LSTM-based Multi-Step SOC Forecasting of Battery Energy Storage in Grid Ancillary Services

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

06-08-2021 / 09-08-2021

CITATION

Ardiansyah, Ardiansyah; Kim, Yeonghyeon; Choi, Deokjai (2021): LSTM-based Multi-Step SOC Forecasting of Battery Energy Storage in Grid Ancillary Services. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.15124416.v1>

DOI

[10.36227/techrxiv.15124416.v1](https://doi.org/10.36227/techrxiv.15124416.v1)

LSTM-based Multi-Step SOC Forecasting of Battery Energy Storage in Grid Ancillary Services

Ardiansyah, Yeonghyeon Kim, Deokjai Choi

Department of Artificial Intelligence Convergence, Chonnam National University, South Korea

Email: ardi@ejnu.net, 206379@jnu.ac.kr, dchoi@jnu.ac.kr

Abstract—Battery energy storage (BES) participation in the grid ancillary services markets is increasing rapidly in recent years. To facilitate optimal participation, the need for accurate BES state-of-charge (SOC) forecasting is indispensable. In grid ancillary services, the development of SOC forecasting models should deal with uncertainties and corresponding stochastic processes that determine the BES SOC periodically. Several traditional and state-of-the-art machine learning (ML) techniques, ranging from decision-tree to deep learning methods, were used to solve this problem. However, developing a multi-step SOC forecasting model remains a challenge in this subject that is essential for optimal BES economic dispatch and unit commitment. Taking advantage of the Long short-term memory (LSTM) deep learning and its variants techniques which are proven to be a robust method for predicting sequentially dependent data in the time-series domain, this paper proposes LSTM-based multi-step SOC forecasting for BES operating in frequency regulation. Various developed models, i.e., Vanilla-LSTM, Vanilla-Gated Recurrent Units (GRU), Bidirectional-LSTM (Bi-LSTM), and Bi-GRU, are evaluated using real-world datasets. The evaluation results show that the developed models outperform the existing methods in terms of root mean square error (RMSE) and mean absolute error (MAE) evaluation metrics.

I. INTRODUCTION

Ensuring the power grid is stable, and the supply matches demand continuously requires decisions that need to be made over a staggering array of time scales. For this problem, the day-ahead energy market allows power grid operators to have enough supply to meet anticipated demand 24 hours in advance. The real-time market does the same thing on an hour-ahead basis. However, many things can happen between the hour-ahead market clearing and actual real-time dispatch, forcing the grid operator to add supply immediately. Therefore, grid ancillary services have been developed to handle the fluctuation problem of supply and demand in those moments.

Ancillary services are expected to quickly provide the reliability needed to respond to variations in supply and demand as well as maintain frequency and voltage. There are currently many types of ancillary services, where the most common services are frequency regulation, voltage control, reactive power supply, and black start or grid restoration [1]. Traditionally these have been provided by contracted regular power generators. However, the conditions of gradual replacement of dispatchable fossil-fuel-based power plants by intermittent renewable power generation demand more flexibility to guarantee the stability of power system operation.

The immense development of smart grid technologies has prompted a shift in the equipment used to handle the challenge,

as mentioned above. One of the most promising elements to provide more flexible ancillary services is battery energy storage (BES) [2]. BES is a good option because of its fast response time feature. Also, it surpasses conventional power plants in ramping capabilities. As a result, the share of BES in the ancillary service markets is increasing rapidly. For example, in the Pennsylvania-New Jersey-Maryland (PJM) regional transmission organization (RTO) ancillary services markets alone, the BES capacity as regulation resources have increased from zero to over 280 MW in the last decade, making up 41% of their regulation procurement capacity [3].

