A Data-Driven Short-Term PV Generation and Load Forecasting Approach for Microgrid Applications

Rohit Trivedi, Member, IEEE, Sandipan Patra, Member, IEEE, and Shafi Khadem, Senior Member, IEEE

Abstract — The data-driven (DD) is a systematic approach to improve the data and model by deriving/adding features to address the problem identified during the iterative loop of forecasting model development. This article proposes a DD framework for forecasting short-term PV generation and load demand. A framework of three stages with a unique contribution in each stage, such as generalizing data preprocessing steps (stage-1), multivariate feature generation and selection (stage-2), and model hyperparameter tuning (stage-3) for further improvement in forecasting is proposed. It focuses on data as well as forecasting models. The whole process is analyzed using the time-series measured data collected from a real-life demonstration project in Ireland. Data preprocessing is generalized for both generation and demand forecasting under the same framework. The relevant features are selected with the help of the proposed random forest sequential forward feature selection algorithm. Hyperparameters are tuned through tree-structured Parzen estimator algorithm for further improvement. In addition, the performance of the classical autoregressive integrated moving average model is compared with the machine learning-based gate recurrent unit, long short term memory, recurrent neural network, and convolutional neural network models. Results show that the data-driven forecasting model framework systematically improves the model performance. The seasonal variation has also a high impact on the model performances.

Index Terms — Data-driven (DD) forecast, hyperparameter tuning, load demand, machine learning, microgrids, PV generation.

I. INTRODUCTION

Forecasting is an essential element in the context of microgrids control and operation. Over the recent years, consumers have shown interest in installing distributed energy resources (DERs), especially photovoltaics (PV) and energy storage and this has populated several scattered DERs at low voltage networks. A challenge has emerged for utilities/microgrid controllers to allow high penetration of DERs and at the same time maintain the grid stability and integrity of the entire electrical system. Accurate forecasting, therefore, can help make the utilities aware of the uncertainty related to production from DERs and the varying load demand so that utilities can anticipate the risk and take the necessary action to circumvent the problem.

The individual residence with/without PV generation inside a microgrid must be taken into account and exactly match the individual profiles of users in a microgrid, forecasting of generation and load demand at the user level becomes prominent. To be precise, the forecasting horizon has a very crucial role in a decision-driven microgrid system. Depending upon the applications, the prediction range can differ significantly. For example, the long-term forecast can help in planning microgrids, the mid-term forecast is useful for resource security and allocation and the short-term forecast (STF) can be instrumental for microgrid control, real-time scheduling of generation and energy market-related operations. The focus of this article is on the short term forecast (from minutes to day-ahead) to augment the control and energy market segment in microgrids.

A. Literature Review

In general, STF models can be classified into two major categories: classical methods and artificial intelligence (AI) based methods. Classical methods include statistical time series [1] and regression-based models [2]. The simple mathematical equations involved in these methods makes them efficient for linear time-series problems but to deal with complex nonlinear forecasting, these methods are inefficient [3]. Hence, AI-based methods that consist of machine learning (ML) enabled models with learning capability have been applied in nonlinear forecasting problems [4]. The application of AI in microgrid control environments has also been reviewed in [5]. Some of the most popular ML models are artificial neural networks (ANNs) [6], random forest (RF) [7], support vector machines [3], and support vector regression (SVR) [8]. Most of the above-mentioned forecasting approaches are model-centric in which emphasis is given to improving forecasting models rather than data. However, data-driven (DD) approaches that focus more on the data might aid in improving the prediction accuracy. To fix this model-centric trend, in recent time, DD approaches have been considered for forecasting PV generation [9], [10] and load demand [3], [8], [11], [12]. Laouafi et al. [8] have focused on improving forecasts by combining different forecasting models.

1) Data-Driven PV Forecasting: Rather than stressing multi-combination forecasting models, Kang et al. [13] have proposed a DD approach that compares the target time series to a set of reference time series and predicts future value based on

Manuscript received 19 April 2022; accepted 26 May 2022. Date of publication 3 June 2022; date of current version 23 September 2022. This work was supported by the Department of the Environment, Climate and Communications. (Corresponding author: Rohit Trivedi.)

The authors are with the International Energy Research Centre, Tyndall National Institute, T12 R5CP Cork, Ireland (e-mail: rohit.trivedi@tyndall.ie; sandipan.patra@tyndall.ie; shafi.khadem@tyndall.ie).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JESTIE.2022.3179961.

