DTU Library

Evaluating Machine-Learning-Based Methods for Modeling a Digital Twin of Battery Systems Providing Frequency Regulation

Kharlamova, Nina; Hashemi, Seyedmostafa

Published in:
IEEE Systems Journal

Link to article, DOI:
10.1109/JSYST.2023.3238287

Publication date:
2023

Document Version
Peer reviewed version

Evaluating Machine-Learning-Based Methods for Modeling a Digital Twin of Battery Systems Providing Frequency Regulation

Nina Kharlamova, *Student Member, IEEE*, and Seyedmostafa Hashemi, *Member, IEEE*

***Abstract*—Battery energy storage systems (BESSs) have the potential to become providers of frequency regulation services in future energy systems. In this case, the reliable operation of BESS has a significant impact on system stability. One of the tools that ensures the secure BESS operation is applying a digital twin to forecast the BESS state of charge (SOC). It allows to predict BESS behavior and identify potential failures or cyberattacks in BESS. In this article, we compare multiple data-driven approaches to forecast the SOC of a BESS based on realistic dataset and real battery operation. Recurrent neural networks, e.g., feedforward neural network, gated recurrent unit, long short-term memory, support vector regression, random forest, and AdaBoost methods are applied to evaluate each method's performance. We compare the methods based on the maximum and average forecast errors, the required computational capacity, as well as expected training speed. To validate the results, in addition to the data gathered from simulations, we used experimental setup with a BESS of 79 kWh providing containment reserve for normal operation, which is a BESS service with a high economical potential in Nordic Region. Furthermore, we provide recommendations for the methods that are suitable to be applied for modeling the digital twin of a utility-scale battery digital twin.**

***Index Terms*—Artificial intelligence (AI), battery energy storage system (BESS), digital twin, frequency regulation, neural network, state of charge (SOC).**

Nomenclature

Constants

η BESS efficiency
W Energy capacity of a BESS
Δt Duration of the control period
N Number of samples in the dataset
MAE Mean absolute error
ME Maximum error

Variables

$\varphi(x_i)$ Input features vector
G A set of high-dimensional linear functions
w, b Coefficients that minimize regularized risk function
V Vector of weights
z_t Update gate
r_t Reset gate
$\tilde{H}_t$ A candidate for a new hidden state
$W_{xz,xr,xh}$ Weight parameters for input data
$W_{hz,hr,hh}$ Weight parameters for a vector of hidden units
$b_{z,r,h}$ Bias vectors

Indices and Sets

f Set of system frequency
SOC Set of state of charge of BESS
$P_{\%,N-1}$ Set of a battery installed capacity share value
h Vector of hidden units
h_{out} Output vector
$\tilde{y}$ Set of real data
X Input vector
x_i Obtained forecast vector
x Sample of authentic values

Manuscript received 1 March 2022; revised 4 July 2022; accepted 9 January 2023. This work was supported by the Danish project "BOSS: Bornholm smart grid secured by grid-connected battery systems" co-founded by Danish Energy Technology Development and Demonstration program under Grant 64018-0618.
(Corresponding author: Seyedmostafa Hashemi.)

The authors are with the Technical University of Denmark, 2800 Lyngby, Denmark (e-mail: ninkhar@elektro.dtu.dk; shtog@dtu.dk).
Digital Object Identifier 10.1109/JSYST.2023.3238287

I. Introduction

Renewable energy sources (RESs) are one of the technologies that enable covering a raising global energy consumption [1]. Battery energy storage systems (BESSs) are a critical part of an electrical grid since they can contribute to a smooth integration of RES [2]. A BESS can provide various services to the grid, such as frequency control, peak shaving, black start, and reactive power support. Frequency control services, in particular, frequency containment reserve for normal operation (FCR-N), are among the BESS services with a high revenue potential in the Nordic Region. Since conventional power plants applied for frequency control are retiring, they have to be replaced with modern ones. The response time of a BESS, which is one of the most important features for frequency regulation services, is lower than conventional power plants [2]. Therefore, BESSs have the potential to become one of the possible replacements for conventional power plants.

A BESS can provide such a critical system service only if it can always ensure a reliable operation. A BESS can be vulnerable to cyber threats that can consciously jeopardize its ability to provide requested functions. Therefore, cybersecurity

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

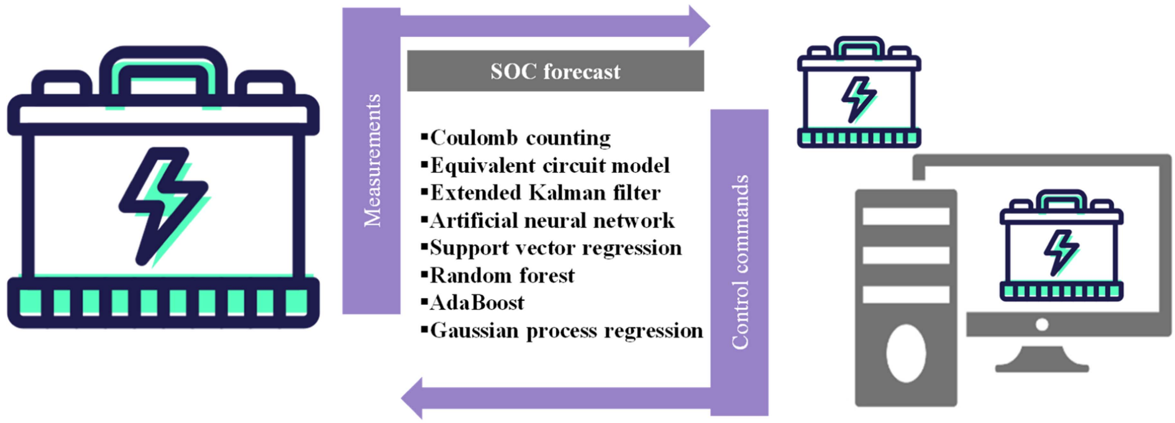

Fig. 1. Methods for a digital twin modeling.

of BESSs is an important concern that should be considered due to the high integration of internet-of-things (IoT) technology [3], [4]. A digital twin of a BESS is an essential tool to enable a safe and predictable functioning of a BESS by estimating the system performance profile. It can be applied for BESS performance monitoring, early fault, and cyberattack detection, improving the efficiency of BESS operation and battery lifetime prognosis [5]. Since frequency regulation is one of the BESS services that has a high economical potential, such a digital twin should be able to reflect the behavior of a BESS providing frequency regulation.

In order to reflect a BESS behavior, one should forecast its state of charge (SOC). The state-of-the-art BESS SOC forecast techniques are model-based, e.g., equivalent circuit models and coulomb counting, which have multiple drawbacks including the need for initial data of the cell, sensing errors, and model inaccuracy [6]. Besides, there are data-driven methods that are robust and accurate according to [3] and [4]. Multiple studies are comparing various data-driven SOC forecasting techniques. For instance, Hasan et al. [7] compared support vector regression (SVR) and artificial neural network (ANN)-based methods for SOC forecasting. Long short-term memory (LSTM) network was applied to predict SOC variation [8]. In [9], the gated recurrent unit (GRU) method was proven to apply to the given dataset for SOC forecast. In [10], SVR, AdaBoost, random forest (RF) along with a fusion model were compared on the same dataset. Sidhu et al. [11] analyzed the results of SVR, RF, and neural-network-based methods for SOC forecast in the electric vehicle (EV) domain. NaitMalek et al. [12] reviewed the methods for SOC forecasting in a microgrid, taking into account ANN, LSTM, and SVR. They compare two methods that are autoregressive integrated moving average and LSTM on the same dataset. However, the comparison did not include other methods that can be applied for SOC forecast.

