IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 February 2024, accepted 22 February 2024, date of publication 29 February 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371883

RESEARCH ARTICLE

Electric Vehicle Battery State of Charge Estimation With an Ensemble Algorithm Using Central Difference Kalman Filter (CDKF) and Non-Linear Autoregressive With Exogenous Input (NARX)

AYODEJI S. OGUNDANA, (Student Member, IEEE), PRANAYA K. TERALA, (Student Member, IEEE), MIGARA Y. AMARASINGHE, (Student Member, IEEE), XUANCHEN XIANG, AND SIMON Y. FOO, (Member, IEEE)

Department of Electrical and Computer Engineering, Florida A&M University–Florida State University College of Engineering, Tallahassee, FL 32301, USA

Corresponding author: Simon Y. Foo (foo@eng.famu.fsu.edu)

ABSTRACT Conventional model-based probabilistic inference methods require increasingly complex models to improve Electric Vehicle (EV) battery State of Charge (SOC) estimation. Deep learning methods gained popularity in recent years with their model free estimations. However, practical constraints such as insufficient training data, model complexity for real time implementation, and generalization on new dataset hinder performance reliability. Another major practical drawback of the data driven deep learning approach is its poor convergence from an unfamiliar initial error state as training dataset does not adequately accommodate these practical error scenarios. This paper proposes an ensemble method that uses a weighted estimate of the Central Difference Kalman Filter (CDKF) and Nonlinear Autoregressive with Exogenous Input (NARX) to accurately estimate SOC in the early stages of degradation. We employ a parallel ensemble estimation method that reduces the estimation bias, improves generalization, accuracy, robustness, and reliability of the estimator. We propose a pre-estimated voting weight to combine the ensemble algorithm and employ the CDKF covariance dependent method as the optimum approach for initializing the ensemble system to achieve a robust convergence performance. The state converges on an average of 136 time-steps when initialized halfway from the true state. The average Mean Absolute Error (MAE) performance of the ensemble method is about 0.5 % with an average training data of about 31,041 time-steps. The model was validated using conventional drive cycle data and was shown to outperform its individual ensemble members and gated Recurrent Neural Networks (RNN) such as Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM) and the Bidirectional LSTM (BiLSTM).

INDEX TERMS State of charge (SOC), battery management system (BMS), bidirectional LSTM (BiLSTM), central difference Kalman filter (CDKF), drive cycles, gated recurrent unit (GRU), long short-term memory (LSTM), nonlinear autoregressive with exogenous input (NARX), SOC convergence.

I. INTRODUCTION

The increasing awareness of the impact of carbon emissions on our environment has led to an upward shift towards electric vehicle (EV) adoption. Many developed countries are looking toward increasing EV adoption by the end of the decade [1], [2]. However, studies have shown that range anxiety is one of the major hinderances to the popular adoption of EV [3], [4], [5]. Nonetheless, studies have suggested that reaching for more accurate and reliable

The associate editor coordinating the review of this manuscript and approving it for publication was Ilaria De Munari.

SOC estimation methods in EV can help alleviate this problem [6], [7], [8].

For EVs to compete with internal combustion engines, charging time has to reduce to the 5-10 minute range [9]. Fast charging tends to momentarily increase battery temperature, hysteresis, and diffusion resistance which adversely affects the ability of simple algorithms to accurately estimate State of Charge (SOC). A robust and accurate SOC estimation algorithm is important to achieve effective fast charging. Moreover, the demanding and sensitive application of EVs require advanced SOC estimation algorithm (SOC estimators) that can enable Battery Management Systems (BMS) efficiently exploit battery pack capabilities [9]. EVs require robust SOC estimation algorithms that can handle the dynamic voltage response of the battery to the EVs aggressive current demand at varying operating temperatures. Maximizing the life cycle of EV batteries also require maintaining the battery at manufacturer's recommended operating SOC. Operating the battery pack outside the recommended SOC can be a safety concern [10]. There has been a lot of research over the past decade on improving the SOC estimation accuracy of BMS. Commonly researched methods are data driven machine learning approach and Bayesian Probabilistic Inference methods. With the recent advancements in deep learning, various deep learning methods have been successfully applied to estimate SOC with admirable accuracy [11], [12]. Gated Recurrent Neural Network (RNN), among other deep learning methods has achieved tremendous success in estimating SOC because of their ability to capture the complex temporal dependencies of battery states [11], [13].

Song et al. combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network [14] to estimate SOC. However, the real time implementation of both deep networks is costly. Vidal et al. [15] achieved an improved SOC accuracy with transfer learning on an LSTM network pretrained on a different battery. However, caution must be taken to avoid domain mismatch and larger datasets are often required to fine tune. Terala et al. proposed a novel high performance Neural Network (NN) SOC estimation approach using Stacked Encoder-Decoder Bi-Directional LSTM [16]. However, they also suffer from fluctuating-SOC [17] effects frequently seen in deep learning applications. In addition, they also require a very long sequence of historical data to produce a stable output [18], [19], [20]. On the other hand, probabilistic inference methods such as Extended Kalman Filters (EKF), Sigma Point Kalman Filter (SPKF), and Particle Filter estimates SOC by using battery models, hence it requires less data for implementation. Plett proposed an Enhanced Self-Correcting (ESC) cell model for implementing EKF [21], [22] and SPKF [23]. Hu et al. also proposed a multiscale framework with EKF to estimate SOC and capacity [24]. The Bayes probabilistic based methods combine cell model with Bayes inference algorithm to eliminate noise uncertainties. However, its accuracy heavily depends on the accuracy of the model developed [16]. Data driven algorithms are preferable to model-based filtering algorithms because they do not require complex model design and parameter identification. However, model based filtering algorithms are better trackers of battery states especially when the state initialization is uncertain [25].

Recent trends have shown tremendous benefits with the hybridization of data driven methods and model-based methods as they are more accurate, robust, and free of fluctuating outputs. He et al. proposed a series Feed-Forward Neural Network with Unscented Kalman Filter (UKF) to filter the fluctuating output SOC [26]. Tian et al. estimated SOC with a LSTM Network in series model with a Kalman filter [27]. Yang et al. also proposed using the UKF to filter out noises in the LSTM output SOC [13]. Similarly, UKF was used to filter out noise output from a Nonlinear Autoregressive with Exogenous Input (NARX) network by Qin et al. [28]. The downside to this series filtering design is that the instability of one estimator may lead to a significant error of the output.

Alternatively, this paper proposes a parallel ensemble algorithm that combines two ensemble members (in this case NARX and Central Difference Kalman Filter (CDKF)) to gain an improved performance than either of its ensemble members by taking advantage of their salient properties [29]. The proposed method estimates the ensemble members concurrently, hence improving the runtime speed of the algorithm. We propose using the closed-loop feed-back NARX network to feedback the output SOC as input at every iteration, hence, mitigating vanishing gradient problem. We also propose recalibrating the NARX network to prevent complete divergence by forcing the feedback SOC at voltage boundary voltages and at specified ensemble variance point. We estimated the weights of the ensembles by using the offline performance inverse of the ensemble member on the Dynamic Stress Test (DST) training. This weight can then be subsequently updated online when the variance of the ensemble members reaches some specified limit indicating a drift in estimations.

The contribution of this paper is summarized as follows:

1. A simple novel method of estimating SOC is proposed using CDKF and a relatively inexpensive NARX which is compatible for online state estimation.

2. An effective convergent method and monitoring technique in an online scenario is proposed.

3. The proposed model achieved an impressive performance accuracy which is robust under various temperatures. The performance was validated with 3 standard drive cycles at 8 discrete temperatures.

The rest of this paper is organized as follows:

Section II discusses the details of the experimental data and Open Circuit Voltage (OCV) data preprocessing method.

Section III explains the proposed model and algorithm.

Section IV discusses the results.

Section V presents the conclusion.

II. DATA AND METHODOLOGY

A. PERFORMANCE METRICS

SOC performances can be viewed in terms of estimation speed, accuracy, robustness, and rate of convergence from inaccurate estimates. The proposed algorithm has a constant computational cost (O(1)), because input at every iteration is fixed. The computation speed will vary only with the BMS processor design. Hence, this paper focuses more on accuracy, robustness, and convergence performance. The Mean Absolute Error (MAE) is used to evaluate the performance accuracy while the number of iterations in seconds is used to evaluate the convergence performance.