Digital Object Identifier 10.1109/JESTIE.2022.3179961.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
an average of their future paths. González Ordiano et al. [14] have suggested a DD forecasting model by only considering the data processing that includes missing values, outlier detection, and imputation for missing values. It does not consider any derived feature to validate any possibility of improvement in the forecasting model. Shi and Eftekharnejad [9] have proposed a DD approach in which post and preprocessing the data, features like resolution and weather parameters are derived and added to the dataset. A day-ahead PV forecasting with two features for lagged and newest observations process is proposed in [15]. It improves the accuracy of prediction by implementing a SVR model. To target the relevant lagged features, a least absolute shrinkage and selection operator based algorithm is discussed in [16], which derives the important lagged values of historical PV time-series data and use those features to predict the day-ahead PV generation using a feed-forward neural network (NN) method. Similarly, Rafati et al. [17] derive the important lagged values by RRelieff (RRF) algorithm and utilize them to predict 15 min PV generation through a single hidden multilayer perceptron (MLP) model.

2) Data-Driven Demand Forecasting: A dynamic mode decomposition method for abstracting features in the load demand time-series data is discussed in [17] which selects particular days of similar load patterns from past values and creates a new input time series to predict future values. A similar-day feature is derived using meteorological data to forecast load demand in [18]. The blend of weather data with load demand is tested against the classical autoregressive integrated moving average (ARIMA) model and found that this hybrid method achieves better accuracy on both ordinary and unordinary (i.e., huge meteorological change in a day) days. Dong et al. [12] utilize the temporal time-series features like the year, month, season, temperature to forecast the load demand with an ensemble ANN-based algorithm which performs better than a backpropagation NN model.

Based on daily and hourly load change (load tracking), new features like load difference of two consecutive hours and a change between the load of the specific hour and a day before were derived in [19] and the important features were ranked according to RRF algorithm. The forecasting model was developed using MLP NN to forecast hour-ahead demand. A two-stage load demand prediction methodology is proposed in [20]. The first stage selects the input features of lagged variables and the second stage forecasts the load. The feature selection method might enhance the performance but the hyperparameters like the number of layers and neurons are randomly selected for the MLP model.

3) Research Gap on DD Approach: Inappropriately, preprocessing the data for forecasting is sometimes misunderstood as a DD approach [21]. In fact, processing the data is an essential step for every forecasting problem and should not be confused with. There is a lack of general data preprocessing methods that can be applicable for both PV generation and load demand forecasting.

In the context of feature selection techniques, mutual information (MI), RRF, correlation-based selection (CFS), and autocorrelation (AC) methods are becoming popular for load demand forecasting problems. While AC and CFS detect only linear correlation, MI and RRF can easily observe nonlinear correlations [20]. Liu et al. [11] have developed a wrapper based greedy forward feature selection technique for selecting features from a pool of features which can be implemented using linear regression (LR) as an estimator. But this algorithm may not work well for nonlinear variables relation and can be computationally expensive due to the involvement of multiple stages of determining feature correlation. Hence, in this article, we propose RF as an estimator instead of LR in the sequential forward feature selection (RF-SFS) algorithm as it can deal with non-linearity and multiple features.

Due to the popularity of the classical ARIMA model and ML models like convolutional neural network (CNN), recurrent neural network (RNN), long short term memory (LSTM) and gate recurrent unit (GRU), [22] these models have further been considered in this article to assess the individual model performance following the DD approach.

ARIMA model works well on stationary data. For forecasting time series, CNN can be a 1-D model containing a hidden convolutional layer working over a 1-D sequence. The following pooling layer consists of capturing the pattern in the data and a dense layer interprets the patterns extracted from its preceding layer. RNN, unlike traditional feed-forward NN, stores the output of hidden layers which is then fed back to the input layer to predict the next value. LSTM has an advanced memory feature over RNN consisting of additional input and output gates with many cells in LSTM that solves the exploding or vanishing gradient problems present in deep NN. GRU is a variant of RNN similar to LSTM. Unlike LSTM, there is no output gate in GRU and consists of an update and a reset gate that facilitate the effective control and flow of information in the cells of the NN. More details of these models can be found in [22].

To mitigate the manual hit and trial approach of hyperparameter tuning of ML models, some algorithms are quite popular such as Grid search along with manual search [23], Random search [23], and Gaussian process (GP) based Bayesian optimization [24]. Of these, grid search suffers from the curse of dimensionality and random search struggles with its nonadaptiveness to different experiments [23]. Bayesian method is one of the advanced sequential search tuning algorithms but originally not applicable for a conditional search space, categorical, or integer values [25]. The ML models with tree-based structures can solve this problem. Hence, tree-structured Parzen estimator (TPE) [26] is used in this article to tune the hyperparameters.

