Multi-Step Ahead Electrical Load Forecasting: An Australian Case-Study

S. Mahdi Noori R.A. Ahmad Attarha Masoume Mahmoodi José Iria Dan Gordon Paul Scott

The Australian National University, College of Engineering, Computing and Cybernetics

***Abstract*—The increasing adoption of distributed energy resources (DER) by residential customers has increased the importance of visibility in electrical distribution networks. Electrical load forecasting of individual customers can help the system operators better anticipate network safety problems and act accordingly. However, high volatility and randomness in individual customer power exchanges with the grid and the massive number of customers in the grid make electrical load forecasting challenging. In addition, telemetry data for different customers are collected at varying intervals, ranging from every 30 minutes to monthly, adding complexity to load prediction. This paper presents our findings in the load forecasting component within an Australian trial project context. We explore various optimisation models and machine learning algorithms for short- and medium-term load forecasting problems and offer clear recommendations for selecting appropriate forecasting techniques. We also introduce an approach for medium-term electrical forecasting problems and, using numerical and statistical experiments, show that it outperforms alternative methods.**

***Index Terms*—Electrical load forecasting, multi-step forecasting, renewable integration, distribution systems**

I. Introduction

The adoption of distributed energy resources (DER) has presented new technical challenges for distribution systems. To address these challenges, dynamic operating envelopes (DOEs) have emerged as a class of techniques for managing constrained distribution network capacity. DOEs allocate time-varying power envelopes either per customer or for regional groups of customers. Project Converge [1] is developing a new approach that integrates aggregator/customer preferences, wholesale market services, and network support into the calculation of operating envelopes. This results in "shaped" operating envelopes tailored to reflect values beyond pure network constraint management. At the core of this approach is an iterative solution to a near real-time optimal power flow (OPF) problem, which relies on accurate forecasts of future customer electrical load consumption.

This paper presents our findings in the load forecasting component of Project Converge. Our study focused on two types of customers: (i) direct trial participants who shared their 30-minute historical data and updated it every 30 minutes, and (ii) other customers connected to the same electrical feeder as trial participants, who shared their 30-minute historical data but only updated it once a month. To solve the OPF problem, we needed to forecast these customers' electrical load for the next day, a task known as multi-step forecasting in the literature [2].

Electrical load forecasting has been a significant research area in recent decades [3]. Prior work has proposed algorithms for forecasting the electrical load of a single customer using various types of input, such as past load observations [4], weather data [5] and calendar data [6]. The existing literature includes several forecasting algorithms, each employing a distinct strategy to utilise the input information to estimate the immediate next time-step load values. Examples of such techniques are the linear regression models [7], random forest regression [8], and extreme gradient boosting [9]. Furthermore, these algorithms are typically utilised within a multi-step ahead forecasting framework to extend the predictions beyond a single future point and forecast a sequence of values in a time series. Examples of such frameworks are Iterated, Direct, and Rectified multi-step ahead forecasting [10].

Unlike the available literature that often assumes access to only certain information, our study considers the impact of various information types because, in practice, electrical utilities can acquire most of the above information at a cost. Also, unlike the existing literature, we consider a wide range of load forecasting techniques for a given real-world setting, which allows us to assess accuracy, computational burden, and sensitivity to the length of the training data of such techniques and provide practical recommendations regarding load forecasting approaches for electrical utilities. In addition, most of the literature in electrical load forecasting is focused on real power prediction [11], while the forecasting of reactive power is essential for our, and most OPF-based, studies. To the best of our knowledge, this is the first work that studies the real and reactive power prediction problem using various inputs and forecasting algorithms using a real-world heterogeneous telemetry electrical load dataset.

This study uses optimisation models and machine learning algorithms to investigate various short-term and medium-term multi-step point-based electrical load forecasting techniques. The assessment involves evaluating the scalability and accuracy of each method through numerical experiments and analytical approaches. The dataset includes data from over 5000 customers in Canberra, Australia, from 2022-2023. Our analysis reveals that, for short-term load forecasting, the models exhibit similar prediction errors due to the high volatility and randomness in individual electricity consumption patterns. Consequently, we recommend adopting a linear regression model with the Iterated algorithm for short-term individual electricity forecasting. This approach effectively mitigates the computational burden associated with more complex regression algorithms and multi-step ahead forecasting methods.

Additionally, our study demonstrates that the performance of these approaches significantly deteriorates as the forecast period lengthens. To address this, we develop a specialised approach for medium-term load forecasting that leverages participant customers as proxy measurements. By analysing correlation and applying the Granger causality test [12], we identify the most suitable participant customers' data for forecasting non-participant customers' load. Using an optimisation model, we then optimise the parameters of a function that maps the selected participant customers and historical data of non-participant customers to future estimations of the non-participant customers' electricity load. Numerical simulations confirm the improvement in forecast accuracy for non-participant customers.

The rest of this paper is organised as follows. Section II introduces the modelling notation and the multi-step ahead load forecasting problem. Section III details the forecasting approaches. Section IV reports the simulation results, and Section V concludes this paper.

II. Problem Formulation

As discussed in the Introduction section, we deal with two classes of problems, i.e., relatively short-term and medium-term forecasting horizons. In the following, we first introduce the modelling notation, and then present the problem formulation.

A. Modelling Notation

Let set $\mathcal{N} := \{0, ..., N\}$ denote the customers that their electrical load[1] has to be forecasted. For the i-th customer, let sets $\mathcal{T}_i^h := \{0, ..., T_i\}$ and $\mathcal{T}_i^f := \{T_i+1, ..., T_i'\}$ denote the time-steps with historical data and the future time-steps, respectively. Let y_{it} denote the i-th customer electrical load at the t-th time step. Throughout this paper, we use bold symbols to differentiate between the variables and parameters in the optimisation models, where parameters are denoted with bold symbols.