B. EXPERIMENTAL DATA

Publicly available CALCE dataset was used for modelling and performance verification [30]. The two important datasets needed are the Open Circuit Voltage (OCV) data and the dynamic drive cycle data. The OCV data was collected by fully charging and discharging the battery at C/20 [30], [31]. The dynamic data set was collected by passing current profiles of DST, US06 and FUDS through 18650 Lithium Iron Phosphate (LFP) batteries at 8 discrete temperatures and the voltage response of the battery was obtained. The average data points collected per temperature for the training data is about 31,041. The nominal capacity of the battery tested is 1.1 Ah, the nominal voltage is 3.3 V, and the upper and lower cutoff (bound) are 3.6 V and 2.0 V respectively. The battery continuous discharge current was limited to 30 A [31].

C. OCV DATA AND MODEL

The OCV is the terminal voltage of the battery when the battery is in equilibrium [32]. The OCV data is required in the output equation of the CDKF. It is experimentally impossible to obtain the OCV for every discrete state. Hence, there is a need to approximate using a polynomial expression. The CALCE OCV data were obtained at various temperatures: −10°C, 0°C, 10°C, 20°C, 25°C, 30°C, 40°C, and 50°C by charging and discharging the test cell at C/20 at the specified temperature [26], [31]. A commonly used approach of estimating the true OCV by taking the average of the charge and discharge OCV [33], [34] was applied. OCV linear function of temperature and SOC was developed as shown in (1) and (2) for possible computation online by BMS. The pseudo-inverse of (1) can be performed offline to obtain variables *v1* and *v2* from estimated true OCVs on the right-hand side of the equation. Having obtained *v1* and *v2*, (2) can then be performed online to estimate OCVs at untested temperatures. Experimental true OCV value at 25°C was used to verify the performance of the Least Square Approximation. An acceptable MAE value of 4.23 mV was obtained as shown in Fig 1(a) to validate the performance of the OCV model. It should be noted that the accuracy of the CDKF highly depends on the accuracy of the OCV approximations.

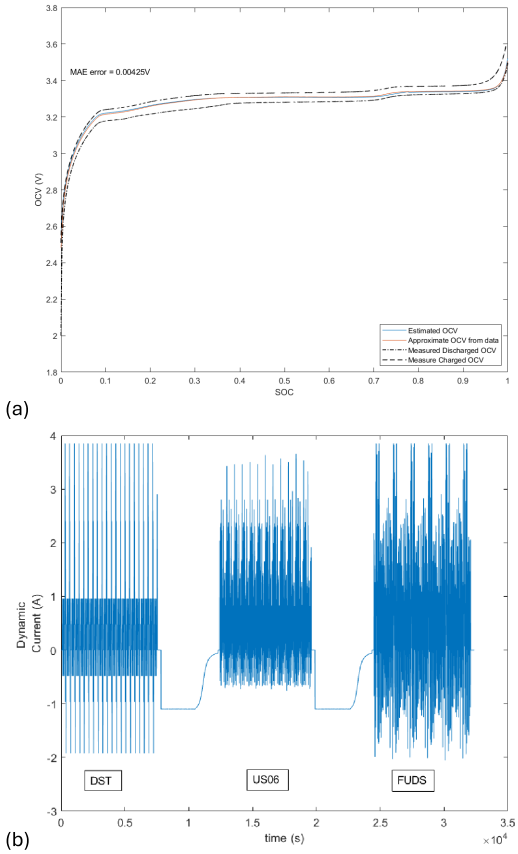


FIGURE 1. (a) OCV plots of the LFP battery at 25°C showing the estimated OCV (solid blue line), approximate OCV from data (solid red line), measured discharged OCV (short black dash line) and measure charged OCV (long black dash line) (b): DUF drive–cycle dynamic charge-discharge profile at 25°C.

$$\begin{bmatrix} 1 & T1 \\ 1 & T2 \\ 1 & T3 \\ 1 & T4 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} v1(z) \\ v2(z) \end{bmatrix} = \begin{bmatrix} OCV1(z, T1) \\ OCV2(z, T2) \\ OCV3(z, T3) \\ OCV4(z, T4) \\ \vdots \end{bmatrix} \quad (1)$$

$$OCV\,(z, T25) = v_1 + (T25 * v_2) \quad (2)$$

where *T1, T2, T3, T4* and *T25* are the test temperatures. *v1, v2* are the estimated parameters. z is the SOC, and *OCV* is the input OCV data.

D. DYNAMIC STRESS TEST (DST)

DST is a 6-minute sequence of power steps with seven discrete power levels designed by the US Advanced Battery Consortium (USABC) to simulate a dynamic charge-discharge regime of an EV battery demand [25], [35]. The DST profile is scaled and repeated with respect to battery and test demand. DST is a good excitation signal for obtaining battery dynamic properties. This dynamic profile shown in Fig 1(b) is used to obtain the preliminary performance and weights of the proposed algorithm.

E. SUPPLEMENTAL FEDERAL TEST PROCEDURE (US06)

US06 is a 596 second test procedure used to simulate an aggressive acceleration of highway driving condition at a top speed of 80.3 mph and an average speed of 48.37 mph over an 8.01-mile distance. The US06 is also referred to as EPA US06 or Supplemental Federal Test Procedure (SFTP) [36], [37]. Similar to other drive cycles, the charge-discharge current can be scaled to the maximum current of the battery under test. The US06 drive cycle is used to verify the performance of the proposed algorithm as discussed in Section IV of this paper.

F. FEDERAL URBAN DYNAMOMETER SCHEDULE (FUDS)

FUDS is an end-to-end 1372 second power dynamic regime estimated using FUDS vehicle time-velocity profile, scalable to the peak power or capacity of the battery under test [35]. This profile can be scaled to battery under test parameter requirements. Fig 1(b) is a plot of the driving profiles scaled for the battery under test. The FUDS drive cycle is also used to verify the performance of the proposed algorithm. The integration of the 3 drive cycles (DST-US06-FUDS) is abbreviated as DUF drive profile in the rest of the paper.

III. ESTIMATION ALGORITHMS AND MODELS

A. NARX

NARX is a recurrent dynamic NN architecture used to model input-output nonlinear systems. NARX uses the present input, past input and past output to model the nonlinear system dynamic [28]. NARX is employed in the battery SOC estimation due to the time series dynamic properties in Lithium-ion batteries [38]. In a NARX network, the output of the network at a given time step is fed back as an input to the network at the next time step, along with its delayed input. This is the major architectural contrast between NARX and other RNN where feedback connections are mainly found in the hidden layers [39]. NARX as a function of the input and output delay expression is shown in (3) below [38], [40]:

$$y\left(n+1\right)=f\left(y\left(n-d_y\right);x\left(n+1\right),\ldots,x\left(n-d_x\right)\right) \quad (3)$$

where $f(.)$ is the mapping function of the NARX, $y(n)$ is the output at time step n, $x(n)$ is the input at time step n, d_x is the number of input delay, and d_y is the number of output(layer) delay. A more explicit form of equation (3) with weights and bias expressions of the hidden layer is shown below in (4).

$$y\left(n+1\right)=f_0\left(b_0+\sum_{h=1}^{N}w_{ho}f_h(b_h+\sum_{i=0}^{d_x}w_{ih}x\left(n-i\right))+\sum_{j=0}^{d_y}w_{jh}y\left(n-j\right)\right) \quad (4)$$

where $f_0(.)$ and $f_h(.)$ are output and hidden layer nonlinear functions, b_0 and b_h are the output and hidden node biases respectively, $w_{ih}$is the weight vector of the input-hidden node. w_{ho} is the weight vector of the hidden to output node and w_{jh} is the weight vector of the layer to hidden node. The NARX design is of two main forms: the Open-Loop NARX (OLN) or Series-Parallel Network and the Closed Loop NARX (CLN) or Parallel Network. For BMS applications, the NARX Network can be used in Open-loop form or the Closed Loop form. Our BMS application requires both forms for effective training, but online implementation requires just the Closed Loop form.

1) OPEN LOOP NARX

This architecture uses target value as feedback input instead of the output from the previous time step Fig. 2(a) below is a network representation of OLN.

