the likely departure time. These forecasted values can be used as the default for smart charging and bidirectional charging strategies. This has the potential to simplify the user experience and increase potential flexibility offering from EVs.

Our exploration into the forecasting model revealed inherent challenges in accurately predicting the future locations of EVs. Regarding the location forecast, the complexity of spatial-temporal dynamics, combined with external factors influencing EV mobility which lack in our input data, contributed to poor forecast results. The nature of the Taxi data, with a large number of possible drop-off/pick-up/charging locations, results in a poor fit with a time-series classification approach. The developed methods are likely to have marked performance improvement when considering a private transport vehicle, which often consider fewer locations; however, the authors could not find appropriate private transport data to use for this analysis. Additionally, the assumptions made to reduce the number of locations in an effort to reduce forecasting error for the location prediction have negatively impacted the trip energy consumption prediction by significantly increasing variability in the energy consumption per trip (which can comprise few or many subtrips). The implications of these assumptions are that for every trip between known locations, the amount of time, distance, and energy consumption can be very different. The balance between simplifying the dataset with location clustering, and limiting the negative impact on the forecast for trip energy consumption, was difficult for the taxi data, but could be limited or avoided by using personal vehicle data.

Funded by the European Union

In contrast, the optimization model has shown success in effectively managing EV battery usage based on accurate energy consumption forecasts. Across the 336 different scenarios, the results obtained with the proposed optimal energy management (User Smart Model) have been compared with two other management strategies, Immediate Charge and CSO Smart Model, and the User Smart Model shows an improvement in all studied KPIs: purchase energy cost, mean purchase energy value, battery degradation associated cost, and total cost.

While the optimization model has demonstrated that with an accurate forecast of weekly energy consumption, it can improve charge management, this highlights the need to further study the forecasting model. The improvement of the forecasting model is a critical step in unlocking the full potential of the optimization model in real-life scenarios.

While the proposed models have been used in the literature for other different but related forecasting activities with better results, the authors think that the nature of the input data (i.e., the use of taxi data instead of private transport) plays a significant role. For future work, the proposed methods should be applied to personal transport data to confirm that this will improve the results.

In conclusion, our study serves as a foundation for future advancements in EV management. The symbiotic relationship between forecasting and optimization models, particularly as it relates to the user-centric approaches, highlights the relationship of accurate predictions and tangible real-world applications which can contribute to both economic and environmental sustainability.

4. Day-ahead building EV charging demand forecast (EATON)

The Centre of Intelligent Power (CIP) at Eaton provides day-ahead EV charging demand forecasting services to customers and in-house projects. These services aim to provide deeper insights into the patterns of EV charging demand loads.

Since EV charging will significantly increase the load on buildings and grids, accurate EV charging power forecasting can help the building administrators better utilise energy storage units, optimise energy usage, and plan for the risks associated with peak power.

The EV charging demand forecasting algorithms developed by Eaton CIP have been designed with a practical approach in mind. Specifically, statistical and ensemble methods have been prioritised for their simple structures, minimal need for feature engineering, and effectiveness in typical forecasting tasks. However, machine learning and deep learning (DL) based models and feature engineering techniques are also being experimented with to improve the performance of EV load forecasting.

This section will summarise the research and experiments conducted for selecting and evaluating EV load forecasting models for day-ahead building EV charging demand forecast.

Funded by the European Union

4.1. State of the Art

4.1.1. Forecast Models

Forecast models analyse historical time series data to predict future development of energy demand. These models capture the relationship between past and future values using parameters, taking into account various aspects of a time series, such as seasonal patterns or current energy output.

EV charging demand forecast models are adaptations of general forecasting models, designed to capture the specific data characteristics of the energy domain. When it comes to EV charging demand forecasting, it is challenging to find a universally accepted method that suits all possible application scenarios. This is because EV charging behaviours can vary under different scenarios. Therefore, our literature review has prioritised models that have low input requirements and have been extensively researched. As we progress with the review, we will gradually shift our focus to machine learning and deep learning-based models. These models possess powerful learning and predicting capabilities, which may further enhance EV charging demand forecasting abilities.

Statistical models

Statistical models have been widely used for univariate time series forecasting and have been successfully applied to EV charging demand forecasting. These models have minimal input requirements, as they only consider the historical data of the time series and do not require other complex factors that influence EV charging power. However, incorporating exogenous predictors into statistical models is also supported and can help improve the accuracy of the forecasts.

AutoRegressive Integrated Moving Average (ARIMA) has been widely used to forecast time series in production and as a benchmark. In [90], the authors proposed time-series seasonal ARIMA models for predicting aggregated EV charging station load. Their main objective was to identify the most suitable seasonal ARIMA model for various scenarios, without taking into account exogenous variables. Another ARIMA model, presented in [91], predicts the electric power consumption of conventional electrical load and the charging demand of EVs. This model utilises daily driving patterns and distances as inputs to estimate the expected charging load profiles, and then employs ARIMA to forecast future charging demand.

Additionally, when dealing with EV charging demand data, it is important to consider and analyse seasonal patterns and recurring patterns and exploit available features when designing forecasting models. [92] combines two kinds of Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model (SARIMAX) together with three kinds of training-forecasting procedures to identify the combination that provides the best long-term forecasts. SARIMAX is an extension of the ARIMA model, and the authors include both seasonality and exogenous variables (e.g., holidays and lagged energy consumption data) in their models to enhance the forecasting ability and accommodate for non-stationary data.

It should be noted that while the examples above utilise ARIMA or its variants as the forecasting models, other statistical models can also be used for predictions. Finding the best model for forecasting is not a one-size-fits-all approach and depends on personal knowledge and preference.

Funded by the European Union

Machine Learning Models

To address the complexities associated with factors that impact EV demand forecasting, machine learning techniques have been introduced to leverage their powerful learning capabilities. One advantage of ML models is their ability to capture hidden non-linear dependencies between load and exogenous influences, enabling them to better understand and learn patterns from the input data.

Random Forest (RF) and Gradient Boosting have recently demonstrated their effectiveness in load forecasting. Gradient Boosting produces a prediction model in the form of an ensemble of weak prediction models, such as simple decision trees, that make minimal assumptions about the data. During training, new models are trained to minimise the loss function of the previous models using gradient descent, typically using mean squared error or cross-entropy. [93] applied supervised machine learning to a dataset from the Netherlands and analysed three regression algorithms: Random Forest, Gradient Boosting, and XGBoost [94]. They aimed to identify the most accurate algorithm and the main influencing parameters. They found that XGBoost performed the best under their experimental conditions, with the most influential parameters being the time of day at which the charging sessions start and the total energy supplied. [95] evaluated various machine learning methods, including Decision Tree, Support Vector Machines (SVM), and Artificial Neural Network (ANN), for EV load forecasting. Their results showed that SVM and ANN generally performed better under their experimental settings, but they also had longer training times.

Compared to statistical models, ML models are generally more complex and require more time for training. Additionally, the accuracy potential of an ML model also depends on selecting appropriate input variables. While ML models and deep learning-based models have stronger learning abilities, they may not necessarily improve forecast accuracy. As indicated in [96], the advantage of one forecasting technique over another highly depends on the use-case and the available data. Therefore, it is worth conducting research to determine the forecasting models that are suitable for specific applications and data, instead of blindly starting with a very complex ML or DL model.

Deep Learning Models

Deep learning has a greater ability than the previous two categories to discover inherent features. It can represent the internal features of EV charging demand data without any prior knowledge, and it achieves superior prediction performance when well designed. The following table presents some examples of deep learning-based forecasting models for EV charging demand prediction.

The reviewed deep learning models show diverse characteristics in terms of inputs, outputs, forecast horizon, deep learning techniques, hyperparameter optimisation, evaluation metrics, and feature selection. This further highlights the complexities of EV charging demand forecasting and the design of DL models. Even though the examples don't use the same deep-learning architectures, they do share some similarities:

1) The input data has more features (than the ones used by the statistical and ML models), and the selected features are diverse.
2) Deep-learning architectures such as Transformer, Encoder-Decoder, and Long Short-Term Memory (LSTM) have been used to enhance learning and forecasting abilities.
3) Hyperparameter optimization and feature selection have been employed to improve the performance of forecasting models.

Funded by the European Union

Table 16. Examples of deep learning-based forecasting models.