B. Key Contributions

This article discusses a DD forecasting framework that contains data preprocessing steps generalized (bringing both PV and load data pre-processing and forecasting under the same framework), followed by a feature selection through the RF-SFS algorithm and completes with hyperparameter tuning by the TPE method. The foremost contributions are as follows.

A systematic DD forecasting framework: The proposed framework systematically presents the steps required in time-series forecasting that include data collection, processing, feature generation, selection, and hyperparameter tuning.
A generalized data preprocessing for both PV and load forecast:
The presented preprocessing steps eliminate the need for
physical parameters like inverter rating, voltage, short-circuit
current, performance ratio, etc., and are valid for load de-
demand data as well. It emphasizes DD modeling rather than
physics-driven modeling.

A unique RF-SFS algorithm for feature selection: The proposed
SFS-RF combination is a novel contribution to PV generation
and demand forecasting problems. According to the enhanced
findings reported in this article, the performance of this com-
bination cannot be overlooked.

Implementation of TPE for hyperparameter tuning: TPE is one
of the advanced algorithms compared to grid search and
random search techniques. The TPE technique has not been
previously developed for Load and PV forecasting appli-
cations. The findings show that TPE greatly increases the
accuracy by picking the most appropriate hyperparameters
for this situation.

II. PROPOSED METHODOLOGY

The proposed data-driven forecasting model framework
shown in Fig. 1 comprises three stages as a workflow package.
1) Stage-1: Data collection, preprocessing, and data selection
(generalized data preprocessing).
2) Stage-2: Feature generation and selection (implement RF-
SFS algorithm, combine the relevant attributes).
3) Stage-3: Best-performing models identification (Imple-
ment HT).

Forecasting models and accuracy evaluation are performed in
each stage to obtain the improvement. The overall methodology
is explained step by step in the following sections with a case
study example, where the time-series measured data has been
collected from a real-life pilot project, StoreNet [27].

A. Stage -1: Data Collection, Preprocessing, and Data
Selection (DPS)

Data is processed initially to improve the quality of data.
Identifying format, timestamps interval, consistency, dealing
with missing data, imputation and resampling with final statistics
analysis are key things to consider during the analysis. The data
preprocessing steps as suggested in [28] have been improved
and generalized for both PV generation and demand profile as
shown in Fig. 2. The steps are elaborated as follows.

Step-1: Initial Data Identification
Post data collection, recording interval, feature labels and the
timestamp format should be observed. For example, in this
work, the data was recorded at 1 min interval and the times-
tamps were in two formats, unix and dd.mm.yyyy hh:mm:ss,
which is then converted to dd/mm/yyyy HH:MM:SS format
to maintain the consistency. Feature labels were also updated
as there were unexpected blank spaces.

Step-2: Consistency Examination
The collected data is reviewed for gaps, repetitiveness, and
duplication. Several repeated little gaps are identified but no
duplicate entries.

Step-3: Invalid and missing data identification
The values for PV and load demand are missing only in the
case when the timestamp is not recorded. No invalid data is
found during the analysis process. The values are in line with
the capacity of PV installation and consumption profiles.

Step-4: Data imputation and resampling
The missing timestamps are first resampled with the 1 min in-
terval, which is the original resolution of the collected dataset.
The variation from previous to next values is not extremely
high and the mean value with linear interpolation is valid.
Hence, the missing values are imputed using the mean and
linearly interpolated to make the observations uniform and
consistent. However, other missing data treatment techniques
depending upon the missing rate and timestamps can be found
in [29].

Step-5: Data verification, aggregation, and statistics
Finally, the dataset is cross-checked for any invalid/corrupted
data which is unexpected in the given time series, for example,
sudden zero value or extremely high value that is beyond
the installed PV capacity and peak load demand. Missing
values are then aggregated based on different granularity, time
depending on the final use of the data. The dataset is resampled
to half-hourly resolution for this research work. A distplot
function of the Seaborn library [30] is used to analyze the
distribution of all observations of multiple variables. Mean,
median, and max value distributions are examined using box-plot based statistics.

The processed data is now ready to forecast but to follow the benchmark models, some exogenous parameters are required. So the weather parameters like solar irradiation, dry bulb, and grass temperature are downloaded from the Irish Meteorological site [31] and fed to the forecasting models and then the accuracy (error) is calculated.