The simple expression for the OLN is shown in (5) below:

$$\hat{y}\left(n+1\right)=F\begin{pmatrix}y(n),y(n-1),\ldots,y\left(n-d_y\right),x(n+1)\\x(n),x(n-1),\ldots,x\left(n-d_x\right)\end{pmatrix} \quad (5)$$

where F(.) is the mapping function of the open loop NARX, $x(n)$ is the input at time step n, $y(n)$ is the target value at time step n, while other variables used are the same as in equations 3 and 4.

2) CLOSED LOOP NARX

This architecture uses output from previous time step as feedback input. Fig. 2(b) is a representation of CLN.

The mathematical expression for the CLN is as shown in (6) below:

$$\hat{y}\left(n+1\right)=F\begin{pmatrix}\hat{y}(n),\hat{y}(n-1),\ldots,\hat{y}\left(n-d_y\right),x(n+1),\\x(n),x(n-1),\ldots,x\left(n-d_x\right)\end{pmatrix} \quad (6)$$

where $\hat{y}(n)$ is the output variable at time step n, while other variables are as represented in equation (3), (4) and (5).

3) NARX NETWORK DESIGN AND TRAINING

Similar to other RNNs, hyperparameter tuning and method of training play a huge role in the model's performance. It is impossible to train the network model for every discrete operating temperature, hence, we chose to train the network at temperature intervals of −10°C, 0°C, 25°C and 50°C. A MATLAB integrated development environment and its time-series libraries were used in training and model implementation of the network. Algorithm 1 provides a step-by-step procedure to effectively train the NARX model for BMS application.

As shown in Algorithm 1,line 7, it is very important to sort the training data chronologically. This can be easily done in MATLAB using *net.divideblock* or *net.divedeind*. Levenberg Marquardt backpropagation is employed in training the network because it is a blend of gradient descent and Gauss Newton algorithm [41]. This makes it relatively faster than the Gauss Newton and harder to get trapped in the local minima. Since models are trained at discrete temperatures, estimating SOC at untrained temperature makes use of the nearest available temperature model.

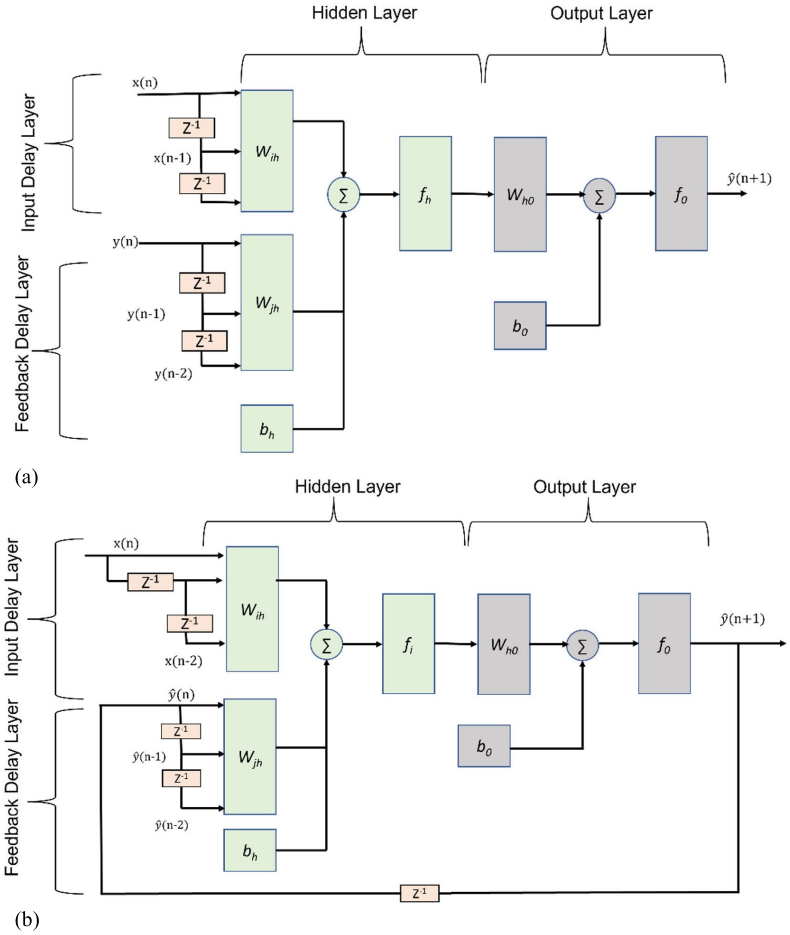


FIGURE 2. (a) Open Loop NARX Network (Series-Parallel Network) (b): Closed Loop NARX (Parallel) Network.

B. CENTRAL DIFFERENCE KALMAN FILTER METHOD (CDKF)

The CDKF maps points through a model and uses a weighted means to estimate the system's state. CDKF is a non-linear Bayesian probabilistic inference model where selected points around the mean of the estimated value is propagated through the function $f(x)$ and the output is weighted to obtain the state estimate and the covariance [42], [43]. We use CDKF rather than EKF because it performs better at lesser or equal computation cost [44], [45], [46], Also, CDKF need no derivatives which can be difficult to compute for some complex nonlinear function [42], [44], [47]. The CDKF, among other SPKF family uses Sterling polynomial interpolation to derive covariance and mean weight function. CDKF has just one tuning parameter hence has relatively simpler implementation and marginally better accuracy in comparison to UKF [23]. CDKF methods require battery model dynamics for the state model. Hence, we applied a robust equivalent circuit model (Enhance Self Correcting) [21] shown in (7) below. The state variables are the SOC ($z(k)$), diffusion current ($i_R(k)$), and hysteresis parameter ($h(k)$). The output variable is the voltage ($v(k)$) as shown in (8).

$$\begin{bmatrix} z[k+1] \\ i_R[k+1] \\ h[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_{RC} & 0 \\ 0 & 0 & A_H \end{bmatrix} \begin{bmatrix} z[k] \\ i_R[k] \\ h[k] \end{bmatrix} + \begin{bmatrix} -\eta[k]\,\Delta t/Q & 0 \\ (1-A_{RC}) & 0 \\ 0 & (A_H-1) \end{bmatrix} \begin{bmatrix} i[k] \\ sgn(i[k]) \end{bmatrix} \tag{7}$$

$$v[k] = \mathrm{OCV}(z[k], T[k]) + M_0 s[k] + Mh[k] - \sum_j R_j i_{R_j}[k] - R_0 i[k] \tag{8}$$

where $A_{RC} = exp{-}(\Delta t/RC), A_H = \exp -(H\eta[k]\,i[k]\Delta t/Q)$, $i[k]$ is the input current, Q is the cell capacity, η is the coulombic efficiency. While the observation parameters are the instantaneous hysteresis parameter (M_0), dynamic hysteresis parameter (M), charge transfer resistances (R_j) and the solution resistance (R_0) respectively. The Resistance-Capacitor function (A_{RC}) in (7) was gotten using system identification. The observation parameters (M_0, M, R_j, R_0) in equation (8) were obtained using least squares to fit

Algorithm 1 NARX Network Training

Input: Drive-Cycle current, voltage response, temperature
Target: SOC
Output: CLN model
//NARX Network Design and Hyperparameters
Network choice type *: Open loop NARX*
Input Nodes*: 2;* No of Targets*: 1*
No of Hidden Layer and nodes*: 1 hidden layer, 5 nodes*
No of Input delay*:2;* No of feedback (layer) delay*: 2*
Activation function*: tansig*
//Weight Training
Backpropagation Algorithm*: Levenberg Marquardt*
Data Preprocessing*:* sort in chronological block form *//* use *net.divideblock* in Matlab
for *Temperature = -10, 0, 10,20,25,30,40 50//*
  Start Training
  while (training conditions are not met*)//* wait here till the conditions (*Error Gradient, Error Correlation, MAE, Iteration)* are met
  Stop training *and* **Close** the Network Loop
  Test Closed Loop Network (CLN) performance
  if (CLN Performance < acceptable range)
    Retrain in CLN //transfer learn in closed loop
  else
    Save the CLN model.//save at various temp.
  end if
end for loop

the difference between the terminal voltage($v(k)$) and the $OCV(z[k], T[k])$ while simultaneously evaluating the best hysteresis dependent factor (H) to minimize the error. The CDKF Bayesian algorithm for estimating the states is as shown in Algorithm 2 below.