Many models exist to determine the optimal participation of BES in ancillary services markets. In many cases, the participation strategy is optimized for only one or a few specific services, such as performing voltage control [4] or frequency regulation while minimizing BES aging [3]. Other models investigate the economic aspects of BES participation in ancillary services that require a cost-benefit analysis [5], [6]. The developed participation strategy is bound to various decision-making activities to find the trade-off between BES operation benefits and cost in pay-for-performance-based ancillary services markets [7]. In this context, the decision needs to take account of the BES state-of-charge (SOC) constraint for their optimal participation. Consequently, accurate information of BES SOC is essential.

It is owing to the development of the internet of things (IoT)-based battery digital twin system [8], battery-related data that are collected using onboard sensing devices can be securely transferred to the cloud-based battery management systems (cloud-BMS) for online state prediction and optimization. With the wealth of available data, machine learning (ML)-based approaches have received renewed attention in solving SOC forecasting problems [9], [10]. In the case of BES in grid ancillary services, SOC forecasting models need to deal with multi-physics conditions to support forecasting accuracy. In [11], several traditional and state-of-the-art ML techniques, ranging from decision-tree-based methods (Decision-Tree (DT), Random-Forest (RF), and Light Gradient Boosting Machine (LightGBM)) to data-driven deep learning approaches (Convolutional Network Network (CNN) and Recurrent Neural Network (RNN)), have been proposed and analyzed. The developed ML models considered the uncertainties of corresponding frequency features that periodically determine the BES SOC. However, developing a multi-step SOC forecasting model remains a challenge in this subject

that is important for optimal BES economic dispatch and unit commitment. Multi-step ahead forecasting is the task of predicting a sequence of values in a sequential time-series manner. Given recent SOC data from a historical time series, a forecasting model can be developed to predict SOC values multiple time steps of the defined forecasting horizon in the future.

Taking advantage of the Long short-term memory (LSTM) deep learning techniques that are proven to be a robust method for predicting sequentially dependent data, this paper proposes and analyzes LSTM-based multi-step SOC forecasting models for BES operating under frequency regulation. Specifically, the main contributions of this paper are as follows:

- 1) We formulate our objective to maximize the received reward of distributed energy resources (DER) producers in a pay-for-performance-based scheme of grid ancillary service.
- 2) We handle SOC forecasting as a multi-step time-series forecasting problem. Four state-of-the-art LSTM variants techniques, i.e., Vanilla-LSTM, Bidirectional-LSTM (Bi-LSTM), Vanilla-Gated recurrent unit (GRU), and Bidirectional-GRU (Bi-GRU), are proposed and analyzed to solve this problem.
- 3) We evaluate our developed multi-step SOC forecasting models along with the existing ML methods in [11] using real-world datasets that publicly available in [12].

We believe that the work presented in this paper will aid in realizing an efficient and meaningful tool for optimal decision-making applications of BES participation in smart grid ancillary services.

The rest of this paper is organized as follows. In the next section, we provide a brief description of our objective and problem formulation. Then, Section III describes the general principles of the LSTM deep learning algorithms applied in this study and describes our methods in developing and optimizing multi-step SOC forecasting models. Section IV provides the evaluation results, and finally, section V concludes this paper.

Fig. 1. Example of grid frequency failure event. The frequency regulation service is then activated at any time to bring the balance back to the grid.

II. OBJECTIVE AND PROBLEM FORMULATION

A. Objective

In frequency regulation, the grid operator holds power capacity by contracted energy producers, activated automatically to bring the balance to the grid if a grid failure event happens. As depicted in Fig. 1, frequency containment reserves (FCR) stops the frequency drift, and the frequency restoration reserves (FRR) bring the frequency back to its nominal value. BES can be used either for FCR or FRR of frequency control due to its technical capabilities [13]. First, the high power capacity of BES ensures enough instantaneous active power that can be discharged for grid frequency support. Second, a fast response time ensures that the time to start providing active power output for grid frequency support is short enough.