To the best of our knowledge, no article analyzes commonly applied data-driven methods for the SOC forecast based on the realistic operation of a BESS providing frequency regulation services. In addition, no article uses the real operation of BESS to evaluate SOC forecast methods. Comparison aims to provide sufficient information about the application of the methods so that one can choose the most suitable method for the intended application.

In this article, we propose a digital twin of a BESS providing FCR-N in Nordic Region. We compared several artificial intelligence (AI)-based methods for SOC forecast on the same dataset. Those are are ANN methods, such as feedforward neural network (FNN), GRU, LSTM; RF, AdaBoost, and SVR. We shortlisted the data-driven methods that can be applied for the BESS SOC forecast, and compared them on the datasets of both simulated and real BESS operation. The methods that can be potentially utilized for a SOC forecast are depicted in Fig. 1.

The rest of the article is organized as follows. In Section II, we study the applications of a BESS digital twin. Section III describes the approaches for the BESS SOC forecast that will be compared in the article. In Section IV, the case study that includes a numerical comparison of described methods on a simulated and real dataset is presented. We compare different approaches based on their accuracy, computational heaviness, speed, and sensitivity to input data.

II. Applications for a BESS Digital Twin

The development of a BESS digital twin as well as forecasting its SOC is an important tasks approached by various researchers. In this section, we overview the potential applications of a BESS digital twin.

A. *Potential Digital Twin Applications*

A digital twin is a tool that was first introduced in the IoT domain [13]. Nowadays its application is extended to other domains. It contributes to Industry 4.0 project realization [14]. The idea behind a digital twin is to predict the main features of a physical object to form its virtual copy so that it can be used for decision-making [14]. There are various applications for a digital twin of a BESS including battery performance evaluation and diagnostics, battery degradation estimation, and cyberattack detection [15]. In order to make a BESS digital twin, one has to forecast the SOC of a BESS since this is a critical parameter to evaluate a BESS state. Furthermore, lifetime prognosis can be

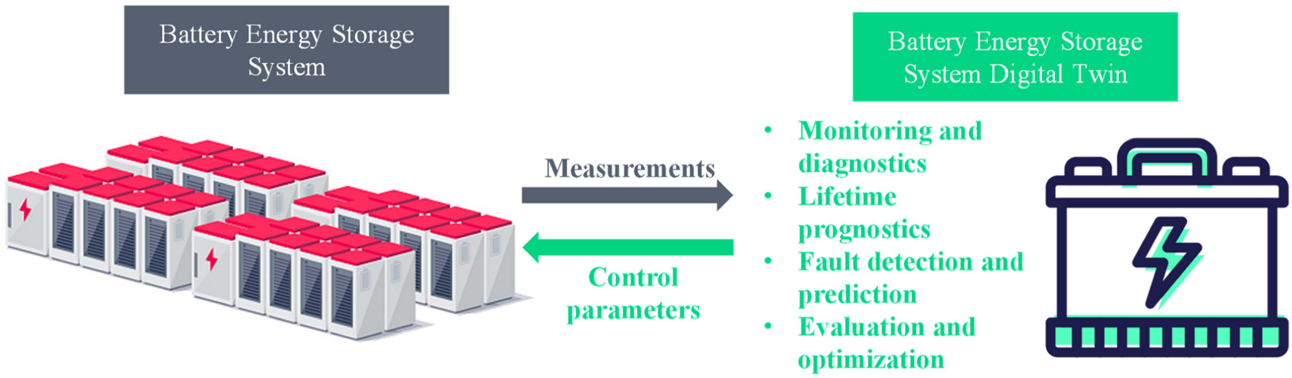

Fig. 2. Methods for a digital twin modeling.

carried out using an state of health (SOH) forecast to improve the quality of SOC prediction towards the end of the battery lifecycle. These parameters cannot be measured and have to be evaluated through voltage, current, and temperature. There are model-based and data-driven estimation techniques used for SOC computation. The work of a BESS can be optimized using a digital twin for battery management [4], [16].

As a result, a digital twin model has to be applicable for both short-term applications, e.g., battery performance monitoring, safety, and cyberattack detection, and long-term applications, e.g., lifetime estimation [5]. A digital twin can considerably improve fault detector performance. An important part of a fault detection system is the early detection of system faults. It allows mitigating system errors at an early stage before they can cause significant system damage [16].

One of the BESS digital twin applications is to develop the most efficient operation strategy for a utility-scale battery taking into account different performance patterns. Furthermore, big data from a BESS operation can be analyzed in order to optimize a system planning process to select the most efficient planning strategy [16]. Digital twins also proved to be useful for a vehicle virtual representation while applied in the EV domain. The development of new EVs and the research related to improving battery efficiency can be speeded up by utilizing predictive testing and smart modeling [13].

Battery diagnostics is one of the primary applications of a digital twin since it allows ensuring safe and reliable BESS operation. There are conventional battery diagnostic techniques that include spectroscopy, physical, and electrochemical methods for an SOC estimation as well as data-driven methods that utilize machine learning (ML) and AI in order to analyze a behavior pattern and detect a system breakage at the early stage. AI-based digital twins have a high potential to be utilized in real-world applications as an online diagnostic technique [5]. Data-driven solutions allow robust system diagnostics onboard minimizing its cost without compromising accuracy. Data-driven approaches detect correlated features of an SOC profile and utilize the data to detect the possible failure [5]. Furthermore, a BESS digital twin can be applied for cyberattack detections. It can be utilized to detect cyberattacks against a BESS. BESS digital twin can mitigate cyberattack negative influence on the system operation by generating pseudomeasurements to substitute the corrupted data with a realistic forecast. It also helps to maintain a BESS observability and reliability of a BESS performance [3]. Another important application of a digital twin is lifetime prognosis that estimates battery's state of health. BESS degradation pattern may vary based on the way the BESS is exploited. Therefore, it is important to evaluate and forecast battery state of health to be aware of a BESS real capacity. The functions of a BESS digital twin are presented in Fig. 2.

A digital twin can be based on one forecasting technique as well as a combination of those. The models can be improved using "what-if" scenarios that would take into account the maintenance and breakage of the particular system component. A digital twin has to consider not only short-term dependencies, such as a charge–discharge cycle, but also long-term dependencies, e.g., aging due to different factors. To evaluate the quality of a digital twin performance, the forecast has to be compared to authentical measurements. However, besides being accurate, the digital twin has to have a moderate computational speed and complexity so that it can be applied for real-time or near real-time applications [14].

B. Digital Twin for a BESS Providing Frequency Control

Frequency control is a critical system service that is aimed to maintain system frequency within the required limit. It is one of the most spread ancillary services provided by transmission system operators ensuring system stability. BESSs have the potential to substitute conventional generators that traditionally provide frequency control service in the grid. BESSs have a lower response time than conventional power plants and, therefore, can be utilized for frequency control [17].