Let $\mathcal{F}$ denote the set of all possible or acceptable functions that map available data to a prediction. For each function $f(x_t^d; \theta_f) \in \mathcal{F}$, let vector $x_t^d \in \mathbb{R}^{1\times(m\times d)}$ collects the prior d observations of m features, such as historical load and calendar data, from the current observation t. The function f is characterised by its parameters $\theta_f \in \Theta_f$, where Θ_f represents the parameter space. To simplify notation, we refer to the parameters of the function f as θ. The load forecasting algorithms discussed in detail in the following sections restrict $\mathcal{F}$ to a specific family of functions, such as affine functions.

In our study, we differentiate between the participant and non-participant customers in the trial based on their historical data availability. To do so, we define sets $\mathcal{N}^p := \{0, ..., N^p\}$ and $\mathcal{N}^n := \{0+1, ..., N^n\}$, where $N^n + N^p = N$, which represent the participant and non-participant customers in the trial, respectively. We also define y_{it}^p and y_{it}^n to denote the i-th participant and non-participant customer electrical load at the t-th time step, respectively. Additionally, we use sets $\mathcal{T} := \{0, ..., T\}$, $\mathcal{T}' := \{T+1, ..., T'\}$ and $\mathcal{T}'' := \{T'+1, ..., T''\}$ denote the time-steps with historical data for both participant and non-participant customers, the time-steps with historical data for only participant customers and the future time-steps, respectively. Figure 1 shows the schematic of data availability of participant and non-participant customers.

[1]We use the term electricity load to refer to the net real power consumption of each customer, which can have a positive or negative value.

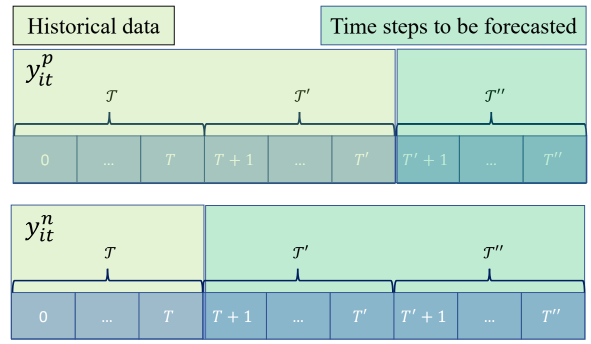

Fig. 1. Schematic of the multi-step short-term and medium-term load forecasting problems for the participant and non-participant customers in the trial.

B. Electrical load forecasting problem formulation

We define the electrical load forecasting problem in our trial as follows:

Given:

$$\boldsymbol{y}_{it} \;\; \forall i \in \mathcal{N}^p, \; \forall t \in \mathcal{T} \cup \mathcal{T}' \text{ and } \boldsymbol{y}_{it} \;\; \forall i \in \mathcal{N}^n, \; \forall t \in \mathcal{T}, \tag{1a}$$

for participant customers:

Estimate:

$$y_{it} \;\; \forall i \in \mathcal{N}^p, \; \forall t \in \mathcal{T}'', \tag{1b}$$

and for non-participant customers:

Estimate:

$$y_{it} \;\; \forall i \in \mathcal{N}^n, \; \forall t \in \mathcal{T}' \cup \mathcal{T}''. \tag{1c}$$

We refer to the participant customer problem, denoted by (1a) and (1b), as short-term electrical load forecasting. In our study, this period is 48 steps, given the 30-minute data resolution for one day. Also, we refer to the non-participant customer problem, denoted by (1a) and (1c), as medium-term electrical load forecasting. In our study, this period is 1440 steps, given the 30-minute data resolution for one month.

In the following sections, we discuss how different load forecasting algorithms deal with the short-term and medium-term electrical load forecasting presented above.

III. Electrical Load Forecasting Algorithms

To solve the short-term and medium-term forecasting problems introduced above, a forecasting approach must first address several key questions:

- What data is available for training the forecasting model? The answer to this question dictates the input to the forecasting approaches.
- How can the selected information be related to future estimations? This aspect is often referred to as the regression model in the literature.
- What is an appropriate loss function to quantify the quality of the estimations? The answer to this question will determine the objective functions used in the forecasting approaches to train and optimise the parameters of the regression models.
- What is the optimal strategy for estimating a customer's load for multiple time steps into the future?

This section presents different responses to these questions and introduces various load forecasting approaches that are distinguished by their unique solutions to these challenges.

A. Training data

1) Individual historical data: The forecasting approaches rely solely on each customer's historical data to predict their future energy usage. These approaches are commonly referred to as autoregressive models in the literature.

2) Weather data: Studies, such such as [13], have shown a strong correlation between the weather data elements such temperature and energy consumption. Thus, we use historical weather data obtained from [14], which includes temperature, humidity, precipitation and solar radiation as proxy measurement in the load forecasting models.

3) Calendar data: In addition to using customers' historical data, calendar data can be incorporated in load forecasting models. Specifically, we consider time-of-day and days-of-week information in our study. To account for time-of-day, we use a trigonometric encoding that maps the time to coordinates on a circle using the following formula:

$$t_x = \sin(2\pi \frac{60t_h + t_m}{1440}), \quad t_y = \cos(2\pi \frac{60t_h + t_m}{1440}), \tag{2}$$

where t_x and t_y represent the circle coordinates, and t_h and t_m represent the hour and minute of the day, respectively. To incorporate days-of-week information, we use a binary variable that takes a value of 1 for weekends and holidays, and 0 otherwise.

