

# Forecast Based Consensus Control for DC Microgrids Using Distributed Long Short-Term Memory Deep Learning Models

Seyed Amir Alavi, *Graduate Student Member, IEEE*, Kamyar Mehran, *Senior Member, IEEE*,  
Vahid Vahidinasab, *Senior Member, IEEE*, João P. S. Catalão, *Senior Member, IEEE*

**Abstract**—In a microgrid, renewable energy sources (RES) exhibit stochastic behavior, which affects the microgrid continuous operation. Normally, energy storage systems (ESSs) are installed on the main branches of the microgrids to compensate for the load-supply mismatch. However, their state of charge (SoC) level needs to be balanced to guarantee the continuous operation of the microgrid in case of RES unavailability. This paper proposes a distributed forecast-based consensus control strategy for DC microgrids that balances the SoC levels of ESSs. By using the load-supply forecast of each branch, the microgrid operational continuity is increased while the voltage is stabilized. These objectives are achieved by prioritized (dis)charging of ESSs based on the RES availability and load forecast. Each branch controller integrates a load forecasting unit based on long short-term memory (LSTM) deep neural network that adaptively adjusts the (dis)charging rate of the ESSs to increase the microgrid endurance in the event of temporary generation insufficiencies. Furthermore, due to the large training data requirements of the LSTM models, distributed extended Kalman filter algorithm is used to improve the learning convergence time. The performance of the proposed strategy is evaluated on an experimental 380V DC microgrid hardware-in-the-loop test-bench and the results confirm the achievement of the controller objectives.

**Index Terms**—two times step ahead (2TSA), DC microgrid, distributed consensus control, forecast based control, LSTM, SoC balancing.

## NOMENCLATURE

$v_i^{ref}$	set-point voltage of the DC-DC converter.
$v_{mg}$	nominal voltage of the microgrid.
$r_i^{drp}$	virtual droop resistance in the primary layer.
$i_i$	output current of the DC-DC converter.
$\Delta v$	maximum acceptable deviation of the voltage.
$P_{max}$	maximum power of the converter.
$v_{min}$	minimum acceptable microgrid voltage.
$T_s$	switching period of the DC-DC converter.
$H_i^{vol}$	DC-DC converter internal loop transfer function.
$H_i^{vcl}$	DC-DC converter closed-loop transfer function.
$E_i^{max}$	capacity of energy storage system $i$ .
$v^{ref}$	voltage set-point for the DC-DC converter.
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	communication graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$ .
$N$	number of nodes in the communication graph.
$a_{ij}$	weight of the edge between two nodes in $\mathcal{E}$ .
$\mathcal{A}$	adjacency matrix of graph $\mathcal{G}$ .
$d_i$	in-degree value of node $i$ in $\mathcal{G}$ .
$\mathcal{D}$	in-degree matrix of graph $\mathcal{G}$ .
$\bar{v}$	consensus voltage vector of the nodes.
$\mathbf{G}_{avg}$	transfer function matrix of consensus protocol.

$\mathbb{I}_N$	identity matrix.
$Q$	steady state matrix of $\mathbf{G}_{avg}$ .
$\delta v_i^{soc}$	SoC balancing correction term.
$\delta v_i^v$	bus voltage regulating correction term.
$T_{outage}$	ESS supply time after energy source outage.
$P_{gen}^{f1}$	first step generation forecast power.
$E_{gen}^{f1}$	first step generation forecast energy.
$P_{L_i}^{f1}$	first step branch load forecast power.
$E_{L_i}^{f1}$	first step branch load forecast energy.
$E_{diff}^{f1}$	load/generation energy difference for the first step.
$P_{L_i}^{f2}$	second step branch load forecast power.
$E_{L_i}^{f2}$	second step branch load forecast energy.
$E_L^{f2}$	sum of the branches load forecast energy.
$C_{L_i}$	load contribution factor of the branch.
$p_{i,t}$	time series data set for forecasting.
$X_{i,t}$	time series instants for training the LSTM network.
$m_t$	number of ensembles in $X_{i,t}$ used for training.
$\Omega(\cdot)$	non-linear mapping of LSTM neuron.
$\Theta(\cdot)$	mean pooling function.
$\theta_{i,t}$	the neural network weight vector.
$\varepsilon_{i,t}$	observation error of the LSTM network output.

## I. INTRODUCTION

**R**enewable energy sources (RES) such as wind turbines (WT) and photovoltaic (PV) panels are the main subjects of research due to the environmental issues and scarcity of fossil fuels. However, the intermittent characteristic of RES results in an imbalance between the power sources and the loads, which affects the delivered power quality [1], [2]. For the better management of the power quality and security issues, energy storage systems (ESSs) are deployed in the microgrids to compensate for the temporary mismatch of supply and demand. Furthermore, in rural areas, where there is no connection to the main grid, ESSs such as batteries are deployed in large quantities as a solution for temporary power stabilization during RES unavailability. ESSs have also been used for critical loading, reliable reserve, and peak load support [3].

The currently developed control systems are mainly concerned with the real-time operation of the microgrid to compensate for the disturbances. Based on this approach, the unit commitment and energy management tasks are processed centrally in the tertiary control layer of the microgrid, which relies on a separate communication network. This separation

between the real-time control layer and the energy management layer limits the resolution of the optimization problems that could be solved if these layers were integrated. As an example, a single day-ahead energy forecast requires the collection of data from smart meters for the previous days and running the computationally intensive algorithms centrally to predict the load. Considering the growing number of smart meters and the amount of data required, this leads to major scalability problems in the tertiary layer management, not to mention the privacy and communication issues.

Distribution of the forecasting tasks to the local controllers has been recently proposed due to advances in the area of edge computing and distributed control systems, which has enlightened new approaches to solving the energy management issues in microgrids [4], [5]. For a distributed microgrid with multiple sources, distributed control architecture is a natural choice compared to the current centralized supervisory control and data acquisition (SCADA) based approaches. The main advantages of distributed controllers are: 1) increased reliability against controller failures; 2) distribution of computation complexity, and 3) robustness in the control system [6]. In the microgrids, distributed control and estimation are mainly implemented in the secondary layer, due to the distributed nature of the RES, and the limitations of the communication network. Optimal neighbor data sharing and multi-agent consensus protocols are the problems of interest in the proposed distributed strategies [6], [7]. Among the available consensus protocols in the multi-agent systems, distributed average consensus (DAC) is the commonly used one, in which the agents agree on the average value of their shared variables from an initial condition [1]. There are a number of important applications for the distributed control systems, such as in power systems [8], industrial automation [9], situational awareness [10], drones control [11], and in self-driving vehicles [12]. Decentralized control approaches are also reported in several works such as [13]–[16]. They are mainly categorized into two types, virtual resistance control [13] and impedance control [16]. Virtual resistance droop control can only be implemented in homogeneous ESSs, without the capability to allocate different frequency components of loads. For example in [13], a decentralized output constrained control algorithm is proposed for single-bus DC microgrids. In [15], the authors have proposed a decentralized controller which has removed the need for an accurate model of the DC microgrid. However, virtual impedance control is able to assign different frequency components of the loads to specific ESSs [14], [16]. The majority of the virtual impedance control methods are based on filters such as works in [17]. Although decentralized methods can solve the real-time control problems in a microgrid, they can not be used in energy forecasting applications, which require a reliable communication network.

Because each ESS can act as both a source or a load in its charging and discharging modes, it is needed to be integrated as part of the microgrid unit commitment solution. Energy management of the ESSs during the microgrid operation is mainly concerned with the stabilization of the state of charge (SoC) levels at different times of the day. In a microgrid with RES, the load and generation profiles are the main players

affecting the SoC of ESSs during the microgrid operation, therefore, load forecasting is an essential step throughout the design of the microgrid control strategy and its energy management plan.

The main approach for ESS management is based on keeping the ESSs fully charged to respond to supply failure, disregarding the load behavior and generation forecast [18]. For centralized small size microgrids, this approach works well, as the ESSs are managed centrally [18]. However, for distributed microgrids with low inertia of the operating point and the intermittent behavior of RES, ESSs are used to establish voltage stability and accurate SoC balancing [3], [19]. This needs a multi-objective control system to provide the operating point stability, and optimal reserve endurance when the distributed generations (DGs) are not available. The multi-objective control problem must be resolved in real-time, considering the scalability of the distributed system. Additionally, since many RES such as PV panels and fuel cells (FC) naturally generate DC, they can easily get interfaced to a DC system [20]. DC microgrids, comparing to the AC ones, require fewer interfacing circuits and also eliminates reactive power and frequency constraints [21], [22]. These fewer constraints lead to a simpler control system with energy losses reduction [23].

Based on the mentioned requirements analysis for the DC microgrids, a robust distributed control system is proposed in this paper, which stabilizes the bus voltages of the microgrid meanwhile balancing the SoC levels of ESSs adaptive to the load and generation forecast of each main branch. Recently, deep neural network models have become a feasible solution for energy forecasting. Deep learning mainly refers to multiple layers of neural networks being stacked, as opposed to shallow learning, relying on stochastic optimization algorithms for training. Several layers provide different abstraction levels that can improve learning performance. In the proposed models, the long short-term memory (LSTM) recurrent neural network (RNN) has unique capabilities for time series sequence [24], as introduced by Hochreiter and Schmidhuber [25]. This has led to innovations in many areas such as speech recognition, image captioning, and dynamic system modeling [26]. Long-term load forecasting is used in power system infrastructure planning, while short-term load forecasting is mainly used for online real-time control of the microgrid operations [27]. Due to the power of LSTM networks for time-series forecasting problems, it has been used in the proposed architecture to implement the distributed forecasting task in the local controllers. The model has been optimally realized using Tensorflow Lite from Google. Furthermore, due to the large training data requirements of the LSTM models, the distributed extended Kalman filter (DEKF) learning algorithm is used to improve the prediction convergence time.

