IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 December 2023, accepted 11 January 2024, date of publication 15 January 2024, date of current version 24 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3354703

RESEARCH ARTICLE

Intelligent Solar Forecasts: Modern Machine Learning Models and TinyML Role for Improved Solar Energy Yield Predictions

ALI M. HAYAJNEH[1], (Member, IEEE), FERAS ALASALI[1], (Member, IEEE), ABDELAZIZ SALAMA[2], (Member, IEEE), AND WILLIAM HOLDERBAUM[3], (Member, IEEE)

[1]Department of Electrical Engineering, Faculty of Engineering, The Hashemite University, Zarqa 13133, Jordan
[2]Department of Electrical and Electronic Engineering, University of Leeds, LS2 9JT Leeds, U.K.
[3]School of Science, Engineering and Environment, University of Salford, M5 4WT Salford, U.K.

Corresponding authors: William Holderbaum (w.holderbaum@salford.ac.uk) and Ali M. Hayajneh (alihayajneh@hu.edu.jo)

This work was supported in part by the Scientific Research and Innovation Support Fund, Ministry of Higher Education and Scientific Research, The Hashemite Kingdom of Jordan, under Grant ENE/1/02/2022; in part by the University of Salford, U.K.; and in part by the Royal Academy of Engineering, U.K., under Grant DIA-2021-18.

ABSTRACT The advancement of sustainable energy sources necessitates the development of robust forecasting tools for efficient energy management. A prominent player in this domain, solar power, heavily relies on accurate energy yield predictions to optimize production, minimize costs, and maintain grid stability. This paper explores an innovative application of tiny machine learning to provide real-time, low-cost forecasting of solar energy yield on resource-constrained edge internet of things devices, such as micro-controllers, for improved residential and industrial energy management. To further contribute to the domain, we conduct a comprehensive evaluation of four prominent machine learning models, namely unidirectional long short-term memory, bidirectional gated recurrent unit, bidirectional long short-term memory, and simple bidirectional recurrent neural network, for predicting solar farm energy yield. Our analysis delves into the impacts of tuning the machine learning model hyperparameters on the performance of these models, offering insights to improve prediction accuracy and stability. Additionally, we elaborate on the challenges and opportunities presented by the implementation of machine learning on low-cost energy management control systems, highlighting the benefits of reduced operational expenses and enhanced grid stability. The results derived from this study offer significant implications for energy management strategies at both household and industrial scales, contributing to a more sustainable future powered by accurate and efficient solar energy forecasting.

INDEX TERMS Solar power forecasting, time series forecasting, Internet of Things, deep neural networks.

I. INTRODUCTION

A. MOTIVATION

Solar photovoltaic (PV) integration into global power systems has increased significantly over the past decade. The majority of these PV facilities are deployed in low-voltage (LV) and medium-voltage (MV) networks, presenting distinct challenges for integrating renewable energy sources (RES) as distributed generation (DG). In distribution networks (DN), these difficulties include reverse power fluxes, voltage violations, and grid stability [1], [2]. Luis et al. [1] conducted a study to assess the impact of forecasting on centralised voltage control for solar generation in distribution systems. Their findings emphasised the significance of accurate forecast data for achieving optimal control settings and highlighted the need for improvements in forecasting tools for predicting solar generation in distribution systems. Therefore, Zang et al. proposed a day-ahead PV power forecasting approach based on deep learning (DL). Their study demonstrated the accuracy and reliability of the approach through the

The associate editor coordinating the review of this manuscript and approving it for publication was Salvatore Favuzza.

utilisation of deep convolutional neural network (CNN) [2]. On the other hand, high levels of PV penetration can cause power and voltage fluctuations due to cloud shadows, as well as an increase in energy losses when reversing power fluxes are significant. In addition, the unpredictable nature of PV power generation, which is influenced by abrupt weather changes, ultimately presents a significant challenge to integrated power infrastructures [3], [4]. Prema et al. conducted an extensive review of forecast models in the context of integrating solar and wind power into main power grids. Their study emphasises the importance of accurate short-term predictive models for grid operation and planning, providing critical insights into the duration of data used and the performance indices of these models [3]. In the process of determining the scheduling of power generation plans and short-term dispatches, it is an essential tool for mitigating the effects of weather-induced power fluctuations. Dimd et al. [4] presented a comprehensive review of machine learning (ML)-based PV output power forecasting models in the context of cold regions such as the Nordic countries and Canada. Their study focused on the impact of meteorological parameters and the effect of snow on prediction model performance, providing important insights and suggestions for model selection of ML. As a solution to this problem, accurate PV power forecasting emerges as a crucial and valuable technology. artificial intelligence (AI) and modern ML techniques have the potential to tackle PV power's limitations. By utilising the capabilities of modern ML methods, PV usage can be increased, thereby improving PV power forecasting performance and stochastic low voltage data at the distribution network application [5]. Therefore, this paper aims to employ and assist various modern and new ML models, including multi-layer deep neural networks (DNNs), bidirectional gated recurrent unit (BiGRU), bidirectional long short-term memory (BiLSTM), simple bidirectional recurrent neural network (BiRNN), and unidirectional LSTM in the context of solar power yield time series forecasting. By evaluating the performance of these models under different hyperparameter settings and exploring their strengths and weaknesses, we seek to identify the most suitable forecasting model for solar energy yield prediction. This research contributes to the development of effective forecasting tools for solar energy yield, ultimately promoting more efficient resource planning and energy management in the rapidly expanding solar power sector.

Furthermore, the rising interest in edge computing and implementing ML models on resource-limited devices, such as micro-controllers unit (MCU) and internet of things (IoT) devices, has encouraged the investigation of Tiny ML (TinyML) for solar power forecasting [6]. TinyML enables the deployment of AI and ML capabilities on small, low-power devices, allowing them to execute complex tasks without the need for remote servers or high-performance hardware. This presents numerous benefits, including reduced latency, enhanced privacy, and energy efficiency. In this scenario, unidirectional LSTM is particularly appealing for deployment on edge devices due to its lower complexity and compatibility with TensorFlow Lite Micro [7]. This renders it an attractive option for practical applications requiring on-device processing and local decision-making capabilities, particularly in remote locations with limited connectivity or where immediate responses are crucial [8]. By evaluating the performance of unidirectional LSTM models for solar power forecasting and edge inference using TinyML, we aim to identify a robust, efficient, and computationally viable solution that can be deployed on resource-constrained devices. This approach not only contributes to more precise and effective solar power forecasting but also encourages the adoption of edge AI solutions in the renewable energy sector, fostering innovative applications and more efficient energy management [9].

B. LITERATURE REVIEW

In the management of new energy generation and consumption, the accuracy of forecasting models is crucial. The implementation of smart grid technology [10] facilitates accurate capacity forecasting, which is essential for such strategies and distribution power networks [5]. The effectiveness of load management techniques, coupled with accurate forecasting, enables DN operators to navigate the challenges presented by ultimately promoting sustainable energy consumption practices. In the context of smart grid applications such as solar generation in low and medium voltage levels [1] and outputs of photovoltaic panel power [2]. The literature provides multiple articles on PV power forecasting that employ techniques such as time series forecasting and neural networks.

Forecasting methods for PV power can be broadly classified into three categories: statistical, physical model, and intelligent methods. Statistical methods or time series forecasting techniques, such as Auto-regressive moving average (ARMA), for forecasting PV power rely on historical data (PV power data), making them suitable for short-term forecasting. The ARMA model is one of the most common statistical models used for load demand and PV power forecasting and does not require meteorological forecasts. Nevertheless, statistical models rely on inputs with stable auto-correlations, such as daily and seasonal periodicities in PV power series, and therefore may exhibit insufficient prediction accuracy on cloudy or rainy days. This issue becomes especially severe when forecasting one day in advance. Recent research, however, has investigated methods to enhance the prediction accuracy of statistical models by integrating multiple forecasting models using ensemble and ML techniques [2], [10]. The meteorological data (weather prediction) are used as input to predict PV power outputs [2]. To establish the correlation between input data (weather data) and the future PV power output, there are two common approaches: analytical equations and soft-computing models under various ML algorithms. However, analytical equations can be difficult to compute due to their complexity, resulting

in high computational costs, especially when the edge deployment is in mind [2]. As such, soft-computing models are more commonly used for PV power forecasting, as they offer a more efficient and effective solution. Ongoing research in this field is aimed at further improving these models and increasing their accuracy. Rodríguez et al. developed an approach for forecasting intra-hour solar photovoltaic energy by combining wavelet-based time-frequency analysis with deep-learning neural networks. This study [10] demonstrated improved accuracy compared to a persistence benchmark model, achieving a validation deviation below 4% in the majority of sample days.

In recent years, AI and ML algorithms, such as support vector machine (SVM), and artificial neural network (ANN), have become increasingly popular for forecasting PV output power due to their ability to effectively capture the highly nonlinear relationship between environmental input parameters and PV power [11]. These models typically use both power and weather parameter measurements as inputs to the forecasting model, although limited high-performing models have been developed that only require measurements of PV power for short-term forecast horizons [4]. ANN, SVM, adaptive neuro-fuzzy (NF) networks and evolutionary optimisation have been used to forecast PV power output by using time series data of PV power and weather forecasting [12], [13]. Semero et al. [12] developed a hybrid approach for accurate forecasting of electricity production in micro-grids with solar photovoltaic installations. Their method combines genetic algorithm, particle swarm optimisation, and adaptive NF inference systems to address the intermittent and uncertain nature of solar power. In [13], a forecast model employing LSTM and neural network (NN) to predict the PV power generation over one step advance with data resolutions up to one hour. Authors in [14] have developed a forecast model based on LSTM and NN for predicting the hourly PV power output over 24 hours. To achieve high performance from SVM or ANN models, it is necessary to determine suitable model parameters via optimisation algorithms and cross-validation. However, when dealing with large-scale samples, the training efficacy of SVM models tends to decrease, resulting in low performance [11]. The study [15] examines a long-term PV power forecast constructed by using feed ANN. However, the proposed approach disregards the effect of past trends on prospective PV output. In short-term forecasting, [16] utilised the extreme learning machine (ELM) to predict PV power for forecast horizons extending from 15 to 60 minutes. The researchers utilised particle swarm optimisation to optimise the ELM. However, ELM with particle swarm optimisation has a complex structure with many model parameters and high computational costs. Therefore, a Gated Recurrent Unit network (GRU) was proposed as an alternative to the commonly used LSTM architecture in RNNs in a study by Cho et al. [17]. GRU, as contrasted with LSTM, has only two gates, resulting in fewer training parameters while preserving high prediction accuracy. This architecture additionally addresses the overfitting issue observed in LSTM models. Although GRU and other DL algorithms have significantly improved prediction accuracy over ML techniques, they may not completely exploit the local characteristics and concealed information present in historical PV data.

DL, with its autonomous feature extraction capabilities, has transformed ML and AI fields, as evidenced in works like [4], [12], [13], [15], [16], [17], [18], and [19]. Models like RNNs, LSTMs, and GRUs have excelled in handling time series data. Our study builds on this foundation, optimizing these models for TinyML applications in edge computing environments [20], [21]. This approach, unique in its focus on low-cost, resource-constrained edge IoT devices, addresses a gap in the existing literature, where the full potential of TinyML for solar energy forecasting remains largely unexplored.