B. Regression models

1) Linear regression: This model adopts a linear process to model the data of each customer by defining the linear function $f_i^{LR} \in \mathcal{F}$ as:

$$y_{it} = f_i^{LR}(x_{it}^d; \theta_i) + \epsilon_{it} = A_i x_{it}^d + b_i + \epsilon_{it}, \tag{3}$$

where $A_i \in \mathbb{R}^{(m \times d) \times 1}$ and $b_i \in \mathbb{R}$ are the slope and the y-intercept of the linear function f_i^{LR}, and ϵ_{it} denotes a stochastic independent and identically distributed residual. This model then proceeds to optimise the parameters of the linear function, i.e., θ_i, by solving the following optimisation problem. When presenting each approach, we use the mean square error as the loss function for simplicity. However, in Sections III-D and Section IV, we will introduce alternative loss functions and analyse their impact on the accuracy and computation speed of each forecasting algorithm, respectively.

$$\theta_i^* = \underset{\theta_i \in \Theta_{f_i^{LR}}}{\arg\min} \sum_{t \in \mathcal{T}_i^h} \Big(\boldsymbol{y}_{it} - f_i^{LR}(\boldsymbol{x}_{it}^d; \theta_i) \Big)^2. \tag{4}$$

2) Random Forest Regression: Random Forest (RF) Regression [15] is a powerful ensemble learning algorithm extensively used for regression tasks in machine learning. The algorithm builds multiple decision trees during the training phase, with each tree being constructed on a random subset of the data and features. This diversity in the data subsets and feature selection reduces the risk of overfitting and significantly enhances the model's generalisation capabilities. When making predictions, each decision tree independently generates its prediction for a new data point. The final prediction from RF is obtained by averaging the individual predictions from all the decision trees. This averaging process smoothens individual decision trees' noise, resulting in a more robust and accurate prediction [16]. Below is a brief summary of the RF algorithm's structure.

Let each element in set $\mathcal{S} := \{1, ..., S\}$ denote a decision tree. Then the RF algorithm states:

$$y_{its} = f_{is}^{FR}(x_{its}^d; \theta_{is}) + \epsilon_{its} \qquad \forall s \in \mathcal{S} \tag{5}$$

Optimising the parameters of each decision tree function, the prediction is then obtained by averaging the predictions from all trees as:

$$y_{it} = \frac{1}{S} \sum_{s=1}^{S} f_{is}^{FR}(x_{its}^d; \theta_{is}^*). \tag{6}$$

3) Extreme gradient boosting (XGBoost): XGBoost is a supervised learning algorithm designed for regression and classification problems [9]. It is based on the Gradient Boosting framework, which sequentially adds weak decision trees to the ensemble and improves the model performance by minimising the loss function through gradient descent. Furthermore, XGBoost includes regularisation techniques to prevent overfitting and improves the model's generalisation performance. Here, we present a brief summary of the XGBoost algorithm's structure. However, we recommend referring to the original research paper [9] and the XGBoost implementation tutorial [17] for a more comprehensive understanding.

Let function $g_k() \in \mathcal{G}$ denote the k-th regression tree, where $\mathcal{G}$ is the set of all possible regression trees. XGBoost models the customer i's electricity load as the following:

$$y_{it} = f_i^{XGB}(x_{it}^d, \theta_i) + \epsilon_{it} = \sum_{k=1}^{K} g_{ik}(x_{it}^d) + \epsilon_{it}, \tag{7}$$

where K is the number of trees used to estimate the load value, and will be determined using the XGBoost algorithm.

Next, this model formulates an optimisation problem to minimise a loss function, minimising the distance between estimations and historical data. In addition, it incorporates a regularisation term to prevent overfitting by considering the complexity of the decision tree. The loss function is formulated as follows:

$$\min \sum_{t\in\mathcal{T}_i^h} \Big(\boldsymbol{y}_{it} - \sum_{k=1}^{K} g(\boldsymbol{x}_{it}^d)\Big)^2 + \sum_{k=1}^{K} w(g_k) \tag{8}$$

where $w(g_k)$ denotes the complexity of the k-th regression tree, and is defined as follows:

$$w(g_k) = \boldsymbol{\gamma} T + \frac{1}{2}\boldsymbol{\beta}\sum_{j=1}^{T} w_j. \tag{9}$$

In (9), T denotes the number of leaf nodes in the tree, w_j is the score of the leaf nodes, and γ and β are the parameter to control the complexity of the tree. In summary, by solving the optimisation (8), we optimise the parameters of function f_i^{XGB} and obtain $\boldsymbol{\theta}_i^*$.

C. Multi-step forecasting strategies

The regression models presented in Section III-B are designed to model the data of each customer by assigning a function $f \in \mathcal{F}$, optimising the parameters of the assigned function, and calculating the next immediate time-step based on the observed value. However, for multi-time step forecasting, it is necessary to develop an algorithm on top of the introduced regression models to estimate a sequence of values for future time steps. In what follows, we present four algorithms that address the multi-step load forecasting problem.