FCR-N is one of the most commonly spread frequency services. It uses a frequency droop to react to the system frequency fluctuation by changing BESS operating point. To support frequency control services, the digital twin has to provide a short-term forecast for the next minutes. Moreover, since digital twin is used for fault or anomaly detection, it is critical to minimize the maximum error (ME) of the forecasted value so that inaccuracy of the forecast will not result in a false warning. In order to test the shortlisted methods applicability for an SOC forecast

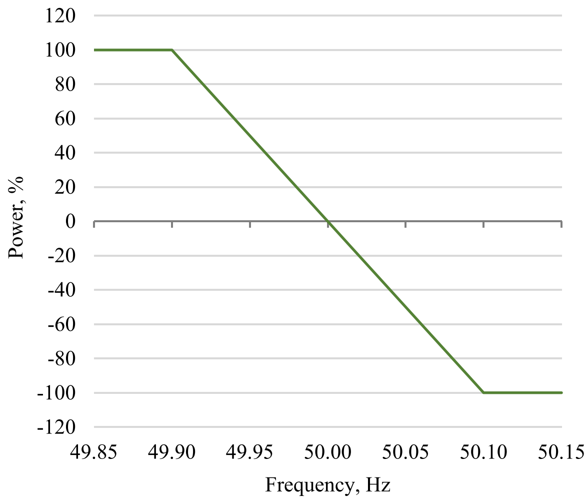

Fig. 3. FCR-N droop.

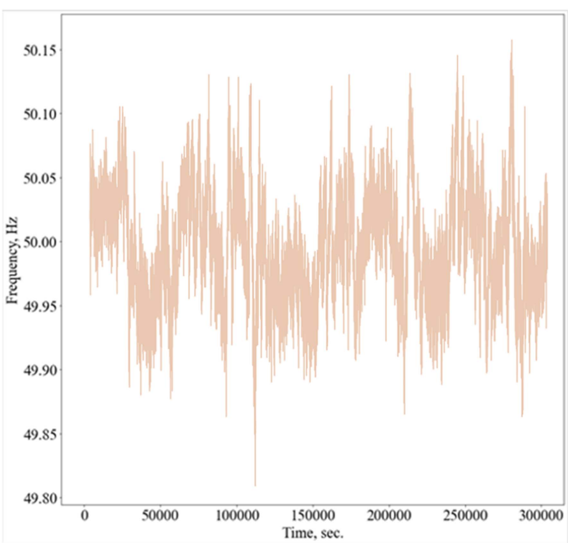

Fig. 4. Frequency dataset for the case study.

for a BESS providing FCR-N, we simulated the operation of a 1 MW/1 MWh utility-scale battery to compare the methods that were used in the literature to forecast SOC. These methods were described in detail in previous section.

In this article, we generate a training dataset based on real frequency data for 10 months, 1 MW/1 MWh utility-scale battery, and a frequency droop depicted in Fig. 3 [18]. It is divided into training (80%) and testing (20%) sets. A total of 48 h fragment of the 8-month training dataset is depicted in Fig. 4. The testing set contains 2 months dataset. Furthermore, the methods were tested for a shorter testing sets of 1 to 6 months due to the natural limitation of the application. Since in many cases the forecast is required for a recently installed BESS, it is important to choose a forecasting method that does not require a big training set for an accurate performance. The input size at all times was 1-h data with a resolution per minute, while the output is a vector of 5 min.

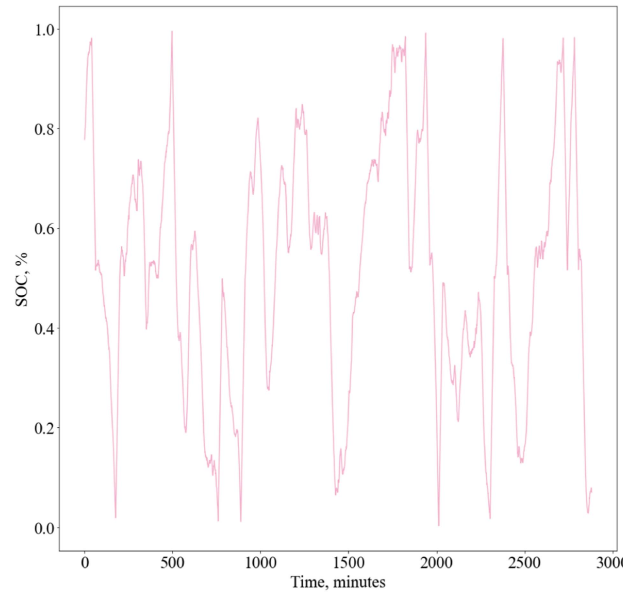

Fig. 5. Two days SOC dataset for the case study.

The data are preprocessed by applying a MinMax normalization with missing data removed from the dataset.

The dataset is obtained using the following formulas (1) and (2) and depicted in Fig. 5

$$\mathrm{SOC}_N = \mathrm{SOC}_{N-1} - \frac{P_{In} \cdot P_{\%,N-1} \cdot \eta \cdot \Delta t}{W}. \tag{1}$$

In which, SOC_{N-1}—SOC on the time step $N-1$, P_{In}—power capacity of a BESS, η—a BESS efficiency, W—energy capacity of a BESS, and Δt—the duration of the control period. The $P_{\%,N-1}$ is the share of a battery installed capacity. The BESS efficiency η depends on multiple factors, such as the operating point, the BESS age, lot-to-lot variation, and mode of a BESS; however, for the given application, we assume that it is equal to 90% [19]. The formula for it is based on the FCR-N droop that is depicted in Fig. 3 [18], where f_N is a frequency of the system

$$P_{\%,N-1} = 50\,000 - 1000 \cdot f_N. \tag{2}$$

III. Methods for BESS Digital Twin Modeling

In this section, we overview the ML methods that can be potentially used to model a BESS SOC proving a detailed description of each method shortlisted for implementation in the case study. These methods can be potentially applied for the BESS digital twin modeling.

A. Support Vector Regression (SVR)

SVR is the method to review and the principle behind it is called a structured risk minimization principle. SVR aims to minimize a generalization error upper bound. A set of high-dimensional linear functions is a base for a regression function using [20]

$$G = w\varphi(x_i) + b \tag{3}$$

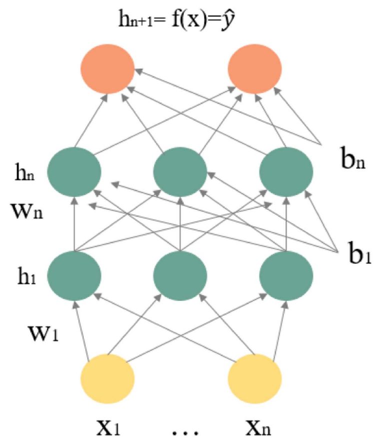

Fig. 6. FNN schematics.

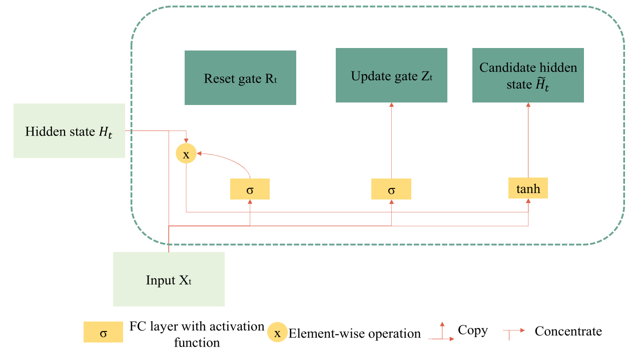

Fig. 7. Structure of GRU.

where $\varphi(x_i)$ represents the features of the input, while *w* and *b* are coefficients calculated minimizing regularized risk function. There are three main parameters to change in order to improve the result of the time-series forecast. First, different kernels allow a finding of a hyperplane in the higher dimensional space. Increasing the number of dimensions is necessary in case it is not possible to find a separating hyperplane in a given dimension. Radial basis function kernel is recommended to be used as a starting point for a forecast since it works the best for a wide range of datasets. The decision boundary that can be also considered as a demarcation line that divides positive and negative examples comes from the support vector machine used for clustering and applied in SVR for a time-series forecast.