Reference	Datasets	Inputs & Output	Forecast Horizon	Models	Hyperparameter optimization	Evaluation Criteria	Feature Selection
[97]	Electric Vehicle Charging Station Data of Boulder, Colorado [98]	**Inputs:** **1)** Aggregated EV charging load (kW/day); **2)** Binary weekend (0 or 1); **3)** Min temperature (◦F) **Time Resolution:1 day** **Outputs:** Aggregated EV charging load	7,30,60, and 90 days ahead (forecast steps: 7,30,60,90 1-day steps)	**Transformer** (compared with ARIMA, SARIM, RNN, and LSTM)	-	RMSE, MAE	-
[99]	EV Charging data within the campus of Georgia Tech, Atlanta, USA (**not publicly available**)	**Inputs** **1)** Charging duration **2)** Energy (kWh) **3)** Greenhouse Gas (GHG) savings (kg) **4)** Gasoline savings (gallons) **5)** cost incurred (USD) **Outputs:** Charging demand of energy (kWh).	-	**EMD–AOA-DLSTM** (EMD: empirical mode decomposition; AOA: arithmetic optimization algorithm; DLSTM: deep long-short term memory)	arithmetic optimization algorithm (AOA)	MAE, MSE, RMSE and accuracy of prediction (Apre)	empirical mode decomposition (EMD).
[100]	synthetic data generated based on statistical info (**not publicly available**)	**Inputs:** historical data including: **1)** arrival time **2)** departure time **3)** trip length Outputs: Aggregated Plug-in Electric Vehicles (PEVs) load (kW)	1 day	**Recurrent Artificial Neural Networks (ANN) with Levenberg Marquardt (LM)** (Compared with Monte Carlo Simulation, MCS)	-	MAE, MAPE, RMSE	-
[101]	a real-world dataset containing 593 charging stations in Germany, covering August 2020 to December 2020 (**not publicly available**) **Time Resolution:15 mins**	**Inputs:** **1)** Occupation (which is a binary flag indicating a charging record of the charging station at the time *t*) **2)** Day of the week **3)** Time of day **4)** Mean Occupation **5-6)** Quantiles (0.25 and 0.75). Data is collected at the charging outlet level. **Time Resolution:15 mins** **Outputs:** **occupation of a charging station/outlet** for each time point of the prediction horizon	8 hours (36 binary outputs with a 15-min interval)	**Deep Fusion of Dynamic and Static Information model (DFDS) which has an Encoder-Decoder structure and a Fusion Component** baselines: Historical Average, K-Nearest Neighbours, Random Forest, Logistic Regression, Support Vector Machine, Gru+Fully, Connected Sequence-2-Sequence.	-	Precision, Recall, F1-Score	Manual analysis (by removing individual features from the DFDS model and measuring the difference in the forecasting performance.)
[102]	EV charging data from the open data portal [103] of the city of Dundee, UK. Data includes Start and finish date/time, KWh (amount of electricity) used, Charge point ID, Location, and Type of Charger. **Time Resolution:1 min**	**Inputs:** **1)** Time of day **2)** Day of week **3)** Weekday/weekend binary indicator **4)** Average charging occupancy rate profile for weekday/weekend **5)** Past charging occupancy states **Time Resolution:10 min** (the dataset has been processed and bucketed with a 10-min interval) Outputs: **a vector of binary values representing the multistep forecasting of the charging occupancy states**	10 mins - 6 hours (**forecast steps:** 10 min is one forecast step, and 6 hours ahead forecast generates 36 binary values)	a new mixed long short-term memory neural network incorporating both historical charging state sequences and time-related features for multistep discrete charging occupancy state prediction. Baselines include logistic regression, SVM, random forest, and Adaboost	Grid Search	MAE, F1 Score	Manual Selection by comparing three groups of features

Funded by the European Union

4.1.2. Datasets

Table 17. Examples of datasets for EV charging sessions forecasting models.

Name	Application Scenarios	Description
Electric Vehicle Charging Dataset [104]	Workplace	This dataset contains information from 3,395 high-resolution electric vehicle charging sessions. The data contains sessions from 85 EV drivers with repeat usage at 105 stations across 25 sites at a workplace charging program. The workplace locations include facilities such as research and innovation centers, manufacturing, testing facilities and office headquarters for a firm participating in the U.S. Department of Energy (DOE) workplace charging challenge. The data is in a human and machine-readable *.CSV format. The resolution of the data is to the nearest second, which is the same resolution as used in the analysis of the paper. It is directly importable into free software.
ACN-Data [105]	Workplace	ACN-Data exists to help researchers access real data around electric vehicle charging. The dataset is made possible by a close collaboration with PowerFlex Systems, which operates Adaptive Charging Networks around the United States. Each entry in the dataset contains information about a single charging session. The dataset contains tens of thousands of charging session data from three sites over a span of several years. We have selected this dataset for our experiment because of its completeness of its data and user-friendly API and web interface for downloading data.
Daily Load Transactions for Electric Vehicles from SAP Labs France [106]	Workplace & Residential	This dataset corresponds to all EV charging transactions of SAP Labs employees France carried out either at their workplace or at home since June 2017. A transaction is the entire charging operation from start to stop (from insertion to removal of the badge, button or cable), including the period of inactivity (when charging is complete, but the vehicle is still plugged in). It should be noted that this dataset is no longer available for downloading.
Electric Vehicle Charging Sessions Dundee [103]	Public	This dataset contains every electric vehicle charging session at Dundee City Council owned charge points. Its data fields include Start and finish date/time, kWh (amount of electricity) used, Charge point ID, Location, and Type of Charger.
Electric Vehicle Charging Transactions by London Borough of Barnet [107]	Public	Data showing Electric Vehicle Charging Point Transactions. Data fields include charging point location, time, total Watt Hours (Wh) and connect time. Data measured in Wh.
Electric Vehicle Charging Station Energy Consumption of Boulder (Colorado, USA) [98]	Public	This dataset shows the energy use, length of charging time, gasoline savings and greenhouse gas emission reductions from all city-owned electric vehicle charging stations. Data are broken out by charging station name/location, transaction date, and transaction start time; 1 row indicates 1 EV charging station transaction.
Electric Vehicle Charging Station Usage of Perth & Kinross (UK) [108]	Public	These are the datasets for Perth & Kinross Council's EV charging stations under the ChargePlace Scotland scheme. It includes anonymous data from each individual charging session.
Electric Vehicle Charging Station Usage (2011.07-2020.12) of City of Palo Alto [109]	Public	This dataset shows the EV Charging Station Usage of City of Palo Alto.

Funded by the European Union

Belib' - Charging points for electric vehicles - Real-time availability [110]	Public	This dataset contains geo-localised data on real-time availability of Charging Points [Bélib'] for electric vehicles, where Belib' represents the network of EV charging terminals. It can be used to scrape some data for charging station occupancy research.
Electric Chargepoint Analysis 2017: Domestics [111]	Residential	The dataset contains experimental statistics on the usage of OLEV-funded domestic charge points in the UK in 2017. This includes details of charging events and the amount of energy supplied.

4.2. Methodology

The research work conducted by the Eaton team is not exhaustive, and initially, we focus on testing the categories of models that possess a certain conciseness and capacities. Two categories of forecasting models have been evaluated, which are statistical models such (e.g., *ARIMA*) and ensemble models (e.g., *Random Forest*). The two categories of models and some selected models will be introduced in the following sub-sections.

4.2.1. Experiment Data

We have selected the ACN-Data [105] as the dataset for our research because it provides relatively complete EV charging sessions for workspace scenarios. Our research focuses on forecasting the EV charging demand of buildings to optimize energy usage. Therefore, we specifically chose scenarios that involve buildings within relatively enclosed environments. Public EV charging stations may exhibit different charging demand patterns, and their operators may not prioritise energy usage optimisation and electricity bill reduction as much as building administrators do. As a result, public charging stations are not within the scope of our current research.

Data from the Caltech site (Site ID: *caltech*) is utilised for our experiment. Caltech is a research university located in Pasadena, CA, US, and it has 54 EVSEs in its campus garage. Although the site is open to the public, the majority of usage comes from faculty, staff, and students. Therefore, it can be used to simulate our target application scenario.

The Caltech dataset consists of 31,424 entries of charging sessions, spanning from April 2018 to September 2021. We have converted the charging sessions into the charging power of EV chargers, and the transformed data is measured in kilowatts (kW) with a time resolution of 5 minutes. In addition, the data are averaged in order to form time series with a 5-min resolution, and the aggregated load imposed by EV charging on the grid at the site level is analysed.

For training models, we reserve the data from 'May 1, 2018' to 'April 30, 2019', and for testing the model performance, we use the data from 'July 1, 2019' to 'May 1, 2020'. However, there is a key difference between this split and the common train-test-splitting approach. In our case, we use the former period to assist some models in hyperparameter tuning, while the latter period is used to simulate the training and prediction processes when the models are actually deployed.

Specifically, for models with parameters that need to be determined in advance, we utilise the data from the former period to aid in decision-making. For example, the hyperparameters of seasonal ARIMA are tuned using the data between 'May 1, 2018' and 'April 30, 2019'. On the other hand, we use the data from the latter period to train the models and make predictions using a moving window.

Funded by the European Union

This simulates the deployed version of our forecasting module, where the models are regularly updated at certain intervals. Actually, the dataset contains more data than we have actually used. This is because we are unsure about the effects of COVID-19 on office attendance policies and EV charging behaviours, and we decided to use the data before the middle of 2020.

Some important settings for our data include:

1) The aggregated EV charging load at the site level has a 5-min time resolution and it is charging power we are going to analyse.
2) *training_resampling_freq = "60T"*: we aggregate the training data to hourly frequency to suit the need of our downstream applications (such as energy usage scheduler)
3) *forecast_resampling_freq = "60T"* and *forecast_horizon = 24* (in terms of *training_resampling_freq*): the output of the model will be a 24-hour ahead forecast, with hourly frequency (i.e., the forecast will be an array containing 24 points)
4) For each date in the period selected ('July 1, 2019' - 'May 1, 2020'), a model will be trained using the *days_in_training_history* worth of data from the past (12:00 of each date is selected as the reference timestamp), and we do forecast for the next 24 hours.

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) have been used as the metrics for measuring the performance of forecasting. MSE measures the average of the squared differences between the predicted and actual values, providing a comprehensive view of a model's accuracy. RMSE is the square root of MSE and is often preferred as it gives a more interpretable measure of error. MAE, on the other hand, calculates the average of the absolute differences between the predicted and actual values, providing a measure of the average magnitude of error. By considering these metrics, we can gain valuable insights into the effectiveness of forecasting models and make informed decisions based on the results.