1) Iterated multi-step (IM) algorithm: The IM forecasting algorithm uses a recursive approach to make multi-step forecasts. It repeatedly forecasts a target variable for multiple time steps ahead, using the previously forecasted values as inputs for subsequent forecasts. In this algorithm, a single model is trained using a regression model to predict the next time step. Then, the same model is used iteratively to predict future time steps, using the forecasted value as input for each step. The IM algorithm can be summarised for each customer i as follows:

Algorithm 1 Iterated multi-step algorithm

• Optimise the parameters of $f_i(\boldsymbol{x}_i^d, \theta_i)$, i.e., calculate $\boldsymbol{\theta}_i^*$
for $t \in \mathcal{T}_i^f$ **do**:
  $y_{it} = f_i(\boldsymbol{x}_i^d; \boldsymbol{\theta}_i^*)$
  Update x_i

2) Direct multi-step (DM) algorithm: Unlike the IM algorithm, this algorithm considers a separate model for each time step that needs to be forecasted. Specifically, this model for customer i is written as:

$$y_{i,t} = f_{i\tau}(\boldsymbol{x}_{i,t-\tau}^d; \theta_{i\tau}) + \epsilon_{it}, \tag{10}$$

where $\tau_i = T_i - t$. It is worth noting that the DM algorithm differs from the IM algorithm regarding the input provided to the regressor function. While the IM algorithm recursively updates the input using the last forecast, the DM algorithm solely relies on historical data as input. Instead, the DM assigns a separate function ($f_{i\tau}$) for each time step and optimises its parameters to obtain $\boldsymbol{\theta}_{i\tau}^*$, using a regression model. The DM algorithm can be summarised for each customer i as follows:

Algorithm 2 Direct multi-step algorithm

for $t \in \mathcal{T}_i^f$ **do**:
  Optimise the parameters of $f_{i\tau}(\boldsymbol{x}_{i,t-\tau}^d, \theta_{i\tau})$: $\boldsymbol{\theta}_{i\tau}^*$
  $y_{it} = f_{i\tau}(\boldsymbol{x}_{i,t-\tau}^d; \boldsymbol{\theta}_{i\tau}^*)$

3) Rectified multi-step (RM) algorithm: The Rectified algorithm is a hybrid approach that combines the advantages of the IM and DM forecasting methods [10]. It aims to produce more accurate forecasts by first generating IM forecasts and then refining them using a DM strategy. To achieve this, the RM algorithm first uses Algorithm 1 to generate a sequence of estimations for future time steps ($\hat{y}_{it}$). It then trains a regression model for each time step using the DM algorithm on the residuals of the IM estimations. The residual is the difference between the actual value and the IM forecast ($\hat{y}_{it}$). The DM model is trained to predict these residuals using the input features $\boldsymbol{x}_{i\tau}$, and generate a sequence of estimations for the errors in the IM forecasts ($\hat{e}_{it}$). The RM algorithm can be summarised for each customer i as follows:

Algorithm 3 Rectified multi-step algorithm

• Generate an initial multi-step estimation forecast using Algorithm 1
for $t \in \mathcal{T}_i^h$ **do**:
  Generate an estimation of the errors using Algorithm 2
  $y_{it} = \hat{y}_{it} + \hat{e}_{it}$

4) Specialised forecasting strategy for non-participant customers: Algorithms 1-3 can be used in both short-term and medium-term multi-step load forecasting. However, our analysis shows a challenge: as the forecast period increases, the performance of these models deteriorates significantly. This challenge becomes particularly pressing in our trial, where we must generate forecasting values for non-participating customers' next 1440 time steps. To deal with this challenge, we leverage real-time telemetry data from participating customers to enhance the forecasts for non-participants. As mentioned in the Introduction section, the data for non-participating customers are updated only monthly, while participants' data are refreshed every 30 minutes. Thus, we have devised an algorithm that utilises the load information from participating customers as a proxy measurement to forecast the load of non-participating customers.

Our modelling process first requires determining which participant customers' data should be used to forecast the electrical load of each non-participant customer. To accomplish this, we begin by conducting a correlation study, which allows

TABLE I

A SUMMARY DIFFERENT FORECASTING MODELLING TECHNIQUES.

	Input			Regression			Multi-step			Loss function		
	historical	weather	calendar	lin.	R.F.	XGB.	Iterated	Direct	Rectified	Ridge	Lasso	MSE
Tech 1	✓			✓			✓			✓		
Tech 2	✓	✓	✓	✓			✓			✓		
Tech 3	✓	✓		✓			✓			✓		
Tech 4	✓		✓	✓			✓			✓		
Tech 5	✓			✓			✓				✓	
Tech 6	✓			✓			✓					✓
Tech 7	✓			✓				✓		✓		
Tech 8	✓			✓					✓	✓		
Tech 9	✓				✓		✓					✓
Tech 10	✓	✓	✓		✓		✓					✓
Tech 11	✓					✓	✓				✓	
Tech 12	✓	✓	✓			✓	✓				✓	

us to rank the participant customers based on their correlation coefficient. Next, we perform a Granger causality test [18] to determine whether a participant customer's load can be used to forecast a non-participant customer's load. It is important to note that while the correlation between two time series measures the strength of their linear relationship, it does not necessarily indicate that one time series can be used to predict the other [19]. To utilise one time series to predict the other, there must be a causal relationship between them, which can be identified using statistical techniques such as the Granger causality test.