C. ENSEMBLE NARX–CDKF

An ensemble algorithm is the combination of two or more domain algorithms in a systematic manner to achieve a stable and more accurate result than its individual algorithm. We use DST drive cycle as the weight performance drive cycle because of its good excitation properties. We propose estimating the ensemble weights of the base learner from DST preprocess offline performance. We also propose using covariance slope of the CDKF as a convergence monitoring system for the ensemble model. A detailed procedure showing the inter-relationship of the base models and the ensemble is shown in Algorithm 3 below.

Fig. 3 shows a block illustration of the offline algorithm. This algorithm is suitable for real time BMS application because the bulk part of the algorithm is evaluated offline before installation to the EV. Fig. 4 shows online part of the algorithm to be embedded on a BMS. (9) below shows the weighted expression for estimating the combination of the

Algorithm 2 CDKF Offline Preprocessing and Online State Estimation

Input: Drive-Cycle current, voltage response, temperature
Output: SOC, diffusion- current, hysteresis, covariance.
// CDKF Offline preprocessing.
Collect the OCV Charge-Discharge of the battery at −10, 0, 10, 20, 25, 30, 40, 50(°C)
Use the Discharge OCV capacity at 25°C as the nominal capacity.
Estimate the model parameters of the ESC Model with the preprocessed drive cycle data.
$x_k = g\,(x_{k-1}, u_{k-1}, w_{k-1})$ // state equation
$y_k = h\,(x_k, u_k, v_k)$ // output equation
$X^+_{k-1} = \left\{\hat{x}^+_{k-1}, \hat{x}^+_{k-1} + \gamma\sqrt{(P^+_{\hat{x},k-1})}, \hat{x}^+_{k-1} - \gamma\sqrt{(P^+_{\hat{x},k-1})}\right\}$ *where* $\gamma = \sqrt{3}$
// CDKF Online Algorithm
$X^{x,-}_{k,i} = g\left(X^{x,+}_{k-1,i}, u_{k-1}, w_{k-1}\right)$
$Y_{k,i} = h\left(X^{x,-}_{k,i,} u_k, w_k\right)$
$\hat{x}^-_k = \sum_{i=0}^{p} \alpha^m_i X^{x,-}_{k,i}$; *where,* $\alpha^m_{i=0} = \frac{1}{2\gamma^2}$, $\alpha^m_{i\neq 0} = \frac{\gamma^2 - L}{\gamma^2}$, $L = length(x_k)$ // prior state
$\hat{y}_k = \sum_{i=0}^{p} \alpha^m_i Y_{k,i}$. *//p = 2∗L, prior output*
$P_{\bar{y},k} = \sum_{i=0}^{p} \alpha^c_i \left(Y_{k,i} - \hat{y}_k\right)\left(Y_{k,i} - \hat{y}_k\right)^T$ // prior output-covariance
$P^-_{\hat{x}\bar{y},k} = \sum_{i=0}^{p} \alpha^c_i \left(X^{x,-}_{k,i} - \hat{x}^-_k\right)\left(Y - \hat{y}_k\right)^T$ //prior cross- covariance
$L_k = P^-_{\hat{x}\bar{y},k} P^{-1}_{\bar{y},k}$ // kalman gain
save: $\hat{x}^+_k = \hat{x}^-_k + L_k\left(y_k - \hat{y}_k\right)$, // state output (SOC).
save: $P^+_{\bar{x},k} = P^-_{\bar{x},k} - L_k P_{\bar{y},k} L^T_k$ // Covariance output
return to Algorithm 3.

estimates of the ensemble members.

$$\mathrm{SOC\,(k)} = \frac{\mathrm{Wn} * \mathrm{NARX\_SOC(k)} + \mathrm{Ws} * \mathrm{CDKF\_SOC(k)}}{\mathrm{Wn} + \mathrm{Ws}} \tag{9}$$

where the *Wn*(1/NARX_MAE) is the contributing weight of the NARX pre-estimated offline on File-2 DST data. Similarly, *Ws (*1/CDKF_MAE) is the contributing weight of the CDKF pre-estimated offline on File-2 DST data. The NARX_SOC(k) is the iterative online estimated SOC at time-step (k) by the NARX model. Similarly, the CDKF_SOC(k) is the iterative online estimated SOC at time-step (k) by the CDKF Algorithm 2. The MAE is used here in weighting to avoid putting more weights on rogue noise outliers that might completely throw off our estimations.

D. MITIGATING PERFORMANCE LOSS

The proposed model does not encompass a battery lifetime model. Hence, as the battery ages, the SOC estimation accuracy drops, and the variance of the ensembles tends to drift apart. However, the covariance of the CDKF, the feedback SOC and the variance of the ensemble are

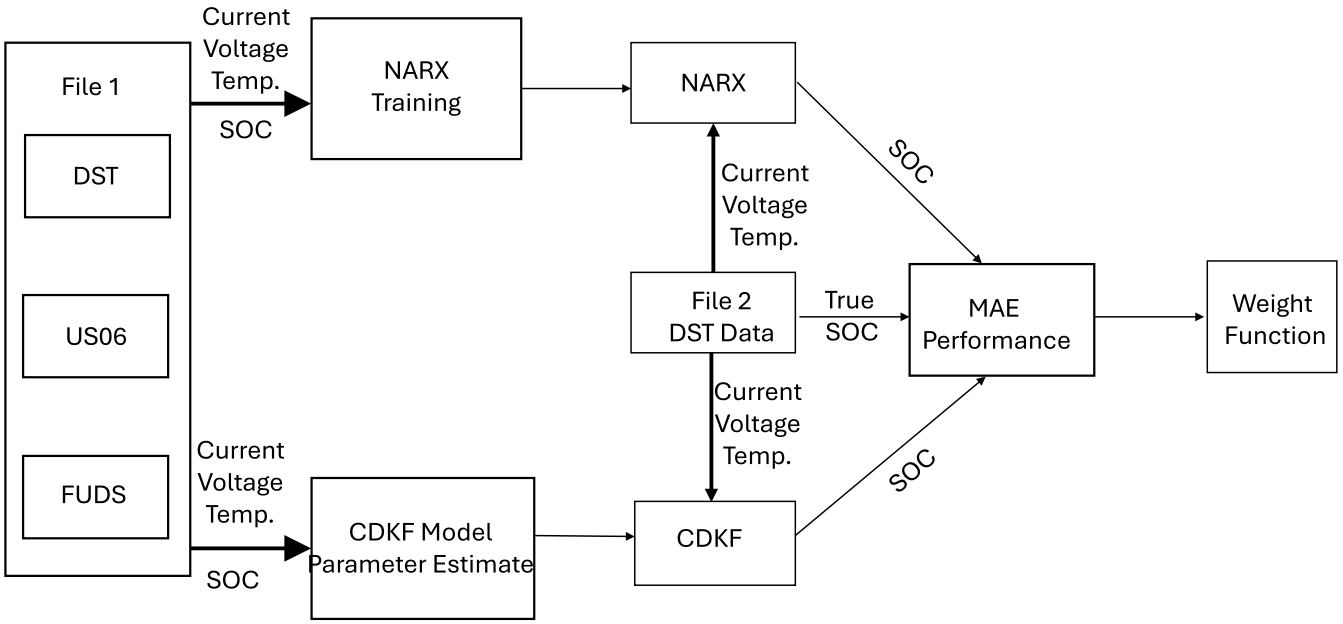


FIGURE 3. Offline block diagram of the ensemble CDKF-NARX algorithm, the thick arrows represent a bus of signals (current, voltage, temperature, and SOC).

Algorithm 3 Ensemble Algorithm

Input: Drive-Cycle current, voltage response, temperature.
Output: SOC, convergence point.

Estimate preprocessed SOC and MAE with NARX model generated from Algorithm 1
Estimate preprocessed SOC and MAE with CDKF algorithm (Algorithm 2)
Compute the running weights *Wn, Ws* of the model algorithms by inverting the offline pre-estimated NARX MAE and CDKF MAE respectively.
while $k \neq 0$ // **Recurring ensemble online estimation**
  if $(k = 1)$ and $(C{=}1)$ // run convergence only if the BMS is starting from off state
    $C = C{+}1$ // avoid running convergence twice
    initialize the SOC using voltage.
    Run CDKF Algorithm 2 to converge the SOC
    Wait here till Algorithm 2 Covariance slope is close to zero and save convergent point
  end
  Estimate online NARX_SOC with the NARX model output from Algorithm 1
  Estimate online CDKF_SOC with CDKF Algorithm (Algorithm 2).
  Compute Estimated *SOC(k)* with Equation (9)
  save *SOC(k)* as SOC Estimate
  if $(N \bmod k = 0)$,
    NARX_SOC = CDKF_SOC = SOC(k)
  if Ensemble Variance is > 0.25, recalibrate the weights to penalize the ensemble member with larger uncertainty
end.

mitigating(correcting) factors to reduce the performance drop. The feedback SOC is used to reinitialize the state of the NARX when the battery is at boundary voltages. The CDKF covariance(uncertainty) estimate is also used iteratively to adjust the kalman weights. The variance of the ensemble members is used to adjust the ensemble weights, using the additive properties of the variance.