4.2.2. Models for Experiments

4.2.2.1. Statistical Models

The following statistical models from the Python *StatsForecast* [112] package have been evaluated in this experiment:

- The seasonal naive model is currently used as a benchmark for EV demand forecasting. Along with *HistoricAverage*, they serve as two baseline models for our evaluation.
- *ARIMA* is widely used for time series forecasting. In this experiment, we use its automatic version for parameter tuning.
- The *CrostonClassic* model may not be suitable for this experiment as it is designed for forecasting time series with intermittent demand. However, we keep this model due to its quick training time.
- *DynamicOptimizedTheta* and The *Holt-Winters'* method are also included to assess their performance.

Seasonal ARIMA

In time series forecasting, *ARIMA* is one of the most widely used approaches. *ARIMA* exploits the autocorrelations in the data. An *ARIMA* model is characterised by 3 terms: *p, d, q* where, *p* is the order

Funded by the European Union

of the Auto Regressive (AR) term, *q* is the order of the Moving Average (MA) term, and *d* is the number of differencing required to make the time series stationary. The full *ARIMA(p,d,q)* model can be written as follows.

$$\hat{y}_t = c + \emptyset_1 y'_{t-1} + \cdots + \emptyset_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t$$

Where y'_t is the *d*-order differenced t[th] member of the time series data, *p* is the maximum lag considered in the AR part, and $\epsilon_t = y_t - \hat{y}_t$ is the forecast error associated at timestamp *t*. c, $\emptyset$ and θ are the intercept term, coefficients of lagged terms, and coefficients of error terms. They are estimated by fitting the model to the data.

SARIMA (p,q,d)(P,Q,D)[s] is an extended version of the *ARIMA* model that takes into account seasonality in the input data. *P*, *Q*, and *D* are similar to their non-seasonal counterparts *p*, *q*, and *d* respectively. They represent the seasonal AR terms, seasonal MA terms, and the order of seasonal differencing. Seasonal differencing is performed in a similar way to regular differencing, but instead of subtracting consecutive terms, the value from the previous season is subtracted. The seasonal length is given by *s*.

In our experiment, we aim for our ARIMA model to handle seasonality while maintaining simplicity by not incorporating exogenous variables. Therefore, we apply SARIMA to the time series data of aggregated EV charging load.

In our experiments, we utilise the automatic version of *AutoARIMA* from the python *StatsForecast* package for parameter tuning to select the best model. While we also manually analyse and select parameters from *ARIMA*, the parameter selection process is subjective and may result in different models. Therefore, we will solely rely on *AutoARIMA* for our experiment.

Holt-Winters' method

Holt-Winters' method, also known as triple exponential smoothing, is an extension of exponential smoothing for series that contain both trend and seasonality. Exponential smoothing uses a weighted average of all past observations where the weights decrease exponentially into the past, and it is suitable for data with clear trends and/or seasonality.

season_length is set to 24 in our experiments.

DynamicOptimizedTheta

The *Theta* method decomposes the seasonally adjusted data into two "theta lines" and uses different techniques to obtain and combine the two theta lines to produce the final forecasts.

In a standard *Theta* model, the first theta line removes the curvature of the data to estimate the long-term trend component. The second theta line doubles the local curvatures of the series to approximate the short-term behaviour [113]. Based on the description of [114], the optimised *Theta* model has been shown to be more accurate than other time series forecasting methods, especially for time series with complex trends and seasonality.

The Dynamic Optimised Theta Model was proposed in [113] and it is a state space model that selects the best short-term theta line optimally and revises the long-term theta line dynamically. The Dynamic Optimised Theta Model generally achieves higher levels of forecasting accuracy than other versions.

Funded by the European Union

CrostonClassic model
It is a method to forecast time series that exhibit intermittent demand.

Seasonal Naïve
It is a method similar to the *Naïve* model uses the last known observation of the same period (e.g., the same month of the previous year) to capture seasonal variations.

HistoricAverage
It is also known as the mean method, and it uses a simple average of all past observations. We set its seasonal length to 24 and use it a benchmark model.

4.2.2.2 Ensemble Models

Ensemble methods combine the predictions of multiple base estimators, which are built using a specific learning algorithm, to improve generalisability and robustness compared to a single estimator. Two well-known ensemble methods, random forest (RF) and gradient-boosted trees are evaluated in our experiments.

Random Forest (RF)
The Random Forest algorithm is designed for trees, using a perturb-and-combine technique. It creates diverse classifiers by introducing randomness in the construction process. The ensemble's prediction is the average of the individual classifiers. More specifically, each tree in the ensemble is built from a bootstrap sample (drawn with replacement) from the training set. When splitting each node, the best split is found from either all input features or a random subset.

Introducing randomness in these ways reduces the variance of the forest estimator. Individual decision trees often overfit due to high variance. The randomness in the forest leads to decision trees with somewhat decoupled prediction errors. Averaging these predictions cancels out some errors. Random forests achieve reduced variance by combining diverse trees.

Random Forests implemented by the python package scikit-learn [115] has been used in our experiment.

LightGBM
Light Gradient Boosting Machine (LightGBM) is a histogram-based gradient boosting algorithm that was developed by Microsoft in 2017 [116]. Its main objective is to improve the scalability of boosted algorithms and reduce computation times. This algorithm stands out from other gradient boosting tree (GBT) algorithms because it constructs trees in a leaf-wise manner instead of a depth-wise manner. With the depth-wise strategy, leaves at the same depth are split at the same time, which can result in unnecessary splitting of leaves with low information. However, with the leaf-wise strategy implemented in LightGBM, only the leaf with the highest information gain is split, which leads to a significant enhancement in efficiency. Moreover, LightGBM utilizes a histogram-based strategy to group attributes before splitting, which is a faster approach compared to the presorting method used by other GBT models.

The LightGBM implementation by [117] has been used and evaluated in our experiment.

Funded by the European Union

4.2.2.3 Pipeline of Transforms

We apply the scikit-learn Pipeline to the ensemble models to transform the input data before feeding it into the models. This allows the ensemble models to better utilise some additional features in the input data. The basic steps included in the pipeline are as follows:

```
steps=[
        ("input_transformer", input_transformer),
        ("reshape_transformer", reshape_transformer()),
        ("regressor", regressor),
    ]
```

Figure 22. Steps in the scikit-learn Pipeline.

The "*input_transformer*" is used to enrich our univariate dataset with additional features, which will be explained in the following content. Then, the "*reshape_transformer*" ensures that the data fed into the regressors have the correct shapes.

Similar to the technique employed in [118] to obtain the trigonometric characteristics, we convert each ordinal time attribute (which is the weekday, hour, and month information coming from each timestamp) into 2 features that collectively represent the same information in a non-monotonic manner. This approach also ensures that there are no abrupt transitions between the first and last values of the periodic range for each sub-feature.

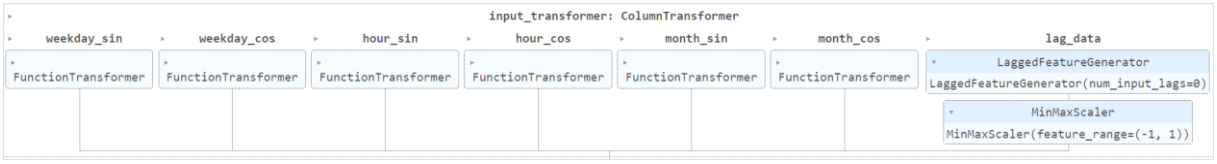


Figure 23. Transforms applied within input_transformer.

The *LaggedFeatureGenerator* is a customised transformer that creates lagged versions of a pandas DataFrame column. It is specifically designed to be used in a scikit-learn pipeline to add lagged values of a column as additional input features. This allows a model to access more information from the past, but it also increases the complexity of the pipelines.

Additionally, when using time-sensitive cross-validation to select the best *num_input_lags* (from the list [0, 12, 24]) for the two ensemble models, it has been found that *num_input_lags=0* leads to better performance. This suggests that either our models do not rely heavily on lagged features, or the current models cannot fully utilise the lagged features. Since our experiments are exploratory, we make *num_input_lags=0* for our current experiments.

4.2.3. Experiment Workflow

The workflow of our experiments is tightly linked with the split of the experiment data. Figure 22 explains how we split our data for running our simulations.

Funded by the European Union

The selected forecasting models are used to provide EV demand forecasts for the next 24 hours, with an hourly frequency. The aggregated EV load data from 'May 1, 2018' to 'April 30, 2019' is utilised to tune the hyperparameters of certain models, such as *ARIMA* and *RF*. This is done to obtain preliminary models that can be directly used for forecasting or require less effort for fine-tuning.

For automatic forecasting models like *AutoARIMA*, our application may not allow us to search for the best parameters each time or accept a model with changing parameters. In such cases, we run *AutoARIMA* on the aforementioned data period to learn about the *(p,q,d)(P,Q,D)* parameters. *AutoARIMA* selects the best parameters based on a provided information criterion (AIC, AICc, BIC, or HQIC). The function searches for possible model and seasonal orders within the given constraints and selects the parameters that minimize the specified metric.

For ensemble models like *RF* and *LightGBM*, considering training time, we train a model using the data from this period and only apply this model to new data when making predictions. We utilise a hyperparameter optimizer and time-sensitive cross-validation to determine their hyperparameters.