The independent electricity market operators such as Korea Power Exchange (KPX) will publish the dynamic frequency regulation (REGD) signal to ask contracted energy producers to follow. Then their participation will be paid using the pay-for-performance-based scheme. In this scheme, the reward depends on their frequency regulation capacity and their accuracy in the signal following. DER energy producers should periodically evaluate the weighting factor between possible rewards and operation costs to optimize the BES economical dispatch and unit commitment. The objective is to maximize the received reward by considering their mismatch penalty and cost of BES operation, depicted as

$$\max_{C,b} CT - \sum_{t=1}^T |b(t) - Cr(t)| - \sum_{t=1}^T f[b(t), SoC(t)], \quad (1)$$

where, $C \geq 0$ is the frequency regulation capacity bidding, $t = (1, 2, 3, \dots, T)$ is the operation time interval and r is the REGD signal (1 by T vector). b is BES charging/discharging (positive for charging) capacity and function $f[b(t), SoC(t)]$ represents the cost of BES operation, which can be calculated by SOC variation value over time.

The change of the BES SOC within the period of frequency regulation can be defined as percentage, expressed as

$$SoC(t) = 100\% \int_{t_{i-1}}^{t_i} \frac{\eta P_{FR}(t)}{E_{nom}} dt, \quad (2)$$

where, η is the BES efficiency, P_{FR} is the BES power output at every moment t according to the REGD signal following, and E_{nom} is the nominal energy capacity of the BES. Therefore, optimal participation is dependent on the serial correlation of coupled resources and corresponding stochastic processes that predetermine the development of BES SOC in a time-series manner. Hence, accurate multi-step SOC forecasting is essential to facilitate optimal BES economic dispatch and unit commitments in frequency regulation service.

B. Multi-Step Forecasting Problem

In general, a multi-step time series forecasting task consists of predicting the next $H = [x_{t+1}, \dots, x_{t+H}]$ values of a historical time series $[x_1, \dots, x_t]$, where $H > 1$ denotes the

forecasting horizon. Several methods are developed for multi-step forecasting problems, where three main strategies are, 1) Direct Multi-step Forecast Strategy, 2) Recursive Multi-step Forecast Strategy, 3) Multiple Output Forecast Strategy [14]. Different ML models have been used to implement the direct strategy for multi-step forecasting tasks, for instance, the nearest neighbors [15], DT and RF [16]. This direct method involves developing a separate model for each forecast time step. Using one model for each time step can add computational problems, especially as the number of time steps to be forecasted increases.

On the other hand, the recursive strategy that is commonly used in the Auto-Regressive Integrated Moving Average (ARIMA) forecasting model [17] involves a single model multiple times. In this case, the prediction for the prior time step is used iteratively to predict the following time step. Unfortunately, when very long-term forecasts at stake and a stochastic setting is assumed, this modeling approach neglects the existence of stochastic dependencies between future values. Thus, prediction errors will be accumulated such that performance can quickly degrade as the prediction time horizon increases.

The multiple output strategy involves developing a model that is capable of predicting the entire forecast sequence in a one-shot manner, depicted as

$$[x_{t+H}, \dots, x_{t+1}] = F(x_t, \dots, x_{t-n+1}) + w, \quad (3)$$

where $t \in \{n, \dots, N - H\}$, $F : \mathbb{R}^d \rightarrow \mathbb{R}^H$ is the input vector-valued function, and $w \in \mathbb{R}^H$ is a noise vector. Hence, the predicted value is no more a scalar quantity but a vector of future values of the time series. The multiple output model is a more complex forecasting approach as they need to learn the dependence structure between inputs, inputs-outputs, and between outputs.

III. LSTM FOR MULTI-STEP SOC FORECASTING

A. General Principles of LSTM and its variants

RNN is a type of neural network family designed to work, learn, and predict sequentially dependent data [18], such as the mentioned multiple output strategy in a multi-step time-series forecasting problem. In RNN, the output can be fed back into the network as inputs creating a recurrent structure. However, RNN suffers from short-term memory that may have a hard time carrying information from earlier time steps to later ones if a sequence is long enough. The LSTM cell is then created as a solution to solve this problem [18]. LSTM has internal mechanisms called gates that can regulate the flow of information for producing a stable model. In addition to utilizing the recurrent connection of the outputs from the previous time-step, gates internal memory operates like a local variable to accumulate state over the input sequence. Thus, the gates can learn which data is important to keep or throw away.