C. CONTRIBUTIONS AND ORGANISATION

This paper contributes to the field of solar power output forecasting by introducing the innovative use of TinyML for real-time, low-cost solar energy yield forecasting on edge IoT devices. This approach is particularly suitable for DNs and residential settings due to its cost-effectiveness and efficiency. Further, we provide a comprehensive comparative study of various ML techniques to predict and improve the PV power output forecasting. Our main contributions are as follows:

- Introduction of the novel use of TinyML on edge IoT devices for real-time, low-cost solar energy yield forecasting. This approach significantly contributes to the field by enabling efficient resource planning and energy management at both household and industrial scales.
- Comprehensive evaluation of four prominent ML models, namely unidirectional LSTM, BiGRU, BiLSTM, and simple BiRNN, for predicting solar farm energy yield.
- Systematic comparison of the performance of these ML models in the context of solar energy yield forecasting, providing valuable insights into their effectiveness and areas for potential improvement.
- Detailed analysis of the impact of hyperparameter selection on the performance of these ML models, providing practical insights for future research and applications.
- Thorough investigation into the computational requirements and resource constraints of implementing the proposed TinyML-based solution on edge IoT devices, emphasising its suitability for real-time, cost-effective forecasting applications.
- Highlighting the inherent challenges and trade-offs of implementing complex DNN architectures for solar power forecasting in the context of edge computing, and showing how TinyML principles can address these challenges, making our approach distinct from

traditional forecasting methods that don't account for computational limitations.

The rest of this paper is organised as follows: In Section II, we describe the methodology employed, including data preprocessing, model development, and evaluation metrics used in our study. Section III delves into the implementation of TinyML for low-cost household power yield prediction, discussing the benefits and challenges associated with deploying ML models on resource-constrained devices. The results and discussion of the performance of various models are presented in Section IV, highlighting the strengths and weaknesses of each approach. Section V concludes the paper, summarising our findings and providing insights into future research directions in this field. Finally, appendix lists all the abbreviations and acronyms that we used in this article.

II. METHODOLOGY

A. THE DATASET

The dataset used in our research was collected from on-site renewable energy facilities located in China, comprising power generation and weather-related information from six wind farms and eight solar stations. The dataset was introduced in [22] and collected at 15-minute intervals over a two-year period from 2019 to 2020. Our work focuses primarily on the dataset related to five on-site solar farms. We use ML models to develop time series forecasting for solar power generation across these five locations, with the trained model tested and validated against actual data from these sites.

The primarily dataset was divided into 70% as a training dataset 10% for validation and 20% for testing. The division of the dataset into training, validation, and testing sets is a common practice in ML. It aims to optimise model performance and generalisability. The training set, being the largest portion, enables the model to learn and capture patterns. The validation set helps in tuning the model's hyperparameters to avoid overfitting and improve its generalisability. The test set, kept separate, provides a final evaluation of the model's performance, simulating real-world scenarios. Our 70/20/10 split is widely accepted in the field as it provides a balance between maximising learning and evaluating the model's generalisability capability. This split can vary depending on factors such as the dataset's size and the specific application.

The dataset includes seven features, with the first six representing weather-related features (i.e., TSI for total solar irradiance, DNI for direct normal irradiance, GHI for global horizontal irradiance, AT for air temperature, ATM for atmospheric pressure, and RH for relative humidity). The last feature represents the solar farm power yield and is the output of the latent relations between the first six features and the power yield [22].

To preprocess the data for our analysis, we first average the dataset based on specific time intervals instead of the original 15-minute intervals. In this approach, every N_{avg} samples, corresponding to 15-minute intervals, are averaged together to obtain the desired interval for forecasting. Next, we prepare the samples for each forecasting task to look back for LB look-back steps and predict the LA look-ahead steps. We test the model configurations for two different grouping setups for the number of features $N = 3$ and $N = 7$.

The choice of N_{avg} and the look-back and look-ahead steps is crucial as it influences the granularity of our forecast and the ability of our models to capture temporal dependencies. For instance, larger N_{avg} would yield coarser time intervals, potentially smoothing out significant fluctuations. On the other hand, smaller N_{avg} could capture more detailed fluctuations but might be more prone to noise and less generalisable.

For the look-back steps, LB, a higher value allows the model to take into account a wider window of past data, helping it identify longer-term patterns or trends. However, this might also increase the model's complexity and computational requirements. As for the look-ahead steps, LA, a higher value would mean forecasting further into the future, which can be more challenging and uncertain.

In this study, the specific values of N_{avg}, LB, and LA were selected through experimentation, considering the trade-off between model performance, computational efficiency, and the specific requirements of solar power forecasting. Further, the feature selection ($N = 3$ or $N = 7$) was determined based on the relevance and contribution of each feature to the power yield, aiming to retain the most informative features while reducing the model's complexity and potential for overfitting.

B. FEATURES SELECTION

Feature selection constitutes a critical phase in devising precise ML models. To streamline this procedure, we concentrated on the correlation between input features and the power yield of each solar farm. We employed the normalised covariance matrix to pinpoint the variables exhibiting a strong correlation with the power yield, allowing us to choose a subset of the most predictive features. This strategy optimises our ML forecasting models by diminishing the number of features employed and centring on the most crucial variables, which, in turn, will enhance the model performance. Furthermore, by recognising the variables strongly correlated with power yield, we will acquire lucid insights into the physical and environmental factors that impact the power generation process [23].

Figure 1 illustrates the normalised covariance matrix between the input features of the dataset. For our analysis, we selected the top 5 cleanest datasets (i.e., The datasets exhibiting high data quality, characterized by accuracy, consistency, minimal noise or errors, and a lack of significant gaps or missing values.) from the available options (sites) listed in [22]. The input features related to solar irradiance have the most significant impact on the output power yield, as seen in the highly correlated top left 3×3 dark sub-heatmap and the correlations with the output power yield in the last heatmap column.

Figure 1 further showcases the correlation between the three input features and the output power yield. In this investigation, we concentrate on two kinds of forecasting models. Firstly, we will employ the first three input features to execute forecasting and examine the impact of altering the hyperparameters of the ML model on the comprehensive performance of the forecasting. Secondly, we will utilise the complete set of features including the time series power data in the datasets to carry out forecasting and undertake a comparative analysis between the two chosen input features grouping models.

C. DATA SCALING

We apply min-max data scaling to minimise errors and aid the learning process. This technique normalises the dataset input features to a specific range based on the minimum and maximum values of the features.

The scaling can be expressed as follows:

$$\hat{y}_i = \frac{y_i - \min(y_i)}{\max(y_i) - \min(y_i)}, \tag{1}$$

where $\hat{y}_i$ is the normalised vector that contains all the data for a certain input feature i, and y_i is the original vector that contains all input feature vectors from the specific dataset.

To reverse the data scaling after the inference (time series forecasting), we use the following equation:

$$y_i = y'_i(\max(y_i) - \min(y_i)) + \min(y_i), \tag{2}$$

where y'_i is the estimated value after the inference of the future value in the time series forecasting application. The min-max scaling technique is efficient and useful in optimising ML processes such as gradient descent algorithms, leading to faster convergence in the learning process. Moreover, the scaling ensures that different models are compared fairly in terms of their performance, especially when measuring the root mean square error (RMSE).

In the subsequent subsections, we explore the BiRNN, BiLSTM, and BiGRU models. Additionally, we discuss the unidirectional LSTM, deferring its discussion to section III to place it within the context of TinyML for edge inference. Our objective is to determine the most fitting model for solar energy yield prediction by analyzing their performance across diverse configurations and considering multiple input features. Moreover, we will evaluate the pros and cons of each model, offering a thorough understanding of their suitability for solar power forecasting applications.

D. BiRNN MODEL

The BiRNN model consists of two simple RNN layers, one processing the input sequence in the forward direction ($\overrightarrow{H}_t$) and the other in the reverse direction ($\overleftarrow{H}_t$). For a given time step t, the RNN equations for both directions are as follows [24]:

$$\overrightarrow{H}_t = \tanh(W_{\overrightarrow{h}} \cdot [\overrightarrow{h}_{t-1}, x_t] + b_{\overrightarrow{h}}), \tag{3}$$

$$\overleftarrow{H}_t = \tanh(W_{\overleftarrow{h}} \cdot [\overleftarrow{h}_{t+1}, x_t] + b_{\overleftarrow{h}}) \tag{4}$$

where x_t is the input vector at time step t, $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are the hidden state vectors in the forward and reverse directions, respectively, and tanh denotes the hyperbolic tangent activation function. The weight matrices $W_{\overrightarrow{h}}$ and $W_{\overleftarrow{h}}$ and the bias vectors $b_{\overrightarrow{h}}$ and $b_{\overleftarrow{h}}$ are the learnable parameters of each RNN layer. In the BiRNN model, the hidden states of both the forward and reverse RNN layers are combined at each time step, providing a better context for predictions [25].

While the bidirectional structure of the BiRNN model allows it to capture both past and future information effectively, it suffers from certain limitations that can affect its performance in solar energy yield forecasting. The most notable drawback of the BiRNN model is its susceptibility to the vanishing gradient problem, which can hinder the model's ability to learn long-range dependencies in the input data. In the following sections, we will discuss the BiGRU and BiLSTM models and demonstrate how their unique features make them ideal candidates for solar energy yield forecasting.

E. BiLSTM MODEL

In this paper, we examine the potential of the LSTM models for forecasting the energy yield of solar farms, considering multiple input features. The LSTM model is especially proficient in managing time series data, owing to its inherent ability to seize long-range dependencies, which is vital for precise forecasting. We utilise seven input features to train our LSTM model, where the last feature signifies the target variable to be predicted. By adjusting the number of look-back steps, we aim to discover the optimal LSTM configuration for effective solar energy yield forecasting.

The Bidirectional LSTM (BiLSTM) model consists of two LSTM layers, one processing the input sequence in the forward direction and the other in the reverse direction. Each LSTM layer is composed of memory cells and three gating units: the input gate (i_t), the forget gate (f_t), and the output gate (o_t). For a given time step t, the LSTM equations for both directions are as follows [24]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{8}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t \odot \tanh(C_t) \tag{10}$$

where i_t is input gate, f_t is the forget gate, and o_t is the output gate, x_t is the input vector at time step t, h_t is the hidden state vector, and C_t is the cell state vector. The sigmoid activation function is represented by σ, while element-wise multiplication is denoted by $\odot$. The weight matrices W_f, W_i, W_C, and W_o correspond to the forget, input, cell state, and output gates, respectively, and the bias vectors b_f, b_i, b_C, and b_o are the learnable parameters of each LSTM layer. The gates i_t, f_t, and o_t control the flow of information through