After identifying which participant customers' data should be used for each non-participant customer (we refer these participant customer by set $\mathcal{N}_i^p$ for non-participant i), we solve the following optimisation problem to optimise the parameters of the function $f_i^{SFS} \in \mathcal{F}$:

$$\theta_i^* = \underset{\theta_i \in \Theta_{f_i}}{\arg\min} \sum_{t \in \mathcal{T}_i^h} \left(\boldsymbol{y}_{it}^n - f_i^{SFS}(\boldsymbol{x}_i'; \theta_i) \right)^2. \tag{11}$$

Parameter $\boldsymbol{x}_i' = [x_{it}^d, x_{it}'']$, where $x_{it}'' = [y_{it}^p \quad \forall i \in \mathcal{N}_i^p]^\intercal$, denotes the historical data of i-th non-participant customer along with the few selected participant customers as real-time proxy measures. Then, for each non-participant customer, the forecasting values for the time step in $\mathcal{T}'$ (where the participant customers data is available) are obtained as follows:

$$y_{it}^n = f_i^{SFS}(\boldsymbol{x}_{it}'; \boldsymbol{\theta}_i^*). \tag{12}$$

For the remaining time steps, i.e., $t \in \mathcal{T}''$, we first forecast the chosen participant customers' load using one of the Algorithms 1-3. Then, we use (12) to forecast the non-participant customers' load. Our proposed algorithm can be summarised for each customer i as follows:

Algorithm 4 Non-participant customers forecasting algorithm

- Identify suitable participant customers to be used for the forecasting
- Use one of the Algorithm 1-3 to forecast the selected participant customers' load
- Optimise the parameters of $f_i^{SFS}(\boldsymbol{x}_{it}', \theta_i)$ by solving (11)

for $t \in \mathcal{T}_i^f$ **do**:
    $y_{it}^n = f_i^{SFS}(\boldsymbol{x}_{it}'; \boldsymbol{\theta}_i^*)$

D. Regression loss functions

Our study examines a set of quadratically convex loss functions that have been extensively employed in the load forecasting literature. Each loss function is an objective function of a particular load forecasting algorithm that measures the distance between the actual data y_i and our prediction based on f. To this end, we define two vectors: $y_i = [y_{it} \quad \forall t \in \mathcal{T}_i^h]$ and $\hat{y}_i(f_{\theta_i}) = [f_i(x_t^d; \theta_i) \quad \forall t \in \mathcal{T}_i^h]$ which capture the i-th customer's historical data and our estimates of it based on function f, respectively. Below, we briefly introduce the loss functions employed in our study.

1) Mean square error:

$$\theta_i^* = \underset{\theta_i \in \Theta_{f_i}}{\arg\min} \ \|\boldsymbol{y}_i - \hat{y}_i(f_{\theta_i})\|_2^2. \tag{13}$$

2) Ridge:

$$\theta_i^* = \underset{\theta_i \in \Theta_{f_i}}{\arg\min} \ \|\boldsymbol{y}_i - \hat{y}_i(f_{\theta_i})\|_2^2 + \boldsymbol{\alpha}_i \|\theta_i\|_2^2, \tag{14}$$

where α_i is a constant used to control regularisation strength.

3) Lasso:

$$\theta_i^* = \underset{\theta_i \in \Theta_{f_i}}{\arg\min} \ \frac{1}{2T_i} \|\boldsymbol{y}_i - \hat{y}_i(f_{\theta_i})\|_2^2 + \boldsymbol{\beta}_i \|\theta_i\|. \tag{15}$$

where β_i is a constant used to control regularisation strength.

Table I summarises the twelve modelling techniques derived from having different inputs, regression models, multi-step forecasting algorithms and loss functions. Section IV uses the same technique numbers when reporting the numerical results.

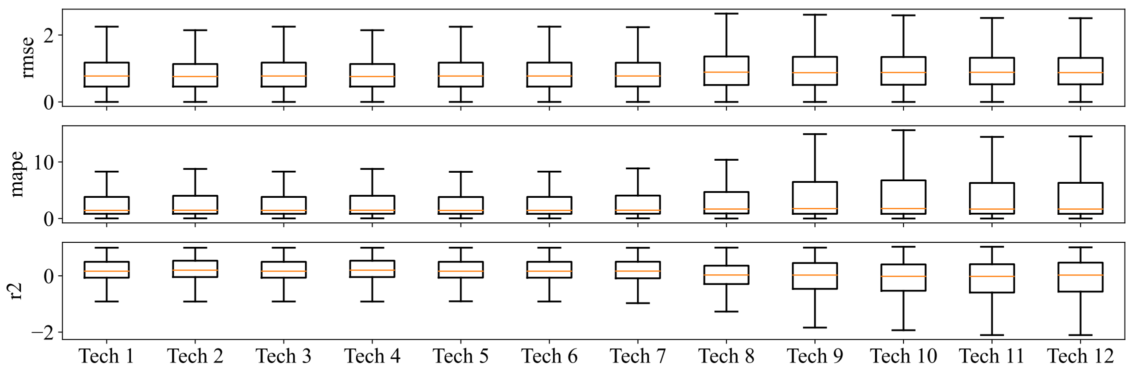

Fig. 2. Boxplot of RMSE, MAPE and R^2 of each technique in the short-term real power load forecasting problem.

IV. Experiment Results

This section presents the outcomes of our numerical experiment, focusing on the accuracy of diverse load forecasting models. To accomplish this, we perform a time-series cross-validation study, which provides a robust performance estimate and enables us to evaluate the models' sensitivity to the length of the training data [20]. Our investigation employs various techniques to assess the suitability of each model for forecasting customers' electrical load. These techniques include time-series visualisation, numerical indices such as root mean square error (RMSE) and the mean absolute percentage error (MAPE), the coefficient of determination (R^2) as a statistical measure [21], and the Kolmogorov-Smirnov test (KST) as a statistical test [22]. Since the telemetry data in our study reports the real and reactive power in kW and kVAR, the RMSE is reported in in kW and kVAR in the subsequent sections. In the following sections, we study the models in the short-term and medium-term forecasting problems, respectively.