B. Stage-2: Feature Generation and Selection (FGS)

This stage processes multivariate forecasting by implementing the reference benchmark models developed in [22]. With the available PV generation and load demand data, several features are derived and explained as follows:

Step-1: Feature generation

First, the entire one-year dataset is split into seasons of the year (Spring, Summer, Autumn, Winter). The features considered here for STF are based on the literature [32], [33] and are categorized into three classes. Their development mechanism is described in the following sections.

Encoded cyclic features: A polar coordinate system is used to encode calendar effect features. A periodical cycle in a polar coordinate system is regarded as a unit circle. The periodical features in this system are encoded as a unit circle and their coordinate system is considered as a unit circle. The periodical features are encoded as a unit circle and their coordinates specify the time. The process of encoding is described by the following:

\[
x = \left[ \begin{array}{c} \cos(g) \\ \sin(g) \end{array} \right], \quad g = \frac{2\pi t}{p} \tag{1}
\]

where \( t \) is the time and \( p \) represents the cycle length. The cycle length \( p = 24 \) for the time of the day, \( p = 7 \) for the day of the week, and \( p = 12 \) for the month of the year. The nomenclature of encoded variables is presented in Table I.

Historical time-series features: For a given time \( t \), load in 24 h is denoted by \( L_{t+48} \) and \( L_t \) is the current load. Similarly, for PV, it is \( P_{t+48} \) and \( P_t \). In terms of historical time series, which is denoted by \( L_{t-i} \) and \( P_{t-i} \), \( i = 0, 1, \ldots, K \), where \( K \) is the number of past values and can be regarded as candidate features to forecast value at \( L_{t+48} \) and \( P_{t+48} \). However, considering all the previous 48 lagged points will increase the input dimension and computational complexity of the model. Thereby, to determine the optimal number of lagged values as candidate features, AC and partial AC (PAC) methods are used.

Weather features: Weather is one of the important factors when forecasting PV generation and load demand. Similar to [34] and [35], three parameters that are frequently used as exogeneous variables are considered here. It is also compelling to assess their impact on forecasting \( L_{t+48} \) and \( P_{t+48} \). These are hence, considered as candidate features and since the weather parameters at \( t + 48 \) are unknown, the parameters are recorded at time \( t \) following the model as presented in [11]. All considered candidate features with their description are presented in Table I.

Step-2: Feature selection

Sequential selection algorithms belong to the family of greedy search algorithms and are mainly used to reduce the \( d \)-dimensional feature set to a \( k \)-dimensional subset, where \( k \leq d \). The ultimate goal of using feature selection algorithms is to determine the optimum number of features relevant to the problem which further helps in improving the computational efficiency or decreasing the generalization error of the forecasting model by eliminating irrelevant features [36]. Mathematically, for a given set \( F = \{ f_1, f_2, \ldots, f_d \} \) of total features \( d \). Selecting the most optimal features means deriving a new subset that contains all the essential parameters i.e., \( X = \{ x_1, x_2, \ldots, x_k \} \) where \( k \leq d \).

Sequential selection algorithm can work in two different ways, forward (SFS) and backward (SBS) to eliminate the relevant and redundant features. SFS begins with a single feature and makes the data model using the given structure. Then it sequentially selects the features that provide higher performance (depending upon the performance metric defined in the structure) and this process is repeated until an optimum number of features is selected. In BFS, the algorithm starts with all features and sequentially eliminates the features that provide the least reduction accuracy. It repeats the process until a set of optimum features is not obtained. BFS has a limitation of feature reevaluation, i.e., once the feature is removed, it is not possible to reevaluate its usefulness and cannot be included in the next iteration [37]. On the other hand, SFS does not have this problem and hence, we consider here for this work. The pseudocode for SFS is extracted from [38] and discussed below.

A RF algorithm is used to make it work as an estimator for selecting suitable features. For the RF regression number of estimators considered is 10 with a coefficient of determination \((R^2)\) as the performance metric in SFS. Thus, we combine the RF estimator and SFS algorithm and propose the RF-SFS algorithm here to deal with nonlinearity and multiple features.

As shown in Fig. 3, the whole training set is divided into six partitions and from the second partition onward, one partition is considered as a test set each time and all previous partitions comprise the training set. The performance considered is a cross-validation score and calculated by averaging the performance of
Algorithm: Pseudo code for SFS algorithm.