When it comes to multi-step time series forecasting, LSTM offers several benefits. First, LSTM can automatically learn features from sequence data as input without the need for lag observations. Second, LSTM is able to map input data directly

Fig. 2. Internal mechanisms to regulate the flow of information in LSTM and GRU cells. In LSTM (top), we have three different gates that control information flow, i.e., forget gate, input gate, and output gate. The forget gate is used to decide what information should be thrown away or kept from the previous cell state c_{t-1} subject to the output of the sigmoid function (closer to 0 = forget, closer to 1 = keep). The input gate is then used to update the cell state, and the output gate is used to decide what the next hidden state should be. The new cell state c_t and the new hidden h_t are then carried over to the next time step. On the other hand, in GRU (b), the cell state is removed and only used the hidden state to transfer information. GRU also has two gates, a reset gate and an update gate with similar functionality to the gates in LSTM.

to a variable-length vector that may represent multiple output time steps. Last, LSTM also supports multiple parallel input sequences for multivariate inputs. The inputs can be presented in a flat structure. Recently, there are several newer generation techniques, i.e., GRU, Bi-LSTM, and Bi-GRU, as variants of the LSTM [19]. To make it easier, we call four variants of LSTM as 1) Vanilla-LSTM, 2) Vanilla-GRU, 3) Bi-LSTM, and 4) Bi-GRU in this paper.

Vanilla-GRU has fewer tensor operations than Vanilla-LSTM. Therefore, they are expected to train faster than Vanilla-LSTM. As depicted in Fig. 2, the update gate in Vanilla-GRU acts similar to the forget and input gate of the Vanilla-LSTM. It decides what information to throw away and what new information x_t to add. The reset gate is another gate that is used to decide how much past information to forget. Furthermore, the idea of Bi-LSTM and Bi-GRU is to aggregate input information in the past and future of a specific time step in Vanilla-LSTM/GRU models. Bi-LSTM/Bi-GRU computes the hidden forward sequence, calculates the backward hidden sequence, and combines both to calculate the output. Therefore, in Bi-LSTM/Bi-GRU, we may be able to preserve information from both past and future at any point in time.

Fig. 3. Example of frequency measurement (top) and the percent value of SOC charging/discharging variation (bottom) on March 7, 2015, of publicly available dataset. The frequency needs to be maintained within the dead-band limits by responding to the frequency deviations from the nominal system frequency. BES is expected to become one of the first activated resorts in frequency regulation to guarantee frequency stability. It can be expressed as a BES power output response following the drop curve signal. A negative frequency deviation leads to BES discharging, while a positive deviation provokes BES charging. The BES SOC variation is represented in percentage (positive for charging, negative for discharging).

B. BES SOC Dataset

In this work, we analyze and compare the performance of four variant LSTM methods for multi-step time-series SOC forecasting in the case of BES operating in frequency regulation services. Representative data that taking into account the corresponding stochastic processes in the real-world environment is needed to make a good SOC forecasting model. Therefore, the real-world SOC dataset of BES operating in frequency regulation service that published online in [12] is used. In this dataset, the percentage value of SOC variation in BES charging and discharging was generated using the real

frequency measurements for the period of four years from 2015 to 2018. The case study corresponds to a frequency regulation service market in the northern Europe area, using a 50Hz frequency instead of 60Hz, like in South Korea or the United States.

Several features such as timestamp, market, frequency, and modified SOC are available in the dataset. From exploratory data analysis (EDA) in [11], the BES SOC variation has a high correlation with the frequency measurement feature. With an almost perfect correlation in mutual information score (0 = no mutual information, 1 = perfect correlation), the frequency measurement becomes the essential feature that needs to be considered to reduce uncertainty and support the SOC forecasting accuracy. Moreover, The 15-minutes and one-hour time scale SOC data were selected considering the BES operation in a multi-objective grid ancillary services with current procurement practices and possible future developments. In total, 140041 samples were taken for the 15-minutes time scale, divided into 69968 samples for the year 2015-2016 data and 70073 samples for the year 2017-2018. Furthermore, the 15-minutes data were transformed to an hourly resolution that produced 35064 samples of the one-hour time scale. An example of frequency measurement and the percent value of SOC variation in a day of BES operation is depicted in Fig. 3.