There is a considerable amount of research in the area of short-term load and generation forecasting. In [28], the autoregressive integrated moving average (ARIMA) model is proposed for intraday load forecasting. In [28], radial basis function (RBF) neural network is used for the short-term load forecasting. Authors in [29] combined the RBF neural network with the adaptive neural fuzzy inference system (ANFIS)

to adjust the prediction by using the real-time electricity price. In [30], the short-term day-ahead forecasting problem is addressed based on a grid method combined with back-propagation (BP) training of RBF neural networks. Authors in [31] also proposed a neural network-based predictor for very short-term load forecasting. The approach considers the load values of the current and previous time steps as the input to predict the load value at the coming time step. In [32], an ensemble of extreme learning machines (ELMs) is used to learn and forecast the total load of the Australian national energy market. The proposed methodology not only made use of the supreme ELM learning efficiency for self-adaptive learning but also used the ensemble structure to mitigate the instability of the forecasts. The  $k$ -nearest neighbor (KNN) algorithm is also reported to be successful for load forecasting in [30]. KNN is a widely used approach due to its computational simplicity, however, training requires considerable feature extraction work. Authors in [33] proposed a dedicated input selection scheme to work with the hybrid forecasting framework using wavelet transformation and Bayesian neural network. In the pioneering work of [27], the LSTM model is proposed for short term residential load forecasting, however, the microgrid stabilization and the effect of energy forecast on it was not considered.

The proposed control strategy has three components: 1) primary layer virtual resistance droop control, 2) secondary layer distributed consensus control for voltage offset correction, and SoC balancing of ESSs, 3) short-term load forecast LSTM distributed learning adapter for predictive (dis)charging of ESSs.

The main contributions of this work can be summarized as the proposal of the following items:

- A distributed consensus control system that stabilizes the bus voltages co-designed to balance the SoC levels of ESSs in a DC microgrid;
- Prioritized (dis)charging controller for ESSs based on short-term energy forecast of the branches to achieve higher endurability for the DC microgrid;
- Integration of the load forecasting unit in the secondary control layer of the microgrid based on LSTM neural network with DEKF learning algorithm;

The rest of this paper is organized as follows. Section II reviews the state of art in energy forecasting methods and shows the advantage of LSTM models for this purpose. Section III introduces the components of DC microgrids. The proposed control strategy is discussed in Section IV in detail. Section V provides information about the proposed load forecast adaptation method and the prioritized (dis)charging logic. The training of the distributed LSTM forecasting models is then described in Section VI. The experimental results for the case study microgrid are demonstrated in Section VII. Finally, the paper is concluded in Section VIII.

## II. ENERGY FORECASTING METHODS IN DIFFERENT HORIZONS

Different factors affect the forecasting performance thus making such prediction a sophisticated process. Among these

factors, the forecasting horizon is the most important decision parameter, which is the future time duration for output forecasting [40]. The main types of forecasting horizons introduced in the literature can be categorized as [41]: very short-term, short-term, medium-term, and long-term.

1) *Very short-term forecasting*: Very short-term forecasting is used in power system and smart grid planning with the prediction period from seconds to several hours min [42].

2) *Short-term forecasting*: This is the most common horizon chosen in the electricity market, where decisions comprise of economic load dispatch and power system operation. It is also useful in the control of renewable energy integrated power management systems, therefore in this paper, short-term load forecasting is selected for the ESS SoC balancing problem. Generally, the temporal horizon is between several hours to seven days [40].

3) *Medium-term forecasting*: Medium-term forecasting spans up to a month ahead as being in this category. It is essential for maintenance scheduling of conventional or solar energy integrated power systems consisting of high-end transformers and different types of electro-mechanical machinery [40].

4) *Long term forecasting*: Long-term forecasts predict scenarios for a month to a year [40]. Such a prediction horizon is suitable for long term power generation, transmission, distribution and solar energy rationing [43], as well as seasonal trends prediction.

### A. Energy Forecasting Methods

The collection of energy data over time results in time series. These time series are stochastic by their nature, therefore deterministic model-based methods such as model predictive control (MPC) lack the suitable performance required. Time series provides statistical information to foresee the nature of the quantified element. These observations are generally recorded overtime at successive points in regular intervals [44]. The main established time series prediction techniques are [41], [45]: exponential smoothing, autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA), RNN, radial basis function neural network (RBFNN), and LSTM neural network.

1) *Exponential smoothing*: The exponential smoothing method or exponentially weighted moving average (EWMA) is a technique that adopts exponential window function for statistical analysis of historical time series data to make predictions. The technique was first formulated by Brown [44] and has since seen many applications. Generally, it allocates an unequal set of weights over equal weights to historical observations, thereby exponentially reducing the data from the most recent to the most distant data points.

2) *Autoregressive moving average model (ARMA)*: ARMA is a time series statistical analysis frequently used in forecasting. The model has been evaluated by many researchers in different applications of forecasting (solar and wind forecasting) and it has consistently performed with good prediction accuracy. The model incorporates two polynomials: AR and MA for forecasting from historical data [35].

TABLE I  
COMPARISON OF THE ENERGY FORECASTING METHODS IN TERMS OF COMPUTATIONAL COMPLEXITY, DATA REQUIREMENTS, OPTIMIZER OPERATION, AND THE EVALUATED FORECAST HORIZON.

Forecasting Method	Computational Complexity	Data Requirements	Optimizer Operation	Evaluated Forecast Horizon
Exponential Smoothing [34]	Very Low	Equally Spaced Samples	Centralized	hourly
ARMA [35]	Low	Equally Spaced Samples	Centralized	monthly
ARIMA [36], [37]	Low	Equally Spaced Samples	Centralized	hourly
MLPNN [38]	Medium	High Resolution, Noise Free	Centralized	monthly
RNN [24]	High	High Resolution, Noise Free	Centralized	daily
RBFFNN [39]	Very High	High Resolution	Distributed	monthly
LSTM [27]	High	High Resolution	Distributed	weekly

3) *Autoregressive integrated moving average (ARIMA)*: ARIMA is also known as the Box-Jenkins model and was developed by George Box and Gwilym Jenkins in 1976 [37]. ARIMA model is an extended version of ARMA and it is a popular time series analysis technique as it supports a standard level of forecast accuracy for short term horizon. Moreover, this model can clip non-stationary values from the analyzed data. Its structure consists of autoregression (AR), integration (I), and moving average (MA) to evaluate and predict time series characteristics [36].

4) *Multi-layer perceptron neural network (MLPNN)*: Many researchers treat the MLPNN model as a benchmark [38]. It is a technique for elementary and effective artificial neural network (ANN) approach to designing and prediction. It is so powerful that this network is used in universal approximation, nonlinear modeling, and complex problems that cannot be solved by an ordinary single-layer neural network [38]. Generally, MLP is a composite of three or more layers of incoherently activating nodes. Therefore, it can correlate the input and output relationship through learning.

5) *Recurrent neural network (RNN)*: RNN is a class of ANN that can learn and process different relationships as well as computational structures. This network provisionally relies on time series data by the feedback system to inherit the previous time step values; demonstrating temporal dynamic characteristics. The model has a simple structure with a built-in feedback loop, which allows it to act as a forecasting engine. RNN output of the concerned neural layer is summed with the next input vector and fed back into the same layer which is the only layer in the entire network. The applications are versatile ranging from speech recognition to driverless cars.

6) *Radial basis function neural network (RBFFNN)*: RBFFNN is a quicker and better approach to machine learning than other ANN approaches. Hence, it is used in approximation, time series prediction, classification, and system control [39]. The structure uses radial basis functions as activation functions. This network generally has two layers. The characteristics are merged with the radial basis activation function in the first layer, and then the output of the first layer is used to compute the same output in the next time step.

7) *Long short-term memory(LSTM) neural network*: The LSTM neural network is a type of RNN. As mentioned, RNNs use previous time events to inform the later ones. RNNs work well if the problem requires only recent information to perform the present task. If the problem requires long term

dependencies, RNN doesn't provide the required performance. The LSTM was designed to learn long term dependencies. It remembers the information for long periods. LSTM was introduced by S Hochreiter, J Schmidhuber in 1997 [25].

Due to the following advantages of LSTM deep learning model, it was chosen in this work to solve the forecasting problem:

- LSTM models are proven to have superior performance when there are long term dependencies in the forecasted times series.
- As LSTM models are deep neural networks, the feature analysis is automatically done through the learning process, therefore, it has an easier application and less error-prone.
- LSTM neural networks support distributed training using the DEKF training algorithm. Therefore it enables its integration with distributed control tasks that work over the neighbor communication system.