**Data:** $F$, $k$

**Result:** $X_k$

**Initialisation:** $x_1 = \arg\max_{f_i \in F} J(f_i)$; $X = \{x_1\}$, $k = 1$;

**While** $k < d$ **do**

$x_{k+1} = \arg\max_{f_i \in F \setminus X_k} S^+(f_i, X_k)$;

$X_{k+1} = X_k \cup x_{k+1}$; $k = k + 1$;

---

**Table II**

| Target variable   | Seasons | Selected feature subsets | Highest CV score |
|-------------------|---------|--------------------------|------------------|
| PV gen            | Spring  | $W^{S}, T^h, T^m, T^{PCS}, P_1$ | 0.9810          |
|                   | Summer  | $W^{S}, T^h, T^m, T^{PCS}, T^{DPCS}, P_1, M^{PCS}, T^m$ | 0.9824          |
|                   | Autumn  | $W^{S}, Y^M, T^{DPCS}, P_1$ | 0.9776          |
|                   | Winter  | $W^{D}, W^{S}, T^h, T^m, M^{PCS}, P_1, D^{PCS}$ | 0.9625          |
| Load dem          | Spring  | $W^{D}, T^h, T^{PCS}, D^{PCS}, M^{PCS}, L_1$ | 0.8816          |
|                   | Summer  | $W^{D}, W^{S}, Y^M, T^h, T^{PCS}, M^{PCS}, D^{PCS}, T^m, L_1$ | 0.8812          |
|                   | Autumn  | $W^{D}, W^{S}, Y^M, T^{PCS}, M^{PCS}, D^{PCS}, T^m, L_1, M^{SUB}$ | 0.8820          |
|                   | Winter  | $W^{D}, W^{S}, Y^M, T^h, T^{PCS}, D^{PCS}, T^m, L_1, M^{SUB}$ | 0.8858          |

---

Fig. 3. Illustration of five-fold cross validation.

Fig. 4. Performance of various feature sets for PV forecast.

Fig. 5. Performance of various feature sets for load demand.

The SFS adds features from the 14 derived features and forms a feature subset in a greedy manner. In each stage, the RF estimator selects the best feature to add based on the cross-validation (CV) score obtained by the estimator. After performing the feature selection experiment separately for PV generation and load demand, as shown in Figs. 4 and 5 for each season, the different combinations of optimal features are selected based on the highest CV score obtained over multiple iterations. For instance, PV generation as a target variable in the spring season, a set of five features has achieved the highest score of 0.9810. Similarly, for load demand, a set of six features has achieved the highest CV score of 0.8816. The CV score with selected feature names is tabulated in Table II.

C. Stage-3: Best-Performing Model Identification (BMI)

Recent advances in configurations of existing forecasting techniques have brought a shift in dealing with traditional classification and regression tasks. Hyperparameter optimization has been a manual process of selecting the optimal number of parameters within an ML model for so long. This hit and trial approach is time-consuming and cumbersome. To eliminate this, we adopt a TPE algorithm, which is a greedy sequential method and computationally more efficient than conventional tuning methods, based on the expected improvement criterion [26].

For a given configuration space $X$, the TPE models $p(x|y)$ by transforming the graph-structured generative process (selecting the number of layers first and then choosing the parameters for each), replacing the configuration distribution with non-parametric densities. By utilizing the different observations $\{x^{(1)}, x^{(2)} \ldots x^{(k)}\}$ in the nonparametric densities, the replacement indicates that the learning algorithm can generate many different densities for the given configuration space $X$. Making use of two such densities, the TPE algorithm defines $p(x|y)$ as [26] given by the following:

$$p(x|y) = \left\{ \begin{array}{ll} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{array} \right\}$$

(2)

where $l(x)$ represents the density generated by using observations $\{x^{(1)}\}$ in a way that corresponding loss $f(x^{(1)})$ is less than $y^*$. The density $g(x)$ is generated by the remaining observations. Unlike the GP algorithm, the TPE selects the $y^*$ as some quantile $\gamma$ of observed values $y$ such that the $p(y < y^*) = \gamma$ that does not require any model for $p(y)$. We utilize the open-source hyperOpt [39] software for hyper-parameter tuning. The ML models are implemented and simulated using the open-source library TensorFlow [40]. The HyperOpt has four essential elements to optimize the hyperparameters, namely, search space, a loss...
function, optimization algorithm (TPE here) and a database of score and configuration history as shown in Fig. 6. Initially, the user defines the search space with a given set of parameters to be tuned, mathematically it is determined by a continuous and convex function. A loss function is to be evaluated for each configuration setting using the number of observations. The optimization algorithm is based on sequential model-based global optimization that finds the best solution for convex optimization problem and the scores for each configuration through the iterative process are stored in the database in a set of tuples (score, configuration). The search space with the highest score is extracted and the new space is redefined for further sampling. This process is repeated until either the overall highest score is achieved or early stopping is enabled.