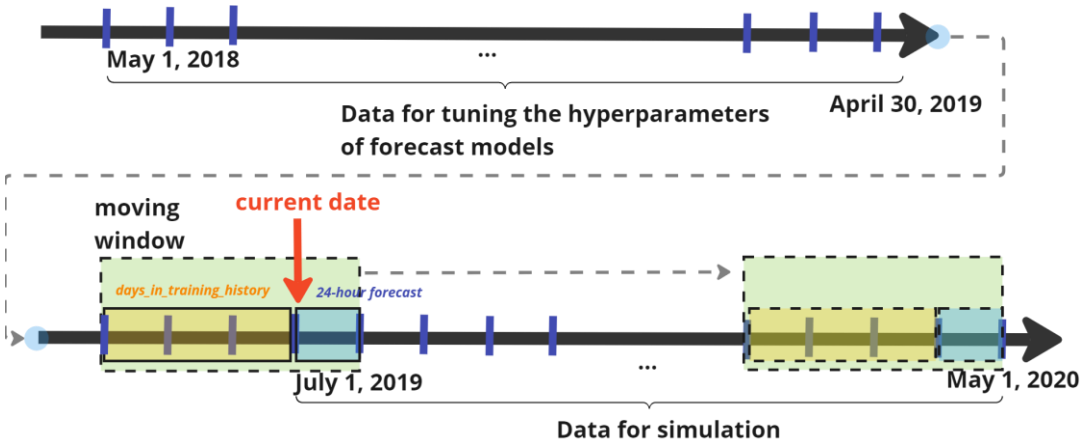


Figure 24. Data splitting for experiments.

To test the performance of the model, we select data from 'July 1, 2019' to 'May 1, 2020' and use a moving window to simulate the actual application of our model. The workflow is as follows:

1. Select a date within the specified period and use it as the current date.
2. Use the previous *day_in_training_history* worth of data before the current date to train the models. If the models are already pre-trained, this step can be skipped, and the models can be used directly for forecasting or refitting.
3. Make predictions for the next 24 hours using the trained models.
4. Record the performance metrics.
5. Move the window one day forwards and repeat the process.

This allows us to assess the model's performance over time and evaluate its effectiveness in real-life scenarios.

To identify the model with the best performance, two training-forecasting procedures have been utilised based on the models for evaluation:

Funded by
the European Union

1. The "static procedure" uses the data from 'May 1, 2018' to 'April 30, 2019' to train models and tune hyperparameters. This procedure is applied to RF, *LightBGM*, and *AutoARIMA*.
2. The "rolling-origin procedure" uses a moving window with a fixed length (determined by *days_in_training_history*) on the data from 'July 1, 2019' to 'May 1, 2020' to select training data. The origin of the moving window is shifted daily to simulate the daily training and forecasting procedures. This procedure is applied to statistical models such as *AutoARIMA* and *DynamicOptimizedTheta* that generally have shorter training time. It is important to note that we use *AutoARIMA* in both procedures to compare the performance of the ARIMA model under different procedures.

We use the averaged MSE, RMSE, and MAE as performance metrics for forecasting.

4.3. Simulation Results

The simulation experiments have been conducted based on the methodology defined in Section 4.2. *day_in_training_history* is set to 7 (days) and all the seasonal lengths are set to 24 (hours).

Based on the static procedure, *AutoARIMA* finds the best ARIMA model *ARIMA(3,1,2)(0,0,2)[24]* on the dataset used for model tuning. *ARIMA(3,1,2)(0,0,2)[24]* will be compared with other statistical models trained using the rolling-origin procedure. Table 18 shows MSE, RMSE, and MAE values of the selected statistical models.

Table 18. MSE, RMSE and MAE of statistical models.

	ARIMA(3,1,2)(0,0,2)[24]	AutoARIMA	HoltWinters	CrostonClassic	SeasonalNaive	HistoricAverage	DynamicOptimizedTheta
MSE	131.13	115.28	105.30	178.19	128.60	171.72	111.33
RMSE	9.66	9.10	8.98	11.44	9.47	11.52	9.20
MAE	6.81	6.03	6.66	8.64	6.30	8.80	6.13

From the results, *AutoARIMA* (trained using “rolling-origin”) and *DynamicOptimizedTheta* outperform the others in terms of MAE, although they slightly lag behind *HoltWinters* in terms of MSE and RMSE. This suggests that *AutoARIMA* (trained using the rolling-origin) and *DynamicOptimizedTheta* generally perform well, but may occasionally have larger forecast errors, resulting in higher MSE and RMSE values. On the other hand, *ARIMA(3,1,2)(0,0,2)[24]* trained using the static procedure does not perform as well as *AutoARIMA* (trained using the rolling-origin). This could be because the EV demand forecast relies on more recent data, making the orders used by *ARIMA(3,1,2)(0,0,2)[24]* slightly outdated.

Furthermore, the *HistoricAverage* model performs the worst, as it assumes that EV charging demand is simply an average of past values. The *CrostonClassic* model also performs poorly, as it is designed for forecasting intermittent patterns.

Last, we have observed that models such as *AutoARIMA* and *DynamicOptimizedTheta*, which consider seasonality and long-term trends, perform better than models that do not consider these factors *(HistoricAverage* and *CrostonClassic*). This highlights the importance of including the analysis of seasonality in the forecasting process. In our experiments, we set the *seasonal_length* for some models to 24 hours for our initial experiments. However, the performance of some models can be further improved by carefully selecting the *seasonal_length* parameter.

Funded by the European Union

To train our forecasting models using ensemble models, we utilise both static and rolling-origin procedures. This allows us to observe their performance under different simulation procedures. However, we first run the static procedure and use grid search to find the best estimators. The best estimators found can be used for EV load forecasting in the static procedure and to determine the hyperparameters of forecasting models in the rolling-origin procedure.

It is important to note that the hyperparameters selected in the static procedure may not be optimal for the rolling-origin procedure. This is because the two procedures use different data with different lengths to fit models. However, to save time and avoid hyperparameter tuning for every simulation date in the rolling-origin procedure, we will use the hyperparameters selected by the static procedure as the starting point for our experiment.

Lastly, the search space for these hyperparameters is defined as follows.

```
if algorithm_id == 1:
    regressor = LGBMRegressor()
    search_space = {
        "learning_rate": [0.1, 0.05, 0.01],
        "max_depth": [3, 5, 7],
        "num_input_lags": [0, 12, 24],
        "num_leaves": [7,15,31],
        "n_estimators": [50, 100, 150, 200],
    }
    model_name = 'LGBMRegressor'
elif algorithm_id == 2:
    regressor = RandomForestRegressor(oob_score=True)
    search_space = {
        "max_depth": [3, 5, 7],
        "num_input_lags": [0, 12, 24],
        "n_estimators": [50, 100, 150, 200],}
    model_name = 'RandomForestRegressor'
```

Figure 25. The search space used by two ensemble methods for hyperparameter tuning.

The optimal value for *num_input_lags* has been determined to be 0. The best models for *LightGBM* and *RF* are: *LGBMRegressor(learning_rate=0.01, max_depth=3, n_estimators=200, num_leaves=7)* and *RandomForestRegressor(max_depth=3, n_estimators=150)*. These hyperparameters will be utilised in the rolling-origin procedure.

The MSE, RMSE, and MAE metrics of the ensemble models are summarized in Table 19.

Table 19. MSE, RMSE and MAE of ensemble models.

	LGBMRegressor (rolling-origin)	RandomForestRegressor (rolling-origin)	LGBMRegressor (static)	RandomForestRegressor (static)
MSE	81.37	71.59	643.20	730.73
RMSE	7.93	7.21	22.78	24.76
MAE	5.65	5.07	20.10	21.24

Compared to the statistical models, the two ensemble models trained using the rolling-origin procedure demonstrate lower forecast errors. This is because we have utilised a greater number of features to train the models, and the ensemble mechanism has enhanced the learning and forecasting capabilities of weak decision trees. On the other hand, the ensemble models trained using the static process exhibit significantly higher errors compared to other models. This could be due to the fact that

Funded by the European Union

the ensemble models trained using the static procedure do not update themselves with the most recent data, and the large amount of historical data used for training does not compensate for the disadvantage of not utilising the latest data. This indicates the importance of tuning hyperparameters and fitting models using the latest data.

It is also important to note that the simulation process for the two ensemble models, using the rolling-origin procedure, took 1.56 days. In contrast, the total time for running the six statistical models, using the same procedure, was only 6.64 hours. The longer simulation time could be attributed to the complexities of additional features and ensemble models.

Therefore, when designing EV demand forecast models, it is recommended to consider any limitations on the training time of models. It is important to choose models that satisfy both accuracy and training time requirements.

Finally, the forecasts and ground truth for the 1st week of July 2019 have been plotted in the following figures, to compare the forecasting performance of models.

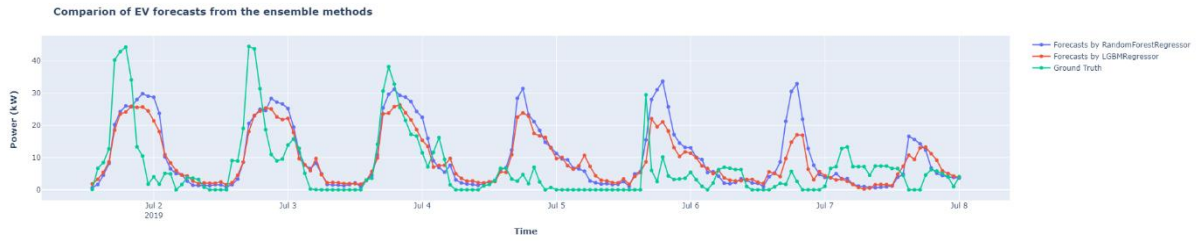


Figure 26. Visualisation of EV forecasts from the ensemble methods

From Figure 26, it can be observed that the two ensemble methods can catch the ups and downs of EV charging power during the day, but there are some mismatches for the time that EV peak power appear. Thus, even though the two ensemble methods outperform other methods, their forecasts are still not perfect. Furthermore, the forecasts from RF seem to be higher than those from the *LightGBM*, making it closer to the ground truth data. This may explain why RF had higher forecast accuracy.