The comparison of different forecasting methods is summarized in Table I. We have chosen the short-term forecasting horizon due to the following reasons:

- The secondary and tertiary control systems for microgrids are usually operating at a high speed, therefore, the energy forecasting horizon can not be very long, otherwise, the adaptiveness of the control parameters to the transient in energy supply and demand would be lost. Therefore, from the control system performance viewpoint, either short-term and very short-term forecasting horizons should be considered.
- The (dis)charging cycles of energy storage systems in a microgrid take place with a time duration of several hours to several days. This is because their capacity is often large and for maintenance and storage health reasons, their (dis)charging currents are limited to increase the lifetime of the energy storage systems.
- The renewable energy sources such as wind turbines and PV panels are intermittent sources dependant on the weather condition, temperate and other environmental parameters. Most of these environmental parameters exhibit dynamics in several hours to a few days. Therefore, it is required to prioritise the (dis)charging of the energy storage systems based on the forecasts for the same time duration.

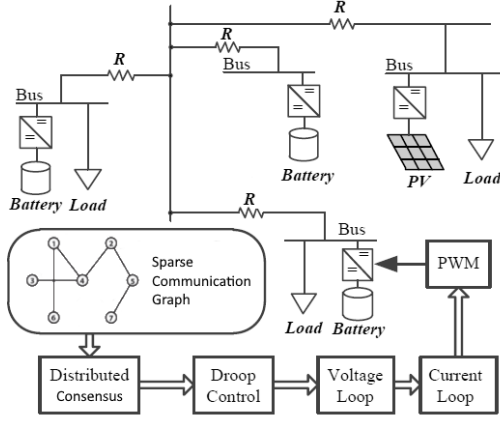


Fig. 1. Schematic of a solar DC microgrid with batteries and PV panels.

### III. RADIAL DC MICROGRID COMPONENTS AND CONFIGURATION

The schematic of a solar DC microgrid is shown in Fig. 1, which consists of battery ESSs, PV panels, and loads. PV panels are connected to the main bus through a voltage controllable boost converter, working in maximum power point tracking (MPPT) mode. The DC loads are connected to the bus through a voltage controllable buck converter and are considered as constant power loads (CPLs). The battery ESSs are connected to the system by DC-DC bidirectional converters and are used to compensate for power mismatch between PVs and loads to regulate the bus voltage.

Conventionally, droop control provides proportional power-sharing among multiple ESSs. The primary control for ESSs satisfies:

$$v_i^{ref} = v_{mg} - r_i^{drp} i_i \quad (1)$$

where  $v_i^{ref}$  is the set-point voltage of the DC-DC converter,  $v_{mg}$  is the nominal voltage of the microgrid,  $r_i^{drp}$  is the virtual droop resistor, and  $i_i$  is the output current of the DC-DC converter. To achieve proportional power-sharing, the virtual resistance is designed based on the following equation for an individual DG:

$$r_i^{drp} = \frac{\Delta v}{P_{max}/v_{min}} \quad (2)$$

where  $\Delta v$  is the maximum acceptable deviation of the microgrid voltage,  $P_{max}$  is the maximum power of the converter, and  $v_{min}$  is the minimum acceptable microgrid voltage. Commonly a 5% deviation is an acceptable threshold [1].

The secondary layer effect on the voltage output of the converter is then adjusted by introducing voltage correction terms for each control objective to equation (1). In the next section, the structure of the secondary layer control is introduced.

### IV. SECONDARY LAYER CONSENSUS CONTROL

In this section, the mathematical model of the secondary control layer for the DC microgrid is developed. First, the voltage correction terms for the control objectives, voltage regulation, and SoC balancing, are introduced, along with the

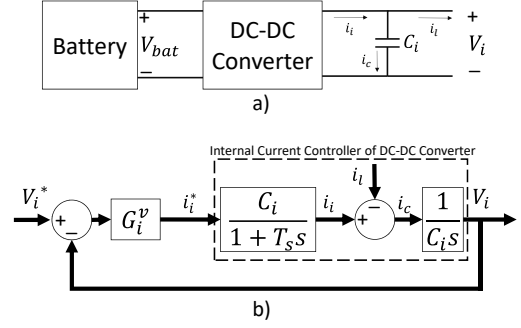


Fig. 2. DC-DC converter model for interfacing batteries to DC microgrid a) converter circuit, b) block diagram of the internal controller.

related SoC dynamics for the batteries. Second, the average consensus protocol for bus voltage regulation is developed. Third, the cooperative control for SoC balancing is developed, and the small-signal stability analysis is described. In the end, the proposed online deep learning framework for load forecast based secondary layer control adjustment is introduced.

#### A. Modified droop control for battery systems

DC-DC converters operate at a high PWM (pulse width modulation) switching frequency with at least one switching interval delay (i.e.,  $T_s$ ) in the current control (CC) mode. In Fig. 2, the diagram of the converter interfacing batteries to the DC microgrid is shown, in which the bus voltage regulation dynamics is designed as an outer-loop between the output voltage of the battery  $v_i^{ref}$ , and the local bus voltage  $v_i$ . The transfer function for the internal loop is given by  $H_i^{vol}$ .

$$H_i^{vcl} = \frac{H_i^{vol}}{1 + H_i^{vol}}, \quad H_i^{vol} = \frac{G_i^v}{sC_i(T_s s + 1)} \quad (3)$$

Therefore, the local bus voltage closed-loop transfer function of the DC microgrid is given by:

$$V = \mathbf{H}^{vcl} V^{ref} \quad (4)$$

A first-order model is used for the battery per-unit energy level charging and discharging:

$$\dot{SoC}_i = -\frac{v_i i_i}{E_i^{max}} \quad (5)$$

where  $E_i^{max}$  is the battery charge capacity of the ES system,  $v_i$  is the bus voltage and  $i_i$  is the converter current. It is assumed that converter loss is negligible.

In the secondary layer control, the voltage reference ( $v_i^{ref}$ ) for the DC-DC converter is set by the modified droop control with two correction terms for each bus controller with battery ESSs, as the following:

$$v_i^{ref} = v_{mg} - r_i^{drp} i_i + \delta v_i^v + \delta v_i^{soc} \quad (6)$$

where  $\delta v_i^{soc}$  is the SoC balancing correction term and  $\delta v_i^v$  is the bus voltage regulating correction term.

### B. Average consensus for voltage regulation

The secondary layer ES control agents are connected via a sparse communication graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ , with nodes  $\mathcal{V} = \{1, \dots, N\}$  and edges  $\mathcal{E}$ . Each graph node represents an ES system and the graph edges represent communication links between them.  $(i, j) \in \mathcal{E}$  if there is an information flow between node  $i$  and node  $j$ . The neighbours of node  $i$  are given by  $\mathcal{N}_i$ , where  $j \in \mathcal{N}_i$ , if  $(i, j) \in \mathcal{E}$ . The graph adjacency matrix is given by

$$\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}, a_{ij} = \begin{cases} 1, (j, i) \in \mathcal{E} \\ 0, \text{ otherwise} \end{cases} \quad (7)$$

The communication graph Laplacian matrix is given by  $\mathbf{L} = \mathbf{D} - \mathcal{A}$ , where  $\mathbf{D} = \text{diag}\{d_i\}$ , and  $d_i = \sum_{j=1}^N a_{ij}$  is the in-degree of the communication network.  $\mathcal{G}$  describes only the communication network between battery ES controller systems. The graph is bidirectional, meaning that each battery ES system can both receive and send information on the same link. Through the communication links, the local consensus controller computes the average values of state variables based on the information from its neighbors  $j \in \mathcal{N}_i$ . Assuming that the microgrid contains  $N$  ES systems, including  $N$  battery systems. The average consensus protocol of each battery local bus voltage through the graph  $\mathcal{G}$  is:

$$\bar{v}_i = v_i + \int \sum_{j \in \mathcal{N}_i} a_{ij} (\bar{v}_j - \bar{v}_i) dt \quad (8)$$

where  $\bar{v}_i$  is the local bus voltage estimation. Therefore, the  $\{\bar{v}_i\}$  are exchanged in the communication network between battery controllers for local bus voltage average consensus protocol. The global dynamics of the distributed average consensus protocol are given by:

$$\dot{\bar{\mathbf{v}}} = \dot{\mathbf{v}} - \mathbf{L}\bar{\mathbf{v}} \quad (9)$$

Applying the Laplace transform yields the following transfer function matrix for the average consensus protocol [1]:

$$\mathbf{G}_{\text{avg}} = \frac{\bar{\mathbf{V}}}{\mathbf{V}} = \frac{s}{(s\mathbf{I}_N + \mathbf{L})} \quad (10)$$

$\bar{\mathbf{V}}$  and  $\mathbf{V}$  are the Laplace transforms of  $\bar{v}$  and  $v$ , respectively.