B. Feedforward Neural Network (FNN)

FNN is a neural network in which the information flow in a neural network is unidirectional. These are also called multi-layered networks of neurons. The structure of FNN consists of input and output layers connected through hidden layers. This type of neural network is a combination of simple models that are sigmoid neurons.

This model is utilized to work with the dataset that cannot be linearly separable. In order to enable the algorithm to do a nonlinear separation, multiple neurons are required. FNN can contain one or more hidden layers that are connected as depicted in Fig. 6. For each connection, there is a weight and bias value assigned. In the preactivation step, the weighted sum of the element from the previous layer and the corresponding bias are computed for all the elements. Furthermore, the output of the preactivation layer serves as an input for an activation function, e.g., sigmoid or tanh. The output is predicted by applying an activation function [21].

C. Gated Recurrent Unit (GRU)

Another method to overview is the GRU method, which is one of the ANN-based methods. To mitigate the weaknesses of a standard RNN, update and reset gates are added to the algorithm. The GRU RNN can be defined through a set of equations. The algorithm exploits sigmoid and tanh functions to form the gates. The Adam optimizer was implemented to find the minimum of the loss function which is as follows (4) [9]:

$$\frac{\sum_{n=1}^{N}\left(\tilde{y} - V \odot h_{\text{out}}\right)^2}{N} \tag{4}$$

where $\tilde{y}$ is the real dataset sample, V is the vector of weights that are updated, h_{out} is the output vector, and N is the number of samples in the dataset.

The gate structure of GRU is expressed using the formulas below. It is depicted in Fig. 7. The input data for each hidden layer consists of an input of the particular length and hidden state from a previous time step. The length of input and output data, as well as a number of hidden layers, hidden states, and a type of an activation function, is defined depending on the type of dataset. The weight matrixes *W* and biases *b* are updated as a result of solving an optimization problem using Adam optimizer minimizing the following [9]:

$$\begin{aligned}
z_t &= \text{sigmoid}\left(XW_{xz} + h_{t-1}W_{hz} + b_z\right)\\
r_t &= \text{sigmoid}\left(XW_{xr} + h_{t-1}W_{hr} + b_r\right)\\
\tilde{H}_t &= \tanh\left(XW_{xh} + \left(r_t \odot h_{t-1}\right)W_{hh} + b_h\right)\\
h &= \tanh\left(z_t \odot \tilde{H}_t + \left(1 - z_t\right) \odot h_{t-1}\right)
\end{aligned} \tag{5}$$

where W_{xz}, W_{hz}, W_{xr}, W_{hr}, W_{xh}, and W_{hh} are weight parameters, and b_z, b_r, and b_h are bias vectors. z_t is update gate, r_t – reset gate, h is a vector of hidden units, and $\tilde{H}_t$ is a candidate for a new hidden state.

D. Long Short-Term Memory (LSTM)

LSTM is the other method to review and it belongs to an RNN type of forecasting algorithm, which is depicted in Fig. 8. LSTM was created by Hochreiter and Schmidhuber in 1997 [22]. A typical LSTM consists of the cell, an input gate, output gate, and forget gate. The gates system allows to save or delete the data using "forget gate level." The main reason for applying these gates is to include very long time lags into a forecast.

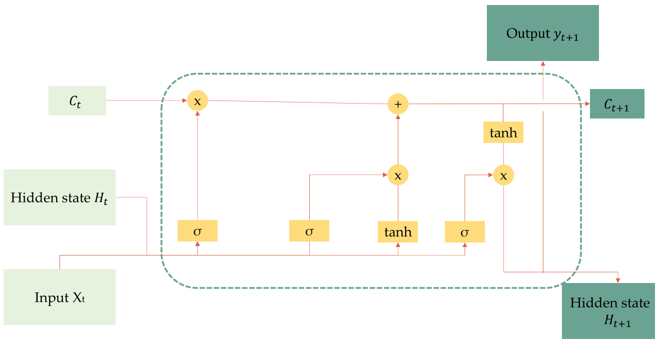

Fig. 8. LSTM schematics.

E. AdaBoost

Adaptive boosting that is also referred to as AdaBoost is an ML algorithm that applies a technique following a decision tree model which depth is equal to 1. This algorithm is aimed to be applied for purposes of classification and regression. In this article, we concentrate on a regression since it can be applied for a time-series forecast. The method increases the weight of those cases that are hard to classify, while those that the algorithm can handle well are given a smaller weight.

The AdaBoost algorithm can be visualized as a forest of stumps that contains one node and two leaves. It combines multiple weak classifiers in order to form a strong classifier, which is a principle behind an AdaBoost algorithm. Initially, the same weights are assigned to every dataset element. The model behind AdaBoost is a decision tree. The tree that contains a node and two leaves is called a stump. This model is applied to a dataset that contains random samples with substitutes from an original dataset with the probabilities corresponding to sample weights [23].

F. Random Forest (RF)

RF is a collection of decision tree algorithms that is an extension of bootstrap aggregation of decision trees. This aggregation of a group of decision trees is made in which each tree is made from a bootstrap sample in the training set. Using RF regression for time-series forecast is a majority prediction across the decision trees. Each decision tree fits a partially different training dataset that has a slightly different performance of the algorithm. A decision tree is formed based on a randomized set of predictors and is decorrelated with others. The L structured based classifier $h(X, \theta_k)$, where $k = 1, \ldots, L$, while θ_k – is a family of identically distributed random vectors that are independent.

All the generated decision trees are combined using bootstrap aggregation. This algorithm randomly chooses n observations with replacing from S_n, where each observation can be selected with a probability of $1/n$. From the pool of bootstrap samples, some of them are selected $(S_n^{\theta 1}, \ldots, S_n^{\theta q})$. To these samples, the previous decision algorithm is applied so that the collection of trees $q : \hat{h}(X, S_n^{\theta^1}), \ldots \hat{h}(X, S_n^{\theta^q})$, in which the output that corresponds to each decision tree $\hat{Y}_q = \hat{h}(X, S_n^{\theta^q})$ [24].

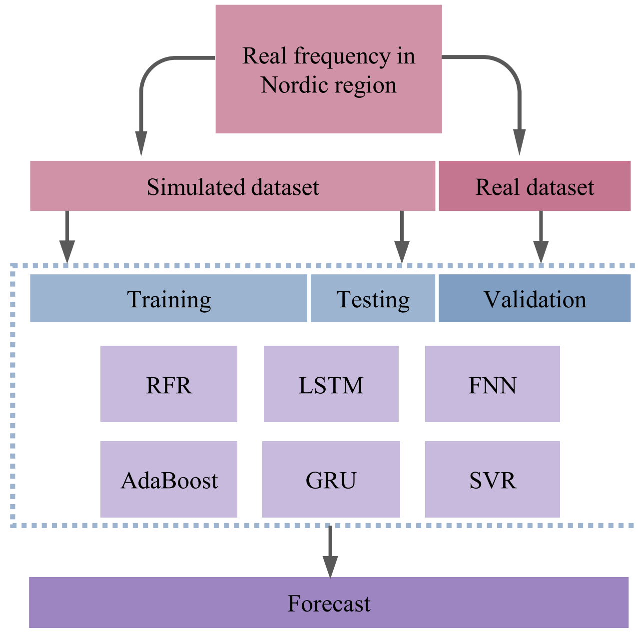

Fig. 9. Flowchart with case study description.

Each tree slightly overfits the training set, however, since it makes these trees different which minimizes the prediction error through having less correlated predictions or prediction errors, it performs better than normal decision tree models.