For the forecasts from statistical models in Figure 27, forecasts from *CrostonClassic* and *HistoricAverage* seem to form straight lines and could not catch the fluctuations of EV charging power. The three statistical methods with better peformance, *AutoARIMA*, *HoltWinters*, and *DynamicOptimizedTheta*, show much better matches than the previous two methods. On the other hands, we observed that some methods, especially the *HoltWinters* methods generated some negative forecasts at the beginning, indicating the needs to include additional steps to ensure positive forecasts.

Funded by the European Union

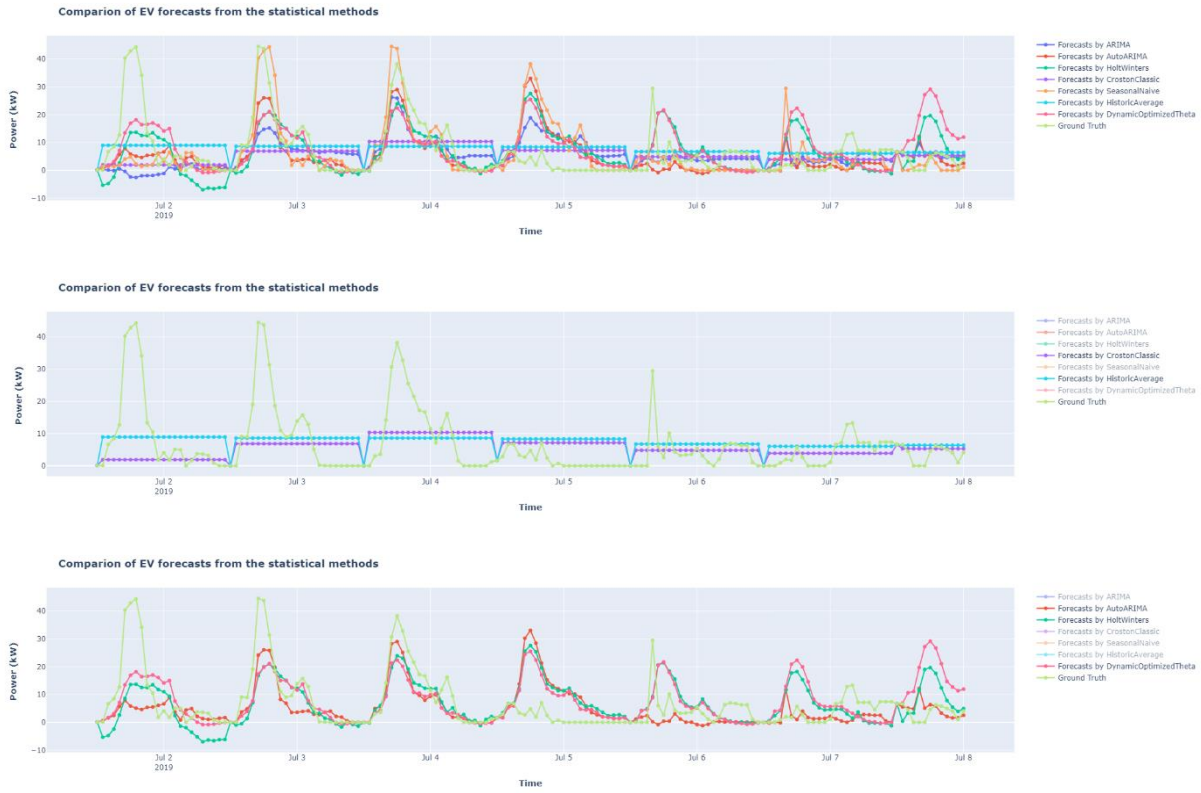


Figure 27. Visualisation of EV forecasts from the statistical methods: (top) forecasts from all models; (middle) forecasts from CrostonClassic and HistoricAverage; (bottom) forecasts from AutoARIMA, HoltWinters, and DynamicOptimizedTheta.

4.4. Summary and Future Work

In this section, the Eaton CIP team has conducted research to evaluate different forecasting models and train-forecasting strategies. Two categories of time series forecasting models, statistical models and ensemble models, were selected to assess their effectiveness. The comparative analysis was performed using data from the Caltech site of the ACN dataset. Forecasts were generated for 24-hour time horizons, using two different training-forecasting procedures.

When compared to the seasonal naive model, the *AutoARIMA* and *DynamicOptimizedTheta* models show improvements in MAE performance by 4.3% and 2.7% respectively. *HoltWinters*'s average RMSE performance is better than that of other statistical models, achieving a 5.2% performance improvement compared to the baseline. The results also showed that the seasonal ARIMA model trained by the roll-origin procedure performs better than the one of the static procedure, indicating the importance of using the most up-to-date data available to train models.

On the other hand, the two ensemble models, Random Forecast and LightBGM, outperform the statistical models. When compared to the seasonal naive model, they show improvements in MAE performance by 10.3% and 19.5% respectively, which are quite higher than the improvements achieved by *AutoARIMA* and *DynamicOptimizedTheta*. This is because they utilise additional features

Funded by the European Union

and have powerful forecasting abilities. However, it is worth noting that the simulation time for these models is longer due to the increased complexities.

For future work, it is recommended to conduct additional experiments by including statistical models that utilize exogenous features. The current comparison may not be entirely fair, as ensemble methods can leverage the transform steps in their pipelines to obtain additional information. The research work conducted so far is exploratory and experience-based, and it could be improved by designing a more rigorous experiment plan. Additionally, it is suggested to include deep learning models such as LSTM in the future comparative analysis to gain a comprehensive view of all the EV charging demand forecasting models.

5. Conclusions

This deliverable demonstrates methods and algorithms to improve forecast of 1) energy prices using a deep-weighted ensemble model, 2) demand from a user perspective with a week-ahead forecast of travel patterns and energy consumption and 3) charging demand from the charge station perspective.

Section 2 discusses the significance of price forecasting for energy aggregators. The effectiveness of charging control by aggregators heavily depends on advanced knowledge of electricity prices, making accurate price forecasting a crucial tool for optimizing participation in demand response programs. This enables aggregators to manage charging loads effectively, meet EV user requirements, facilitate TSO/DSO regulation services, and maximize overall benefits. Traditional statistical methods, which often rely on average cases, prove inadequate for accurately forecasting dynamic electricity price variations due to their fluctuating nature. To address this knowledge gap, we introduced the Deep-Weighted Ensemble Model (DWEM) for forecasting energy prices to manage EV charging at the aggregator level. We formulated the problem and presented a comprehensive mechanism for developing the DWEM model. Testing was conducted on a publicly available dataset from the Texas electricity market, specifically the ERCOT (Electric Reliability Council of Texas), representing wholesale electricity prices in the Houston zone. Results were evaluated against the XGBoost, Linear Regression, Light Gradient Boosting Model, and Multivariate LSTM models in terms of accuracy, mean square error (MSE), and mean absolute error (MAE). The DWEM achieved the highest accuracy of 87.41% compared to these different models. Additionally, we considered user and auto modified hyperparameter tuning mechanisms for standard, stack, and DWEM models and conducted comparative studies with and without outlier scenarios. The results demonstrated that the proposed DWEM outperformed individual standard and stack LSTM models in terms of accuracy, MSE, and MAE.

Section 3 presents a novel tool to provide users the required inputs to perform optimal smart and bidirectional charging automatically any time they arrive at a charging station. This is done by predicting the future trip locations and energy consumption of the EV for the next week. The conceptual interest of this tool is to provide users with the information they need to optimally manage their EV based on their mobility patterns. If implemented this can provide economic savings in form of a reduction in the total cost of the charging sessions and improved battery state of health care.

Drawing from a dataset that includes travel locations and trip energy consumption, the goal of the forecasting tool is to first predict a timeseries of the locations that the EV is going to visit the following

Funded by the European Union

week. Second, given the location profile of the vehicle, the tool trains a second model which tries to precisely estimate the energy consumption of the EV. At first glance it would seem that only knowing the trip consumption would be enough for the optimization tool; however, knowing the location of the vehicle is important too. In real environments, different locations can bring particularities that can be accommodated by considering locations: the location may or may not have EVSEs, different pricing strategies and/or CSO operators, etc. We used taxi data to perform the analysis because personal transport vehicle data was not available.

The results show that the method is feasible; however, the use of taxi data is not well suited to asses the quality of the forecast because the taxi data has many pick-up/drop-off/charging locations in comparison to a typical personal transport vehicle which includes far fewer charging locations to consider, leading to a variability increase across geographical domain. This led to issues with performance and limted the ability to assess the quality of the forecast. Taxi data is more suitable for overall city mobility demand forecast, considering it aggregated and not individual taxi driver profile. While not as relevant for personal transport, the use of taxi data has interesting applications for the work done in FLOW Work Package 5 or to consider from a charging station perspective. Future work should implement the proposed forecasting method using private vehicle data to confirm. Furthermore, private vehicles fleet also includes repetitive trips such us from home to workplace and vice versa. This could lead to an increase of precision when predicting the vehicle position and its energy consumption.

To operationalize the tool, the second part of the work is to take the location and energy consumption data and implement an optimization model, defined as User Smart Model, to determine the optimal charging pattern including desired SOC. In contrast to typical smart charge strategies implemented by CSOs, this model takes into consideration future usage of the vehicle to make optimal decisions in the present. The performance of using this methodology (i.e., User smart charge) has been tested against two other methodologies (“immediate” and “CSO smart charging”), showing promising results in the form of economical savings and battery health improvement.