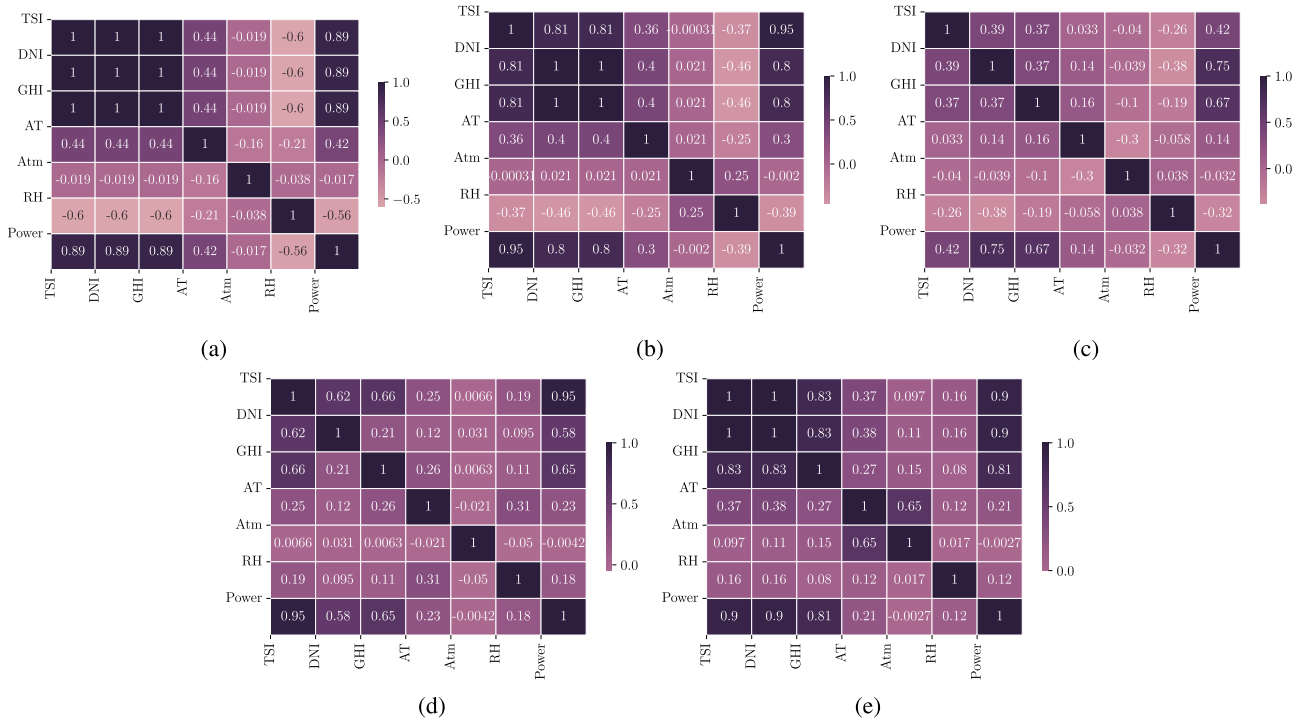


FIGURE 1. (a) Solar station site 1 (Nominal capacity = 30MW), (b) Solar station site 2 (Nominal capacity = 130MW), (c) Solar station site 3 (Nominal capacity = 30MW), (d) Solar station site 4 (Nominal capacity = 50MW), (e) Solar station site 5 (Nominal capacity = 130MW).

the memory cells, while $\tilde{C}_t$ is a temporary cell state used for updating the cell state C_t. The hidden state h_t is updated using the output gate and the cell state [26].

In the BiLSTM model, the hidden states of both the forward and reverse LSTM layers are combined at each time step, providing a better context for predictions. In our study, we train the BiLSTM model with seven input features, adjusting the number of look-back steps to optimise the model's performance for solar energy yield forecasting.

In our study, we train the BiLSTM model with $N = 3$ and $N = 7$ input features, adjusting the number of look-back (LB) steps to optimise the model's performance for solar energy yield forecasting.

F. BiGRU MODEL

We also explore the applicability of the BiGRU model for predicting the energy yield of solar farms based on multiple input features. The BiGRU model, which consists of two GRU layers processing the input sequence in both forward and reverse directions, addresses the vanishing gradient problem commonly encountered in RNNs. This characteristic makes it a promising candidate for handling time series data. We utilise seven input features to train our BiGRU model, where the last feature serves as the target variable to be predicted. By adjusting the number of look-back steps, we aim to determine the optimal BiGRU configuration for effective solar energy yield forecasting.

The BiGRU model consists of two GRU layers, one processing the input sequence in the forward direction and the other in the reverse direction. Each GRU layer has two gating units: the update gate (z_t) and the reset gate (r_t). For a given time step t, the GRU equations for both directions are as follows [24]:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{11}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{12}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \tag{13}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{14}$$

where x_t represents the input vector at time step t, and h_t denotes the hidden state vector. The sigmoid activation function is denoted by σ. Element-wise multiplication is represented by the symbol $\odot$. The weight matrices W_z, W_r, and W_h correspond to the update, reset, and candidate hidden state gates, respectively. The bias vectors b_z, b_r, and b_h are the learnable parameters associated with these gates in the GRU layer. z_t and r_t are the update and reset gates, which control the flow of information through the hidden state. $\tilde{h}_t$ is the candidate hidden state, a temporary value that helps in updating the hidden state [27]. The hidden state is updated using a combination of the previous hidden state and the candidate hidden state, weighted by the update gate. In the BiGRU model, the hidden states of both the forward and reverse GRU layers are combined at each time step, providing a better context for predictions. In our study, we train the BiGRU model with seven input

features, adjusting the number of look-back steps to optimise the model's performance for solar energy yield forecasting.

G. MODELS COMPARISON FOR SOLAR POWER FORECASTING

In this section, we discuss the key differences between the four ML models—BiGRU, BiLSTM, BiRNN, and Unidirectional LSTM—in the context of power yield time series forecasting.[1] The primary distinctions between these models lie in their architecture, ability to handle time dependencies, complexity, and susceptibility to the vanishing gradient problem as illustrated in Table 1.

BiGRU, an RNN variation, addresses short and long-term time dependencies. This model, with medium complexity and lower vulnerability to the vanishing gradient issue, holds potential for time series forecasting tasks, including power yield predictions by capturing temporal dynamics.

BiLSTM, another RNN variant, manages short and long-term time dependencies effectively. Its high complexity and significantly diminished susceptibility to vanishing gradient problems make it a preferred choice for time series forecasting tasks, such as power yield predictions. BiLSTM efficiently captures complex patterns, enhancing forecasting performance.

BiRNN, capable of handling short-term time dependencies, struggles with long-term dependencies due to low-to-medium complexity and severe susceptibility to the vanishing gradient problem. Though applicable to time series forecasting tasks, its performance may lag behind BiGRU and BiLSTM, particularly when long-term dependencies are crucial.

Unidirectional LSTM, adept at handling short and long-term time dependencies, lacks BiLSTM's bidirectional information flow. Consequently, it might not grasp all relevant patterns, potentially resulting in less accurate forecasts compared to BiLSTM. However, its lower complexity could render it suitable for edge devices with limited computational resources.

In summary, for power yield time series forecasting, models like BiGRU, BiLSTM, and Unidirectional LSTM are likely to deliver better performance due to their ability to capture both short- and long-term dependencies in the data. While BiRNNs might face limitations in handling long-term dependencies. The choice between BiGRU, BiLSTM, and Unidirectional LSTM will depend on the specific requirements of the forecasting task and the trade-offs between complexity and performance.

H. PERFORMANCE METRICS

In order to measure the performance of time series forecasting, we employ two types of measures. To measure the accuracy of the prediction, we use the RMSE as a measure of error between the predicted values and the actual values [11].

$$\mathrm{RMSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}\left(y_i - \hat{y}_i\right)^2}, \tag{15}$$

where $\boldsymbol{y}, \hat{\boldsymbol{y}}$ are the actual and the predicted vector of readings, respectively. The vector $\boldsymbol{y} = \{y_1, y_2, \ldots, y_{N_s}\}$ represents the times series values at time i where $i = 1, 2, \ldots, N_s$. N_s is the number of samples in the time series. In subsequent discussions, we will use $e_i = y_i - \hat{y}_i$ to represent the error in prediction at a specific time step and e to denote the average error in the forecast.

In order to capture how well our model can predict future values, we use the determinant coefficient R^2. The determinant coefficient is the proportion of the variance in the dependent variable that is predictable from the independent variable. We can write the determinant coefficient as follows

$$R^2(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1 - \frac{\sum_{i=1}^{N_s}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_s}(y_i - \bar{y})^2}, \tag{16}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{N_s} y_i$ and $\sum_{i=1}^{N_s}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N_s}\epsilon_i^2$ is the residual sum of squares.

III. TinyML FOR LOW-COST HOUSEHOLD POWER YIELD PREDICTION

TinyML, an emerging field at the intersection of ML and embedded systems, offers effective tools for executing ML models on resource-limited devices like MCUs [21]. In the context of our study, TinyML has been utilised to establish an optimised, real-time solar power yield prediction mechanism for low-cost household solar farms. The power of ML is brought to the edge, directly at the source of data, thereby enhancing prediction efficiency, reducing latency, and ensuring data privacy.

The TinyML deployment process commences with data collection from the hardware where the inference engine is required. This data is utilised to train the ML model, which is then implemented directly on the MCU for inference in subsequent iterations. A notable challenge in unlocking the full potential of ML for IoT systems is the fragmentation of the MCU market, and the absence of a unified standard for TinyML implementation. To address these, we utilize TensorFlow Lite Micro, which has become synonymous with TinyML. Most practical ML model implementations now rely on the TFLite libraries [21].

TFLite Micro provides the necessary features for enabling ML on IoT devices. It assumes that the model, input data, and output arrays are already in memory and performs computations on these arrays directly. The TFLite Micro framework uses an interpreter to load the data structure that defines the ML model. This design choice allows for the model to be easily updated without recompiling the firmware on the IoT device. Particularly, the use of an LSTM model in conjunction with TinyML presents an innovative approach in the realm of low-cost, real-time solar power yield forecasting.

[1]The full discussion of Unidirectional LSTM is deferred to section III to place it within the context of TinyML inference on edge devices, as we conduct a more in-depth analysis to study its applicability for deployment on resource-constrained, low-cost devices.

TABLE 1. Comparison of LSTM, BiGRU, BiLSTM, and BiRNN.

Model	Architecture	Directionality	Time Dependency	Complexity [24]	Vanishing Gradient [24]
LSTM	Long short-term memory	Unidirectional	Short- and long-term	Medium	Significantly reduced
BiRNN	Bidirectional RNN	Bidirectional	Short-term	Low-medium	Severe
BiGRU	Bidirectional GRU	Bidirectional	Short- and long-term	Medium	Reduced
BiLSTM	Bidirectional LSTM	Bidirectional	Short- and long-term	High	Significantly reduced

This approach not only serves as a potential solution to overcome the challenges posed by resource-constrained settings but also sets a novel precedent in employing ML techniques for such applications. By deploying the LSTM model on edge devices via TinyML, we are essentially bringing the power of ML directly to the source of data, thereby enhancing the efficiency of prediction tasks, reducing latency, and ensuring data privacy.

The advancements in TinyML research have demonstrated its effectiveness across various domains, including human activity recognition and classification and time series forecasting in diverse scenarios [21]. In this context, we believe that TinyML will be instrumental in shaping the future of smart grids and solar power time series forecasting. Consequently, we have developed an evaluation framework for power yield forecasting in solar farms. However, the scarcity of household-specific datasets poses a limitation for implementing our ML models. To overcome this challenge, this study aims to establish pre-trained models using the aforementioned ML architectures, which can later be adapted for local household solar farms through transfer learning (TL) techniques [28], [29]. The application of TL will ultimately simplify the adoption of such forecasting methods, streamlining the development operations (DevOps) process for seamless ML operations (MLOps) [21].

In addition to TL, which enhances the training experience for forecasting tasks, federated learning (FL) also plays a vital role, especially when applying forecasting to privacy-related data. FL provides an advantage in scenarios where data privacy is concerned and data cannot be moved to a centralised location due to regulatory restrictions or security concerns [30]. FL, much like TL, can offer a streamlined process for the development and implementation of forecasting models. It allows for the leveraging of distributed data sources, enabling a more secure view of the data and thus enhancing the accuracy and robustness of our forecasting models. Moreover, when combined with TL, FL can potentially accelerate the learning process as the models can benefit from previously learned knowledge and apply it across different yet related tasks. This powerful combination can bring forth a new era in ML-powered forecasting, where models are not only effective and efficient but also respectful of data privacy and security requirements.