For a balanced communication graph with a spanning tree, the steady-state gain of the average consensus protocol is given by the averaging matrix:

$$\lim_{s \rightarrow 0} \mathbf{G}_{\text{avg}} = \mathbf{Q}, \text{ where } [Q]_{ij} = \frac{1}{N} \quad (11)$$

The final value theorem shows that for a vector of step inputs, the elements of  $\bar{\mathbf{x}}(t)$  converge to the global average of the steady-state values  $\mathbf{v}^{ss}$ :

$$\lim_{t \rightarrow \infty} \bar{\mathbf{v}}(t) = \lim_{s \rightarrow 0} \mathbf{G}_{\text{avg}} \lim_{t \rightarrow \infty} s\mathbf{v} = \mathbf{Q}\mathbf{V}^{ss} = \mathbf{v}^{ss} \quad (12)$$

To maintain the average battery local bus voltage at the rated value  $v_{mg}$ , a PI controller is used. Then, the local bus voltage correction term in (6) is then computed as:

$$\delta v_{1i} = H_i (v_{mg} - \bar{v}_i), H_i = k_{Pi}^{\bar{v}} + \frac{k_{Ii}^{\bar{v}}}{s} \quad (13)$$

where  $H_i$  is the PI controller,  $k_{Pi}^{\bar{v}}$  and  $k_{Ii}^{\bar{v}}$  are proportional and integral PI gains, respectively. This PI controller regulates the average value of the local bus voltages of the battery DC-DC converter output to the rated microgrid voltage. Thus, the bus voltage offset from the primary droop control is compensated.

Another consensus control balances the SoC level among the batteries. The data of  $\{SoC_i\}$  are exchanged between neighboring ESSs. The correction term  $\delta v_{2i}^b$  in (5) is defined as:

$$\delta v_{2i}^b = G_i^b \sum_{j \in \mathcal{N}_i} a_{ij} (SoC_j - SoC_i), G_i^b = k_{Pi}^{SoC} \quad (14)$$

where  $k_{Pi}^{SoC}$  is the control gain of the SoC cooperative balancing control.

### C. Small signal stability analysis of the control system

The output currents of the ES converters is derived from multiplying the bus voltages with the bus admittance matrix, constructed based on the line and load impedances:

$$\mathbf{I} = \mathbf{Y}\mathbf{V} \quad (15)$$

The total SoC level dynamics is summarized in vector form based on (5):

$$\mathbf{E} = \mathbf{M}\mathbf{Y}\mathbf{V}, \quad \mathbf{M} = \text{diag}\left\{-\frac{v^{mg}}{E_i^{max}s}\right\} \quad (16)$$

The total multi-variable form of the closed-loop secondary and primary control system dynamics is then described as the following, derived from equations (6), (13), (14), and (16):

$$\mathbf{V} = \left( (\mathbf{H}^{\text{vcl}})^{-1} + (\mathbf{G}^b \mathbf{L} \mathbf{M} + \mathbf{r}_{drp}) \mathbf{Y} + \mathbf{H} \mathbf{G}_{\text{avg}} \right)^{-1} ((\mathbf{H} + \mathbf{I}_N) \mathbf{v}_{mg}) \quad (17)$$

where  $\mathbf{G}_{\text{avg}}$  is the transfer function of the voltage average consensus protocol;  $\mathbf{v}_{mg}$  is the nominal microgrid voltage,  $\mathbf{Y}_{net}$  is the microgrid grid admittance matrix,  $\mathbf{I}_N$  is an  $N \times N$  identity matrix.

$$\begin{aligned} \mathbf{V} &= [V_1, V_2, \dots, V_p]^T, \mathbf{r}_{drp} = \text{diag}\{r_i^{drp}\} \\ \mathbf{H} &= \text{diag}\{H_i\}, \mathbf{H}^{\text{vcl}} = \text{diag}\{H_i^{\text{vcl}}\} \\ \mathbf{G} &= \text{diag}\{G_i^b\} \end{aligned} \quad (19)$$

To analyze the stability of the total dynamics in (17), it is assumed that the microgrid voltage reference voltage is:

$$\mathbf{v}_{mg} = (v_{mg}/s) \mathbf{1}_N \quad (20)$$

where  $\mathbf{1}_N \in \mathbb{R}^{N \times 1}$  is the vector with all elements equal to one. Using the final value theorem, the steady-state values of the total microgrid dynamics are derived. By defining the steady-state total bus voltage vector,  $\mathbf{v}^{ss}$ , the final value is:

$$\mathbf{v}^{ss} = \lim_{s \rightarrow 0} \left( s (\mathbf{H}^{\text{vcl}})^{-1} + s (\mathbf{G}^b \mathbf{L} \mathbf{M} + \mathbf{r}_{drp}) \mathbf{Y} + s \mathbf{H} \mathbf{G}_{\text{avg}} \right)^{-1} ((s \mathbf{H} + s \mathbf{I}_N) \mathbf{v}_{mg}) \quad (21)$$

Based on the work in [1], it can be shown that:

$$\text{The final steady state value : } \langle \mathbf{v}^{ss} \rangle = v_{mg} \quad (22)$$

## V. PREDICTIVE (DIS)CHARGING OF ESSs BASED ON LOAD/GENERATION FORECAST

The primary and secondary layer control strategies described in the previous sections can maintain the first two control objectives (i.e. voltage offset compensation and balancing SoC of ESSs), without the resiliency considerations in the event of generation disconnection. Now this question must be answered: How to increase the time duration for the microgrid to continue its operation without any generation supply. The answer is in the reliability and resiliency of the microgrid. This time duration is dependant on the SoC of the batteries and the load profile on branches. As the exact instant of time for generation outage is unknown, the mathematical model of the resiliency should be independent of this instant. Therefore, in this paper a load forecast based approach is proposed to increase the endurance time of the microgrid based on the following non-restrictive basic assumptions:

**Assumption 1.** *The microgrid is based on a single bus architecture and has a radial load distribution on branches.*

**Assumption 2.** *The capacities of the installed ESSs on the main branches are known to the forecasting adapter unit of the distributed controllers. By knowing the capacity, the outage time can be computed as  $T_{outage} = 10h$ , which is the time that the ESS supplies the branch on which installed after the supply outage. For example, a 1 kWh storage for a branch with a peak load of 100 W leads to  $T_{outage} = 10h$ .*

**Assumption 3.** *Each ESS distributed controller has access to the load prediction for all of the branches in the microgrid. DGs are connected to the single bus architecture via a branch, therefore, the controllers have access to the generation forecast, additionally.*

The load/generation forecast is predicted based on LSTM deep neural networks due to their advantages, discussed earlier. The amount of training data gets large when all of the controllers broadcast their information to the other controllers, which increases the burden on the communication network significantly. Therefore in the next section, a distributed learning algorithm is proposed that is based on a DEKF learning model with the neighbor communication.

By knowing the load/generation forecast for each branch of the DC microgrid, a dynamic program can be formulated which prioritizes the (dis)charging of ESSs based on the corresponding load forecast of the branch. In this paper a two-time steps ahead (2TSA) programming method is proposed for the dynamic programming of the ESSs. In the first time step, the distributed controllers compute the excess power or supply shortage of distributed generation based on the generation forecast of each branch. Then, the controllers distributively compute the (dis)charging prioritizes based on the second time step prediction and will adjust the SoC balancing distributed controller accordingly. For the simplicity of the presentation, a daily time step is assumed for the results section. In the following, the mathematical model of the prioritizing function is developed.

The variables used in the proposed 2TSA dynamic model are defined in Table II. The programming model consists of

TABLE II  
DESCRIPTION OF THE VARIABLES FOR THE PROPOSED 2TSA DYNAMIC PROGRAMMING MODEL.

Variable	Definition
$P_{gen}^{f1}, E_{gen}^{f1}$	First step generation forecast, power and energy
$P_{L_i}^{f1}, E_{L_i}^{f1}$	First step branch load forecast, power and energy
$E_{diff}^{f1}$	Load/generation energy difference for the first step
$P_{L_i}^{f2}, E_{L_i}^{f2}$	Second step branch load forecast, power and energy
$E_L^{f2} = \sum E_{L_i}^{f2}$	Sum of the branches load forecast energy
$C_{L_i} = \frac{E_{L_i}^{f2}}{E_L^{f2}}$	Load contribution factor of the branch

two steps, excess/supply shortage energy calculation step, and the (dis)charging prioritization based on the load forecast in the second step. In the first step, distributed LSTM models provide the load forecast and the generation of the branches. This information is available at every distributed controller because LSTM training was run distributively, according to the learning framework provided in the next section. Therefore, the variables  $P_{gen}^{f1}$ ,  $E_{gen}^{f1}$ ,  $P_{L_i}^{f1}$ , and  $E_{L_i}^{f1}$  are computed for the first step. Knowing the generation and load energy forecast, the different show how much energy is available for (dis)charging the storage in the first step:

$$E_{diff}^{f1} = E_{gen} - \sum E_{L_i}^{f1} \quad (23)$$

The objective is now to assign this energy difference to the ESSs, to increase the endurability of the microgrid, in the event of generation failure. Therefore, a load contribution factor is defined,  $C_{L_i} = \frac{E_{L_i}^{f2}}{E_L^{f2}}$ , that forms the basis for prioritized (dis)charging of the ESSs. The priorities are ordered from the branch with the highest load contribution to the branch with the lowest load contribution, in the second step:

$$E_{max_i}^{new} = \frac{C_{L_i} \times E_{diff}^{f1} + SoC_i \times E_{max_i}^{old}}{E_{diff}^{f1} + \sum (SoC_i \times E_{max_i}^{old})} \quad (24)$$

The new calculated value for the maximum SoC of the ESS, adapts the parameter  $E_i^{max}$  in equation (5):

$$SoC_i = -\frac{v_i i_i}{E_{max_i}^{new}} \quad (25)$$

Then, the proposed secondary layer SoC balancing controller distributively regulates the balanced charging of ESSs with the new priorities, defined by the  $E_{max_i}^{new}$ . The new outage time,  $T_{outage}$  is re-calculated with the new SoC condition for each branch, according to Assumption 2:

$$T_{outage} = \frac{P_{max_i}}{SoC_i \times E_{max_i}^{new}} \quad (26)$$

Results in Section VII provides the analysis on the prioritized SoC balancing for a case study 380 V DC microgrid.