A. Simulation Platform

The numerical simulations are carried out using Python scripts, with optimisation models and machine learning algorithms implemented through the *scikit-learn* [23] and *sk-forecast* [24] Python packages. All the codes are accessible online in an open-source package at [25]. The experiments are performed on a server equipped with a 128-core CPU and 128GB RAM.

B. Results in the short-term forecasting problem

Figure 2 displays the boxplot of RMSE, MAPE, and R^2. Additionally, Tables II and III present the median values of each index for each technique in the short-term forecasting problem for real and reactive power predictions, respectively. In addition, these tables provide the average KST results for different techniques. To obtain the KST values, we subject the residuals of the predictions of each model to the KST, comparing them with a normal distribution for each customer and period of the cross-validation. Subsequently, we use an indicator function that returns a value of one if the p-value of the test is greater than 0.05 and zero otherwise. Finally, we calculate the average output of the indicator function for each model and report the percentage in Table II. We also report the average time for training a model and generating predictions in Table II in microseconds.

First, we investigate the impact of including weather and calendar data as proxy measures on the accuracy of different models. Comparing Tech 1 to Tech 4 in Fig 2 and Tables II and III, we observe that incorporating weather data does not significantly improve model accuracy. However, including calendar data results in up to 2% and 30% enhancement in predicting real and reactive power, respectively, in our experiments. Next, we explore the effect of the loss function choice on model accuracy. Comparing Tech 1, Tech 5, and Tech 6, all utilising the Iterated method with the same input, we find that the choice of the loss function has a negligible effect on average model accuracy.

Additionally, we assess the use of Direct and Rectified approaches compared to the Iterated method (Tech 1, Tech 7, and Tech 8). The Iterated method outperforms the other two, with the Rectified method demonstrating the worst performance despite its higher computational cost. We next examine the effect of regression models on load forecasting accuracy by comparing Tech 1, Tech 9, and Tech 12. Surprisingly, the linear regression model outperforms the more complex Random Forest and XGBoost regression algorithms. Furthermore, Fig 3 shows the medium of RMSE, MAPE and R^2 of each technique for each period of the time-series cross-validation. We observe a consistent performance improvement for all techniques as the length of the training data increases.

Based on the results in Fig 2 and Tables II and III, we find that the prediction errors in all models are relatively close to each other. This similarity is attributed to the high volatility and randomness in individual electricity consumption patterns. Thus, considering the computational burden of more complex regression algorithms and multi-step ahead approaches, we recommend using the linear regression model with the Iterated algorithm for the short-term single-customer electricity forecasting problem. Our analysis also shows that

TABLE II
SUMMARY OF THE NUMERICAL RESULTS IN THE SHORT-TERM REAL POWER LOAD FORECASTING PROBLEM

	RMSE	MAPE	R^2	KST	Time
Tech 1	0.815	1.554	0.234	11.586	12
Tech 2	0.795	1.608	0.265	12.221	13
Tech 3	0.815	1.554	0.234	11.586	13
Tech 4	0.795	1.608	0.265	12.221	15
Tech 5	0.815	1.550	0.235	11.348	20
Tech 6	0.815	1.554	0.234	11.572	15
Tech 7	0.816	1.612	0.228	12.476	159
Tech 8	0.936	1.841	0.095	12.076	321
Tech 9	0.935	2.253	0.073	6.134	15157
Tech 10	0.933	2.298	0.029	6.338	15286
Tech 11	0.928	2.127	0.006	9.524	618
Tech 12	0.923	2.114	0.042	8.724	650

TABLE III
SUMMARY OF THE NUMERICAL RESULTS IN THE SHORT-TERM REACTIVE POWER LOAD FORECASTING PROBLEM

	RMSE	MAPE	R^2	KST
Tech 1	0.290	0.895	0.005	0
Tech 2	0.309	1.067	-0.119	0
Tech 3	0.290	0.895	0.005	0
Tech 4	0.309	1.067	-0.119	0
Tech 5	0.290	0.875	0.007	0
Tech 6	0.290	0.895	0.005	0
Tech 7	0.375	1.211	-0.629	0
Tech 8	0.362	1.124	-0.387	0
Tech 9	0.342	0.707	-0.181	0
Tech 10	0.343	0.710	-0.208	0
Tech 11	0.320	0.824	-0.110	0
Tech 12	0.310	0.818	-0.201	0

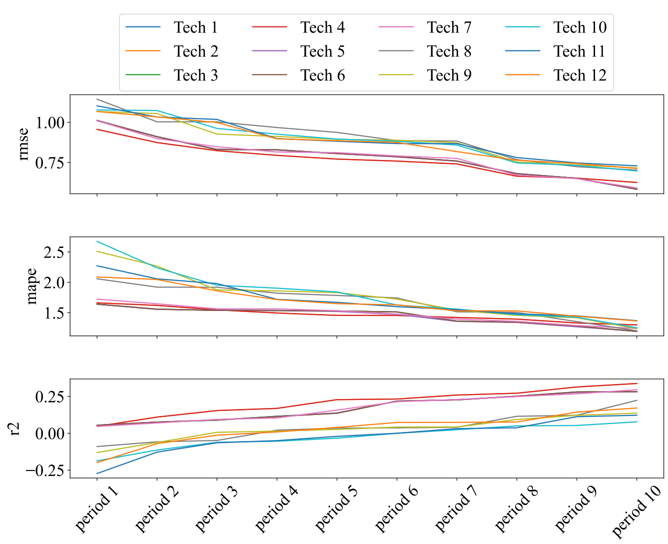

Fig. 3. Medium of RMSE, MAPE and R^2 of each technique for each period of the time-series cross validation in the short-term real power load forecasting problem.

the prediction of customers' reactive power usage proves to be more intricate compared to their real power consumption. This observation is particularly evident through the outcomes of statistical tests, as demonstrated in Tables II and III. Despite our dataset indicating customers' reactive power utilisation remaining below one kVAR, as elaborated in [26], it is crucial to acknowledge that the proliferation of LEDs and battery-powered devices equipped with switch-mode power supplies has led to a progressive rise in customers' reactive power consumption. In light of this trend, the imperative to develop specialised algorithms tailored for accurate customer reactive power predictions becomes evident, especially when such data is intended for integration into load flow and OPF studies.