E. CONVERGENCE DESIGN

It is sometimes the case that EV batteries need replacement or for some reason there is power loss to the BMS. Hence, it is highly important for BMS SOC estimator to return to an acceptable error boundary close to the true SOC as quickly as possible. It would be expected of a superfast charging EV SOC to converge as quickly as possible to prevent overcharging. However, online BMS cannot truly verify its SOC convergence accuracy because the true SOC state is unknown and can only be estimated. We addressed this issue by using the CDKF covariance to drive the estimate towards convergence, while using the covariance slope to indicate the convergence point. Our decision to use CDKF as the convergence algorithm is based on its relative pre-evaluated convergence performance of the NARX and the CDKF as seen in Table 1.

IV. RESULTS AND DISCUSSION

The performance of our proposed ensemble SOC estimator is evaluated on accuracy, convergence speed, and robustness. We verified the performances using the 2 CALCE dynamic data files, each consisting of the 3 drive cycles (DST, FUDS, US06). File-1 was used for system Identification of battery model for the CDKF and weight training for the NARX. The DST part of File-2 was used for offline ensemble weight estimation. The rest of File 2 (FUDS, US06) of the

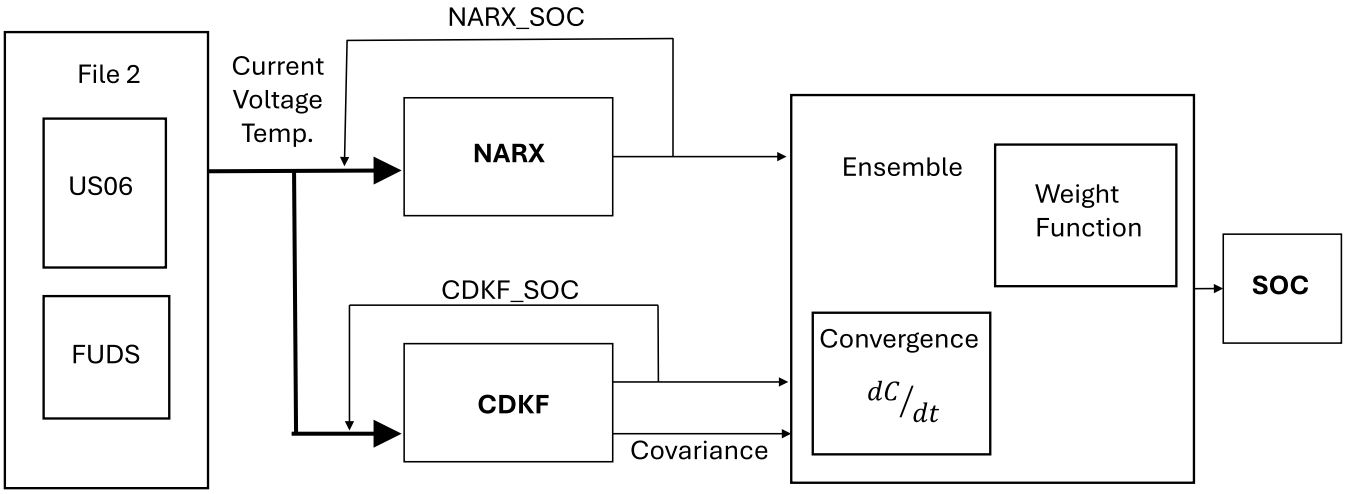


FIGURE 4. Online block diagram of the ensemble CDKF-NARX algorithm. The thick arrows represent a bus of signals (current, voltage, temperature and SOC).

TABLE 1. Offline convergence performance test result on DUF drive cycle.

Temp(°C)	NARX Convergence duration (seconds)	CDKF Convergence duration (seconds)
-10	8723	40
0	18500	49
10	72	45
20	8568	48
25	7343	49
30	7425	50
40	7438	50
50	--	55
Average	**9326**	**48**

data was used for testing the performance of the proposed method. The MAE was used here as the performance metric because it has a 1:1 scale when compared with the actual SOC. Hence, making error interpretation easier. We first evaluated the convergence of the ensemble members at various temperatures to decide the best approach to design the initialization.

A. NARX CONVERGENCE TEST RESULT

Even though the RNN algorithms are good estimators, their ability to converge while starting from a wrong SOC is rarely evaluated in literature. Convergence is technically irrelevant in OLN because OLN uses the last known true SOC to converge after one iteration, and the last known true SOC is unavailable for online applications. Hence, we tested the CLN convergence performance alongside the CDKF convergence performance on DUF drive cycles at all available temperatures. We choose the minimum acceptable convergence limit to an absolute error value of 0.03. The closed loop NARX converges better at low SOC for most of the test temperatures. Fig. 5(a) shows the observed typical convergence pattern of the CLN. The CLN was tested at all available temperatures with an initial offset of 0.5 (50%) from the actual SOC. Convergent points at all tested temperatures

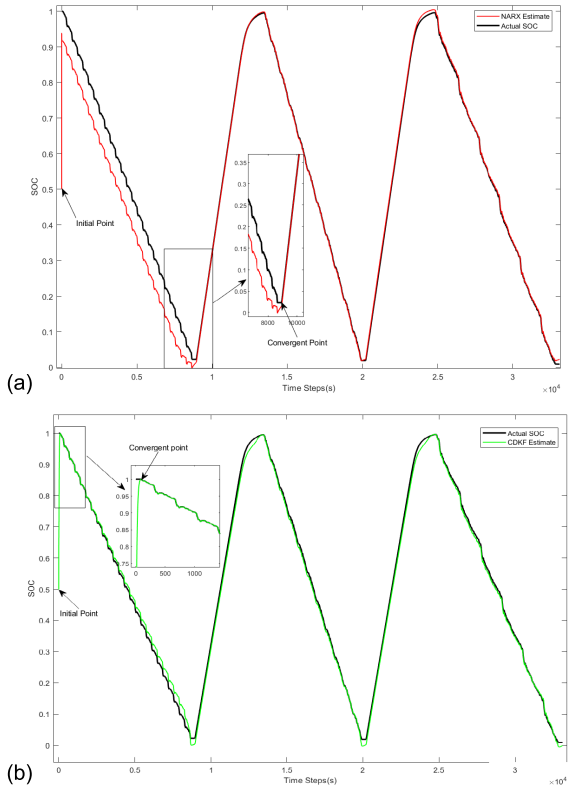


FIGURE 5. (a) NARX convergence test at 20°C, inset shows the convergent point of the CLN. (b) CDKF convergence test at 20°C, inset shows the convergent point of the CDKF.

are provided in Table 1. Our interpretation of the convergence of NARX is that the NARX learned to associate the very low voltage of the battery to zero percent SOC. We also believe that poor convergence of CLN stems from the lack of similar initialization condition in the training data. In conclusion, the NARX convergence property cannot be trusted in BMS