## VI. DISTRIBUTED LOAD FORECAST BASED ON LSTM DEEP NEURAL NETWORKS

At each node  $i$ , the load forecast LSTM predictor sequentially receives  $\{p_{i,t}\}_{t \geq 1}$ ,  $p_{i,t} \in \mathbb{R}$ , and matrices,  $\{X_{i,t}\}_{t \geq 1}$ , defined as  $X_{i,t} = \begin{bmatrix} x_{i,t}^{(1)} & x_{i,t}^{(2)} & \dots & x_{i,t}^{(m_t)} \end{bmatrix}$ , where  $x_{i,t}^{(l)} \in \mathbb{R}^p, \forall l \in \{1, 2, \dots, m_t\}$  and  $m_t \in \mathbb{Z}^+$  is the number of columns in  $X_{i,t}$ .  $p_{i,t}$  is the sampled load power/generation power, and  $X_{i,t}$  is the sampling timestamps used for training the LSTM network, which changes with time  $t$ , as microgrid operation continues. In this paper, a daily sampling timestamp set  $(\{0, 1, \dots, 24\})$  is used, but a longer or shorter training set can also be used, for example weekly or monthly. In this network, each node  $i$  aims to learn a certain relation between the desired value  $p_{i,t}$  and  $X_{i,t}$  signals. After receiving  $X_{i,t}$  and  $p_{i,t}$  samples, each node  $i$  first updates its belief about the relation and then exchanges an updated information with its neighbours. This information exchange helps faster training and results in a more accurate model, due to collective training of the multi-agent system. After receiving  $X_{i,t}$ , each node  $i$  estimates the next signal  $p_{i,t+1}$  as  $\hat{p}_{i,t+1}$ . Based on  $p_{i,t+1}$ , each node  $i$  calculates the loss function  $loss(p_{i,t+1}, \hat{p}_{i,t+1})$  at time instance  $t + 1$ .

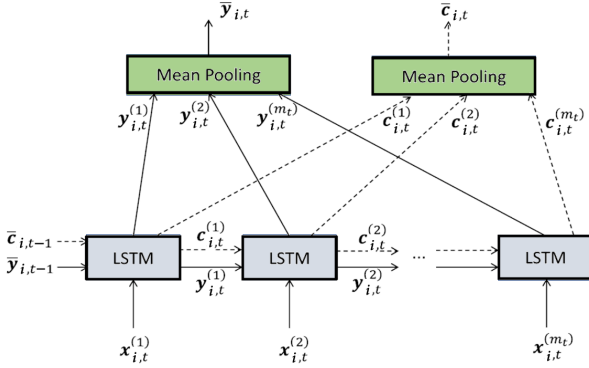


Fig. 4. Distributed LSTM neural network structure for each node. LSTM units learn the time series sequences with their internal recurrent feedback path and memories.

The data flow chart in the layers of the LSTM neural network is illustrated in Fig. 3. The input layer consists of convolution and pooling layers for feature extraction. At first, the input data are injected as multidimensional time-series into the neural network. The convolution layer applies different filters to the input data and the pooling layer compresses the output of the previous steps using mean pooling. LSTM neurons learn the history output from previous layers by their internal recurrent loops. Finally, the output is derived for the predicted time series in the next time step. Mean pooling or average pooling layer performs down-sampling by dividing the input into rectangular pooling regions and computing the average values of each region in the input matrix.

Each node  $i$  generates an estimate  $\hat{p}_{i,t}$  using the LSTM architecture. The LSTM network architecture without peephole connections is used in this paper. The input  $X_{i,t}$  is first fed to the LSTM architecture as illustrated in Fig. 4, where the internal equations of the neural network are given in [47].

Given the outputs of LSTM for each column of  $X_{i,t}$ , shown in Fig. 4, the estimate for each node  $i$  is generated as:

$$\hat{p}_{i,t} = w_{i,t}^T \bar{y}_{i,t} \quad (27)$$

where  $w_{k,t} \in \mathbb{R}^n$  is a vector of the regression coefficients and  $\bar{y}_{k,t} \in \mathbb{R}^n$  is a vector obtained by taking average of the LSTM outputs for each column of  $X_{k,t}$ , i.e., known as the mean pooling method, as described in Fig. 4.

By simplifying this model in Fig. 4 with the LSTM equations in [47], the following nonlinear state space form for each node  $i$  will be derived:

$$\begin{aligned} \bar{c}_{i,t} &= \Omega(\bar{c}_{i,t-1}, X_{i,t}, \bar{y}_{i,t-1}) \\ \bar{y}_{i,t} &= \Theta(\bar{c}_{i,t}, X_{i,t}, \bar{y}_{i,t-1}) \\ \theta_{i,t} &= \theta_{i,t-1} \\ p_{i,t} &= w_{i,t}^T \bar{y}_{i,t} + \varepsilon_{i,t} \end{aligned} \quad (28)$$

where  $\Omega(\cdot)$  and  $\Theta(\cdot)$  represent the nonlinear mappings performed by the consecutive LSTM units and the mean pooling operation as in Fig. 4, and  $\theta_{i,t} \in \mathbb{R}^{n_\theta}$  is the neural network weight vector. Furthermore,  $\varepsilon_{i,t}$  represents the error in observations and it is a zero mean Gaussian random variable with variance  $R_{i,t}$ . The DEKF algorithm is used for distributed training the LSTM network. DEKF was proposed for distributed learning of neural networks and has a higher learning performance and a faster convergence rate comparing to the commonly used stochastic gradient descent (SGD) method. The details of the DEKF training method are discussed in [47].

Due to the choice of short-term energy forecasting horizon, the predictions models are only valid for the same time duration chosen for the horizon. Therefore to have a robust control system that is able to continuously stabilize the microgrid and prioritise the (dis)charging of the energy storage systems, the prediction models should be built regularly at the end of the last forecasting horizon. This operation consists of data collection and re-training of the LSTM deep neural models, which provides robustness against the prediction model uncertainties due to the short-time forecasting horizon.



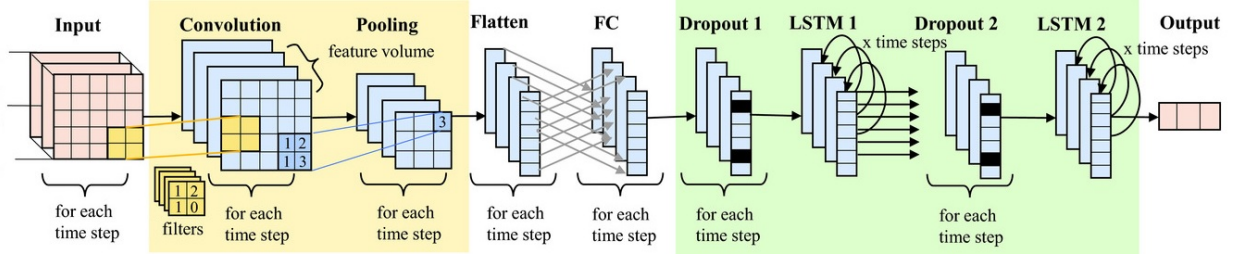


Fig. 3. Data processing flow chart in the LSTM neural network [46]. The input layer consists of convolution and pooling layers for feature extraction. At first the input data are injected as multidimensional time-series into the neural network. The convolution layer applies different filters to the input data and the pooling layer compresses the output of the previous steps using mean pooling. LSTM neurons learn the history output from previous layers by their internal recurrent loops. Finally, the output is derived for the predicted time series in the next time step.

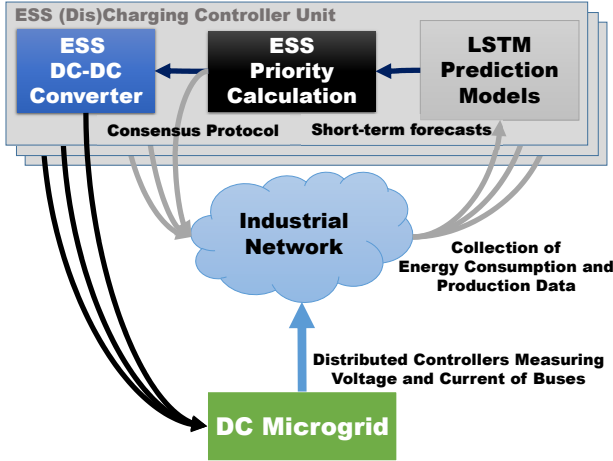


Fig. 5. Summary of the operations in the proposed distributed forecast-based consensus control system. The ESS controller unit consists of three components: LSTM prediction models, ESS (dis)charging priority calculation, and the ESS interfacing DC-DC converter.

## VII. EXPERIMENTAL RESULTS

To validate the performance of the proposed control strategy, we have used a 380 V data center DC microgrid real-time HIL test bench with 5 buses as the main branches. The test bench consists of the dSPACE real-time simulators (SCALEXIO and MicrolabBox), and the Internet of things (IoT) embedded controllers for each bus of the case study, which run the forecast based control tasks. Load and generation forecast models are optimized using "Tensorflow lite for microcontrollers" deep learning framework released by Google<sup>TM</sup>. Load forecasting models are developed in Python programming language, and Keras deep learning interface is used for online training of the LSTM models using the DEKF algorithm.