IV. Case Study

In order to evaluate the applicability of ML methods to develop a BESS digital twin, we compare the ML algorithms described in the previous sections. The flowchart of the case study in Fig. 9 shows the steps required to evaluate the methods. It contains obtaining simulated and real datasets, preprocessing obtained dataset, and applying it for training six different ML-based algorithms. Furthermore, we test the algorithm on the simulated dataset and validate it on the real dataset obtained using an experimental setup in the laboratory.

A. Metrics for Comparison

The algorithms' accuracy is evaluated using mean absolute error (MAE) and ME that are computed using following formulas, respectively [8], in which x_i is the obtained forecast vector, while x is the sample of authentic values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \tag{6}$$

$$\text{ME} = \max |x_i - x| . \tag{7}$$

The dataset was utilized to train and test the algorithms mentioned above. First, the algorithms were trained on datasets of different sizes to forecast quality. The accuracy of the forecasts was compared based on the average and ME observing the influence of a dataset on the forecast quality. The training set of 8 months was used in the combination with testing sets

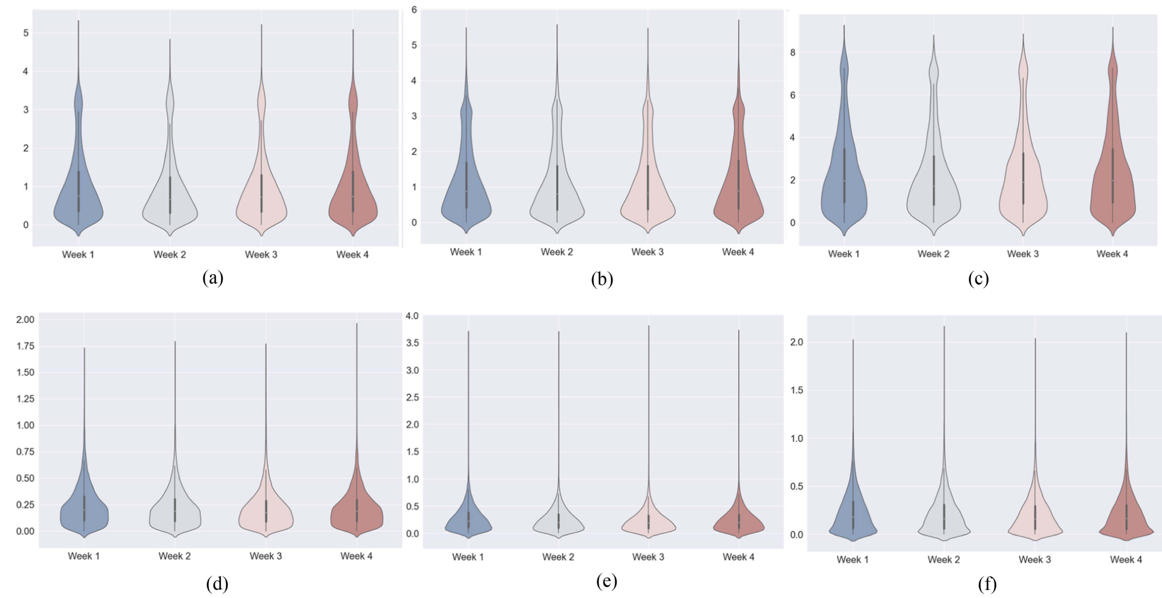

Fig. 10. Comparison of the errors for the week 1–4 training set forecast. (a) Error distribution for different weeks, GRU, %. (b) Error distribution for different weeks, LSTM, %. (c) Error distribution for different weeks, SVR, %. (d) Error distribution for different weeks, AdaBoost, %. (e) Error distribution for different weeks, FNN, %. (f) Error distribution for different weeks, RF, %.

corresponding to different weeks. The results of these tests are summarized in this section.

B. SVR Performance Evaluation

The first method that was applied to a dataset is an SVR. This method can be considered computationally heavy since the training based on a 1-month dataset took 12 807 s. The accuracy provided for a first-minute forecast can be considered sufficient since ME does not exceed 4.5%, while the average one is 3.577%. MAE is also low and on average is equal to 0.544%, while the maximum is equal to 1.5%. Nevertheless, the growth of ME for the next minutes' forecast is considerable. For example, for minute 3 maximum ME exceeds 10%, while for minute 5 its average is equal to 15.43% with the minimum possible ME equal to 14.822% for a forecast based on 6-month training data so the accuracy of the method is considered low.

In order to estimate the method's sensitivity toward the testing set, the errors appearing while testing during different weeks were depicted in Fig. 10. We observe that week-to-week variation is relatively small can be neglected. Despite the maximum MAE for a fifth-minute forecast being relatively low and equal to 1.421, it is not recommended to apply SVR for a SOC forecast since the ME is too high to rely on it for fault or anomaly detection.

C. FNN Performance Evaluation

FNN is a second method that was evaluated using the training and testing set mentioned above. The computational effort is considered to be low since it takes 2022 s to run a month-based forecast, while it takes 15 121 s to run 6-month training-set-based forecast. The average ME for a first-minute forecast is equal to 3.688%, while the maximum ME remains below 4.2%. The average MAE is 0.428% as well as the maximum MAE is 0.671%. The accuracy of FNN is slightly higher than SVR while requiring significantly fewer resources. The forecast of the next 3 min is done with an ME below 10% and MAE below 1%, however, the fifth minute is forecasted with an average ME of 15.277% and an average MAE is equal to 1.459%. This ME is too high to consider a forecast reliable.

The sensitivity of the method to different weeks' testing sets can be considered low as depicted in Fig. 10. The method can be used for up to 3 min forecast in case of the limited availability of the computational capacity.

D. GRU Performance Evaluation

GRU is the type of ANN method that has low computational heaviness since it requires only 2430 s and can be carried out using a laptop. In order to make 1-month-based forecast it takes 2430 s. Six-month training-set-based forecast is done within 5700 s. The average ME of the first-minute forecast is equal to 5.805%, which is higher than the ME of previous methods; however, the ME of the fifth-minute forecast remains equal to or below 10%. Along with that, the average MAE is equal to 1.12% and 1.738% for the first and fifth minutes, respectively. The best scenario is achieved for a 4-month training set, in which the ME is 5.789% and 9.241% for the first and fifth minutes, respectively. MAE is 1.321% and 1.738% for the first and fifth minutes, respectively, in case we consider the best scenario minimizing the ME of the last minute. In case we choose the best case based on the lowest MAE, it is reached using a 5-month forecast with the ME of 4.739% and MAE of 0.765% for the first minute. The ME for the fifth minute is equal to 10.449% along with a MAE of 1.863%. The algorithm is not sensitive to variations in the testing data as shown in Fig. 11.

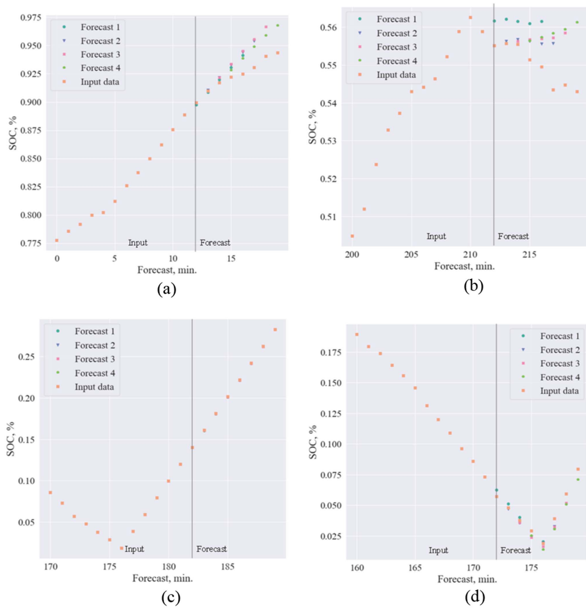

Fig. 11. Forecast by RF for different cases. (a) SOC forecast for 0–20 min, %. (b) SOC forecast for 200–220 min, %. (c) SOC forecast for 170–190 min, %. (d) SOC forecast for 160–180 min, %.