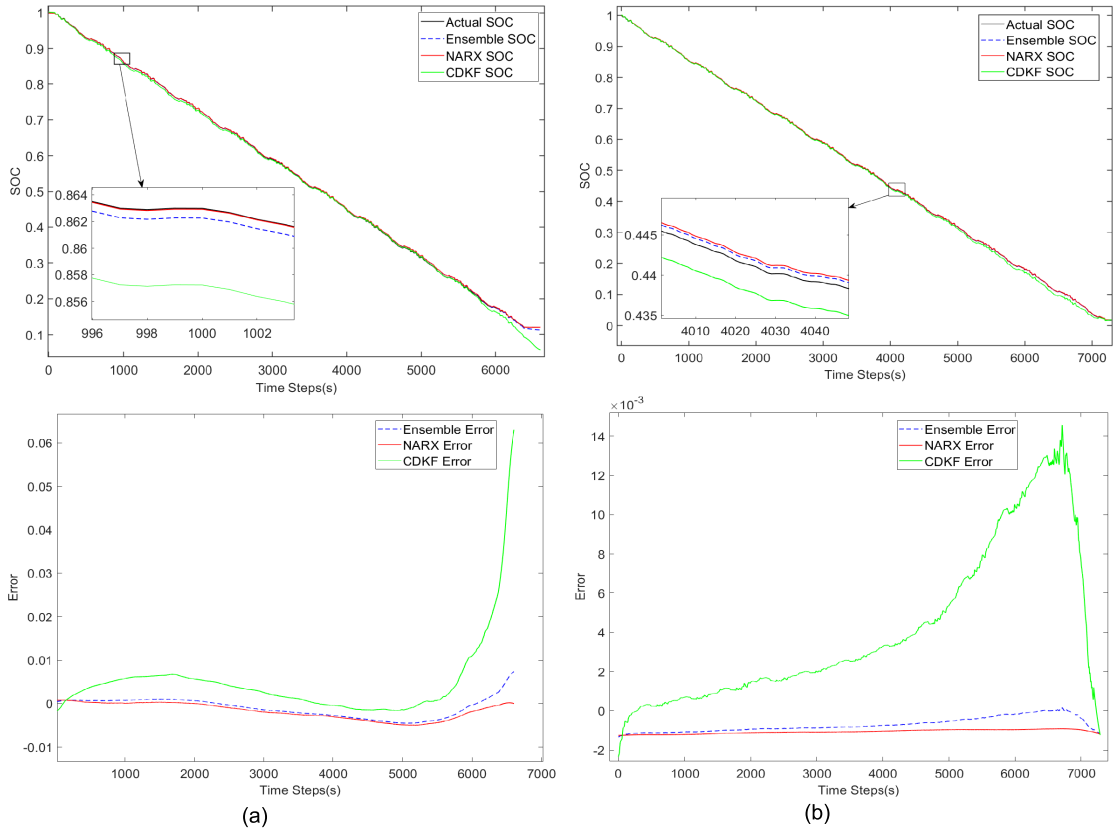


FIGURE 6. (a) Steady-state performance plot and error plot of ensemble estimates on US06 drive cycle at 0°C (b) Steady-state performance plot and error plot of ensemble estimates on US06 drive cycle at 50°C.

application as complete discharge cannot be guaranteed in BMS daily operation.

B. CDKF CONVERGENCE TEST RESULT

Similar to the NARX convergent test experiment, convergent experiment was performed on the CDKF algorithms under the same condition. The actual SOC is at 1 (100%) while the initialized SOC is at 0.5 (50%). It can be seen in Fig. 5(b) that the estimated SOC converges quickly to the true SOC after 87 iterations. This shows that the covariance properties of the CDKF enable quick convergence. Hence, this validates our reasoning behind the use of covariance property of the CDKF as an initialization method. Table 1 shows the convergence performance comparison between CDKF and NARX. It illustrates the time it took for the algorithms to converge to an AE of less than 3% SOC at various temperatures. The AE is the absolute value of the difference between the true SOC and the estimated SOC, while the convergence time is the number of iterations it took for our algorithm to come to less than 3% of the actual SOC. It should be noted that the algorithm would not absolutely converge on the exact SOC since the battery is in continuous operation and the true state changes with time. The presence of noise in the system is also another reason the actual state will never be reached. However, the error should be within our acceptable tolerance limit, usually estimated with consideration of the battery's overcharging and undercharging allowance limits. It took the NARX on average about 9326 seconds to converge, while the CDKF converges on an average of 48 seconds as shown in Table 1. The CDKF performs well in convergence, because of its ability to estimate covariance errors. As seen in Table 1, there is no convergence at 50°C for the NARX algorithm, and we consider this an outlier in computing the average NARX convergence. However, it can still be concluded with the available data that training NARX on conventional drive cycle will not perform well when initialization is far off. However, NARX provides better estimation stability than the CDKF when the estimate is in the ballpark of the truth. Once again, the true SOC will not be known in a working BMS, therefore knowing the convergence performance of the estimator by absolute error is impossible. We therefore propose using the correlation property of the CDKF covariance and convergence. Since the estimated covariance always tends to remain constant when the SOC converges, we use the slope of the covariance to determine the time of convergence. We use a slope value of 0.0001 on error bounds ($SOC \pm 3 * \sqrt{P}$) as the point of convergence. Table 2 shows the convergence performance of the proposed online convergence estimation method. The proposed method was tested at 0°C, 10°C, 20°C, 30°C, 40°C and 50°C on FUDS drive cycle. The average convergence time was 136 seconds with an absolute error

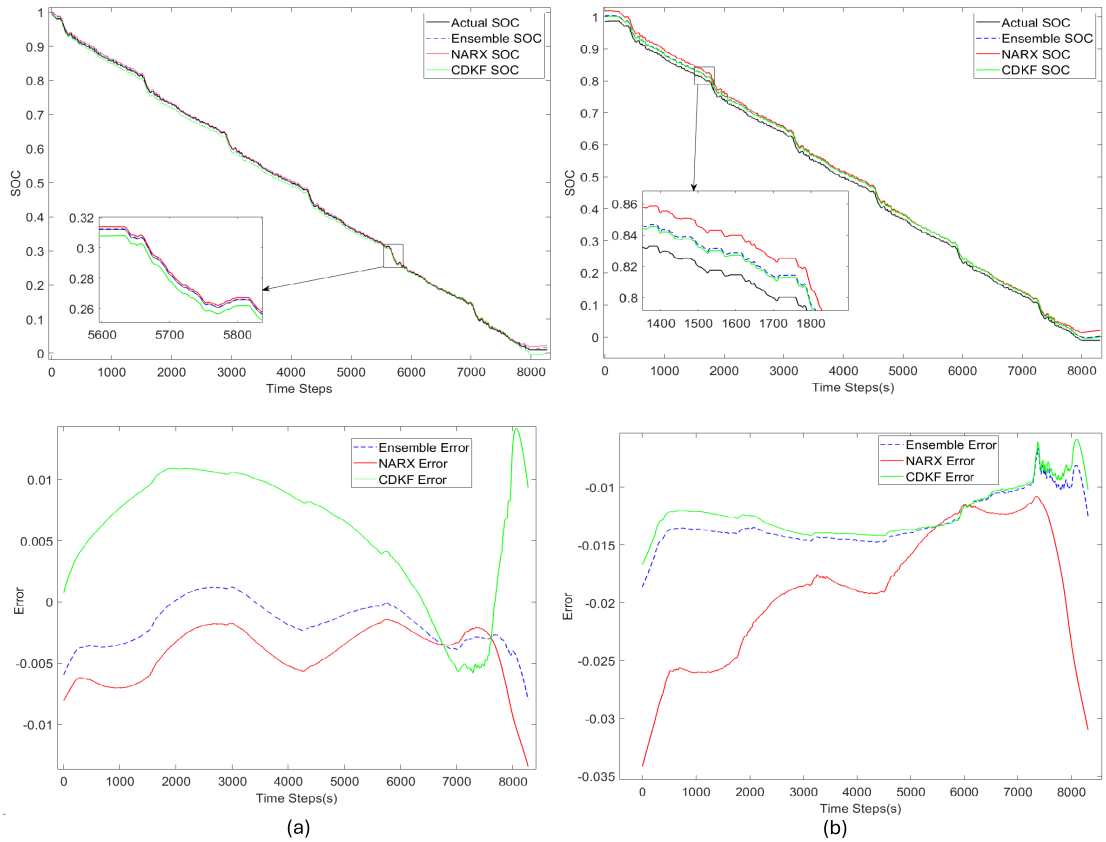


FIGURE 7. (a)Steady-state performance plot and error plot of ensemble estimates FUDS at 20°C. (b) Steady-state performance plot and error plot of ensemble estimates on FUDS drive cycle at 40°C.

TABLE 2. Online convergence test result (slope < 0.0001) on FUDS.

Temp (°C)	Convergent Time (seconds)	Absolute Error at Converged Point
0	124	0.00057
10	188	0.0047
20	123	0.0011
30	63	0.00064
40	128	0.0016
50	190	0.0034
Average	**136**	**0.002**

of 0.002 (0.2%), this is acceptable as it is well within our absolute error acceptance limit and executed in just over 2 minutes.