In this setup, a Raspberry Pi computer with the supporting communication modules for the gateway operation is used. This gateway receives the data from the measurement nodes via MQTT (Message Queuing Telemetry Transport) protocol. Thingsboard<sup>®</sup> software implements the MQTT broker, which is used for data archiving and processing. By using the mentioned protocols and devices, the microgrid control system cost becomes considerably cheaper than other smart metering technologies such as IEC 61850. The architecture of the test

bench and the communication graph is shown in Fig. 7. The laboratory setup is also shown in Fig. 6.

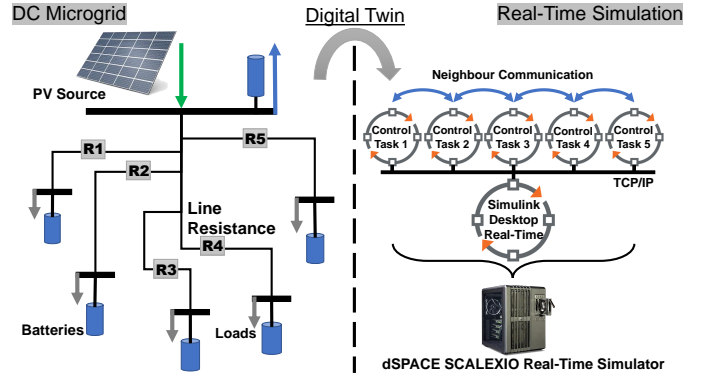


Fig. 7. Real-time simulation architecture of the distributed controllers with neighbor communication graph, and the 380 V DC microgrid.

On each bus, a 400 Wh battery ESS is installed with the nominal maximum power at 400 W that leads to nominal  $T_{outage} = 1h$ , according to the calculation in Assumption (2). Furthermore, to show the effectiveness of the prioritized (dis)charging method, the maximum load of each bus is different as shown in Table III. Also, the initial SoC of ESSs is set to different values (i.e. 80%, 90%, and 100%) as shown in the same table. Line and controller parameters are also provided in Table IV.

The microgrid is emulated for three consecutive days using the real residential sample PV data from 11 July to 13 July 2014 of the UK Power Network [48]. The PV generation forecasting LSTM model is trained by the whole month data of July 2014 from the same source.

For this experiment, the whole month of July 2014 is used for the training, testing, and validation. The forecasting horizon was decided to be three consecutive days, and the dataset is divided into 80%, 10%, and 10% for the LSTM model training, testing and, validation purposes, respectively, and the data selection for each subset was done randomly.

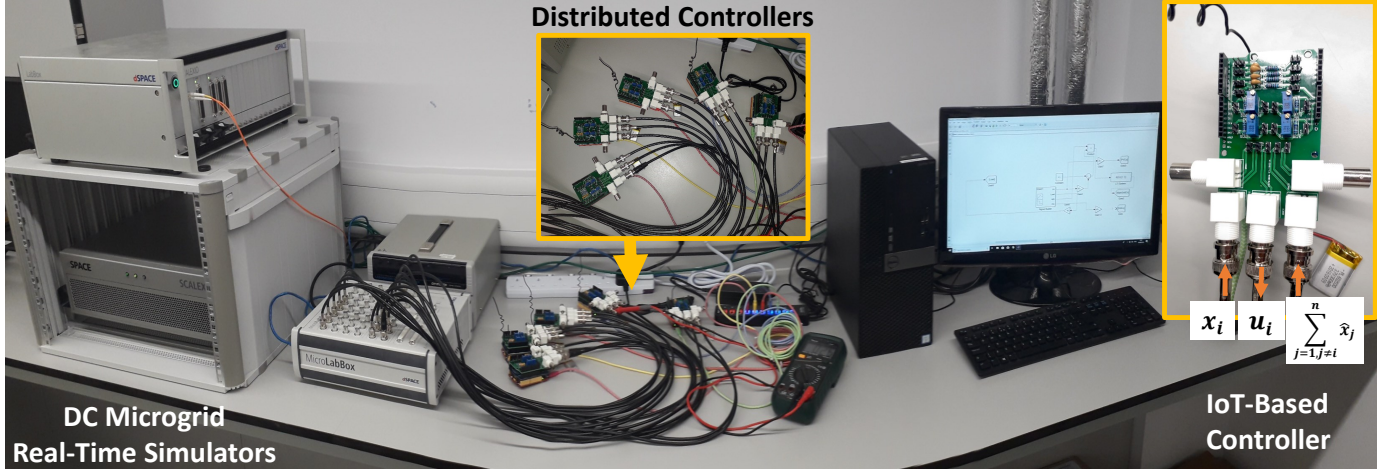


Fig. 6. Laboratory HIL test bench used for the performance analysis of the proposed forecast based distributed control strategy. The test bench consists of dSPACE real-time simulators (SCALEXIO and MicroLabBox), and IoT embedded controllers for each bus of the case study, which run the forecast based control tasks. Load and generation forecast models are optimized using "Tensorflow lite for microcontrollers" deep learning framework released by Google. The embedded controllers are based on the Seeduino development boards with ATSAMD21G18 32-Bit ARM Cortex M0+ CPU.

TABLE III  
MAXIMUM LOAD POWER OF EACH BUS AND THE INITIAL SOC OF ESSs.

Bus Number	Constant Power Load (W)	Initial SoC (%)
1	110	80%
2	120	100%
3	120	80%
4	130	100%
5	120	90%

The load profile dynamics are generated following the total load profile from the same source to allow testing different transient conditions. For each day, the total load of the buses increases from 20% to 100% and then decreases from 100% to 20%. This allows testing the performance of the proposed method during peak load and low load values in the daily forecast horizon. Fig. 8 shows the PV generation and the load power. To highlight the advantages of the proposed control strategy, the experimental results are derived in two configurations:

- **Configuration A:** Traditional droop control, which is a decentralized method commonly used for comparison.
- **Configuration B:** Proposed forecast based distributed control system using LSTM energy forecasting method.

TABLE IV  
PARAMETERS OF THE DISTRIBUTED CONTROLLERS  
FOR THE HIL SIMULATION.

$R_{dc}$	10 $\Omega$	$k_{P_i}^{SoC}$	5000	$k_{P_i}^v$	500
$L_{dc}$	7 $\mu H$	$r$	0.2533	$k_{I_i}^v$	10

The microgrid is emulated for three consecutive days with the initial  $E_{max_i}$  of 400 Wh for the first day as outlined in Table III. On the third day, the PV source is disconnected and the microgrid continues its operation only by the battery ESSs. The disconnection happens after 12 hours.

In configuration A, only the droop control system stabilizes the microgrid, in which the local droop controller acts based

on local measurements only. The results for configuration A are shown in figures Fig. 9, Fig. 11, and Fig. 13. As can be seen in Fig. 9, the ESSs run out of energy between 14h to 19h on the third day, one by one. This is due to the disconnection of the PV sources at time 12h.

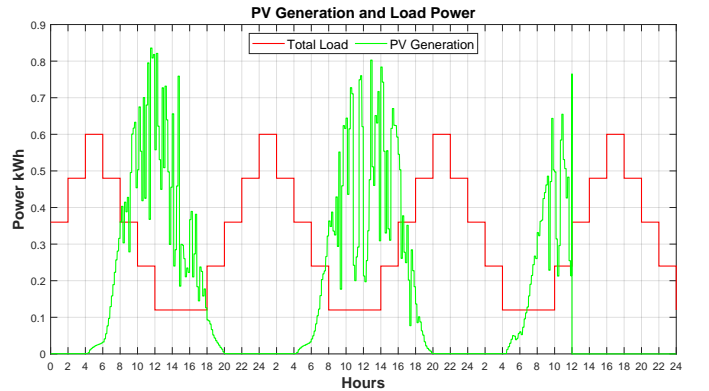


Fig. 8. Case study PV generation and total load profile. The total load profile was generated as a multi-step one to better show the response of the control system to disturbances such as fast load switching.

After the ESSs are depleted, the branch is switched off, therefore the voltages drop to zero as shown in Fig. 11. The output power of the ESSs is also shown in Fig. 13 and Fig. 13. It can be observed that there is a voltage offset of 1 V from the nominal 380 V due because of the droop controllers.

The results for configuration B, the proposed control strategy, is shown in Fig. 10, Fig. 12, and Fig. 14. In this strategy,  $E_{max_i}^{new}$  of the second day, are calculated using equation (24), based on the third-day forecast of distributively trained LSTM models. As shown in Fig. 10 and Fig. 13, the proposed prioritised (dis)charging has changed the power balance in the second and third day, to increase the priority of bus 4 ESS for getting charged with a higher rate than the others. This is

because the load on bus 4 is higher comparing to the other buses. Also the charging rate of ESS at bus 1 is decreased due to the lower load comparing to the buses 2, 3, and 5.