To forecast day-ahead building electric vehicle charging demand, the Eaton CIP team conducted research using a real EV charging dataset. The purpose was to evaluate different forecasting models and train-forecasting strategies. They assessed two categories of time series forecasting models: statistical models and ensemble models, along with two training-forecasting strategies: static and rolling-origin procedures. The results indicate that ensemble methods outperform statistical models, although they may require longer training time. Among the statistical models, *AutoARIMA*, *DynamicOptimizedTheta*, and *HoltWinters* are effective in forecasting EV charging demand while maintaining simplicity. Therefore, the choice of EV charging forecasting model should be based on the specific requirements of real EV demand forecasting applications.

Funded by the European Union

6. Bibliography

[1] M. Umar, X. Ji, D. Kirikkaleli and A. A. Alola, "The imperativeness of environmental quality in the United States transportation sector amidst biomass-fossil energy consumption and growth," *Journal of Cleaner Production,* vol. 285, p. 124863, 2 2021.

[2] K. M. Tan, J. Y. Yong, V. K. Ramachandaramurthy, M. Mansor, J. Teh and J. M. Guerrero, "Factors influencing global transportation electrification: Comparative analysis of electric and internal combustion engine vehicles," *Renewable and Sustainable Energy Reviews,* vol. 184, p. 113582, 9 2023.

[3] M. Muratori, M. Alexander, D. Arent, M. Bazilian, P. Cazzola, E. M. Dede, J. Farrell, C. Gearhart, D. Greene, A. Jenn, M. Keyser, T. Lipman, S. Narumanchi, A. Pesaran, R. Sioshansi, E. Suomalainen, G. Tal, K. Walkowicz and J. Ward, "The rise of electric vehicles—2020 status and future expectations," *Progress in Energy,* vol. 3, p. 022002, 3 2021.

[4] A. Dash, "Adapting to electric vehicles value chain in India: The MSME perspective," *Case Studies on Transport Policy,* vol. 12, p. 100996, 6 2023.

[5] W. Zou, Y. Sun, D.-c. Gao, X. Zhang and J. Liu, "A review on integration of surging plug-in electric vehicles charging in energy-flexible buildings: Impacts analysis, collaborative management technologies, and future perspective," *Applied Energy,* vol. 331, p. 120393, 2 2023.

[6] M. S. Mastoi, S. Zhuang, H. M. Munir, M. Haris, M. Hassan, M. Usman, S. S. H. Bukhari and J.-S. Ro, "An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends," *Energy Reports,* vol. 8, p. 11504–11529, 11 2022.

[7] B. Singh, "Federated learning for envision future trajectory smart transport system for climate preservation and smart green planet: Insights into global governance and SDG-9 (Industry, Innovation and Infrastructure)," *National Journal of Environmental Law,* vol. 6, p. 6–17, 2023.

[8] Y. Ou, N. Kittner, S. Babaee, S. J. Smith, C. G. Nolte and D. H. Loughlin, "Evaluating long-term emission impacts of large-scale electric vehicle deployment in the US using a human-Earth systems model," *Applied Energy,* vol. 300, p. 117364, 10 2021.

[9] P. Poveda-Martínez, R. Peral-Orts, N. Campillo-Davo, J. Nescolarde-Selva, M. Lloret-Climent and J. Ramis-Soriano, "Study of the effectiveness of electric vehicle warning sounds depending on the urban environment," *Applied Acoustics,* vol. 116, p. 317–328, 1 2017.

[10] P. Ahmadi, "Environmental impacts and behavioral drivers of deep decarbonization for transportation through electric vehicles," *Journal of Cleaner Production,* vol. 225, p. 1209–1219, 7 2019.

Funded by the European Union

[11] L. Wang, V. Nian, H. Li and J. Yuan, "Impacts of electric vehicle deployment on the electricity sector in a highly urbanised environment," *Journal of Cleaner Production,* vol. 295, p. 126386, 5 2021.

[12] C. S. Demoulias, K.-N. D. Malamaki, S. Gkavanoudis, J. M. Mauricio, G. C. Kryonidis, K. O. Oureilidis, E. O. Kontis and J. L. Martinez Ramos, "Ancillary Services Offered by Distributed Renewable Energy Sources at the Distribution Grid Level: An Attempt at Proper Definition and Quantification," *Applied Sciences,* vol. 10, p. 7106, 10 2020.

[13] K. E. Forrest, B. Tarroja, L. Zhang, B. Shaffer and S. Samuelsen, "Charging a renewable future: The impact of electric vehicle charging intelligence on energy storage requirements to meet renewable portfolio standards," *Journal of Power Sources,* vol. 336, p. 63–74, 12 2016.

[14] S. Rahman, I. A. Khan, A. A. Khan, A. Mallik and M. F. Nadeem, "Comprehensive review & impact analysis of integrating projected electric vehicle charging load to the existing low voltage distribution system," *Renewable and Sustainable Energy Reviews,* vol. 153, p. 111756, 1 2022.

[15] X. Lu, K. Li, H. Xu, F. Wang, Z. Zhou and Y. Zhang, "Fundamentals and business model for resource aggregator of demand response in electricity markets," *Energy,* vol. 204, p. 117885, 8 2020.

[16] S. S. Kholerdi and A. Ghasemi-Marzbali, "Interactive Time-of-use demand response for industrial electricity customers: A case study," *Utilities Policy,* vol. 70, p. 101192, 6 2021.

[17] S. Hussain, R. R. Irshad, F. Pallonetto, I. Hussain, Z. Hussain, M. Tahir, S. Abimannan, S. Shukla, A. Yousif, Y.-S. Kim and H. El-Sayed, "Hybrid coordination scheme based on fuzzy inference mechanism for residential charging of electric vehicles," *Applied Energy,* vol. 352, p. 121939, 12 2023.

[18] B. Zhang, W. Hu, D. Cao, A. M. Y. M. Ghias and Z. Chen, "Novel Data-Driven decentralized coordination model for electric vehicle aggregator and energy hub entities in multi-energy system using an improved multi-agent DRL approach," *Applied Energy,* vol. 339, p. 120902, 6 2023.

[19] A. Petrucci, F. K. Ayevide, A. Buonomano and A. Athienitis, "Development of energy aggregators for virtual communities: The energy efficiency-flexibility nexus for demand response," *Renewable Energy,* vol. 215, p. 118975, 10 2023.

[20] S.-F. Yang, S.-W. Choi and E.-B. Lee, "A Prediction Model for Spot LNG Prices Based on Machine Learning Algorithms to Reduce Fluctuation Risks in Purchasing Prices," *Energies,* vol. 16, p. 4271, 5 2023.

[21] A. Tariq, S. A. A. Kazmi, G. Ali and A. H. U. Bhatti, "Multivariate stochastic modeling of plugin electric vehicles charging profile and grid impact analysis," *Sustainable Energy, Grids and Networks,* vol. 36, p. 101155, 12 2023.

Funded by the European Union

[22] L. Fu, T. Wang, M. Song, Y. Zhou and S. Gao, "Electric vehicle charging scheduling control strategy for the large-scale scenario with non-cooperative game-based multi-agent reinforcement learning," *International Journal of Electrical Power & Energy Systems,* vol. 153, p. 109348, 11 2023.

[23] A. Hafeez, R. Alammari and A. Iqbal, "Utilization of EV Charging Station in Demand Side Management Using Deep Learning Method," *IEEE Access,* vol. 11, p. 8747–8760, 2023.

[24] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron and A. Dubrawski, "Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx," *International Journal of Forecasting,* vol. 39, p. 884–900, 4 2023.

[25] E. Sarmas, E. Spiliotis, E. Stamatopoulos, V. Marinakis and H. Doukas, "Short-term photovoltaic power forecasting using meta-learning and numerical weather prediction independent Long Short-Term Memory models," *Renewable Energy,* vol. 216, p. 118997, 11 2023.

[26] M. Yamasaki, R. Z. Freire, L. O. Seman, S. F. Stefenon, V. C. Mariani and L. dos Santos Coelho, "Optimized hybrid ensemble learning approaches applied to very short-term load forecasting," *International Journal of Electrical Power & Energy Systems,* vol. 155, p. 109579, 1 2024.

[27] A. Feroz Mirza, M. Mansoor, M. Usman and Q. Ling, "Hybrid Inception-embedded deep neural network ResNet for short and medium-term PV-Wind forecasting," *Energy Conversion and Management,* vol. 294, p. 117574, 10 2023.

[28] S. Abimannan, E.-S. M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla and D. Satheesh, "Ensemble Multifeatured Deep Learning Models and Applications: A Survey," *IEEE Access,* vol. 11, p. 107194–107217, 2023.

[29] R. Ahmed, V. Sreeram, R. Togneri, A. Datta and M. D. Arif, "Computationally expedient Photovoltaic power Forecasting: A LSTM ensemble method augmented with adaptive weighting and data segmentation technique," *Energy Conversion and Management,* vol. 258, p. 115563, 4 2022.

[30] T. Mauldin, A. H. Ngu, V. Metsis and M. E. Canby, "Ensemble Deep Learning on Wearables Using Small Datasets," *ACM Transactions on Computing for Healthcare,* vol. 2, p. 1–30, 12 2020.

[31] G. Benedetto, E. Bompard, A. Mazza, E. Pons, R. Jaboeuf, P. Tosco and M. Zampolli, "Impact of bidirectional EV charging stations on a distribution network: a Power Hardware-In-the-Loop implementation," *Sustainable Energy, Grids and Networks,* vol. 35, p. 101106, 9 2023.