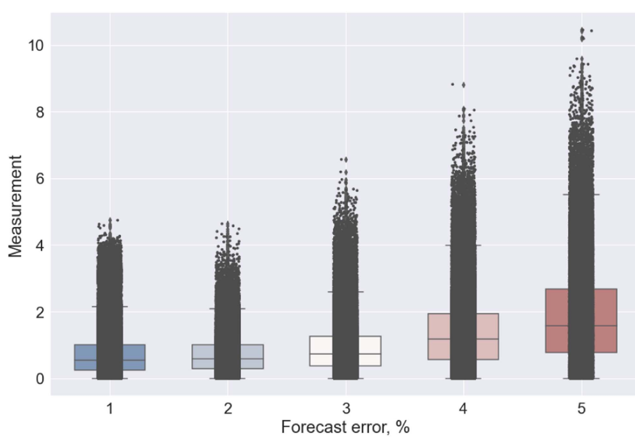

Fig. 12. Error distribution for 5-month training set for 5 min ahead forecast, GRU.

In Fig. 12, we depict an error distribution for a GRU forecast with a training set of 5 months for each minute of 5 min ahead forecast. We observe that for the first 3 min the forecast is of quite high quality with an MAE of approximately 1%, while 4 and 5 min are forecasted with a higher error.

E. LSTM Performance Evaluation

LSTM is a type of RNN that can be applied for an SOC forecast. LSTM performance is relatively similar to GRU, it takes 1455 s to carry out a 1-month input-based forecast, which can be considered as a low computational heaviness. The accuracy of the LSTM method is also similar to GRU and is considered high since the first-minute forecast average MAE is equal to 1.168%, while ME is 6.219%. All 5 min can be forecasted with an ME below 10%, while the average ME for all the training set sizes is 10.092% and an average MAE is 1.757 %. The trend of sensitivity of the ME toward input size is the same as GRU. Increasing the training set size generally leads to a lower MAE and ME. The best forecast from a minimum ME for a first-minute viewpoint is obtained using 6-month input with an ME below 5% and 10.5% for the first and fifth minute, respectively. The minimum MAE appears using 4-month-based forecasts with a value of 1.068% and ME of 6% and 9.23% for the first and the fifth minute, respectively.

Concluding, the high accuracy and relatively low computational time make the LSTM method suitable for application for digital twin modeling, being similar to GRU. Therefore, an additional investigation would be required to choose between these methods for a practical application.

F. AdaBoost Performance Evaluation

AdaBoost has the highest potential to be applied for a BESS digital twin, despite it is relatively computationally heavy compared to the previously mentioned methods. It requires 3573 s to forecast an SOC based on one month's input data, which is considered as average. However, the accuracy of the forecast is considerably higher than RNN-based methods or SVR, especially for a first-minute forecast. Thus, the average ME and MAE for a first-minute forecast are 2.484% and 0.219%, respectively. The sensitivity to a dataset size is very little since the ME for the first and fifth minute of one month-based forecast is 2.6% and 10.768%, respectively, while the same values for the best of 5-month input are 2.371% and 9.941%. As we observe, despite this method producing a very accurate first-minute forecast, for the fifth minute the ME remains around 10%. Nevertheless, the improvement of the forecast can be noticed through a lower MAE, which average is equal to 1.12%. The algorithm is not sensitive to variations in the testing data as shown in Fig. 11. We conclude that AdaBoost can be recommended for application in a BESS digital twin modeling since it is very accurate and reasonably computationally heavy. An additional advantage of the method is that even a 1-month training set is enough to obtain an accurate forecast.

G. RF Performance Evaluation

The last method to be discussed is RF. This method is the most computationally heavy since it requires 14540 s to forecast SOC based on a 1-month forecast. However, it is also the most accurate method of all the abovementioned. The average ME and MAE for the first-minute forecast are 2.301% and 0.212%, respectively. Moreover, the same values for the fifth-minute forecast are 1.045% and 10.133%. By using only one-month input data, we obtain a forecast with an ME for the first and fifth minute of 2.159% and 10.298%, respectively. Three months of input data are required to achieve the same performance by AdaBoost. In case a 3-month training set is used for RF, the accuracy of the obtained result cannot be reached by any method with any input in the framework of this case study. The best-case scenario from the smallest ME and MAE for the first minute is obtained for an 8-month dataset with the values of 2.345% and 0.203%, respectively. For the fifth minute, the ME is 10.085% and MAE is equal to 1.014%. The increasing length

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

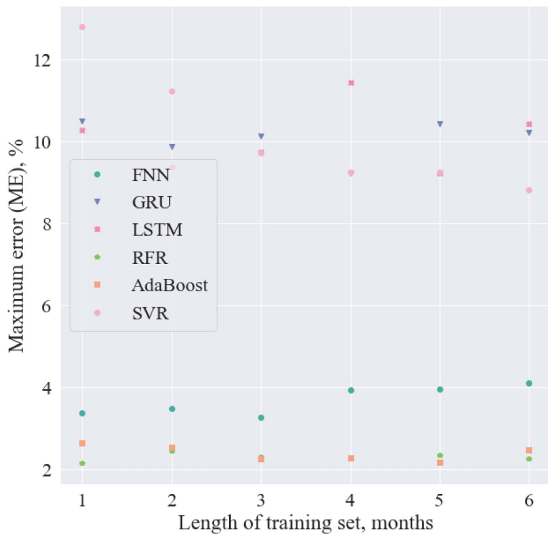

Fig. 13. Comparison of the ME for a first minute forecast for different methods based on the training sets with different length.

of the training set results in decreasing ME and MAE for a first-minute forecast.

Since RF shows the best performance among the algorithms under consideration, we take a closer look at the forecast that it produces. Fig. 11 depicts the 5 min ahead forecast for different cases. We observe that for the situations in which the variables follow the same trend, the forecast is very precise while in case the trend is changing, small variations might appear. Nevertheless, one can observe that the quality of the forecast is quite high for most cases. The most complicated situation for the forecast is when a BESS is changing between charging and discharging each couple of minutes.

H. Methods Comparison

We have looked in detail at each method's performance features. This section is devoted to the methods comparison. We provide a qualitative comparison to understand the general features of each method. Moreover, we analyze the best-case scenario for a BESS providing frequency control dataset in order to numerically compare the methods.

We consider the sensitivity of the method toward the training set size. As depicted in Fig. 13, we find out that the sensitivity of GRU and LSTM is the highest for a 1-min forecast. FNN and SVR methods are relatively stable for a first-minute forecast; however, as can be seen in Table I, it is evident that they provide a low-quality forecast for the fifth minute. For RF and AdaBoost, the sensitivity is lowest with the highly accurate forecast for the first minute; however, it results in the same ME with GRU and LSTM for the fifth minute.