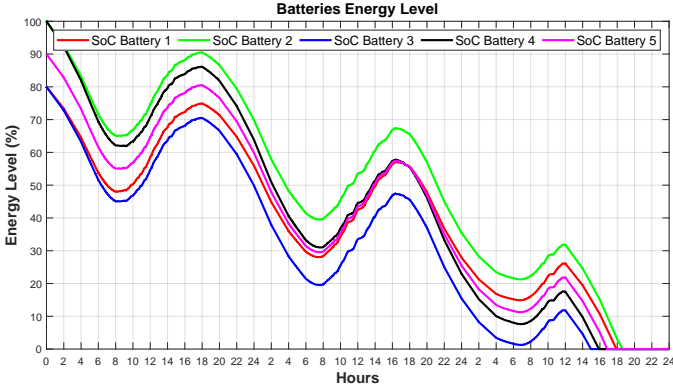


Fig. 9. Configuration A: SoC of ESSs. After the PV outage in the third day at 12h, the ESS has run out of energy from 14h to 19h.

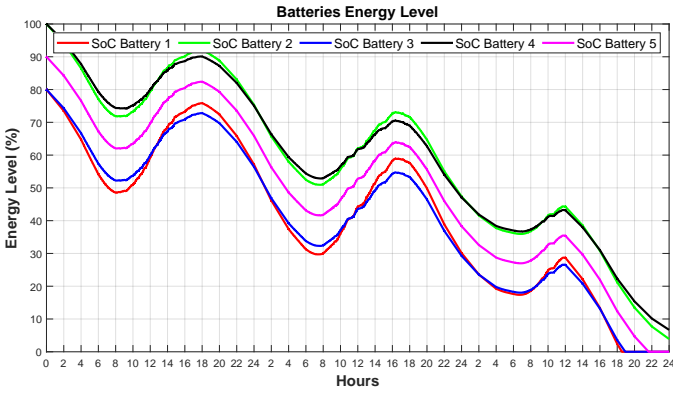


Fig. 10. Configuration B: SoC of ESSs. After the PV outage on the third day at 12h, the ESS has run out of energy from 18h to the following day. This shows how the proposed forecast-based control increased the endurance of the microgrid in cases of a supply outage.

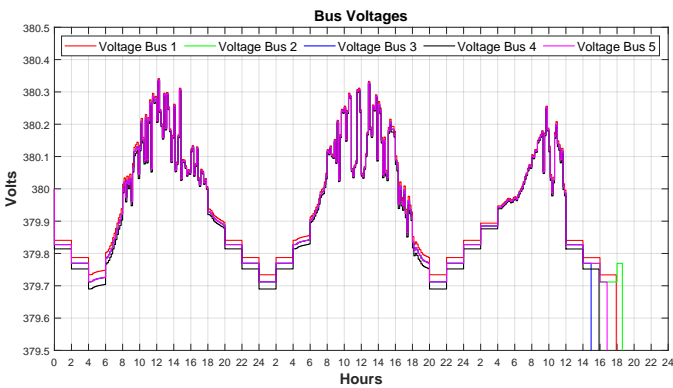


Fig. 11. Configuration A: Voltage of buses. There is a large voltage offset of 1 V from the nominal 380 V due because of the droop controllers. The voltage becomes 0 after the branch ESS has run out of energy.

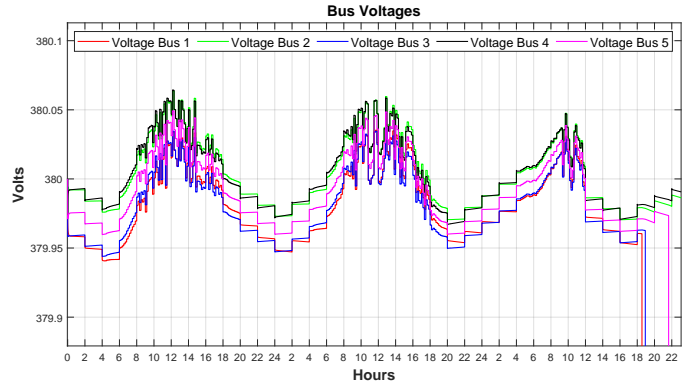


Fig. 12. Configuration B: Voltage of buses. The voltage offset is considerably lower comparing to the droop controllers, less than 0.1 V (90% less). The voltage becomes 0 after the branch ESS has run out of energy.

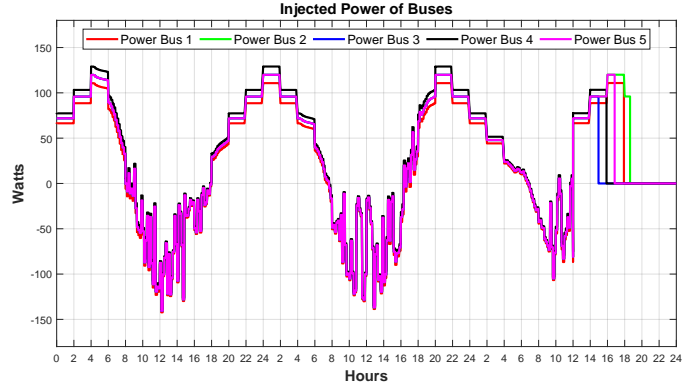


Fig. 13. Configuration A: Injected power of ESSs with the local droop controllers.

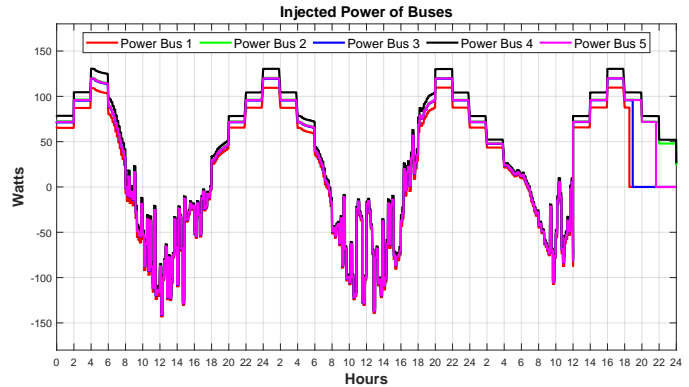


Fig. 14. Configuration B: Injected power of ESSs with the proposed forecast based control system.

On the third day, it can be seen in Fig. 10 that the ESSs run out of energy after 18h. This is because of the prioritized (dis)charging method that has distributed the PV energy on the second day based on the load forecast of the third day. Furthermore, the voltage offset is considerably lower comparing to the droop controllers, less than 0.1 V (90% less).

The voltage becomes 0 after the branch ESS has run out of energy.

The increase in the continuity of the microgrid operation confirms that the resiliency and the endurability of the microgrid are increased by at least 4 hours of longer operation after the fault on PV generation. Furthermore, the voltage of the microgrid is stabilized with lower offset comparing to the decentralized droop control system as shown in Fig. 11 and Fig. 12.

## VIII. CONCLUSION

A novel distributed load forecast based control for DC microgrids was presented in this paper. The proposed control strategy achieves the following objectives:

- Stabilizing the bus voltages co-designed to balance the SoC levels of ESSs in a DC microgrid.
- Prioritized (dis)charging controller for ESSs based on short-term energy forecast of the branches to achieve higher endurability for the DC microgrid.
- Integration of the load forecasting unit in the secondary control layer of the microgrid based on LSTM neural network with DEKF learning algorithm.

The load and generation profiles are predicted using LSTM deep learning models. Due to the large training data requirements of LSTM models, DEKF distributed learning algorithm is used to improve the prediction convergence time. Hardware in-the-loop real-time simulation results confirm the validity of the proposed control strategy for an islanded 380 V DC microgrid. The proposed 2TSA algorithm can get enhanced by considering different ESSs characteristics, in the future.