### III. MODEL EVALUATION

Since the dataset is categorized according to the seasons, the training and testing are split in a way that the last day of each season is considered for testing and the remaining samples are considered for training the models out of which 10% are reserved for validation to achieve a day-ahead (short-term) forecast as described in Table III.

The models are evaluated on normalized RMSE (nRMSE) values [41]. For PV forecast, the formula is given by the following:

$$
nRMSE (\%) = 100 \sqrt{\frac{\sum_{i=1}^{N} \left( \frac{\hat{P}_i - P_i}{P_{\text{installed}}} \right)^2}{P}} \quad (3)$$

Similarly, for load demand forecast, it is given by the following:

$$
nRMSE (\%) = 100 \sqrt{\frac{\sum_{i=1}^{N} \left( \frac{\hat{P}_i - P_i}{P_{\text{installed}}} \right)^2}{P}} \quad (4)$$

where $\hat{P}_i$ and $P_i$ represent predicted and measured power at time $i$, $P_{\text{installed}}$ is the total installed PV capacity, $N$ is the total number of samples, and $\bar{P}$ is the mean of all observations.

There are two more metrics included to better understand the performance of the DD approach in different models, namely, normalized mean bias error (nMBE) and the forecast skill score. The nMBE metric indicates if there is a significant tendency to systematically underforecast or overforecast, which is termed as bias [42]. The positive and negative values imply over and underforecast, respectively. It is indeed useful for network operators as the understanding will allow them to better allocate the resources in the dispatch process and compensate for the errors and is given by the following [43]:

$$
nMBE (\%) = 100 \left( 1 - \frac{1}{N} \sum_{i=1}^{N} \left( \frac{\hat{P}_i - P_i}{P_{\text{max}}} \right) \right) \quad (5)$$

where $P_{\text{max}}$ is the maximum power among all observations.

**TABLE III**

| Seasons | Training & validation (10%) | Testing dataset |
|---------|----------------------------|-----------------|
| Spring  | 01/03/2019 to 30/05/2019    | 31/05/2019      |
| Summer  | 01/06/2019 to 30/08/2019    | 31/08/2019      |
| Autumn  | 01/09/2019 to 29/11/2019    | 30/11/2019      |
| Winter  | 01/12/2019 to 28/02/2020    | 29/02/2020      |

**IV. RESULTS AND DISCUSSION**

Statistical and ML-based models are considered and evaluated here for each stage. The entire three-stage framework is simulated for each season of the year and the errors are calculated for every stage.

**nRMSE**: Fig. 7 shows the performance and evaluation through nRMSE. ARIMA being the classical model, there are no hyperparameters like other advanced ML algorithms. Hence, the results for ARIMA are considered only until stage 2. Following the forecasting framework, the error is sequentially reduced in each stage and for both forecastings. Significant improvement appears for PV generation forecasting [see Fig. 7(a)] and mainly during the summer months. This is also followed by spring. This confirms the initial validation of the effectiveness of the proposed framework. For example, GRU shows the best performance for both PV and load forecasting in spring [see Fig. 7(b)]. The error, which was 13.9% in stage 1, is decreased to 9% in stage 3 for PV forecasting. For load forecasting, the error is reduced by 6.4% in stage 3 compared to stage 1. For summer, an improvement of 14.16% is observed in the LSTM model giving only 5.34% error in the final stage and performing the best for PV forecasting. For load prediction, RNN (12.6%), LSTM (12.5%), and GRU (12%) are quite close to each other.
Similarly, in autumn, RNN performs best for PV and LSTM for load forecasting. For winter, the LSTM model outperforms all the other models giving only 0.56% of error for predicting PV output whereas RNN surpasses other models by giving the lowest 9% error to forecast day-ahead load demand.

From the training time perspective, as shown in Fig. 8, it can be observed that GRU and LSTM have the least training time among all the models. In contrast, RNN has taken the most time to train the model for both PV and load forecasting.

Forecast plots are shown in Fig. 9 for the summer season. Inferred from the above analysis, LSTM is the best-performing PV forecast model and hence three consecutive stages are shown in Fig. 9(a). Similarly, GRU is the best-performing model for load forecasting in the summer season. The performance of these three stages is shown in Fig. 9(b). Thus, it further validates the effectiveness of the proposed forecasting framework.