C. Model Development

In the experiment, we set the timestamp as the time-series index. The percent value of BES SOC variation and its corresponding frequency data are used for the input data. The dataset is then divided for training and testing sets in proportions of 70% and 30%. Also, 25% of the training set is used for the hyper-parameters validation during the model development. Moreover, MinMax scaling with the range from 0 to 1 is used to normalize the dataset. Then, the missing values are replaced with linear interpolation.

We develop sequential models architecture for multi-steps SOC forecasting based on Vanilla-LSTM, Bi-LSTM, Vanilla-GRU, and Bi-GRU techniques. The input of developed models is in a three-dimensional (3D) array. The first dimension represents the number of sample data points. The second

Fig. 4. LSTM is configured as the forecasting model to predict the multi-step output of BES SOC. n historical data points are used as inputs. If the inputs are 48 data points in a 15-minutes time scale (total of 720 minutes), it means that the models make predictions based on the last 720 minutes history of SOC variation. The outputs are the SOC every next 15-minutes of the entire forecast horizon.

dimension represents the time steps of look-back periods, and the third dimension represents the number of features. The look-back period defines how many previous time steps are used to predict the subsequent time step. As depicted in Fig. 4, the models will not produce output vector sequences directly. Instead, the models consist of the encoder to read and encode the input sequence and the decoder that predicts each element in the output sequence of the forecasting horizon.

In this work, we set 48 data points for the number of look-back periods representing twelve-hour prior periods for the 15-minutes dataset and two days before for the one-hour dataset, respectively. For tuning the developed models and avoid overfitting, we employ Adam optimizer [20] and the dropout regularization [21]. A fully connected layer called the TimeDistributed Dense layer is then used for outputting the forecasting results. The implementation of forecasting models are using Keras 2.3.1 deep learning application programming interface (API) [22] with Tensorflow 2.1.0 [23] as the backend in the Python 3.7.6 environment.

IV. EVALUATION

A. Model Optimization

The type of predictive modeling problem imposes constraints on the kind of loss function that can be used. One of the standard loss functions for a regression-type problem is the mean squared error (MSE). Since time-series forecasting is part of a regression type problem, the MSE loss function is used to evaluate the network that is minimized by the Adam optimization algorithm. Furthermore, the networks are trained in a specified number of epochs to the training and validation dataset. Each epoch is then partitioned into groups of input-output pattern pairs called batches. The batch size defines the number of patterns that the network is exposed to before the weights update within an epoch. Note that the batch size must be multiple 8, and the fine-tuning can be done based on the performance observation. Therefore, we set the batch size as 8, 16, and 32 in the experiment. Fig. 5 depicts the example of model optimization results based on the configured parameters in Table I.

TABLE I
PARAMETER SETTING FOR THE MODEL OPTIMIZATION.

Parameter	Value/Range
Developed models	LSTM, GRU, Bi-LSTM, and Bi-GRU
Training data	70% of total data points
Validation data	25% of total training data points
MinMax normalization	Enabled with range from 0 to 1
Missing data imputation	Enabled using linear interpolation
Regularization	Dropout = 0.2 in each layer
Number of units/sequence	64 (128 for Bi-LSTM/Bi-GRU)
Number of maximal epochs	150
Batch size	8, 16, 32
Early stop to avoid overfitting	Yes with patience = 30
Number of look-back periods	48 data points

Fig. 5. Result of the model optimization for Bi-LSTM. It is shown that both training and validation losses are decreasing, with the optimal number of epochs are around 10. Since we applied the dropout regularization in each layer to improve testing accuracy, the training loss can be slightly higher than the validation loss. It should be noted that the regularization is only applied in the training stage, not in the validation stage.