## REFERENCES

- [1] S. A. Alavi, K. Mehran, Y. Hao, A. Rahimian, H. Mirsaedi, and V. Vahidinasab, "A distributed event-triggered control strategy for dc microgrids based on publish-subscribe model over industrial wireless sensor networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4323–4337, 2019.
- [2] H. Arasteh, M. Kia, V. Vahidinasab, M. Shafie-khah, and J. P. Catalão, "Multiobjective generation and transmission expansion planning of renewable dominated power systems using stochastic normalized normal constraint," *International Journal of Electrical Power and Energy Systems*, vol. 121, p. 106098, Oct. 2020.
- [3] R. Zhang, B. Hredzak, and T. Morstyn, "Distributed Control with Virtual Capacitance for the Voltage Restorations, State of Charge Balancing and Load Allocations of Heterogeneous Energy Storages in a DC Datacenter Microgrid," *IEEE Transactions on Energy Conversion*, 2018.
- [4] S. A. Alavi, K. Mehran, and Y. Hao, "Optimal Observer Synthesis for Microgrids With Adaptive Send-on-Delta Sampling Over IoT Communication Networks," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2020.
- [5] S. A. Alavi, M. Javadipour, and K. Mehran, "Microgrid Optimal State Estimation Over IoT Wireless Sensor Networks With Event-Based Measurements," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Oct. 2019, pp. 4145–4150.
- [6] T. Wang, D. O'Neill, and H. Kamath, "Dynamic Control and Optimization of Distributed Energy Resources in a Microgrid," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2884–2894, Nov. 2015.
- [7] M. M. Gulzar, S. T. H. Rizvi, M. Y. Javed, U. Munir, and H. Asif, "Multi-agent cooperative control consensus: A comparative review," *Electronics (Switzerland)*, vol. 7, no. 2, 2018.
- [8] A. K. Singh, R. Singh, and B. C. Pal, "Stability analysis of networked control in smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 381–390, Jan. 2015.
- [9] K. Wen and Z. Geng, "Modelling and analysis of distributed networked control systems," *IET Control Theory and Applications*, vol. 6, no. 9, pp. 1304–1312, 2012.
- [10] S. Amir Alavi, A. Rahimian, K. Mehran, and J. Alaleddin Mehr Ardestani, "An IoT-Based Data Collection Platform for Situational Awareness-Centric Microgrids," in *Canadian Conference on Electrical and Computer Engineering*, vol. 2018-May. IEEE, 2018, pp. 1–4.
- [11] Y. L. Wang, C. C. Lim, and P. Shi, "Adaptively Adjusted Event-Triggering Mechanism on Fault Detection for Networked Control Systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 2299–2311, Aug. 2017.
- [12] L. Herrera, R. Murawski, F. Guo, E. Inoa, E. Ekici, and J. Wang, "PHEVs charging stations, communications, and control simulation in real time," in *2011 IEEE Vehicle Power and Propulsion Conference, VPPC 2011*. IEEE, Sep. 2011, pp. 1–5.
- [13] C. Wang, J. Duan, B. Fan, Q. Yang, and W. Liu, "Decentralized High-Performance Control of DC Microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3355–3363, May 2019.
- [14] Q. Xu, X. Hu, P. Wang, J. Xiao, P. Tu, C. Wen, and M. Y. Lee, "A Decentralized Dynamic Power Sharing Strategy for Hybrid Energy Storage System in Autonomous DC Microgrid," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5930–5941, Jul. 2017.
- [15] J. Peng, B. Fan, J. Duan, Q. Yang, and W. Liu, "Adaptive decentralized output-constrained control of single-bus DC microgrids," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 424–432, Mar. 2019.
- [16] Y. Hu, X. Wang, Y. Peng, J. Xiang, and W. Wei, "Distributed Finite-Time Secondary Control for DC Microgrids with Virtual Impedance Arrangement," *IEEE Access*, vol. 7, pp. 57 060–57 068, 2019.
- [17] J. Xiao, P. Wang, L. Setyawan, and Q. Xu, "Multi-level energy management system for real-time scheduling of DC microgrids with multiple slack terminals," *IEEE Transactions on Energy Conversion*, vol. 31, no. 1, pp. 392–400, Mar. 2016.
- [18] X. Lei, T. Huang, Y. Yang, Y. Fang, and P. Wang, "A bi-layer multi-time coordination method for optimal generation and reserve schedule and dispatch of a grid-connected microgrid," *IEEE Access*, vol. 7, pp. 44 010–44 020, 2019.
- [19] R. Han, J. M. Guerrero, M. Tucci, A. Martinelli, and G. Ferrari-Trecate, "Plug-and-Play Voltage/Current Stabilization DC Microgrid Clusters with Grid-Forming/Feeding Converters," in *Proceedings of the American Control Conference*, vol. 2018-June. IEEE, Jun. 2018, pp. 5362–5367.
- [20] D. Kumar, F. Zare, and A. Ghosh, "DC Microgrid Technology: System Architectures, AC Grid Interfaces, Grounding Schemes, Power Quality, Communication Networks, Applications, and Standardizations Aspects," *IEEE Access*, vol. 5, pp. 12 230–12 256, 2017.
- [21] T. Dragičević, X. Lu, J. C. Vasquez, and J. M. Guerrero, "DC Microgrids - Part II: A Review of Power Architectures, Applications, and Standardization Issues," *IEEE Transactions on Power Electronics*, vol. 31, no. 5, pp. 3528–3549, May 2016.
- [22] T. Samad and A. M. Annaswamy, "Controls for Smart Grids: Architectures and Applications," *Proceedings of the IEEE*, vol. 105, no. 11, pp. 2244–2261, Nov. 2017.
- [23] D. J. Becker and B. J. Sonnenberg, "DC microgrids in buildings and data centers," in *INTELEC, International Telecommunications Energy Conference (Proceedings)*. IEEE, Oct. 2011, pp. 1–7.
- [24] H. Shi, M. Xu, and R. Li, "Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.
- [25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [26] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, and F. Wu, "Efficient parallel framework for HEVC motion estimation on many-core processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2077–2089, 2014.
- [27] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.
- [28] X. Cao, S. Dong, Z. Wu, and Y. Jing, "A data-driven hybrid optimization model for shortterm residential load forecasting," in *2015 IEEE International Conference on Computer and Information Technology*. Liverpool, U.K: CIT/IUCC/DASC/PICOM, 2015, pp. 283–287.
- [29] Z. Yun, Z. Quan, S. Caixin, L. Shaolan, L. Yuming, and S. Yang, "RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 853–858, 2008.
- [30] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite k-nearest neighbor model for day-ahead load forecasting with limited

- temperature forecasts,” in *IEEE Power and Energy Society General Meeting*, vol. 2016-Novem, Boston, MA, USA, 2016, pp. 1–5.
- [31] Q. Pang and M. Zhang, “Very short-term load forecasting based on neural network and rough set,” in *2010 International Conference on Intelligent Computation Technology and Automation, ICICTA 2010*, vol. 3, Changsha, China, 2010, pp. 1132–1135.
  - [32] R. Zhang, Z. Y. Dong, Y. Xu, K. Meng, and K. P. Wong, “Short-term load forecasting of Australian national electricity market by an ensemble model of extreme learning machine,” *IET Generation, Transmission and Distribution*, vol. 7, no. 4, pp. 391–397, 2013.
  - [33] M. Ghofrani, M. Ghayekhloo, A. Arabali, and A. Ghayekhloo, “A hybrid short-term load forecasting with a new input selection framework,” *Energy*, vol. 81, pp. 777–786, 2015.
  - [34] P. Ji, D. Xiong, P. Wang, and J. Chen, “A study on exponential smoothing model for load forecasting,” in *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, 2012.
  - [35] M. David, F. Ramahatana, P. Trombe, and P. Lauret, “Probabilistic forecasting of the solar irradiance with recursive ARMA and GARCH models,” *Solar Energy*, vol. 133, pp. 55–72, Aug. 2016.
  - [36] F. Wang, K. Li, L. Zhou, H. Ren, J. Contreras, M. Shafie-khah, and J. P. Catalão, “Daily pattern prediction based classification modeling approach for day-ahead electricity price forecasting,” *International Journal of Electrical Power & Energy Systems*, vol. 105, pp. 529–540, Feb. 2019.
  - [37] R. Haiges, Y. Wang, A. Ghoshray, and A. Roskilly, “Forecasting Electricity Generation Capacity in Malaysia: An Auto Regressive Integrated Moving Average Approach,” *Energy Procedia*, vol. 105, pp. 3471–3478, May 2017.
  - [38] D. Tien Bui, K.-T. T. Bui, Q.-T. Bui, C. V. Doan, and N.-D. Hoang, “Hybrid Intelligent Model Based on Least Squares Support Vector Regression and Artificial Bee Colony Optimization for Time-Series Modeling and Forecasting Horizontal Displacement of Hydropower Dam,” in *Handbook of Neural Computation*. Elsevier, 2017, pp. 279–293.
  - [39] H. Yang, J. Long, P. He, and Y. Zheng, “Short-Term Load Forecasting of Microgrid Based on Grey Correlation Analysis and Neural Network Optimized by Mind Evolutionary Algorithm,” in *2019 IEEE PES Innovative Smart Grid Technologies Asia, ISGT 2019*. Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 2738–2742.
  - [40] U. K. Das, K. S. Tey, M. Seyedmahmoudian, S. Mekhilef, M. Y. I. Idris, W. Van Deventer, B. Horan, and A. Stojcevski, “Forecasting of photovoltaic power generation and model optimization: A review,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 912–928, Jan. 2018.
  - [41] R. Ahmed, V. Sreeram, Y. Mishra, and M. Arif, “A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization,” *Renewable and Sustainable Energy Reviews*, vol. 124, p. 109792, May 2020.
  - [42] Y. Ren, P. Suganthan, and N. Srikanth, “Ensemble methods for wind and solar power forecasting—A state-of-the-art review,” *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 82–91, Oct. 2015.
  - [43] M. K. Behera, I. Majumder, and N. Nayak, “Solar photovoltaic power forecasting using optimized modified extreme learning machine technique,” *Engineering Science and Technology, an International Journal*, vol. 21, no. 3, pp. 428–438, Jun. 2018.
  - [44] S. Sobri, S. Koochi-Kamali, and N. A. Rahim, “Solar photovoltaic generation forecasting methods: A review,” *Energy Conversion and Management*, vol. 156, pp. 459–497, Jan. 2018.
  - [45] A. Garulli, S. Paoletti, and A. Vicino, “Models and Techniques for Electric Load Forecasting in the Presence of Demand Response,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1087–1097, May 2015.
  - [46] I. Roesch and T. Günther, “Visualization of Neural Network Predictions for Weather Forecasting,” *Computer Graphics Forum*, vol. 38, no. 1, pp. 209–220, Feb. 2019.
  - [47] T. Ergen and S. S. Kozat, “Online Training of LSTM Networks in Distributed Systems for Variable Length Data Sequences,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 5159–5165, Oct. 2018.
  - [48] UK Power Networks, “Photovoltaic (PV) Solar Panel Energy Generation data,” 2014.