Fig. 10(a) shows the nMBE in the case of PV forecast. Like nRMSE, the bias is reducing here as progress from stage 1 to 3. More specifically, the autumn season has the lowest bias tendency. However, it is interesting to note that the bias in the summer (overforecast) and winter (underforecast) seasons is opposite. One of the reasons could be that the summer (winter) months have very high (low) PV production and thus prediction models track these dynamics and bias follow the trends according to the historical and independent features’ time-series values. For all cases, the stage 3 process can significantly reduce the bias error. The bias for the best model (LSTM) in summer and winter is 0.96% and –11.1%, respectively. For the spring (GRU) and autumn (RNN), the bias is 2.46% and 0.65%, respectively.

In the case of load forecast [shown in Fig. 10(b)], the bias is also gradually reducing from stages 1 to 3. Similar trends appear here, as is found for the PV forecasting. Demand is high (low) in winter (summer) months, thus the forecast models
follow the trend and result in an over (under) forecast. The bias in the winter for the best model (RNN) is 5.78%, which is an overforecast. Models in spring (GRU), summer (GRU), and autumn (RNN) seasons have the bias of 2.33%, -2.3%, and 0.67%, respectively. The bias is mostly independent of forecasting models and depends upon the seasons or specifically the training data given to the models.

**Skill Score:** Considering ARIMA as the least-performing baseline model, the SS of other models has been presented for different seasons, as shown in Fig 11. It depicts the accuracy improvement in other models compared to ARIMA. Similar to nRMSE and nMBE, successful improvement appears in SS for both PV and load forecasting methods. These significant improvements again validate the importance of the proposed forecasting framework.

For PV forecast, all models have improved [see Fig. 11(b)] accuracy however, surprisingly, in load forecast CNN and LSTM have a negative improvement over the baseline model. It is noteworthy that the neural network model performance (CNN and LSTM) during load forecast has degraded as shown in Fig. 11(b). It also indicates the sensitivity of the models. It might happen due to the complexities within the internal architecture like model type, size, optimization process, and data complexity which is a further point of investigation in future research.

**V. Conclusion**

In this article, a DD forecasting framework has been proposed that does not only focus on data but also on models. The approach also brings the generation and demand forecastings under the same framework. Data preprocessing steps have been generalized for stage-1. The error is evaluated for each model and individual application. Stage-2 focuses on carefully selecting features through the proposed RF-SFS algorithm. The RF works as an estimator and it sequentially simulates to find specific combinations of features and selects a set with the best performance. Stage-3 tunes the hyperparameters in the models by the proposed TPE algorithm mainly focusing on optimizing the model performance. The numbers of neurons, layers, window length, and batch size are selected for each forecasting task and benchmark models are updated with new parameters. Forecasting on new models is performed and errors are recorded. In concluding the results, the following are the key takeaways from this article:

1) The proposed data-driven forecasting model framework systematically improves the model performance. Baseline model evaluation, feature development, selection, and hyperparameter tuning gradually reduce the error.

2) For each season and application, RF-SFS algorithm selects a different set of independent variables and thus, the appropriate selection of features improves in forecasting models.

3) The model performance also varies and can be biased depending upon the seasons. One universal forecasting model cannot be fixed for the entire year and seasonal/variable models should be developed for better and more accurate forecasting.

4) Newly installed PV in the geographical region in which this forecasting research has been conducted can get benefit from the results and responsible control entities can better match the supply and demand.

5) Enabling graphics processing unit support during model training and forecasting can significantly reduce the training time. However, it still depends on the training data, model complexity and different hyperparameters involved in it.

6) Future research intends to focus on making the AI explainable by utilizing models’ interpretability or explainable modeling.

**ACKNOWLEDGMENT**

This work is a part of MiFIC project and the authors in IERC thankfully acknowledge the support from the Department of the Environment, Climate and Communications.

**REFERENCES**

[1] N. Elamin and M. Fukushige, “Modeling and forecasting hourly electricity demand by SARIMAX with interactions,” *Energy*, vol. 165, pp. 257–268, 2018.

[2] T. Fang and R. Lahdelma, “Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system,” *Appl. Energy*, vol. 179, pp. 544–552, 2016.

[3] A. Yang, W. Li, and X. Yang, “Short-term electricity load forecasting based on feature selection and least squares support vector machines,” *Knowl.-Based Syst.*, vol. 163, pp. 159–173, 2019.

[4] M. Q. Raza and A. Khosravi, “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,” *Renewable Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, 2015.