A. UNIDIRECTIONAL LSTM MODEL FOR EDGE INFERENCE

The Unidirectional LSTM model is designed to process the input sequence in the forward direction ($\overrightarrow{H}t$) only. This model is used to compare its performance with the previously mentioned bidirectional models, as well as to facilitate inference on edge devices using TinyML and TensorFlow Lite Micro since TensorFlow Lite Micro only supports unidirectional LSTM layers [31]. For a given time step t, the LSTM equations are as follows [24]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{17}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{18}$$
$$\tilde{C}t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{19}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{20}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{21}$$
$$h_t = o_t \odot \tanh(C_t), \tag{22}$$

where x_t represents the input vector at time step t. The hidden state vector is denoted by h_t, and the cell state vector is represented by C_t. The functions σ and tanh are the sigmoid and hyperbolic tangent activation functions, respectively. f_t, i_t, and o_t are the forget, input, and output gates of the LSTM layer, which control the flow of information through the cell state. $\tilde{C}_t$ is the candidate cell state, a temporary value that helps in updating the cell state. The weight matrices W_f, W_i, W_C, and W_o correspond to the forget, input, output, and candidate cell state, while the bias vectors b_f, b_i, b_C, and b_o are the learnable parameters associated with these gates in the LSTM layer.

The unidirectional LSTM model efficiently captures past information through the cell state vector, which helps to alleviate the vanishing gradient problem to a certain degree. However, it doesn't consider future information like bidirectional models do. Despite this, unidirectional LSTM's lower complexity makes it apt for deployment on edge devices, and its compatibility with TensorFlow Lite Micro renders it a compelling choice for real-world applications needing on-device processing.

B. UNIDIRECTIONAL LSTM NETWORK ARCHITECTURE

Building on a solid foundation, we present a DL model for forecasting solar power generation using a stacked unidirectional LSTM architecture. By integrating L2 regularisation, dropout, and batch normalisation techniques, the model enhances both performance and generalisation capabilities. As illustrated in Figure 2 (a), the network comprises two LSTM layers with (LSTM1) N_1 and (LSTM2) N_2 units, capturing temporal relationships in input data. L2 regularisation, with a strength of 0.001, counters overfitting by penalising large weight values, while dropout layers

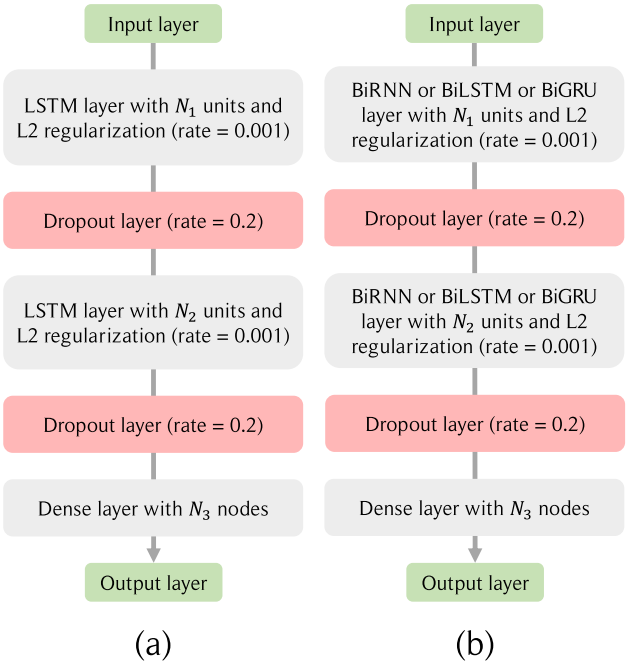


FIGURE 2. (a) Unidirectional LSTM network architecture and (b) BiRNN, BiLSTM, and BiGRU networks architecture.

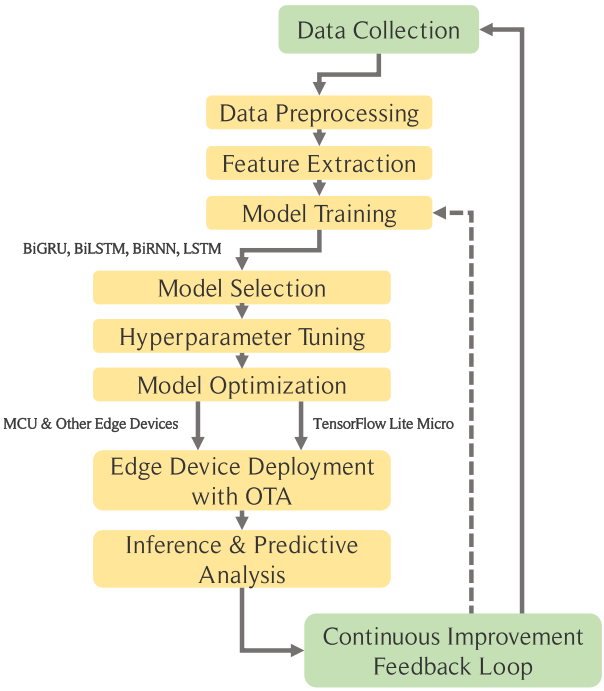


FIGURE 3. Simplified TinyML MLOps workflow for solar energy forecasting.

with a rate of 0.2 bolster robustness and curb overfitting. Batch normalisation layers standardise the activations of the LSTM layers, boosting training efficacy and mitigating overfitting. A Dense layer with N_3 units maps the LSTM outputs to a lower-dimensional space, and the output layer produces the final solar power generation forecasts. In essence, our proposed model smoothly integrates a stacked unidirectional LSTM architecture with L2 regularisation, dropout, and batch normalisation techniques, creating a robust and well-generalised model apt for solar power generation forecasting.

C. BiRNN, BiLSTM, AND BiGRU NETWORKS ARCHITECTURE

To make a fair comparison with the Unidirectional LSTM, the BiRNN, BiLSTM, and BiGRU models adopt the same DL architecture. As depicted in Figure 2 (b) this consists of two bidirectional layers, with (BiRNN1 or BiLSTM1 or BiGRU1) N_1 and (BiRNN2 or BiLSTM2 or BiGRU2) N_2 number of units. L2 regularisation (with a strength of 0.001) and dropout layers (with a rate of 0.2) are employed across all three models,[2] as well as batch normalisation layers for normalising the activations of the bidirectional layers. A Dense layer with N_3 number of units maps the bidirectional layer outputs to a lower-dimensional space, and the output layer produces the final solar power generation forecasts. This configuration enables an equitable comparison of their performance with the Unidirectional LSTM.

[2]To further elucidate our choice of parameters, it's pertinent to note that the selection of 0.001 for L2 regularization and 0.2 for dropout rate was not arbitrary. This decision has been formulated after an extensive grid search process and trial and error for fine tuning the model and training process to ensure no model overfitting.

D. TinyML MLOps PIPELINE

To summarise the deployed operations, our approach towards solar energy forecasting incorporates a comprehensive MLOps pipeline, specially designed for TinyML applications. This pipeline is outlined in Figure 3 and consists of several steps, each contributing to the efficient and accurate functioning of the forecasting model.

The process initiates with *Raw Data Collection* from diverse sources, including solar irradiance sensors and weather stations. Following this, the collected data is subjected to *Data Preprocessing* to ensure data integrity and consistency. Upon preprocessing, we carry out *Feature Extraction and Engineering*, where pertinent features are selected to be included in the forecasting model. These features encompass historical solar irradiance data, weather conditions, and other relevant factors.

The *Model Training* phase follows, involving the experimentation with various ML architectures. Post model selection, *Hyperparameter Tuning* is conducted to refine the chosen model's performance. The model then undergoes *Model Optimisation* to strike a balance between model complexity and computational efficiency, a critical step in the realm of TinyML. Subsequently, the optimised model is *Deployed on Edge Devices* via TensorFlow Lite Micro for real-time inference and predictive analysis.

In addition to this, *Over-The-Air (OTA) Updates* play a significant role in ensuring that the edge devices run the

TABLE 2. Hyperparameters and regularisation parameters used in the model.

Parameter	Value
Optimiser	Adam
Learning rate	0.001
Batch size	64
Epochs	{64, 128, 254}
Hidden dense layers activation function	ReLU
Output layer activation function	Linear
L2 regularisation (LSTM)	0.001
Dropout (after LSTM and dense layers)	0.2
Loss function	MSE

most updated version of the model, permitting remote model management and updates. The pipeline concludes with a *Continuous Improvement Feedback Loop*, facilitating the continuous collection of new data, evaluation of our model's performance, and model updating as needed. This feedback loop ensures that our model remains current with the latest data and trends, thereby maintaining its predictive accuracy and efficiency over time.

IV. RESULTS AND DISCUSSION

This section discusses the results of employing various ML models and elaborates on the key hyperparameters and regularisation techniques used in the training process. The hyperparameters of the different models are summarised in Table 2. The use of these specific hyperparameters and regularisation parameters, as discussed earlier, ensures a robust and stable learning process.

The comparative analysis of the different ML models addresses the disparities in their performance, and provides insights into their relative merits and limitations. These variations in performance are particularly significant when considering the deployment of these ML models on edge devices for low-cost domestic applications. A detailed exploration of this aspect will help us understand the practical limitations of the chosen models, as well as the potential enhancements required for future implementations.

A. MODELS PERFORMANCE COMPARISON

Figure 4 presents the results of the effect of the choice of *LB* on the overall performance of the system for two performance metrics, RMSE and R^2. The figure displays the performance of three types of bidirectional models—BiGRU, BiLSTM, and BiRNN—using both 3 and 7 input features. The first row of subfigures corresponds to the models using 3 input features, while the second row represents those using 7 input features.

A noticeable trend across the subfigures is that the models tend to perform relatively similarly after 5 look-back time steps, regardless of the number of input features used. This suggests that increasing the look-back step size beyond 5 steps may not significantly improve the models' forecasting performance. As such, we can optimise the computational resources by limiting the look-back step size to around 5 steps without compromising the accuracy of the solar power forecasting.

Figure 5 showcases the outcomes for various ML model configurations with three input features and a 4-hour look-back period (i.e., 8 look-back steps with a half-hour step size for solar power forecasting for half an hour ahead) for a site with a nominal power of 50MW. The figure includes six subfigures, illustrating the performance of three kinds of bidirectional models—BiRNN, BiGRU, and BiLSTM—in terms of R^2 and RMSE, along with their respective error distributions.

The marked epochs in Figure 5 represent key ''elbow points'' during the training of the models. The ''elbow point'' in a training curve typically signifies the point at which further training begins to yield diminishing improvements in the error rate. In other words, this is the point at which the models start to converge, and training beyond these epochs leads to relatively minor reductions in the error. This observation is particularly crucial from a computational efficiency standpoint, as it indicates an optimal stopping point that can prevent excessive computational resource usage and overfitting.

Identifying these ''elbow points'' is an empirical process based on monitoring the model's performance throughout the training period. The depicted epochs were determined to be the most effective in our case. This technique is widely used in the ML community to optimise training and prevent overfitting, a critical aspect for practical applications.

Our study of the error distribution results, illustrated in Figure 5, demonstrates that our models' error patterns diverge from the conventional characteristics of a normal distribution. Nevertheless, it's key to underline that despite these variations, our models don't exhibit a considerable bias, and the error variance stays within permissible boundaries. Importantly, the mean error is around zero, emphasising that our models, by and large, avoid over or under fitting.

These findings illustrate that, despite the deviations from a normal distribution in error patterns, the models can still deliver precise, unbiased forecasts of solar power production and almost follows kernel normal distribution. This understanding stands out as it demonstrates the models' deftness in dealing with the intrinsic unpredictability tied to solar power generation. It highlights their capability to provide trustworthy, data-driven results when interacting with intricate, real-world datasets. Here, it is essential to clarify that the objective behind this bias test analysis was not to show which error distribution is suitable for the forecasting task but to underscore the models' unbiased nature. The results show that, regardless of the observed error distribution, these models offer unbiased forecasts which emphasise the robustness and reliability of these models in solar power yield prediction.