B. Performance Evaluation Metrics

The forecasting errors ϵ_t are estimated by the deviation between actual observation and the prediction of SOC value from time to time. The mean absolute error (MAE) and the root mean squared error (RMSE) widely used for error evaluation metrics were employed to quantify the statistical quality of forecasts for testing data points. Depicted as

$$MAE = \frac{1}{T} \sum_{t=1}^T \epsilon_t, \text{ and} \quad (4)$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T \epsilon_t^2}. \quad (5)$$

C. Performance Comparison

We analyze and compare the performance of our developed SOC forecasting models with the existing ML methods described in [11]. The existing algorithms are included DT, RF, LightGBM, CNN, and RNN. We re-implemented the existing ML methods using Keras 2.3.1 API for CNN and RNN deep learning models, Scikit-learn [24] library for DT and RF regression models, and its own library for LightGBM model [25].

The example of experimental results are presented in Fig. 6 and the performance comparison of all forecasting models are shown in Table II. According to the evaluation using MAE and RMSE performance metrics, the Bi-GRU model achieved the best score for forecasting the change of BESS SOC in both 15-minutes and hour-ahead time scales.

Fig. 6. Example performance results of DT technique versus our proposed techniques on the 15-minutes time scale of the BES SOC testing set. It is shown that the Bi-LSTM and Bi-GRU methods can precisely predict the SOC variation of the forecasting horizon and outperform the DT-based approach in multi-step forecasting of BES SOC.

TABLE II
PERFORMANCE EVALUATION RESULT USING TESTING DATA.

Models	MAE	RMSE
DT 15-min	3.3797	5.3930
RF 15-min	2.3652	3.7271
LightGBM 15-min	2.4036	3.8889
CNN 15-min	2.6439	4.3826
RNN 15-min	2.9348	4.5273
Bi-GRU 15-min	2.0281	3.4199
Bi-LSTM 15-min	2.2519	3.4981
DT one-hour	14.0884	19.3813
RF one-hour	8.7977	13.4765
LightGBM one-hour	9.3875	13.6441
CNN one-hour	8.9080	14.3442
RNN one-hour	9.1430	14.3381
Bi-GRU one-hour	8.7160	13.2724
Bi-LSTM one-hour	8.7251	13.3865

V. CONCLUSION

In this paper, multi-step SOC forecasting based on LSTM and its variants techniques were developed. Moreover, a comparison between the proposed methods with several traditional and state-of-the-art ML algorithms was conducted using the real-world BES SOC data. Finally, the performance evaluation of the algorithms was carried out on 15-minutes and one-hour time scales. The results demonstrate that all the developed models outperform all decision-tree-based (DT, RF, LightGBM), CNN, and RNN learning methods on both evaluation time scales. Furthermore, the Bi-GRU model achieves the best performance based on MAE and RMSE evaluation metrics.

We believe forecasting of BES SOC can potentially provide an efficient and meaningful tool for decision-making applications in grid ancillary services, which maximize the received reward of DER producers by considering the cost of BES operation in a pay-for-performance-based scheme. However, considering BES degradation to decision processes that may be determined using the state of health (SOH) prediction should

also be investigated for such applications in future research.

ACKNOWLEDGMENT

This research is supported by Brain Korea (BK) 21 AI Convergence Human Resource Education and Research Center at Chonnam National University, Gwangju, South Korea.