In Table I, we analyze the best scenario for each dataset and compare the performance of each method. We observe that the best method from the accuracy viewpoint is RF method that is closely followed by AdaBoost. Nevertheless, AdaBoost is much less computationally heavy while it provides relatively similar performance. If we compare less computationally heavy

TABLE I
BEST SCENARIO SUMMARY

Method	Dataset length, months	CPU time, s	ME, 1 min, %	ME, 5 min, %	MAE, 1 min, %	MAE, 5 min, %
SVR	3	38422	3.594	15.592	0.271	1.339
FNN	3	2022	3.265	15.029	0.351	1.513
GRU	5	5032	4.739	10.449	0.765	1.863
LSTM	4	6213	6.151	11.452	1.068	1.644
AdaBoost	4	9782	2.426	9.712	0.219	1.104
RF	4	65402	2.290	9.951	0.206	1.029

TABLE II
SCENARIO PERFORMANCE ACCORDING TO THE CPU TIME (S)

	SVR	FNN	GRU	LSTM	Ada Boost	RF
1 month	12 807	2022	2430	1455	3573	14 540
3 months	38 422	7140	4758	4025	8134	48 858
6 months	73 277	1 5121	5645	10 483	15 719	125 069

methods, which are FNN, GRU, and LSTM, the best performance is shown by the GRU. It shows the best combination of low computational effort while maintaining a relatively accurate forecast for the first to fifth minute forecasts.

As mentioned above, accuracy is not the only criterion based on which we choose the methodology. Table II summarizes the performance of each algorithms on the same high-performance computer. One can observe the duration of the computational process as well as the resources required for the training and testing for the best case according to Table II. The most computationally heavy forecast is done by RF, which provides the best quality forecast. AdaBoost shows a balance between the computation time and the accuracy of the forecast, and RF provides a better quality longer term forecast. LSTM, GRU, and FNN have similar requirements for computation, while GRU shows the best performance out of three. SVR appears to be both computationally heavy and inaccurate.

The multifactorial analysis of the methods is presented in Table III. We compared the methods qualitatively. We estimated what is the minimum training set size needed to obtain the performance of no more than 1% MAE and no more than 10% of the ME for the forecast of the first minute. The size of the minimum dataset is shown in column 4 of Table III. We observe that only in the case of RF and AdaBoost, a 1-month training set is enough to obtain the forecast of the accuracy mentioned above. For the FNN method that was showing good performance for a first-minute forecast, it was not possible to provide a forecast within 10% of ME for a fifth-minute forecast for any size of a training set.

The minimum dataset needed to obtain a forecast with ME of 5% for the first-minute and 10% for a fifth-minute forecast is shown in column 5 of Table III. RF and AdaBoost are also showing good performance; however, they are relatively computationally heavy as shown in Table II. GRU and LSTM perform

TABLE III
SOC FORECASTING METHODS AND QUALITATIVE COMPARISON

Method	Accuracy	Computational heaviness	Minimum dataset size for accuracy, 1–10%	Minimum dataset size for accuracy, 5–10%
SVR	Low	High	–	–
FNN	Low	Low	3	–
GRU	Average	Low	5	5
LSTM	Average	Low	7	6
Ada Boost	High	Average	1	4
RF	Very high	High	1	4

with sufficient accuracy without being computationally heavy. SVR and FNN do not provide sufficient accuracy for domain implementation.

Overall, we conclude that RF and AdaBoost provide the most accurate forecast using the minimum size of a dataset of 1 month. It is recommended to use a dataset of 4 months for an RF and an AdaBoost, if possible. The weakness of applying these methods to solve a practical problem is requirements for a high computational power. In the case, such a capacity is not available, GRU and LSTM can be applied. It is to be noted that RF provides a more accurate forecast while AdaBoost is less computationally heavy. The Gaussian process regression was too computationally heavy to run on both the laptop and using high-performance computing; therefore, it was decided to remove it from the comparison.

I. Validation on a Real Dataset

In order to prove the applicability of the methods for the real test case, the results are validated using the real SOC measurements of a laboratory BESS at the Technical University of Denmark. The setup is a part of the Bornholm smart grid secured by grid connected battery systems BOSS project. For that purpose, the testing setup of a BESS containing the energy storage of 79 kWh and the converter of 55 kVA is selected and FCR-N service is applied to the BESS for 24 h. The results of the comparison are used for choosing the best method for a cyberattack detection algorithm development for a BESS designed as a part of the BOSS project: Bornholm smart grid secured by grid-connected battery systems. The experimental setup is shown in Fig. 14. The dataset of SOC with a minute resolution containing 1440 samples is used to validate the quality of the forecast for 1 month training set. The results aligned with the previous observations in which AdaBoost and RFR methods have shown the best performance, while FNN, LSTM, and GRU performed with slightly lower accuracy.

The best result was achieved by the RFR method with ME of 3.67% and MAE equal to 0.41%. Furthermore, the second-best performance by AdaBoost resulted in an ME of 3.69% and MAE of 0.40%. The third best result was obtained by the FNN method with an ME and MAE of 2.58% and 0.56%, respectively. LSTM with ME of 4.12% and MAE of 1.35% along with GRU with ME

Fig. 14. Experimental test setup at the Technical University of Denmark.

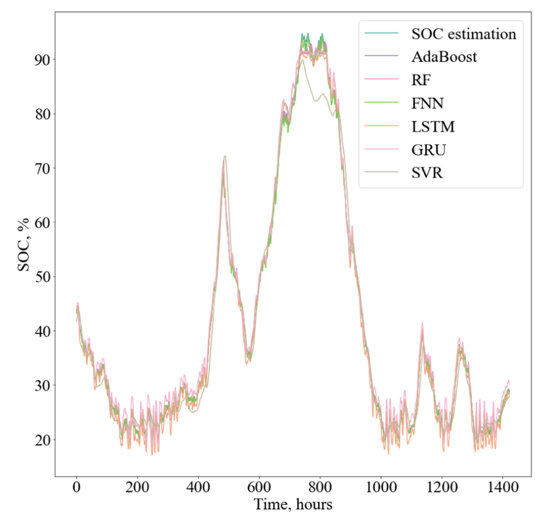

Fig. 15. Validation of SOC forecast using laboratory setup.

of 4.77% and MAE equal to 1.28% can be used as an alternative to the mentioned above methods; however, their accuracy is lower. SVR shows the highest ME of 11.25% and MAE of 1.99% is not sufficient for this application. Overall, the validation has proved that the AdaBoost method shows a reasonable balance between the accuracy and computational heaviness. However, RFR, FNN, LSTM, and GRU are also applicable due to their relatively high accuracy. The forecasts are depicted in Fig. 15.

Furthermore, during the validation step, we proved that the dataset based on realistic frequency measurement in the combination with the droop characteristics can be used to train the algorithm that will be further applied to a real BESS performance forecast with relatively high accuracy.

V. CONCLUSION

A digital twin of a BESS is necessary to ensure the safe and economically efficient operation of utility-scale BESS. The main forecasted value to model a BESS digital twin is SOC. ANN approaches are proved to be applicable for forecasting.

We tested them in the area of BESS digital twin modeling in this article. In the case study, we applied a dataset based on a BESS operation to evaluate the performance of each minute for a short-term, 5-min ahead forecast of the behavior of a BESS providing FCR-N, which a frequency regulation service in the Nordic Region. Moreover, the forecast accuracy was validated further using the real dataset obtained from an experimental setup.