Drawing from these insights, we propose the selection of the most fitting bidirectional model be guided by a combination of aspects - predictive accuracy, available computational resources, and the type and volume of input features. This

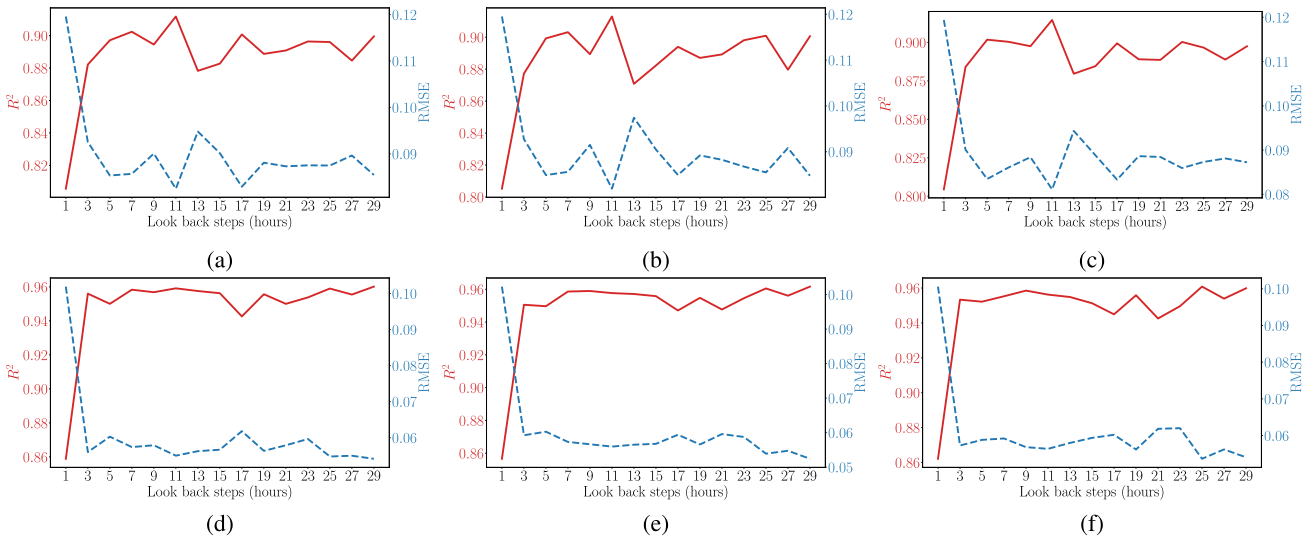


FIGURE 4. RMSE and R^2 for time series forecasting for 1/2 hour ahead on site 1 using site 1 dataset trained model. The figures in the first row correspond to the number of used input features N = 3, and the figures in the second row correspond to the number of used input features N = 7. From left to right: BiGRU, BiLSTM, and BiRNN models. Hyperparameters: Adam optimiser, ReLu activation functions for the hidden layers, linear activation function for the output layer, learning rate = 0.001. Dashed line for RMSE (Left red axis) and solid for R^2 (Right blue axis).

considered approach aids in more customised optimisation of solar power forecasting, consequently improving the proficiency and practicality of renewable energy systems. Furthermore, recognising and managing the deviations from normal distribution in error patterns enables a deeper comprehension of model performance and promotes a consistent enhancement of our forecasting models' robustness.

Figure 6 showcases time series power yield forecasting outcomes from multiple ML models. These models, tested for a site with a 50MW nominal power, utilize either 3 or 7 input features and adopt an 8-hour look-back period, effectively meaning 8 steps each of a half-hour interval to forecast the solar power for the subsequent half-hour. Three bidirectional models, namely BiRNN, BiGRU, and BiLSTM, form the core focus of this analysis.

A closer look at Figure 6 brings to light a discernible trend: models leveraging the full suite of 7 features ($N = 7$) tend to have a performance edge over those restricted to just 3 features ($N = 3$). One critical factor behind this superior performance is the inclusion of the 'actual power yield' feature in the seven-feature setup. These features selection can be used as evidence of the correlation coefficient analysis and features selection criterion that we studied before in Figures 1 and 4. That said, it's vital to acknowledge the practical considerations that govern feature selection in real-world scenarios. Often, the availability of certain features, especially time-series data for the actual power yield, might be restricted due to various reasons – from data collection challenges to resource constraints. The encouraging takeaway from our analysis is the model's capability to churn out reliable and relatively accurate forecasts even when it's fed with a pared-down feature set. This demonstrates the model's resilience and adaptability, and signifies that solar farms, even those with fewer measurement tools or data constraints, can still harness robust forecasting models. This adaptability underscores the model's value, especially in settings where resources might be limited or data acquisition might pose challenges.

Furthermore, the assertions and insights drawn from Figure 6 find resonance in the comprehensive results that are presented in Table 3. The table, set for a detailed discussion in the subsequent table, lays bare the numeric performance metrics across an array of scenarios – ranging from different feature inputs to season-based variations. This reinforces the findings from the figure but also offers a multi-dimensional understanding of the model's performance.

Table 3 presents the performance of four different models—LSTM, BiRNN, BiGRU, and BiLSTM—across three different seasons' results (Winter, Summer, and both). The performance is measured using various metrics, including training and test R^2, training and test RMSE, error variance (σ_e^2), and the expectation of error ($E[e]$). The results are shown for two different feature sets ($N = 3$ and $N = 7$). As observed in the table, a more insightful analysis can be as follows:

1) The BiRNN model demonstrates a comparable performance in terms of test R^2 to other more complex models such as BiLSTM, despite its own lower complexity. This hints towards BiRNN's effectiveness in learning from the provided data without requiring complex architectures. However, one should also bear in mind the significant vanishing gradient problems associated with BiRNN, which might limit its ability to learn long-term dependencies if the data involves intricate temporal patterns.

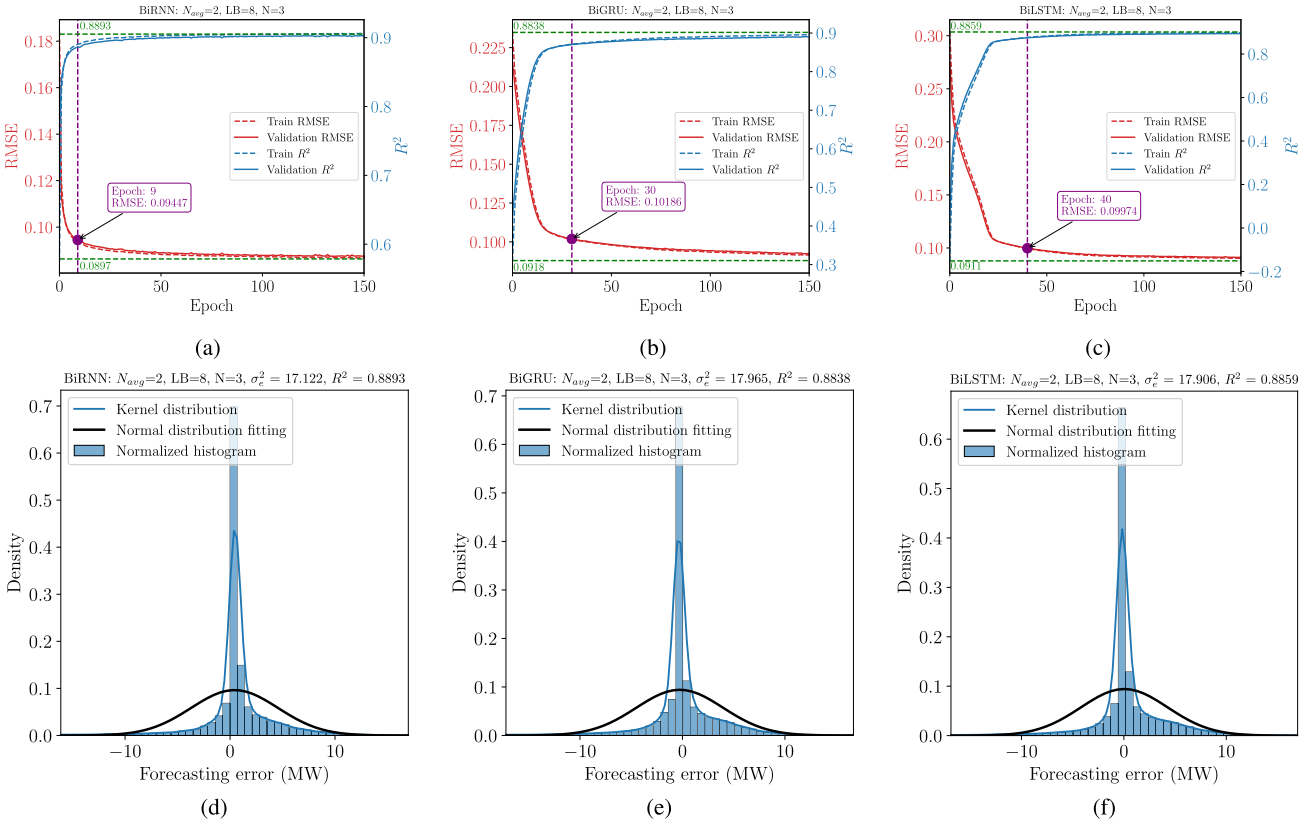


FIGURE 5. Results for various setups of the ML models for three input features and look-back period of 4 hours (i.e., 8 look-back steps with half an hour step size for solar power forecasting for half an hour ahead) for a site with nominal power of 50MW.

2) The uni-directional LSTM and BiLSTM models have architectures designed to tackle the vanishing gradient problem, leading to a significantly reduced effect. However, their performance does not show a clear advantage over the simpler BiRNN model. This could be due to the nature of the solar power yield data, where the benefits of the more sophisticated architectures are not as pronounced. It's also worth noting that the BiLSTM model has higher complexity, which might lead to increased computational costs and longer training times.
3) The BiGRU model strikes a balance between complexity and performance, with diminished vanishing gradient problems in comparison to BiRNN. Even though it doesn't noticeably outperform the BiRNN model, it offers a competitive performance against the others. Therefore, if there's a need to balance performance with computational efficiency in the realm of solar power yield forecasting, the BiGRU model could be an apt choice.
4) All the models show consistent performance improvement when the number of input features increases from $N = 3$ to $N = 7$, suggesting that a more comprehensive feature set can boost the forecasting performance, irrespective of the model architecture. This observation could be harnessed in future researches by including weather forecasting data as additional input features, potentially elevating the accuracy of solar power yield predictions without necessitating prior knowledge of the exact power yield time series data.
5) The seasonal effect, as reflected in the performance table, reveals that the models generally perform better in Summer compared to Winter. This can be attributed to the differences in weather patterns, cloud coverage, and solar radiation between these two seasons. Therefore, when evaluating the performance of different models and choosing the most suitable one for solar power yield forecasting, the seasonal effect should be taken into account.

In conclusion, despite the challenges posed by vanishing gradients, the BiRNN model's performance, coupled with its simplicity, stands out. However, in choosing the most suitable model for forecasting, it is imperative to consider other factors, such as model complexity, computational cost, seasonal impacts, and the specific attributes of the solar power yield data. The BiGRU model serves as a balanced alternative, and the augmentation of features can generally enhance the performance across all architectures.