REFERENCES

- [1] K. Oureilidis, et al., "Ancillary services market design in distribution networks: Review and identification of barriers," *Energies*, vol. 13, no. 4, pp. 917, 2020.
- [2] L. Maeyaert, L. Vandeveld, and T. Döring, "Battery storage for ancillary services in smart distribution grids," *J Energy Storage*, Vol. 30, 2020.
- [3] B. Xu, et al., "Optimal battery participation in frequency regulation markets," *IEEE Trans Power Syst*, vol. 33, no. 6, pp. 6715, Nov. 2018.
- [4] M. Zeraati, M.E.H. Golshan, and J.M. Guerrero, "Distributed control of battery energy storage systems for voltage regulation in distribution networks with high PV penetration," *IEEE Trans Smart Grid*, vol. 9, no. 4, pp. 3582, Nov. 2018.
- [5] G. Celli, et al., "Relieving tensions on battery energy sources utilization among TSO, DSO, and service providers with multi-objective optimization," *Energies*, vol. 14, pp. 239, 2021.
- [6] G. Rancilio, et al., "Grid-scale BESS for ancillary services provision: SOC restoration strategies," *Applied Sciences*, vol. 10, no. 22, pp. 4121, 2020.
- [7] B. Xu, et al., "A comparison of policies on the participation of storage in us frequency regulation markets," *2016 IEEE Power Energy Soc. Gen. Meeting*, 2016.
- [8] Ardiansyah, Y. Kim, and D. Choi, "IoT system architecture for battery digital twin," *Proc. of KNOM Conference 2021*, pp. 40-43, 2021.
- [9] M-F. Ng, et al., "Predicting the state of charge and health of batteries using data-driven machine learning," *Nat Mach Intell*, vol. 2, pp. 161, 2020.
- [10] B. Wu, et al., "Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems," *Energy and AI*, Vol. 1, 2020.
- [11] A. Mashlakov, et al., "Multi-timescale forecasting of battery energy storage state-of-charge under frequency containment reserve for normal operation," *16th Int. Conf. on the European Energy Market (EEM)*, pp. 1-8, 2019.
- [12] (2019) "BESS-SOC-forecasting datasets," [Online]. Available: <https://github.com/aleksei-mashlakov/BESS-SOC-forecasting>
- [13] Y-K. Wu and K-T. Tang, "Frequency support by BESS – Review and analysis," *Energy Procedia*, Vol. 156, pp. 187-191, 2019.
- [14] G. Bontempi, S.B. Taieb, and Y.A.L. Borgne, "Machine Learning strategies for time series forecasting," *eBISS 2012 Lecture Notes in Business Information Processing*, vol. 138, 2013.
- [15] J.J. Flores, et al., "Evolving nearest neighbor time series forecasters," *Soft Comput*, vol. 23, no. 3, pp. 1039–1048, 2019.
- [16] D. Vassallo, et al., "Analysis of random forest modeling strategies for multi-step wind speed forecasting," *Energies*, vol.13, no.20, pp. 5488, 2020.
- [17] G.D. Nugraha, et al., "Lambda-based data processing architecture for two-Level load forecasting in residential buildings," *Energies*, vol. 11, no.4, pp. 772, 2018.
- [18] H. Hewamalage, C. Bergmeir, K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *arXiv:1909.00590v5 [cs.LG]*, 2020.
- [19] S. Boubaker, et al., "Deep neural networks for predicting solar radiation at Hail region, Saudi Arabia," *IEEE Access*, vol. 9, pp. 36719, 2021.
- [20] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980v9 [cs.LG]*, 2017.
- [21] N. Srivastava, et al., "Dropout: a simple way to prevent neural networks from overfitting" *J Mach Learn Res*, vol. 15, no.1, pp. 1929-1958, 2014.
- [22] F. Chollet et al., Keras 2.3.1, 2019, [online] Available: <https://keras.io>.
- [23] M. Abadi, et al., "Tensorflow: A system for large-scale machine learning", *12th USENIX Symp. on Operating Systems Design and Implementation*, pp. 265-283, 2016.
- [24] F. Pedregosa, et al., "Scikit-learn: Machine Learning in Python", *J Mach Learn Res*, vol. 12, pp. 2825-2830, 2011.
- [25] G. Ke, et al., "LightGBM: A highly efficient gradient boosting decision tree". *Adv. in Neural Inf. Processing Systems* 30, pp. 3149-3157, 2017.