C. MAE PERFORMANCE RESULTS

The performance of our algorithms is reported when offset initialization is at 0% and 50%. The zero offset initialization means that we started estimation at the true SOC, while the 50% offset initialization means that we started SOC at 0.5 when true SOC is 1.

1) ZERO OFFSET INITIALIZATION MAE PERFORMANCE (STEADY STATE PERFORMANCE)

The zero offset initialization performance is also regarded as the steady-state MAE performance, this is used to analyze the performance of the algorithms when the estimate is stable within the vicinity of the true SOC. The performances were evaluated using the US06, FUDS drive cycle as seen in Fig. 6 (for US06) at 0°C, and 50°C respectively. The stability and robustness of the ensemble model can be clearly seen in the error plots shown in Fig. 6 a and b (US06). In both cases, CDKF deviates a little from the actual SOC at about 6000-time steps. This deviation can be seen to pull the ensemble estimates a little away from the actual SOC but not enough to cause a significant error estimate.

Deviation from the actual SOC can also be clearly seen in the NARX and the CDKF estimates, performance test on FUDS drives cycles at 20°C as seen Fig. 7a.

However, this deviation has little to no negative effect on the ensemble estimate due to the uncorrelated counteracting errors of the CDKF and NARX as clearly seen in its error plots. Similar deviation pattern is observed in the estimation at 40°C on FUDS drive cycle in Fig. 7b. The NARX in this case is seen to deviate more from the true state as seen. The ensemble and the NARX estimates are closer to the true state. Even though the CDKF is seen to be a little bit

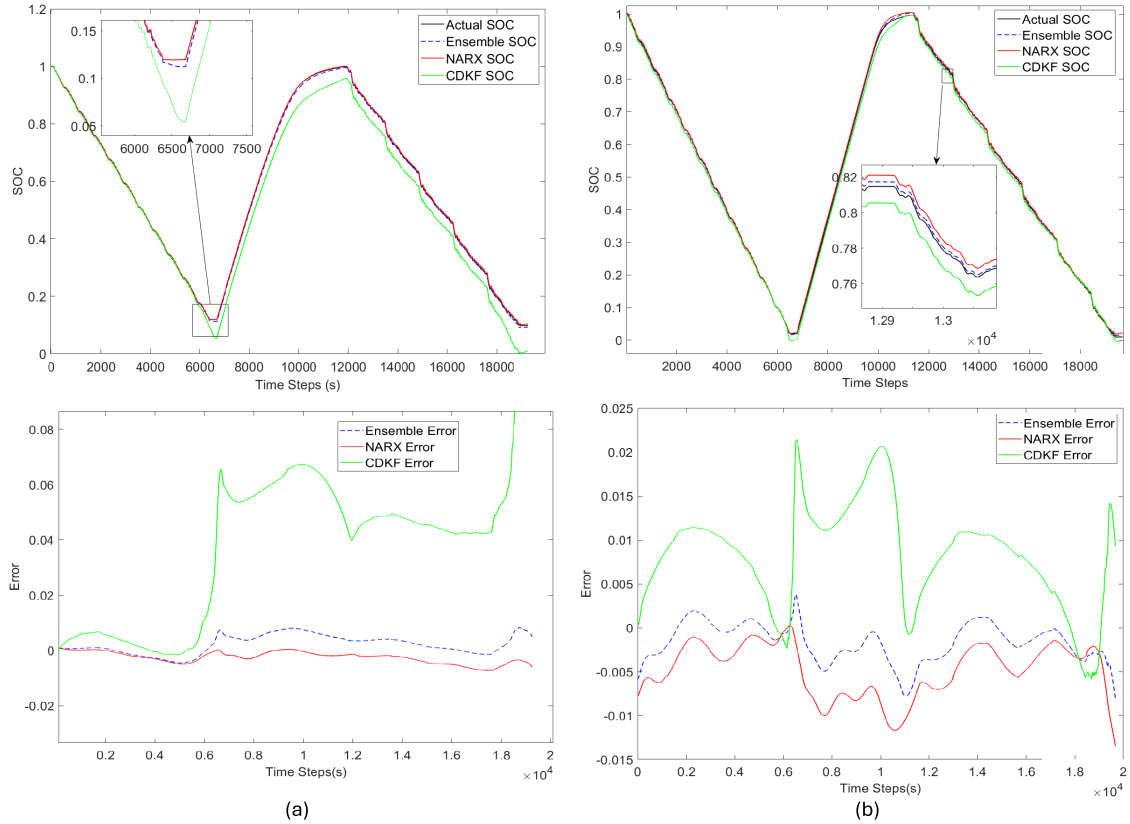


FIGURE 8. (a) Steady-state performance plot and error plot of ensemble estimates on US06-FUDS at 0°C. (b) Steady-state performance plot and error plot of ensemble estimates on US06-FUDS at 20°C.

TABLE 3. Steady state performance on drive cycles.

	US06			FUDS			US06 -FUDS		
Temp(°C)	CDKF MAE	NARX MAE	Ensemble MAE	CDKF MAE	NARX MAE	Ensemble MAE	CDKF MAE	NARX MAE	Ensemble MAE
N10	0.0052	0.0533	0.0080	0.0097	0.0764	0.0096	0.0063	0.0217	0.0090
0	0.0054	0.0019	0.0019	0.0513	0.0041	0.0027	0.0100	0.0012	0.0033
10	0.0042	0.0061	0.0054	0.0027	0.0054	0.0044	0.0133	0.0050	0.0060
20	0.0076	0.0028	0.0014	0.0071	0.0042	0.0021	0.0144	0.0045	0.0022
25	0.0046	0.0059	0.0042	0.0028	0.0058	0.0034	0.0062	0.0110	0.0062
30	0.0061	0.0049	0.0029	0.0029	0.0104	0.0038	0.0065	0.0102	0.0051
40	0.0021	0.0172	0.0099	0.0122	0.0189	0.0130	0.0021	0.0172	0.0113
50	0.0041	0.0011	0.0007	0.0046	0.0018	0.0018	0.0062	0.0005	0.0015
Average	**0.0049**	**0.0117**	**0.0043**	**0.0117**	**0.0159**	**0.0051**	**0.0081**	**0.0089**	**0.0056**

more accurate than the ensemble in this test case, this is the rare cost paid for stability as the accuracy of the CDKF cannot always be guaranteed over the NARX at other test temperature as seen Table 3. Table 3 summarizes the zero initial offset performance (steady state performance) of the estimators (CDKF, NARX, Ensemble) on US06 drive cycle. It shows that the ensemble performs better than the NARX and the CDKF in most cases and on average. The average MAE performance of the ensemble is 0.0043 while the average MAE that of the CDKF and NARX are 0.0049 and 0.0117. Table 3 also shows that the ensemble algorithm also outperforms both ensemble members on FUDS drive cycle. The average MAE performance of the ensemble was 0.0051, while that of NARX is 0.0159 and that of the CDKF is 0.0117.

The performance on US06-FUDS with a charging phase in between the drive cycles was also evaluated. Fig. 8 shows the tracking performance of the ensemble and the ensemble members at 0°C and 20°C. The NARX and the ensemble estimates were closer to the true SOC in the 0°C test case while the CDKF is seen to deviate from the true SOC. The uncorrelated oscillatory error of the ensemble member at the 20°C test case is shown in Fig. 8b error