B. PERFORMANCE ON EDGE DEVICES FOR HOUSEHOLD FORECASTING

In this section, we present the results associated with evaluating the unidirectional LSTM model. The model testing on edge devices is carried out on an ESP32-S3 MCU,

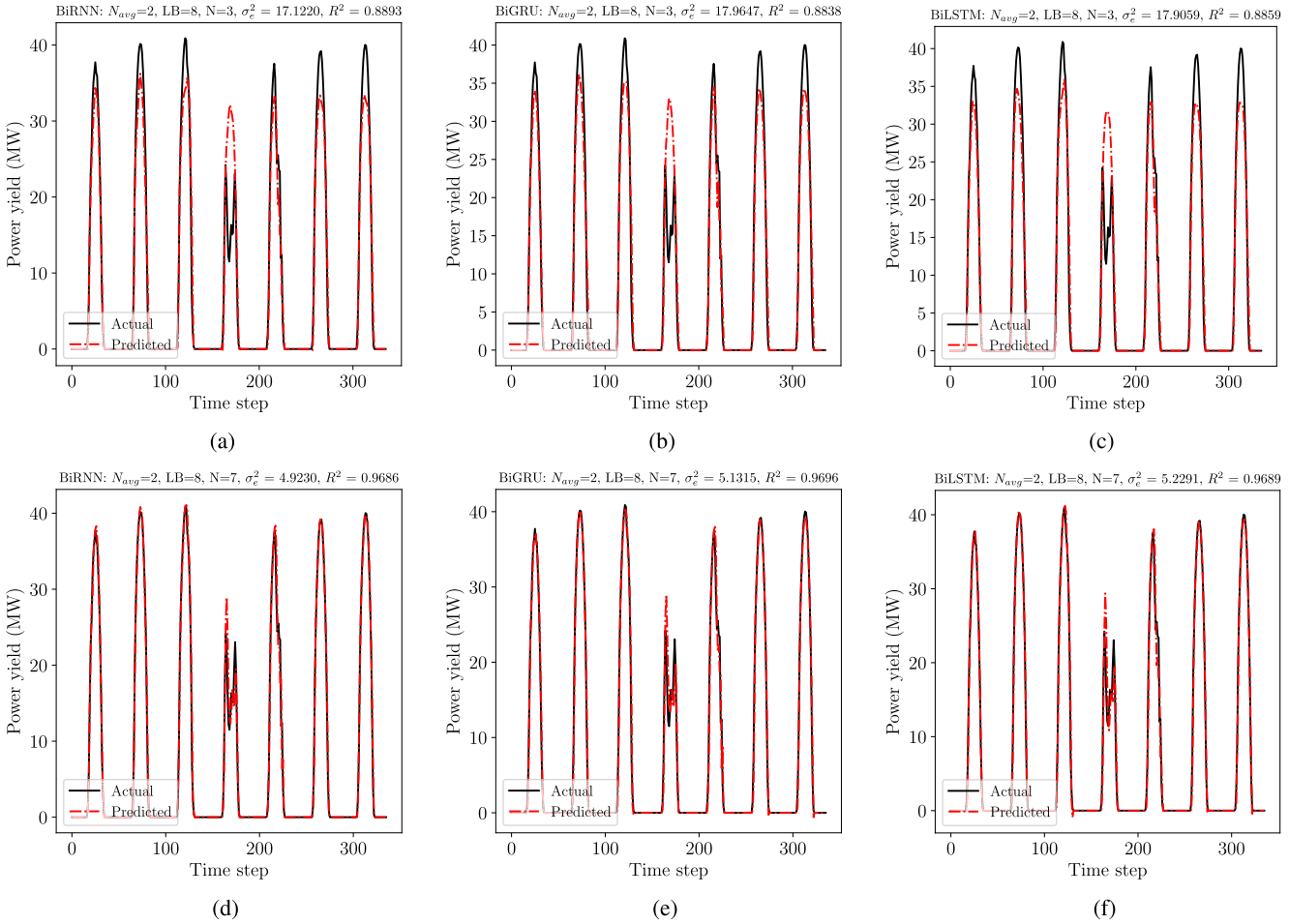


FIGURE 6. Results for various setups of the ML models for 3 and 7 input features and look-back period of 4 hours (i.e., 8 look-back steps with half an hour step size for solar power forecasting for half an hour ahead) for a site with nominal power of 50MW.

TABLE 3. Comparison of LSTM, BiGRU, BiLSTM, and BiRNN architectures and performance for solar power yield forecasting for $N = 7$ and 3 input features and a look-back period of 4 hours (i.e., $LB = 8$ steps with half an hour step size for solar power forecasting for half an hour ahead) for a site with nominal power of 50MW.

Season	Performance metric	Training R^2	Test R^2	Training RMSE	Test RMSE	σ_e^2	$E[e]$	Training R^2	Test R^2	Training RMSE	Test RMSE	σ_e^2	$E[e]$
	ModelNumber of features	N=3						N=7					
Summer	LSTM	0.913	0.931	0.079	0.072	14.710	-0.145	0.974	0.970	0.043	0.046	4.385	-0.062
	BiRNN	0.914	0.933	0.079	0.071	14.564	0.252	0.976	0.980	0.041	0.038	4.084	-0.099
	BiGRU	0.906	0.927	0.082	0.074	15.734	0.175	0.971	0.977	0.046	0.041	4.908	0.124
	BiLSTM	0.908	0.928	0.081	0.073	15.393	-0.198	0.972	0.976	0.045	0.042	4.852	0.010
Winter	LSTM	0.885	0.881	0.094	0.093	20.145	-0.027	0.965	0.964	0.052	0.052	6.111	0.025
	BiRNN	0.888	0.874	0.093	0.098	19.502	-0.317	0.965	0.967	0.052	0.051	6.044	0.208
	BiGRU	0.884	0.863	0.094	0.102	20.245	-0.259	0.963	0.965	0.053	0.052	6.387	-0.093
	BiLSTM	0.883	0.862	0.095	0.102	20.306	-0.280	0.962	0.959	0.054	0.056	6.666	0.196
All	LSTM	0.900	0.886	0.086	0.091	17.524	-0.107	0.972	0.974	0.046	0.044	4.812	-0.101
	BiRNN	0.901	0.889	0.086	0.090	17.122	0.430	0.970	0.969	0.047	0.049	4.923	-0.450
	BiGRU	0.896	0.884	0.087	0.092	17.965	-0.267	0.970	0.970	0.047	0.048	5.132	0.006
	BiLSTM	0.897	0.886	0.087	0.091	17.906	0.040	0.969	0.969	0.048	0.048	5.229	-0.006

which boasts a dual-core XTensa LX7 processor operating at 240 MHz and 512 kilobytes of internal Static random-access memory (SRAM). The choice of the ESP32-S3 stems from its affordability and IoT-readiness, as it encompasses built-in WiFi and Bluetooth capabilities, as well as a dual-core architecture that enables running multiple threads. This facilitates the collection of footprint measurements of the ML model while isolating background processes, providing a clearer understanding of the TinyML performance for cost-effective solar energy yield forecasting.

In Figure 7, the performance of unidirectional LSTM models with 3 and 7 input features and a 4-hour look-back period (i.e., 8 look-back steps with an hour step size for solar power forecasting for half an hour ahead) for a site with

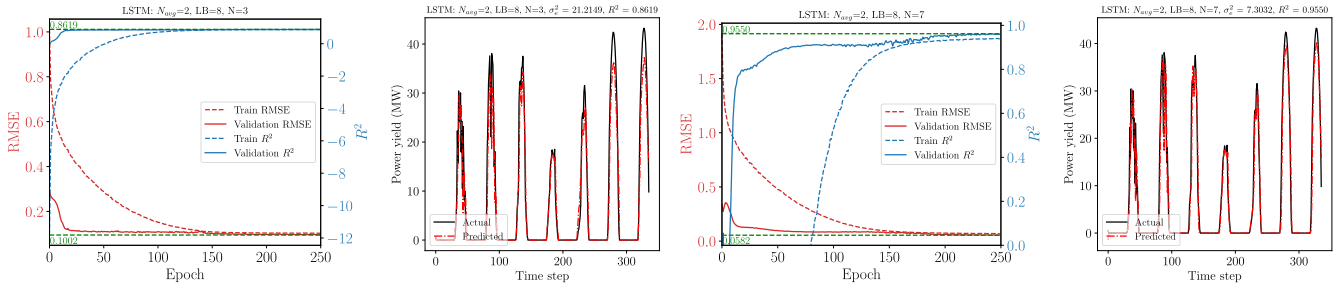


FIGURE 7. Results for various setups of the unidirectional LSTM ML models for 3 and 7 input features and look-back period of 4 hours (i.e., 8 look-back steps with an hour step size for solar power forecasting for half an hour ahead) for a site with nominal power of 50MW.

a nominal power of 50MW is displayed. Comparing these outcomes to the previously discussed bidirectional models (BiRNN, BiGRU, and BiLSTM), it's clear that unidirectional LSTM models exhibit similar performance levels. Yet, one notable difference is the LSTMs' tendency to converge more gradually than other models. This slower convergence can be a trade-off when considering their use on edge devices. Although offering comparable accuracy levels, their extended training times might pose concerns for resource-limited devices. Nevertheless, the benefits of unidirectional LSTM models in terms of computational efficiency and edge device compatibility make them an appealing option for solar power forecasting.

Table 4 presents the performance comparison of LSTM models for solar power forecasting on edge IoT devices for various setups of the unidirectional LSTM ML models for 7 input features and a look-back period of 8 hours (i.e., 8 look-back steps with an hour step size for solar power forecasting for half an hour ahead) for a site with a nominal power of 30 MW. In the table, various hyperparameters, such as the number of LSTM cells and look-back steps, are adjusted to understand their impact on forecasting performance, model flash size, and inference rate. The performance metrics considered include train and test RMSE, train and test R^2, model flash size (Bytes), and inference rate (Hz).

The selection of optimal hyperparameters is critical in achieving an effective balance between model performance and computational efficiency, especially when deploying these models on edge IoT devices with limited computational resources. As observed in the table, a more insightful analysis can be as follows:

1) *Look-back steps:* This represents the number of preceding time steps that are considered as input features for the LSTM model to forecast. In our study, different look-back periods (8 and 4) are evaluated. Models using a 4-step look-back consistently outperform those with an 8-step look-back. This indicates that using a shorter sequence of historical data effectively captures the most influential temporal patterns for accurate forecasting. The performance improvement might be due to focusing on more recent and hence more relevant information. Alternatively, it may be the case that longer look-back steps increase the model's complexity, leading to less optimal results.
2) *Number of LSTM cells:* This refers to the complexity of the LSTM model, denoting the number of hidden units or memory cells in the LSTM layers. We test with different configurations, such as 128 + 64 + 32, 64 + 32 + 24, and so forth. Models with a higher number of LSTM cells generally exhibit improved performance, as indicated by lower train and test RMSE values. However, there is a trade-off. More LSTM cells necessitate more computations, which result in a decrease in the inference rate. Hence, for an edge IoT device with limited computational resources, there's a need to select an optimal number of LSTM cells that balances forecasting accuracy and computational efficiency.
3) *Train and Test Metrics (RMSE and R^2):* These metrics evaluate the model's performance on both the training and testing datasets. Lower RMSE values and higher R^2 values signify better model performance. The table reveals that the test R^2 values are generally higher than the corresponding train R^2 values, which implies our models are not overfitting and can generalise well to unseen data.
4) *Model Flash Size:* This is the amount of storage the model requires, a vital factor when deploying ML models on edge IoT devices, which typically have limited memory. As the complexity of the LSTM model (number of LSTM cells) increases, the model flash size also expands. Thus, it's crucial to find a model with the right level of complexity that fits within the device's storage constraints for successful deployment on edge devices.
5) *Inference Rate:* This denotes the computational speed or how many inferences the model can make per second (Hz). As the LSTM model's complexity (number of LSTM cells) increases, the inference rate generally decreases. Simpler models with fewer LSTM cells and look-back steps are computationally more efficient and provide higher inference rates.

In conclusion, based on these results, an LSTM model with 64 cells and a 4-step look-back period appears to

TABLE 4. Performance comparison of LSTM models for solar power forecasting on edge IoT devices for various setups of the unidirectional LSTM ML models for 7 input features and look-back period of 4 and 2 and hours (i.e., 8 and 4 look-back steps with half an hour step size for solar power forecasting for half an hour ahead) for a site with nominal power of 30MW.