C. Results in the medium-term forecasting problem

This section investigates the performance of different approaches in our medium-term forecasting problem for the non-participant customers, including our proposed method. Figure 4 displays the time series data for the forecasted values and the telemetry data for a specific non-participant customer. The first (top) figure illustrates the prediction using Tech 1 to Tech 4. All these techniques fail to provide accurate predictions, as confirmed by the average results reported in Table IV. The KST values for these techniques are zero, and their R^2 values are significantly far from 1.

The second figure compares techniques that differ solely in their loss functions. Similar to the results in the previous section, we observe that the choice of the loss function has a negligible effect on the techniques' performance. The third figure compares the techniques based on their multi-step ahead algorithm. We notice that the Direct approaches offer more accurate predictions compared to the Iterated and Rectified approaches. Finally, the bottom figure in Fig.4 compares our proposed approach with other approaches that use different Regression models. Our approach outperforms other methods for this specific customer, and overall, it marginally surpasses all other approaches, as shown in Table IV.

V. CONCLUSION

This paper studied the accuracy of the computational burden of several time-series multi-step ahead forecasting approaches on the individual electricity forecasting day-ahead and month-ahead problems. The approaches methods include various inputs, including historical, weather, and calendar data, as well as distinct regression models, such as linear, random forest, and XGBoost regression. Additionally, the study explored different multi-step algorithms, including Iterated, Direct, and Rectified algorithms, while employing distinct loss functions such as LSE, Lasso, and Ridge for parameter optimisation.

For the day-ahead load forecasting problem, the analysis revealed that all studied techniques yielded predictions with comparable errors. Among the different regression models, linear regression showcased the shortest computational time, reducing computation time by 98%. Furthermore, the Iterated algorithm exhibited superior efficiency over other multi-step algorithms, reducing computation time by 92%. Interestingly,

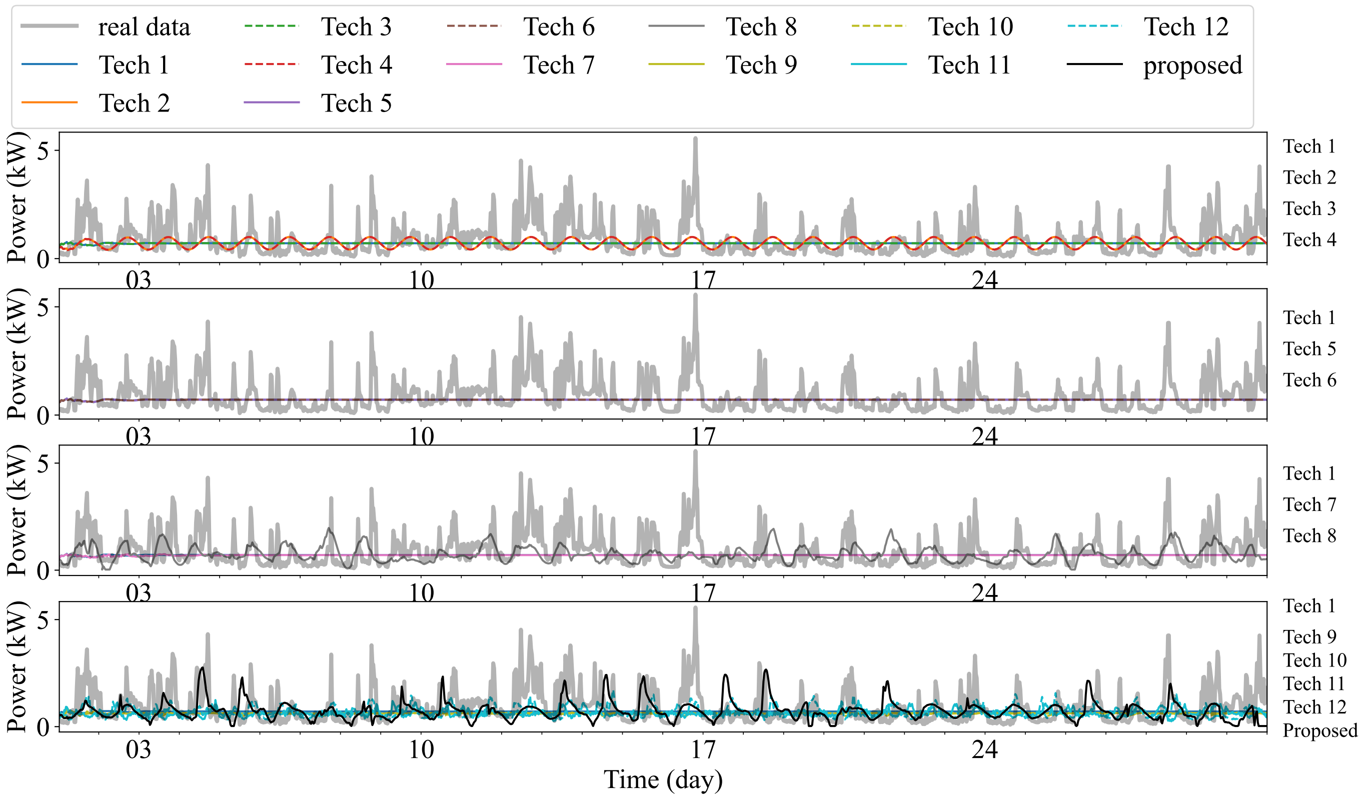

Fig. 4. Time-series values of real power connection point exchange of an specific customer from 2023/04/01 to 2023/04/31, and the prediction values using different forecasting techniques.