[32] T. Unterluggauer, J. Rich, P. B. Andersen and S. Hashemi, "Electric vehicle charging infrastructure planning for integrated transportation and power distribution networks: A review," *eTransportation,* vol. 12, p. 100163, 5 2022.

[33] S. Hussain, Y.-S. Kim, S. Thakur and J. G. Breslin, "Optimization of Waiting Time for Electric Vehicles Using a Fuzzy Inference System," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, p. 15396–15407, 9 2022.

Funded by the European Union

[34] H. Lin, J. Dang, H. Zheng, L. Yao, Q. Yan, S. Yang, H. Guo and A. Anvari-Moghaddam, "Two-stage electric vehicle charging optimization model considering dynamic virtual price-based demand response and a hierarchical non-cooperative game," *Sustainable Cities and Society,* vol. 97, p. 104715, 10 2023.

[35] S. Hussain, S. Thakur, S. Shukla, J. G. Breslin, Q. Jan, F. Khan, I. Ahmad, M. Marzband and M. G. Madden, "A Heuristic Charging Cost Optimization Algorithm for Residential Charging of Electric Vehicles," *Energies,* vol. 15, p. 1304, 2 2022.

[36] R. Naji EL idrissi, M. Ouassaid, M. Maaroufi, Z. Cabrane and J. Kim, "Optimal Cooperative Power Management Framework for Smart Buildings Using Bidirectional Electric Vehicle Modes," *Energies,* vol. 16, p. 2315, 2 2023.

[37] S. Hussain, S. Thakur, S. Shukla, J. G. Breslin, Q. Jan, F. Khan and Y.-S. Kim, "A two-layer decentralized charging approach for residential electric vehicles based on fuzzy data fusion," *Journal of King Saud University - Computer and Information Sciences,* vol. 34, p. 7391–7405, 10 2022.

[38] S. Hussain, M. A. Ahmed and Y.-C. Kim, "Efficient Power Management Algorithm Based on Fuzzy Logic Inference for Electric Vehicles Parking Lot," *IEEE Access,* vol. 7, p. 65467–65485, 2019.

[39] S. Hussain, K.-B. Lee, M. A. Ahmed, B. Hayes and Y.-C. Kim, "Two-Stage Fuzzy Logic Inference Algorithm for Maximizing the Quality of Performance under the Operational Constraints of Power Grid in Electric Vehicle Parking Lots," *Energies,* vol. 13, p. 4634, 9 2020.

[40] S. Hussain, M. A. Ahmed, K.-B. Lee and Y.-C. Kim, "Fuzzy Logic Weight Based Charging Scheme for Optimal Distribution of Charging Power among Electric Vehicles in a Parking Lot," *Energies,* vol. 13, p. 3119, 6 2020.

[41] J. Y. Yong, W. S. Tan, M. Khorasany and R. Razzaghi, "Electric vehicles destination charging: An overview of charging tariffs, business models and coordination strategies," *Renewable and Sustainable Energy Reviews,* vol. 184, p. 113534, 9 2023.

[42] A. Mohammad and F. Mahjabeen, "Revolutionizing solar energy: The impact of artificial intelligence on photovoltaic systems," *International Journal of Multidisciplinary Sciences and Arts,* vol. 2, 2023.

[43] P. Jiang, Y. Nie, J. Wang and X. Huang, "Multivariable short-term electricity price forecasting using artificial intelligence and multi-input multi-output scheme," *Energy Economics,* vol. 117, p. 106471, 1 2023.

[44] T. Mazhar, R. N. Asif, M. A. Malik, M. A. Nadeem, I. Haq, M. Iqbal, M. Kamran and S. Ashraf, "Electric Vehicle Charging System in the Smart Grid Using Different Machine Learning Methods," *Sustainability,* vol. 15, p. 2603, 2 2023.

[45] R. Li and X. Song, "A multi-scale model with feature recognition for the use of energy futures price forecasting," *Expert Systems with Applications,* vol. 211, p. 118622, 1 2023.

Funded by the European Union

[46] M. Heidarpanah, F. Hooshyaripor and M. Fazeli, "Daily electricity price forecasting using artificial intelligence models in the Iranian electricity market," *Energy,* vol. 263, p. 126011, 1 2023.

[47] C. McHugh, S. Coleman and D. Kerr, "Technical Indicators and Prediction for Energy Market Forecasting," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[48] A. Wagner, E. Ramentol, F. Schirra and H. Michaeli, "Short- and long-term forecasting of electricity prices using embedding of calendar information in neural networks," *Journal of Commodity Markets,* vol. 28, p. 100246, 12 2022.

[49] T. Ulgen and G. Poyrazoglu, "Predictor Analysis for Electricity Price Forecasting by Multiple Linear Regression," in *2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, 2020.

[50] A. Zaidi, "Investigation of domestic level EV chargers in the Distribution Network: An Assessment and mitigation solution," 2022.

[51] A. Saint-Pierre and P. Mancarella, "Active Distribution System Management: A Dual-Horizon Scheduling Framework for DSO/TSO Interface Under Uncertainty," *IEEE Transactions on Smart Grid,* vol. 8, p. 2186–2197, 9 2017.

[52] C. Jin, J. Tang and P. Ghosh, "Optimizing Electric Vehicle Charging: A Customer's Perspective," *IEEE Transactions on Vehicular Technology,* vol. 62, p. 2919–2927, 9 2013.

[53] T. M. Christensen, A. S. Hurn and K. A. Lindsay, "Forecasting spikes in electricity prices," *International Journal of Forecasting,* vol. 28, p. 400–411, 4 2012.

[54] J. P. Bharadiya, "Exploring the Use of Recurrent Neural Networks for Time Series Forecasting," 2023.

[55] R. Zhao, R. Yan, J. Wang and K. Mao, "Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks," *Sensors,* vol. 17, p. 273, 1 2017.

[56] L. Zhang, T. Ji, S. Yu and G. Liu, "Accurate Prediction Approach of SOH for Lithium-Ion Batteries Based on LSTM Method," *Batteries,* vol. 9, p. 177, 3 2023.

[57] H. Strobelt, S. Gehrmann, H. Pfister and A. M. Rush, "LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks," *IEEE Transactions on Visualization and Computer Graphics,* vol. 24, p. 667–676, 1 2018.

[58] O. Alkadi, N. Moustafa, B. Turnbull and K.-K. R. Choo, "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks," *IEEE Internet of Things Journal,* vol. 8, p. 9463–9472, 2020.

Funded by the European Union

[59] M. O'Connell, L. Moreira-Matias and W. Kan, *ECML/PKDD 15: Taxi Trajectory Prediction (I),* Kaggle, 2015.

[60] M. F. Luis Moreira-Matias, *Taxi Service Trajectory - Prediction Challenge, ECML PKDD 2015,* UCI Machine Learning Repository, 2013.

[61] A. Crivellari and E. Beinat, "LSTM-Based Deep Learning Model for Predicting Individual Mobility Traces of Short-Term Foreign Tourists," *Sustainability,* vol. 12, p. 349, 1 2020.

[62] F. Li, Z. Gui, Z. Zhang, D. Peng, S. Tian, K. Yuan, Y. Sun, H. Wu, J. Gong and Y. Lei, "A hierarchical temporal attention-based LSTM encoder-decoder model for individual mobility prediction," *Neurocomputing,* vol. 403, p. 153–166, 8 2020.

[63] A. Rossi, G. Barlacchi, M. Bianchini and B. Lepri, "Modelling Taxi Drivers' Behaviour for the Next Destination Prediction," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, p. 2980–2989, 7 2020.

[64] Z. Yi, X. C. Liu, R. Wei, X. Chen and J. Dai, "Electric vehicle charging demand forecasting using deep learning model," *Journal of Intelligent Transportation Systems,* vol. 26, p. 690–703, 8 2021.

[65] A. Orzechowski, L. Lugosch, H. Shu, R. Yang, W. Li and B. H. Meyer, "A data-driven framework for medium-term electric vehicle charging demand forecasting," *Energy and AI,* vol. 14, p. 100267, 10 2023.

[66] F. Pallonetto, C. Jin and E. Mangina, "Forecast electricity demand in commercial building with machine learning models to enable demand response programs," *Energy and AI,* vol. 7, p. 100121, 1 2022.

[67] C. Scott, M. Ahsan and A. Albarbar, "Machine learning for forecasting a photovoltaic (PV) generation system," *Energy,* vol. 278, p. 127807, 9 2023.

[68] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena,* vol. 404, p. 132306, 3 2020.

[69] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, p. 1735–1780, 11 1997.

[70] J. R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, p. 81–106, 3 1986.

[71] L. Breiman, *Machine Learning,* vol. 45, p. 5–32, 2001.

[72] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, p. 273–297, 9 1995.

[73] IEA, "Global EV Outlook 2023," 2023. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2023.

Funded by the European Union

[74] E. Akhavan-Rezai, M. F. Shaaban, E. F. El-Saadany and A. Zidan, "Uncoordinated charging impacts of electric vehicles on electric distribution grids: Normal and fast charging comparison," in *2012 IEEE Power and Energy Society General Meeting*, 2012.

[75] M. T. Hussain, D. N. B. Sulaiman, M. S. Hussain and M. Jabir, "Optimal Management strategies to solve issues of grid having Electric Vehicles (EV): A review," *Journal of Energy Storage,* vol. 33, p. 102114, 1 2021.

[76] P. Olivella-Rosell, R. Villafafila-Robles and A. Sumper, "Impact Evaluation of Plug-in Electric Vehicles on Power System," in *Plug In Electric Vehicles in Smart Grids*, Springer Singapore, 2014, p. 149–178.