Look back # steps	N_1+N_2+N_3 # of units	Train RMSE	Test RMSE	Train R^2	Test R^2	Model flash size (Bytes)	Inference rate (Hz)
8	128 + 64 + 32	0.0384	0.0631	0.9798	0.9475	102,848	19.05
8	64 + 32 + 24	0.0494	0.0580	0.9662	0.9560	51,592	57.14
8	32 + 16 + 20	0.0524	0.0581	0.9617	0.9552	38,176	103.16
8	16 + 8 + 16	0.0531	0.0576	0.9611	0.9558	34,304	197.24
8	8 + 4 + 8	0.0528	0.0577	0.9612	0.9559	33,656	396.37
4	128 + 64 + 32	0.0480	0.0568	0.9691	0.9569	89,032	41.85
4	64 + 32 + 24	0.0503	0.0546	0.9655	0.9590	36,680	97.27
4	32 + 16 + 20	0.0528	0.0536	0.9624	0.9608	22,792	206.80
4	16 + 8 + 16	0.0550	0.0543	0.9592	0.9593	18,920	420.56
4	8 + 4 + 8	0.0564	0.0547	0.9568	0.9587	17,752	826.19

offer the best balance between predictive performance and computational efficiency. It's therefore an excellent candidate for deployment on edge IoT devices for solar power forecasting. This configuration achieves a test R^2 of 0.9590, demonstrating high predictive accuracy. Simultaneously, it respects the computational and storage limitations of edge devices, making it suitable for real-world applications in both industrial and residential scenarios.

C. PRACTICAL IMPLICATIONS FOR INDUSTRIAL AND HOUSEHOLD APPLICATIONS

This investigation's outcomes bear several pragmatic implications for both industrial and domestic applications. Enhancing the precision of solar energy forecast through the proposed LSTM models can support the general effectiveness and solidity of energy networks, diminishing the dependency on conventional power origins and promoting the incorporation of renewable energy.

1) INDUSTRIAL APPLICATIONS

In industrial environments, precise solar energy forecasting is essential for optimising energy utilisation and reducing operational expenses. By harnessing the studies models, industries can more effectively arrange their energy-intensive procedures during periods of heightened solar energy generation, lessening their reliance on grid-supplied electricity and minimising their energy expenditures. Furthermore, solar energy forecasting can assist utilities in managing energy demand more proficiently, resulting in enhanced grid stability and decreased energy squandering.

2) HOUSEHOLD APPLICATIONS

For household users, solar energy forecasting can play a crucial role in optimising solar panel usage and energy storage systems. Homeowners can employ forecasts to organise their energy consumption, guaranteeing that they exploit the solar energy generated by their panels to the greatest extent with low-cost solutions. For instance, households can schedule energy-intensive tasks, such as charging electric vehicles or operating appliances, during times when solar energy production is anticipated to be high. Moreover, precise solar energy forecasts can aid homeowners in determining when to store solar energy in their batteries, enabling them to utilise stored energy during periods of low solar generation or elevated electricity prices.

3) EDGE IoT DEVICES

Deploying unidirectional LSTM models on edge IoT devices for solar power forecasting can cultivate a more decentralised and cost-effective energy management ecosystem. By performing solar power forecasting directly on edge devices like smart meters or home energy management systems, households can reap the benefits of real-time forecasting without relying on external servers or cloud services, minimising latency and safeguarding data privacy. Furthermore, edge devices can interact with other smart appliances within the residence to fine-tune energy usage habits, fostering a greener and more energy-conscious living space.

D. LIMITATIONS AND FUTURE RESEARCH DIRECTIONS

Despite the promising results obtained, we can address several limitations that should be carefully taken into account, which also pave the way for future research directions.

1) FEATURE SELECTION

Although the current study considers multiple features for predicting solar energy yield, additional features or feature engineering techniques might further improve the prediction accuracy of the models. Other input features may also consider wind speed, cloud cover, time of day, seasonal changes and forecasted weather conditions as they are commonly used as features in solar panel power forecasting.

2) MODEL SELECTION

The study concentrates on comparing certain types of ML models. Nonetheless, it merits noting that other ML or DL models might potentially provide superior performance in solar energy forecasting. Alternative models, such as hybrid models that amalgamate different ML or DL algorithms, or innovative ML or DL architectures explicitly crafted for time series forecasting, should also be contemplated.

3) HYPERPARAMETER TUNING

The tuning of hyperparameters is a key facet of our methodology. It is particularly critical in the context of TinyML applications and edge inference, where resource constraints necessitate highly efficient and optimised models.

A careful choice of hyperparameters such as learning rate, number of hidden layers and units, and look-back steps, can significantly influence the forecasting model's performance. For instance, the learning rate orchestrates the magnitude of adjustments made to the model's weights during learning. An optimally selected rate ensures effective reduction in training loss, while inappropriate rates can cause unstable training or slow convergence. Among the hyperparameters, the learning rate stands out in significance as underscored by [32] and [33] where the learning rate often emerges as the most influential hyperparameter, with its fine-tuning being of paramount importance when employing stochastic gradient descent.

Furthermore, the model's complexity, as determined by the number of hidden layers and units, should be carefully chosen to effectively capture underlying patterns in the data without leading to overfitting, a critical concern in TinyML. The choice of look-back steps impacts the temporal dependencies the model can learn, striking a balance between richer historical data integration and computational load.

In this study, guided by the significance of learning rate as emphasised in [32] and [33], we employed grid search and random search for hyperparameter tuning across the BiGRU, BiLSTM, BiRNN, and unidirectional LSTM models. Specifically, the learning rate was carefully selected from a log-scale set of $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ during grid search. The objective was to pinpoint hyperparameter combinations conducive to optimal solar power yield forecasting losses and ensuring the models' suitability for edge inference.

While our hyperparameter exploration was thorough, we acknowledge that advanced optimisation techniques like genetic algorithms, particle swarm optimisation, and Bayesian optimisation could further refine model performance. By more effectively navigating the hyperparameter space, these techniques can pinpoint optimal values, and their integration into future studies could bolster the performance of our forecasting models.

4) EDGE DEVICES

To ensure that the LSTM model used for low-cost household forecasting on edge devices is applicable and efficient in practice, it is necessary to conduct extensive evaluations on various edge device configurations. Different edge devices may have different computational capabilities and hardware specifications, which can affect the performance of the LSTM model. Moreover, the training and inference times of the LSTM model may vary depending on the edge device's processing power and memory capacity.

5) IMPACT OF SOLAR FARM SIZE ON EDGE-BASED FORECASTING

The size of solar installations significantly impacts the predictability of solar power yield and, consequently, the complexity and performance of forecasting models deployed on edge devices. The application of TinyML techniques for these scenarios, considering their inherent constraints and opportunities, is crucial.

Industrial applications often feature solar farms that span vast areas, leading to an inherent averaging effect on solar irradiance due to this extensive coverage. Weather-induced localised fluctuations in solar power yield are averaged over the large solar farm area, resulting in relatively stable and predictable power output. This stability can be effectively harnessed by TinyML models operating on edge devices in these setups, simplifying the task of forecasting peak production periods, typically between 7:00 AM to 11:00 PM on weekdays, and aligning them with high-energy-demand operations.

In contrast, household applications typically have limited areas for solar panel installation, making them more prone to yield fluctuations due to variable weather conditions. This situation introduces higher variability and unpredictability in solar power yield, thereby complicating the task of forecasting peak and off-peak production hours. Common residential peak hours are from 5:00 PM to 8:00 PM on weekdays and on weekends, while off-peak hours are usually on weekdays before 4:00 PM and after 9:00 PM, with weekends having a more flexible off-peak schedule. It is worth mentioning that the off-peak and on-peak periods specified pertain to average timings in the USA. It's notable that the distinction between off-peak and on-peak periods can vary between summer and winter, the geographical area as well as the time throughout the year [34], [35]. Such a challenge necessitates TinyML models to incorporate a broader and more dynamic range of information, including real-time weather forecasts and local shading conditions, among other features, to generate accurate predictions on edge devices.

In both contexts, our proposed TinyML models are capable of predicting solar energy yield effectively, accounting for these distinct dynamics. For industrial applications, the models can leverage the relative stability of solar yield to provide reliable forecasts on edge devices, aiding in optimising energy-intensive processes. In contrast, for residential settings, our models can adeptly manage the additional complexities by integrating a wider array of predictive features, ensuring precise and efficient predictions even on resource-constrained edge devices.

Moreover, it is essential to evaluate the appropriateness of the LSTM architecture for implementation on edge devices. Notably, TensorFlow Lite Micro currently supports only simple unidirectional LSTMs. This implies that more intricate LSTM architecture variants, such as BiRNNs, BiLSTMs, and BiGRUs, might not be compatible with edge devices utilising TensorFlow Lite Micro. Hence, it becomes

imperative to consider the constraints and capabilities of TensorFlow Lite Micro when devising LSTM models for low-cost household forecasting on edge devices.

Based on the limitations identified, several avenues for future research can be explored:

6) EXPANDING THE DATASET

It is important to investigate new results based on some new datasets with different characteristics and climate conditions and regions. This will give more insight into the applicability of employing ML on edge devices to improve the design space of AI-powered smart solar systems and generally smart grids.

7) ADVANCED FEATURE ENGINEERING IN TinyML

The adoption of advanced feature engineering techniques can enhance the predictive power of TinyML models for solar power forecasting on edge devices. These techniques could include incorporating additional data sources, calculating derived features like moving averages, standard deviations over rolling windows, or using time-lagged values to capture complex temporal patterns. External data such as weather conditions could also be integrated for enriched context.

However, the introduction of complex features increases computational demands, which can pose a challenge for resource-limited edge devices. Balancing the need for model performance and computational efficiency is therefore crucial. This could involve employing advanced feature selection methods to retain only the most impactful features, keeping the model simple and computationally feasible. Thus, the fine-tuning of feature engineering, in line with hyperparameter tuning, is key to maximizing model performance under the constraints of edge inference.

8) EXPLORING ALTERNATIVE MODELS

While our study primarily utilized unidirectional LSTM models for their effectiveness in time-series forecasting and compatibility with TinyML in edge IoT devices, we also included bidirectional models like BiGRU and BiLSTM for a more comprehensive understanding of sequential data. This diverse model selection not only aligns with current computational capacities but also sets a benchmark for future research. As technology progresses, especially in edge computing, investigating more advanced models, which could become deployable on such devices, will be a key area of future work. This approach lays the groundwork for ongoing advancements in solar energy yield forecasting.

Potential alternatives could include CNNs, valuable for spatial data patterns, and hybrid models like CNN-LSTM or Transformer-LSTM, merging the benefits of both architectures. Nevertheless, the increased computational complexity of these alternatives should be considered, especially in the context of TinyML applications with resource limitations. Therefore, model selection should harmonise computational efficiency and predictive performance. In many instances, simpler models like the LSTM may offer the best trade-off, delivering solid forecasting outcomes without unnecessary complexity, as exemplified by the models in this study. Hence, the advantages of more complex models should be carefully examined against their computational costs.

TABLE 5. List of abbreviations.