TABLE IV
SUMMARY OF THE NUMERICAL RESULTS IN THE MEDIUM-TERM REAL POWER LOAD FORECASTING PROBLEM

	RMSE	MAPE	R^2	KST	Time
Tech 1	1.003	2.274	0.002	0	68
Tech 2	0.964	2.423	0.078	0	73
Tech 3	1.003	2.274	0.002	0	70
Tech 4	0.964	2.423	0.078	0	75
Tech 5	1.003	2.277	0.001	0	77
Tech 6	1.003	2.274	0.002	0	68
Tech 7	0.969	2.110	0.042	0	169
Tech 8	1.044	2.373	-0.088	0	4041
Tech 9	1.152	5.485	-0.063	0	18818
Tech 10	1.084	5.473	-0.032	0	18449
Tech 11	1.331	7.259	-0.460	0	903
Tech 12	1.151	6.505	-0.144	0	910
Proposed	0.871	1.872	0.1.03	0.989	86

including calendar data contributed to a maximum of 2% and 30% improvement in accuracy of real and reactive power prediction, respectively, while the impact of weather data on prediction accuracy was negligible. Moreover, integrating both calendar and weather data during the training process did not significantly affect the processing time.

However, when tackling the more challenging month-ahead forecasting problem, none of the investigated techniques provided accurate results. All approaches failed the KST test in all experiments, resulting in a coefficient of determination (R^2) close to zero or even negative values in some approaches. To address the limitations of existing approaches, our proposed method for month-ahead forecasting demonstrated notable improvements, outperforming alternative techniques.

ACKNOWLEDGEMENT

This work was done in the framework of the project Converge. This project received funding from the Australian Renewable Energy Agency (ARENA) as part of ARENA's Advancing Renewables Program.

REFERENCES

[1] Converge project. *[Online]. Available: https://arena.gov.au/projects/project-converge-act-distributed-energy-resources-demonstration-pilot/.*

[2] Sarah Hadri, Mehdi Najib, Mohamed Bakhouya, Youssef Fakhri, and Mohamed El Arroussi. Performance evaluation of forecasting strategies for electricity consumption in buildings. *Energies*, 14(18):5831, 2021.

[3] Corentin Kuster, Yacine Rezgui, and Monjur Mourshed. Electrical load forecasting models: A critical systematic review. *Sustainable cities and society*, 35:257–270, 2017.

[4] Mahmoud A Hammad, Borut Jereb, Bojan Rosi, Dejan Dragan, et al. Methods and models for electric load forecasting: a comprehensive review. *Logist. Sustain. Transp*, 11(1):51–76, 2020.

[5] Tao Hong, Pu Wang, and Laura White. Weather station selection for electric load forecasting. *International Journal of Forecasting*, 31(2):286–295, 2015.

[6] Peter Lusis, Kaveh Rajab Khalilpour, Lachlan Andrew, and Ariel Liebman. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Applied energy*, 205:654–669, 2017.
[7] Sanford Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.
[8] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
[10] Souhaib Ben Taieb, Rob J Hyndman, et al. *Recursive and direct multi-step forecasting: the best of both worlds*, volume 19. Department of Econometrics and Business Statistics, Monash Univ., 2012.
[11] Hanjiang Dong, Jizhong Zhu, Shenglin Li, Wanli Wu, Haohao Zhu, and Junwei Fan. Short-term residential household reactive power forecasting considering active power demand via deep transformer sequence-to-sequence networks. *Applied Energy*, 329:120281, 2023.
[12] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.
[13] Kshitij Sharma, Yogesh K Dwivedi, and Bhimaraya Metri. Incorporating causality in energy consumption forecasting using deep neural networks. *Annals of Operations Research*, pages 1–36, 2022.
[14] Bureau of meteorology. *[Online]. Available: http://www.bom.gov.au*.
[15] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.
[16] Mark R Segal. Machine learning benchmarks and random forest regression. 2004.
[17] XGBoost tutorial. *[Online]. Available: https://xgboost.readthedocs.io/en/ stable/tutorials/index.html*.
[18] Cees Diks and Valentyn Panchenko. A new statistic and practical guidelines for nonparametric granger causality testing. *Journal of Economic Dynamics and Control*, 30(9-10):1647–1669, 2006.
[19] Rob J Hyndman. Forecasting, causality and feedback. *International Journal of Forecasting*, 39(2):558–560, 2023.
[20] Christoph Bergmeir and José M Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.
[21] Daniel J Ozer. Correlation and the coefficient of determination. *Psychological bulletin*, 97(2):307, 1985.
[22] Frank J Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.
[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
[24] Joaquin Amat Rodrigo and Javier Escobar Ortiz. skforecast, 7 2023.
[25] S. Mahdi Noori R. A. and D. Gordon. Converge load forecasting package. *https://pypi.org/project/converge-load-forecasting/*, Nov. 2022.
[26] Jackson Hannagan, Rhys Woszczeiko, Thomas Langstaff, Weixiang Shen, and John Rodwell. The impact of household appliances and devices: Consider their reactive power and power factors. *Sustainability*, 15(1):158, 2022.