[5] R. Trivedi and S. Khadem, “Implementation of artificial intelligence techniques in microgrid control environment: Current progress and future scopes,” *Energy AI*, vol. 8, 2022, Art. no. 100147.

[6] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, “Short term electricity load forecasting using a hybrid model,” *Energy*, vol. 158, pp. 774–781, 2018.

[7] C. Li, Y. Tao, W. Ao, S. Yang, and Y. Bai, “Improving forecasting accuracy of daily enterprise electricity consumption using a random forest based on ensemble empirical mode decomposition,” *Energy*, vol. 165, pp. 1220–1227, 2018.

[8] A. Laouafi, M. Mordjaoui, S. Haddad, T. E. Boukelia, and A. Ganouche, “Online electricity demand forecasting based on an effective forecast combination methodology,” *Electron. Power Syst. Res.*, vol. 148, pp. 35–47, 2017.

[9] G. Shi and S. Eftekharnejad, “Data-driven solar generation forecast considering temporal characteristics of data,” in *Proc. IEEE Texas Power Energy Conf.*, 2020, pp. 1–6.
et al. (Member, IEEE) received the M.Sc. degree in smart grids and energy systems without weather data," in *Comput. Sci., Res. Develop.*, vol. 32, no. 1, pp. 237–246, 2017.

M. Rana, A. Rahman, and J. Jin, “A data-driven approach for forecasting state level aggregated solar photovoltaic power production,” in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

A. Rafati, M. Joorabian, E. Mashhour, and H. R. Shaker, “High dimensional very short-term solar power forecasting based on a data-driven heuristic method,” *Energy*, vol. 219, 2021, Art. no. 119647.

N. Mohan, K. P. Soman, and S. Sachin Kumar, “A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model,” *Appl. Energy*, vol. 232, pp. 229–244, 2018.

X. Cao, S. Dong, Z. Wu, and Y. Jing, “A data-driven hybrid optimization model for short-term residential load forecasting,” in *Proc. IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable Autonomic Secure Comput. Pervasive Intell. Comput.*, 2015, pp. 283–287.

A. Rafati, M. Joorabian, and E. Mashhour, “An efficient hour-ahead electrical load forecasting method based on innovative features,” *Energy*, vol. 201, 2020, Art. no. 117511.

I. Koprisinska, M. Rana, and V. G. Agelidis, “Correlation and instance based feature selection for electricity load forecasting,” in *Knowl.-Based Syst.*, vol. 82, pp. 29–40, 2015.

E. Sharma, “Energy forecasting based on predictive data mining techniques in smart energy grids,” *Energy Inform.*, vol. 1, no. 1, pp. 367–373, 2018.

R. Trivedi and S. Khadem, “Peak demand management and schedule optimisation for energy storage through the machine learning approaches,” in *Proc. 19th Int. Conf. Smart Technol.*, 2021, pp. 157–163.

J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

J. Snoeck, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2951–2959.

T. Yu and H. Zhu, “Hyper-parameter optimization: A review of algorithms and applications,” 2020, pp. 1–56. [Online]. Available: http://arxiv.org/abs/2003.05689

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyperparameter optimization,” in *Proc. Neural Inf. Process. Syst.*, 2011, pp. 1–9.

M. S. K. Khadem, M. Bahlool, M. Daoud, A. Saif, and A. Majumdar, “StoreNet,” in *Proc. ETIP SNET Regional Workshop*, 2019, pp. 1–10. [Online]. Available: https://www.researchgate.net/publication/356588094_StoreNet

A. Livera et al., “Data processing and quality verification for improved photovoltaic performance and reliability analytics,” *Prog. Photovolt. Res. Appl.*, vol. 29, no. 2, pp. 143–158, 2021.

H. Kang, “The prevention and handling of the missing data,” *Korean J. Analyt. Chem.*, vol. 5, pp. 402–406, May 2013.

“seaborn, distplot — Seaborn 0.11.2 documentation.” Accessed: Nov. 25, 2021. [Online]. Available: https://seaborn.pydata.org/generated/seaborn.distplot.html

“Historical data - Met Eireann - The Irish meteorological service.” Accessed: Nov. 17, 2021. [Online]. Available: https://www.met.ie/climate/available-data/historical-data

X. Zhang, J. Wang, and K. Zhang, “Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by cuckoo search algorithm,” *Electron. Power Syst. Res.*, vol. 146, pp. 270–285, 2017.

A. Bakshshai and R. Stull, “Electric load forecasting for Western Canada: A comparison of two non-linear methods,” *Atmosphere, Ocean*, vol. 50, no. 3, pp. 352–363, 2012.