Abbreviation/Acronym	Full Name
AI	Artificial intelligence
ANN	Artificial neural network
ARMA	Auto-regressive moving average
AT	Air temperature
ATM	Atmospheric pressure
Bi	Bidirectional
BiGRU	Bidirectional gated recurrent unit
BiLSTM	Bidirectional long short-term memory
BiRNN	Bidirectional recurrent neural network
CNN	Convolutional neural network
DevOps	Development operations
DG	Distributed generation
DL	Deep learning
DN	Distribution networks
DNI	Direct normal irradiance
ELM	Extreme learning machine
FL	Federated learning
GHI	Global horizontal irradiance
GRU	Gated recurrent unit
IoT	Internet of things
LA	Look-ahead
LB	Look-back
LSTM	Long short-term memory
LV	Low-voltage
MCU	Micro-controllers unit
ML	Machine learning
MLOps	Machine learning operations
MSE	Mean square error
MV	Medium-voltage
MW	Mega-watt
NF	Neuro-fuzzy
NN	Neural network
OTA	Over-the-air
PV	Photovoltaic
RES	Renewable energy sources
RH	Relative humidity
RMSE	Root mean square error
RNN	Recurrent neural network
SRAM	Static random-access memory
SVM	Support vector machine
TFLite	Tensor-flow lite
TinyML	Tiny machine learning
TL	Transfer learning
TSI	Total solar irradiance

9) INCORPORATING CLIMATE CHANGE

Moving forward, it's essential to acknowledge the influence of changing climate aspects on solar power yield prediction models. The inescapable progression of climate change is reshaping weather patterns and solar irradiance, consequently impacting solar power production. Integrating these shifting climate factors in future investigations will contribute to a more durable and reflective forecasting model, accounting for long-term ecological shifts. This might entail the use of dynamic models designed to adjust to climate variable modifications, or introducing climate change projections as an extra feature in the model.

V. CONCLUSION

This study underscores the transformative potential of combining advanced ML methodologies with TinyML for solar energy yield prediction in low-cost, household-level solar farms. Our research delivered in-depth evaluation of four ML architectures (BiGRU, BiLSTM, BiRNN, and unidirectional LSTM), offering valuable guidance on their applicability and performance under the constraints of edge devices. Our work broadens the understanding of deploying smart, cost-efficient solutions in IoT environments and emphasises the necessity to consider the limitations of edge devices when choosing suitable ML architectures.

While our models exhibit promising accuracy, it is worth noting that the efficacy of our solution might vary based on factors like dataset characteristics, edge device capabilities, and the specificities of solar installations. For instance, the size and type of solar installations, whether household or industrial, significantly influence the predictability of solar yield. As larger installations can provide averaged, more stable outputs, edge-device models may find such contexts easier for prediction. Smaller, household installations, with their inherent yield variability, pose a more complex forecasting challenge.

Future research directions include exploration of other ML or DL models, innovative hybrid architectures for time-series prediction, and the integration of advanced hyperparameter tuning methods to enhance solar energy yield prediction accuracy. Ultimately, our work paves the way for improved resource planning and energy management in solar energy systems, promoting a more sustainable and efficient energy landscape at both the household and industrial levels.

APPENDIX

TABLE OF ABBREVIATIONS AND ACRONYMS

See Table 5.

REFERENCES

[1] L. González-Sotres, P. Frías, and C. Mateo, "Techno-economic assessment of forecasting and communication on centralized voltage control with high PV penetration," *Electr. Power Syst. Res.*, vol. 151, pp. 338–347, Oct. 2017.

[2] H. Zang, L. Cheng, T. Ding, K. W. Cheung, Z. Wei, and G. Sun, "Day-ahead photovoltaic power forecasting approach based on deep convolutional neural networks and meta learning," *Int. J. Electr. Power Energy Syst.*, vol. 118, Jun. 2020, Art. no. 105790.

[3] V. Prema, M. S. Bhaskar, D. Almakhles, N. Gowtham, and K. U. Rao, "Critical review of data, models and performance metrics for wind and solar power forecast," *IEEE Access*, vol. 10, pp. 667–688, 2022.

[4] B. D. Dimd, S. Völler, U. Cali, and O.-M. Midtgård, "A review of machine learning-based photovoltaic output power forecasting: Nordic context," *IEEE Access*, vol. 10, pp. 26404–26425, 2022.

[5] W. Holderbaum, F. Alasali, and A. Sinha, *Energy Forecasting and Control Methods for Energy Storage Systems in Distribution Networks, Predictive Modelling and Control Techniques*, 1st ed. Cham, Switzerland: Springer, 2023.

[6] M.-N. Nguyen, M.-H. Pham, Y.-K. Wu, and N.-T. Nguyen, "An optimizing method based on combining edge computer and long short-term memory networks applied to solar power forecasting," in *Proc. IEEE Int. Future Energy Electron. Conf. (IFEEC)*, Nov. 2021, pp. 1–6.

[7] P. Warden and D. Situnayake, *TinyML: Machine Learning With Tensorflow Lite on Arduino and Ultra-Low-Power Microcontrollers*. New York, NY, USA: O'Reilly Media, 2019.

[8] A. M. Hayajneh, S. Aldalahmeh, S. A. R. Zaidi, D. McLernon, H. Obeidollah, and R. Alsakarnah, "Channel state information based device free wireless sensing for IoT devices employing TinyML," in *Proc. 4th IEEE Middle East North Afr. Commun. Conf. (MENACOMM)*, Dec. 2022, pp. 215–222.

[9] M. Elsisi, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, "Reliable industry 4.0 based on machine learning and IoT for analyzing, monitoring, and securing smart meters," *Sensors*, vol. 21, no. 2, p. 487, Jan. 2021.

[10] F. Rodríguez, I. Azcárate, J. Vadillo, and A. Galarza, "Forecasting intra-hour solar photovoltaic energy by assembling wavelet based time-frequency analysis with deep learning neural networks," *Int. J. Electr. Power Energy Syst.*, vol. 137, May 2022, Art. no. 107777.

[11] W. Holderbaum, F. Alasali, and A. Sinha, "Short term load forecasting (STLF)," in *Energy Forecasting and Control Methods for Energy Storage Systems in Distribution Networks*, 1st ed. Cham, Switzerland: Springer, 2023.

[12] Y. K. Semero, J. Zhang, and D. Zheng, "PV power forecasting using an integrated GA-PSO-ANFIS approach and Gaussian process regression based feature selection strategy," *CSEE J. Power Energy Syst.*, vol. 4, no. 2, pp. 210–218, Jun. 2018.

[13] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, "Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism," *IEEE Access*, vol. 7, pp. 78063–78074, 2019.

[14] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast," *IEEE Access*, vol. 8, pp. 172524–172533, 2020.

[15] H. Eom, Y. Son, and S. Choi, "Feature-selective ensemble learning-based long-term regional PV generation forecasting," *IEEE Access*, vol. 8, pp. 54620–54630, 2020.

[16] M. K. Behera, I. Majumder, and N. Nayak, "Solar photovoltaic power forecasting using optimized modified extreme learning machine technique," *Eng. Sci. Technol., Int. J.*, vol. 21, no. 3, pp. 428–438, Jun. 2018.

[17] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[18] H. Sharadga, S. Hajimirza, and R. S. Balog, "Time series forecasting of solar power generation for large-scale photovoltaic plants," *Renew. Energy*, vol. 150, pp. 797–807, May 2020.

[19] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, vol. 236, pp. 1078–1088, Feb. 2019.

[20] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-enabled frugal smart objects: Challenges and opportunities," *IEEE Circuits Syst. Mag.*, vol. 20, no. 3, pp. 4–18, 3rd Quart., 2020.

[21] S. A. R. Zaidi, A. M. Hayajneh, M. Hafeez, and Q. Z. Ahmed, "Unlocking edge intelligence through tiny machine learning (TinyML)," *IEEE Access*, vol. 10, pp. 100867–100877, 2022.

[22] Y. Chen and J. Xu, "Solar and wind power data from the Chinese state grid renewable energy generation forecasting competition," *Sci. Data*, vol. 9, no. 1, p. 577, Sep. 2022.

[23] W. Holderbaum, F. Alasali, and A. Sinha, "Case study: Low voltage demand forecasts," in *Energy Forecasting and Control Methods for Energy Storage Systems in Distribution Networks*, 1st ed. Cham, Switzerland: Springer, 2023.

[24] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive Into Deep Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Mar. 1997.

[26] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

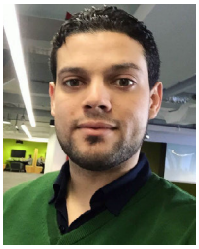
[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[29] Q. Chen, Z. Zheng, C. Hu, D. Wang, and F. Liu, "On-edge multi-task transfer learning: Model and practice with data-driven task allocation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1357–1371, Jun. 2020.

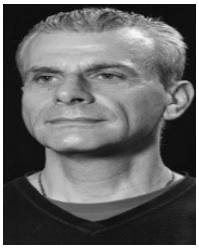
[30] A. Salama, A. Stergioulis, A. M. Hayajneh, S. A. R. Zaidi, D. McLernon, and I. Robertson, "Decentralized federated learning over slotted Aloha wireless mesh networking," *IEEE Access*, vol. 11, pp. 18326–18342, 2023.
[31] TensorFlow. (2021). *TensorFlow Lite Micro for Microcontrollers*. Accessed: Apr. 27, 2023. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers
[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[33] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, 2nd ed. Cham, Switzerland: Springer, 2012, pp. 437–478.
[34] (2021). *Electricity in the United States—Industry Definitions*. Accessed: Oct. 11, 2023. [Online]. Available: https://www.eia.gov/todayinenergy/detail.php?id=42915
[35] (2023). *What Are the Off-Peak and Peak Electricity Hours?* Accessed: Oct. 11, 2023. [Online]. Available: https://freedomsolarpower.com/blog/what-are-the-off-peak-and-peak-electricity-hours

ALI M. HAYAJNEH (Member, IEEE) received the B.Sc. and M.Sc. degrees from the Jordan University of Science and Technology (JUST), Irbid, Jordan, in 2010 and 2014, respectively, and the Ph.D. degree from the University of Leeds, Leeds, U.K. He is currently with the Department of Electrical Engineering, Faculty of Engineering, The Hashemite University, Zarqa, Jordan. He is also the Director of the Innovation and Entrepreneurial Projects Centre, The Hashemite University. His current research is funded by the Royal Academy of Engineering through two programs: 1) Transfer Systems through Partnerships (TSP) and 2) Distinguished International Associate (DIA) in the fields of smart agriculture, drone-assisted micro irrigation, and tiny machine learning on the edge IoT devices. His current research interests include drone-assisted wireless communications, public safety communication networks, backscatter communication, DL, power harvesting, stochastic geometry, device-to-device (D2D), machine-to-machine (M2M) communications, the modeling of heterogeneous networks, cognitive radio networks, cooperative relay networks, edge computing, and reinforcement learning.

FERAS ALASALI (Member, IEEE) received the Ph.D. degree in electrical power engineering from the University of Reading, in 2019. He is currently the Director of the Renewable Energy Center, The Hashemite University, Jordan. He is also an Assistant Professor with the Department of Electrical Engineering with more than six years of experience in optimal and predictive control models for energy storage systems and LV network applications. His research interests include control models for distributed generation and LV networks, load forecasting, and power protection systems. In addition, he is currently working on applying emerging technologies, such as machine learning and optimization methods to optimally simulate network loads, design protection systems for micro and smart grids, and solve different engineering problems.



ABDELAZIZ SALAMA (Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from Tripoli University, Tripoli, Libya, in 2009, and the M.Sc. degree in communication, control, and digital signal processing from the University of Strathclyde, Glasgow, U.K., in 2017. He is currently pursuing the Ph.D. degree with the University of Leeds, Leeds, U.K. His research interests include federated learning, autonomous systems, and sensing. He worked for nine years at local and international firms, in several positions in the areas of telecommunication engineering, information technology, and management.



WILLIAM HOLDERBAUM (Member, IEEE) has been with the University of Glasgow, the University of Reading, Manchester Metropolitan University, and Aston University. He is currently a Professor of control engineering with the University of Salford, U.K. He has played major leadership roles in research, whilst maintaining a very strong international reputation and an extensive list of publications and the Ph.D.'s supervision. He has applied his control expertise to several applications, particularly rehabilitation engineering and energy transmission, storage for electrical systems, and power systems.