The analysis showed that the RF and AdaBoost provide the most accurate forecast with low sensitivity to a training set size, however, being more computationally heavy. The best result was obtained using a 4-month training set in the combination with the RF algorithm of ME and MAE for a first-minute forecast of 2.29% and 0.21%, respectively. The same values for a fifth-minute forecast were 9.95% and 1.03%, respectively. The lowest ME of 9.84% for a fifth-minute forecast was obtained using 8-months-based RF forecast. The highly accurate forecast that took the least time to produce was done in 3573 s using the 1-month training set and AdaBoost method having an ME below 2.60% and MAE below 0.25%. The best performance for the method with the minimum computational time was a 2-month input-based LSTM forecast for ME for the first and fifth minutes below 6.5% and 9.5%, respectively.

Overall, the most accurate method for given datasets was RF, while AdaBoost was combining relatively short computation time and accuracy. Furthermore, these methods allowed obtaining an accurate forecast from a training set of a limited size. Therefore, these methods are recommended for a digital twin of BESS used for frequency regulation, in case a high accuracy is required. If there is hardware limitation, LSTM and GRU can be utilized compromising the accuracy. The increasing training set size along with choosing RF and AdaBoost as a forecasting technique significantly improves the quality of the forecast for the first 3 min, however, for the fifth minute, it was not possible to bring ME below 9.5%.

The validation using dataset executed from experimental setup has supported the results obtained using the realistic dataset. Moreover, the assumption that realistic data can be used to provide a relatively accurate forecast for a BESS in case of the lack of real measurements was proved.

REFERENCES

[1] B. Mohseni-Gharyehsafa, J. A. Esfahani, K. C. Kim, and H. Ouerdane, "Soft computing analysis of thermohydraulic enhancement using twisted tapes in a flat-plate solar collector: Sensitivity analysis and multi-objective optimization," *J. Cleaner Prod.*, vol. 314, Sep. 2021, Art. no. 127947, doi: 10.1016/J.JCLEPRO.2021.127947.

[2] Z. Hameed, S. Hashemi, and C. Traeholt, "Site selection criteria for battery energy storage in power systems," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, Aug. 2020, pp. 1–7, doi: 10.1109/CCECE47787.2020.9255678.

[3] N. Kharlamova, S. Hashemi, and C. Træholt, "Data-driven approaches for cyber defense of battery energy storage systems," *Energy AI*, vol. 5, Sep. 2021, Art. no. 100095, doi: 10.1016/j.egyai.2021.100095.

[4] N. Kharlamova, S. Hashemi, and C. Traholt, "The cyber security of battery energy storage systems and adoption of Data-driven methods," in *Proc. IEEE 3rd Int. Conf. Artif. Intell. Knowl. Eng.*, 2020, pp. 188–192, doi: 10.1109/AIKE48582.2020.00037.

[5] B. Wu, W. D. Widanage, S. Yang, and X. Liu, "Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems," *Energy AI*, vol. 1, Aug. 2020, Art. no. 100016, doi: 10.1016/J.EGYAI.2020.100016.

[6] S. Cho, H. Jeong, C. Han, S. Jin, J. H. Lim, and J. Oh, "State-of-charge estimation for lithium-ion batteries under various operating conditions using an equivalent circuit model," *Comput. Chem. Eng.*, vol. 41, pp. 1–9, 2012, doi: 10.1016/j.compchemeng.2012.02.003.

[7] A. S. M. J. Hasan, J. Yusuf, and R. B. Faruque, "Performance comparison of machine learning methods with distinct features to estimate battery SOC," in *Proc. IEEE Green Energy Smart Syst. Conf.*, Nov. 2019, pp. 1–5, doi: 10.1109/IGESSC47875.2019.9042399.

[8] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6730–6739, Aug. 2018, doi: 10.1109/TIE.2017.2787586.

[9] Z. Huang, F. Yang, F. Xu, X. Song, and K. L. Tsui, "Convolutional gated recurrent unit-recurrent neural network for state-of-charge estimation of lithium-ion batteries," *IEEE Access*, vol. 7, pp. 93139–93149, 2019, doi: 10.1109/ACCESS.2019.2928037.

[10] Q. Wang, "Battery state of charge estimation based on multi-model fusion," in *Proc. Chin. Automat. Congr.*, 2019, pp. 2036–2041, doi: 10.1109/CAC48633.2019.8997253.

[11] M. S. Sidhu, D. Ronanki, and S. Williamson, "State of charge estimation of lithium-ion batteries using hybrid machine learning technique," in *Proc. IECON Ind. Electron. Conf.*, 2019, pp. 2732–2737, doi: 10.1109/IECON.2019.8927066.

[12] Y. NaitMalek, M. Najib, M. Bakhouya, and M. Essaaidi, "Embedded real-time battery state-of-charge forecasting in micro-grid systems," *Ecol. Complexity*, vol. 45, Jan. 2021, Art. no. 100903, doi: 10.1016/J.ECOCOM.2020.100903.

[13] G. Bhatti, H. Mohan, and R. R. Singh, "Towards the future of smart electric vehicles: Digital twin technology," *Renewable Sustain. Energy Rev.*, vol. 141, May 2021, Art. no. 110801, doi: 10.1016/J.RSER.2021.110801.

[14] S. Singh, M. Weeber, and K. P. Birke, "Advancing digital twin implementation: A toolbox for modelling and simulation," *Procedia CIRP*, vol. 99, pp. 567–572, Jan. 2021, doi: 10.1016/J.PROCIR.2021.03.078.

[15] X. Qu, Y. Song, D. Liu, X. Cui, and Y. Peng, "Lithium-ion battery performance degradation evaluation in dynamic operating conditions based on a digital twin model," *Microelectron. Rel.*, vol. 114, Nov. 2020, Art. no. 113857, doi: 10.1016/J.MICROREL.2020.113857.

[16] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, and D. U. Sauer, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *J. Energy Storage*, vol. 30, Aug. 2020, Art. no. 101557, doi: 10.1016/J.EST.2020.101557.

[17] S. Zhang, H. Liu, F. Wang, T. Yan, and K. Wang, "Secondary frequency control strategy for BESS considering their degree of participation," *Energy Rep.*, vol. 6, pp. 594–602, Dec. 2020, doi: 10.1016/J.EGYR.2020.11.183.

[18] "Technical requirements for frequency containment reserve provision in the nordic synchronous Area" *Eur. Network Transmiss. Syst. Operators Electricity*, 2022. Accessed: Aug. 10, 2021. [Online]. Available: https://www.svk.se/contentassets/4f9ae6d3a8e449c291c9ddda105c7c72/fcr-technical-requirements-14.03.2022.pdf

[19] Y. Chen, X. Yang, D. Luo, and R. Wen, "Remaining available energy prediction for lithium-ion batteries considering electrothermal effect and energy conversion efficiency," *J. Energy Storage*, vol. 40, Aug. 2021, Art. no. 102728, doi: 10.1016/J.EST.2021.102728.

[20] P. Pai, K. Lin, C. Lin, and P. Changc, "Time series forecasting by a seasonal support vector regression model," *Expert Syst. Appl.*, vol. 37, no. 6, pp. 4261–4265, 2010.

[21] N. Kumar, "Deep learning: Feedforward neural networks explained," Accessed Dec. 7, 2021, [Online]. Available: https://hackernoon.com/deep-learning-feedforward-neural-networks-explained-c34ae3f084f1?ref=hackernoon.com

[22] S. Hochreiter and J. Schmidhuber, "Long short term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[23] J. Ajeeth, "Implementing the adaboost algorithm from scratch," Accessed Dec. 7, 2021, [Online]. Available: https://www.kdnuggets.com/2020/12/implementing-adaboost-algorithm-from-scratch.html

[24] Y. Li et al., "Random forest regression for online capacity estimation of lithium-ion batteries," *Appl. Energy*, vol. 232, pp. 197–210, 2018.