[77] T. Montes, F. Pinsach Batet, L. Igualada and J. Eichman, "Degradation-Conscious Charge Management: Comparison of Different Techniques to Include Battery Degradation in Electric Vehicle Charging Optimization," 2023.

[78] A. Leippi, M. Fleschutz and M. D. Murphy, "A Review of EV Battery Utilization in Demand Response Considering Battery Degradation in Non-Residential Vehicle-to-Grid Scenarios," *Energies,* vol. 15, p. 3227, 4 2022.

[79] F. G. Venegas, M. Petit and Y. Perez, "Active integration of electric vehicles into distribution grids: Barriers and frameworks for flexibility services," *Renewable and Sustainable Energy Reviews,* vol. 145, p. 111060, 7 2021.

[80] M. Zade, Z. You, B. Kumaran Nalini, P. Tzscheutschler and U. Wagner, "Quantifying the Flexibility of Electric Vehicles in Germany and California—A Case Study," *Energies,* vol. 13, p. 5617, 10 2020.

[81] W. J. Requia, M. Mohamed, C. D. Higgins, A. Arain and M. Ferguson, "How clean are electric vehicles? Evidence-based review of the effects of electric mobility on air pollutants, greenhouse gas emissions and human health," *Atmospheric Environment,* vol. 185, p. 64–77, 7 2018.

[82] F. Pinsach Batet, R. Valdés Martin, L. Igualada and C. Corchero, "An optimal solution for a smart charging station of light electric vehicles," in *2022 International Conference on Smart Energy Systems and Technologies (SEST)*, 2022.

[83] L. Igualada, C. Corchero, M. Cruz-Zambrano and F.-J. Heredia, "Optimal Energy Management for a Residential Microgrid Including a Vehicle-to-Grid System," *IEEE Transactions on Smart Grid,* vol. 5, p. 2163–2172, 7 2014.

[84] L. Zhao and V. Aravinthan, "Strategies of residential peak shaving with integration of demand response and V2H," in *2013 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2013.

[85] M. Uddin, M. F. Romlie, M. F. Abdullah, S. Abd Halim, A. H. Abu Bakar and T. Chia Kwang, "A review on peak load shaving strategies," *Renewable and Sustainable Energy Reviews,* vol. 82, p. 3323–3332, 2 2018.

Funded by the European Union

[86] R. Tu, Y. (. Gai, B. Farooq, D. Posen and M. Hatzopoulou, "Electric vehicle charging optimization to minimize marginal greenhouse gas emissions from power generation," *Applied Energy,* vol. 277, p. 115517, 11 2020.

[87] J. Dixon, W. Bukhsh, C. Edmunds and K. Bell, "Scheduling electric vehicle charging to minimise carbon emissions and wind curtailment," *Renewable Energy,* vol. 161, p. 1072–1091, 12 2020.

[88] A. Rabiee, A. Ghiasian and M. A. Chermahini, "Long term profit maximization strategy for charging scheduling of electric vehicle charging station," *IET Generation, Transmission & Distribution,* vol. 12, p. 4134–4141, 9 2018.

[89] EAFO, "Electric vehicle model statistics," 2023. [Online]. Available: https://alternative-fuels-observatory.ec.europa.eu/policymakers-and-public-authorities/electric-vehicle-model-statistics. [Accessed 27 November 2023].

[90] H. M. Louie, "Time-series modeling of aggregated electric vehicle charging station load," *Electric Power Components and Systems,* vol. 45, no. 14, pp. 1498-1511, 2017.

[91] M. H. Amini, A. Kargarian and O. Karabasoglu, “ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation,” *Electric Power Systems Research,* vol. 140, no. Elsevier, pp. 378-390, 2016.

[92] M. Straka and Ľ. Buzna, "Use cases and introductory analysis of the dataset collected within the large network of public charging stations," *Lecture notes in networks and systems,* pp. 203-213, 2019.

[93] A. Lucas, R. Barranco and N. Refa, "EV Idle Time Estimation on Charging Infrastructure, Comparing Supervised Machine Learning Regressions," *Energies,* vol. 12, no. 2, p. 269, 2019.

[94] xgboost developers, "XGBoost Documentation," 2022. [Online]. Available: https://xgboost.readthedocs.io/en/stable/. [Accessed 30 Nov. 2023].

[95] S. Xydas, C. Marmaras, L. Cipcigan, A. Hassan and N. Jenkins, "Electric Vehicle Load Forecasting using Data Mining Methods," in *IET Hybrid and Electric Vehicles Conference 2013 (HEVC 2013)*, London, 2013.

[96] L. Dannecker, Energy Time Series Forecasting: Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain, Springer Vieweg, 2015.

[97] S. Koohfar, W. Woldemariam and A. Kumar, "Prediction of Electric Vehicles Charging Demand: A Transformer-Based Deep Learning Approach," *Sustainability,* vol. 15, no. 3, p. 2105, 2023.

[98] City of Boulder, Colorado, "Electric Vehicle Charging Station Data," [Online]. Available: https://open-data.bouldercolorado.gov/search?collection=Dataset&q=Electric%20Vehicle%20Charging%20Station%20Data. [Accessed 30 Nov. 2023].

Funded by the European Union

[99] J. Shanmuganathan, A. A. Victoire, G. Balraj and A. Victoire, "Deep Learning LSTM Recurrent Neural Network Model for Prediction of Electric Vehicle Charging Demand," *Sustainability,* vol. 14, no. 16, p. 10207, 2022.

[100] H. Jahangir, H. Tayarani, A. Ahmadian, M. A. Golkar, J. Miret, M. Tayarani and H. Gao, "Charging demand of Plug-in Electric Vehicles: Forecasting travel behavior based on a novel Rough Artificial Neural Network approach," *Journal of Cleaner Production,* vol. 229, pp. 1029-1044, 2019.

[101] A. Sao, N. Tempelmeier and E. Demidova, "Deep Information Fusion for Electric Vehicle Charging Station Occupancy Forecasting," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, IN, USA, 2021.

[102] T.-Y. Ma and S. Faye, "Multistep electric vehicle charging station occupancy prediction using hybrid LSTM neural networks," *Energy,* vol. 244, p. 123217, 2022.

[103] "Dundee City Council Open Data Portal," [Online]. Available: https://data.dundeecity.gov.uk/. [Accessed 30 11 2023].

[104] M. BRYANT, "Electric Vehicle Charging Dataset," 2022. [Online]. Available: https://www.kaggle.com/datasets/michaelbryantds/electric-vehicle-charging-dataset. [Accessed 30 Nov. 2023].

[105] Caltech, "ACN-Data -- A Public EV Charging Dataset," [Online]. Available: https://ev.caltech.edu/dataset. [Accessed 30 Nov. 2023].

[106] Open Data Réseaux Énergies, "Transactions quotidiennes des charges des véhicules électriques de fonction de SAP Labs France," 05 July 2023. [Online]. Available: https://www.data.gouv.fr/fr/datasets/transactions-quotidiennes-des-charges-des-vehicules-electriques-de-fonction-de-sap-labs-france/#/resources. [Accessed 30 Nov. 2023].

[107] London Borough of Barnet, "Electric Vehicle Charging Transactions," 26 May 2023. [Online]. Available: https://www.data.gov.uk/dataset/16c7326b-57fe-4803-88f8-9286c387f68a/electric-vehicle-charging-transactions. [Accessed 30 Nov. 2023].

[108] Perth and Kinross Council, "Open Data," 28 Sep. 2023. [Online]. Available: https://data.pkc.gov.uk/search?collection=Dataset&q=EV%20. [Accessed 30 Nov. 2023].

[109] City of Palo Alto, "Electric Vehicle Charging Station Usage (July 2011 - Dec 2020)," [Online]. Available: https://data.cityofpaloalto.org/dataviews/257812/electric-vehicle-charging-station-usage-july-2011-dec-2020/. [Accessed 1 Dec. 2023].

[110] Direction de la Voirie et des Déplacements - Ville de Paris, "Belib' - Points de recharge pour véhicules électriques - Disponibilité temps réel," [Online]. [Accessed 1 Dec. 2023].

Funded by the European Union

[111] Department for Transport (UK), "Electric Chargepoint Analysis 2017: Domestics," 13 Dec. 2018. [Online]. Available: https://www.data.gov.uk/dataset/5438d88d-695b-4381-a5f2-6ea03bf3dcf0/electric-chargepoint-analysis-2017-domestics. [Accessed 1 Dec. 2023].

[112] Nixtla, "StatsForecast," [Online]. Available: https://nixtla.mintlify.app/statsforecast/index.html. [Accessed 1 Dec. 2023].

[113] J. A. Fiorucci, T. R. Pellegrini, F. Louzada, F. Petropoulos and A. B. Koehler, "Models for optimising the theta method and their relationship to state space models," *International Journal of Forecasting,* vol. 32, no. 4, pp. 1151-1161, 2016.

[114] Nixtla, "Optimized Theta Model," [Online]. Available: https://nixtla.mintlify.app/statsforecast/docs/models/optimizedtheta.html. [Accessed 1 Dec. 2023].

[115] "scikit-learn - Machine Learning in Python," [Online]. Available: https://scikit-learn.org/stable/index.html. [Accessed 1 Dec. 2023].

[116] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, LongBeach,CA,USA, 2017.

[117] Microsoft Corporation, "LightGBM," [Online]. Available: https://lightgbm.readthedocs.io/en/stable/. [Accessed 1 Dec. 2023].

[118] scikit-learn, "Time-related feature engineering," [Online]. Available: https://scikit-learn.org/stable/auto_examples/applications/plot_cyclical_feature_engineering.html. [Accessed 1 Dec. 2023].

Funded by the European Union