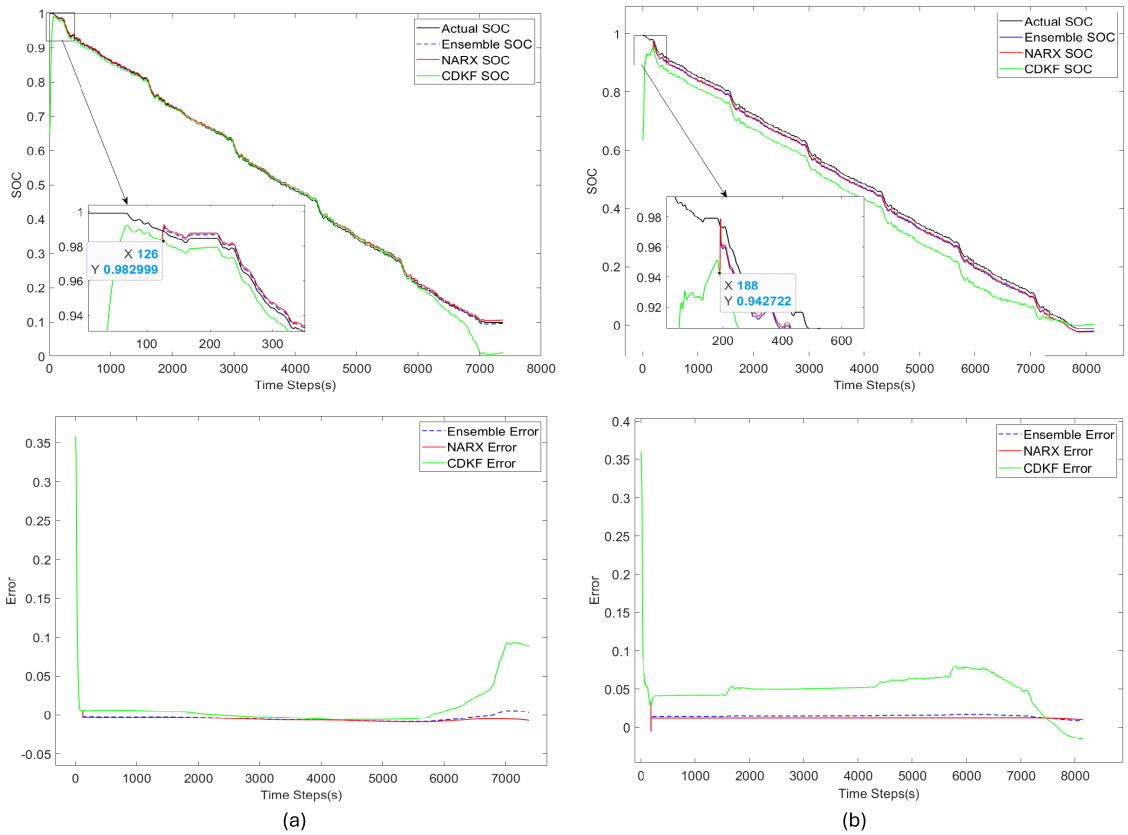


FIGURE 9. (a) Offset performance plot and error plot on FUD at 0°C. (b) Offset performance plot and error plot on FUDS at 50°C.

plot. It enhanced the accuracy of the ensemble method over the ensemble member's accuracy. Table 3 shows that the ensemble outperformed the NARX and the CDKF with an MAE of 0.0022 while the CDKF and the NARX are 0.0144 and 0.0045. The average ensemble result of the US06-FUDS outperforms the ensemble members with an average MAE performance of 0.0056 as shown in Table 3.

2) 50% OFFSET INITIALIZATION MAE PERFORMANCE (OFFSET PERFORMANCE)

The performance of our algorithm was also evaluated when the initial estimate is forced to 50% (0.5) while the true SOC was at 100%. This is used to simulate an extreme initialization error scenario that could occur in a BMS. Fig. 9 shows the behavior of all 3 algorithms with respect to the actual value. All 3 algorithms used the CDKF covariance for initialization, which explains why only the yellow plot (CDKF) was the only visible plot in the beginning of Fig. 9 plots. Fig. 9a is the convergence performance plot and error plot at tested temperature of 0°C, the inset plots of Fig. 9 shows that right after the convergence axis point (126, 0.98), the NARX estimates jumps closer to the actual SOC, hence improving the ensemble estimates. Similar behavior is observed at test temperature 50°C as shown in the performance plots Fig. 9. The inset plot of Fig. 9b shows that the NARX algorithm performs well near the convergent axis point (188, 0.94). This again illustrates the contributing benefit of the NARX to the ensemble algorithm. Table 4 below shows that the ensemble method also outperforms the NARX and the CDKF at an uncertain initialization scenario. The table summarizes the performances of the algorithms at all the tested temperatures. It shows that the average MAE Performance of the ensemble algorithms is 0.0074, while that of the NARX and CDKF are 0.0075 and 0.0213 respectively.

D. PERFORMANCE COMPARISON WITH GRU, LSTM, AND BILSTM

We benchmarked our proposed model with standard GRU, LSTM, and BiLSTM, and performed exhaustive searches for the optimum layer structure at various discrete temperatures using MATLAB deep network designer application. As the battery dynamics changes with temperature, so does the network structure and its optimum hyper-parameter. Hence, in some cases, various optimum network structures were achieved for varying temperatures. The optimum optimization algorithm used in this case is the Adaptive Moment optimizer. The network structure for the GRU at 25°C consist of a sequence input layer, 100 gru layer, tanh layer, fully connected layer, leaky relu layer, fully connected layer, clipped-relu layer and a regression layer. The network structure for the LSTM at 25°C consists of a sequence input layer, 128 LSTM layer, 10 fully connected layers, a fully

TABLE 4. MAE convergence performance on FUDS drive cycles at an 0.5 offset initialization.

Temp (°C)	Ensemble MAE Performance	NARX MAE Performance	CDKF MAE Performance	GRU MAE Performance	LSTM MAE Performance	BiLSTM MAE Performance
0	0.0059	0.0069	0.0129	0.1617	0.3428	0.2443
10	0.0076	0.0054	0.0219	0.1271	0.2898	0.0956
20	0.0040	0.0032	0.0174	0.0850	0.2651	0.0698
30	0.0052	0.0080	0.0102	0.0903	0.2650	0.0786
40	0.0052	0.0070	0.0147	0.0608	0.2753	0.0615
50	0.0166	0.0142	0.0506	0.0560	0.2711	0.0936
Average	**0.0074**	**0.0075**	**0.0213**	**0.0968**	**0.2849**	**0.1072**

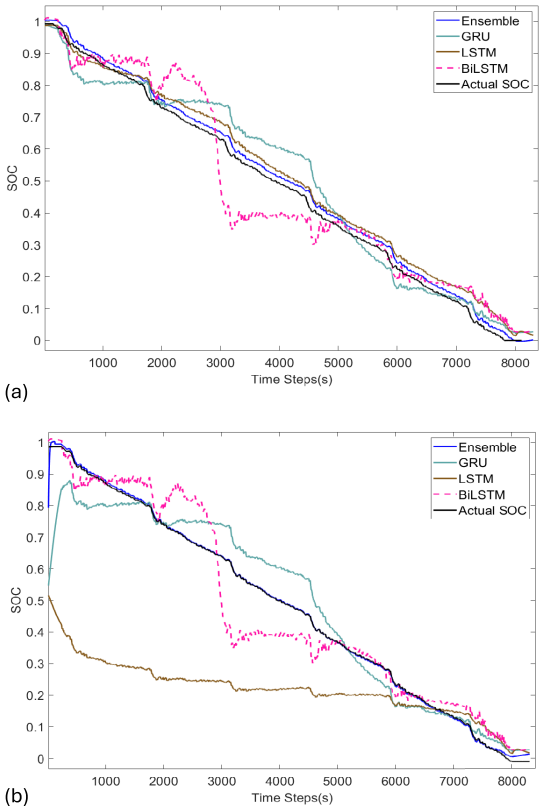


FIGURE 10. (a) SOC performance comparison of ensemble method with GRU, LSTM and BiLSTM (b) SOC convergence performance comparison of ensemble method with GRU, LSTM and BiLSTM.

connected layer, and a regression layer. The BiLSTM network structure at 25°C contains a sequence input layer, 80 BiLSTM layers, a relu layer, 20 fully connected layers, a relu layer, a fully connected layer, and a regression layer.

Table 4 shows that convergence performance of the ensemble model outperforms the gated-RNNs. The average MAE convergence performance of the ensemble is 0.0074 while the GRU, LSTM and the BiLSTM MAE are 0.0968, 0.2849 and 0.1072 respectively.

The steady state performance of the GRU, LSTM, BiLSTM and the ensemble on FUDS is shown in Table 5

TABLE 5. Steady state MAE performance comparison with GRU, LSTM and BiLSTM on FUDS drive cycles.

Temp (°C)	GRU	LSTM	BiLSTM	Ensemble
N10	0.0847	0.0873	0.0813	0.0096
0	0.0988	0.0810	0.0988	0.0027
10	0.1021	0.0300	0.0982	0.0044
20	0.0771	0.0120	0.0678	0.0021
25	0.0734	0.0181	0.0828	0.0034
30	0.076	0.0260	0.0765	0.0038
40	0.0489	0.0341	0.0635	0.0130
50	0.0487	0.0308	0.0881	0.0018
Average	**0.1016**	**0.0532**	**0.1095**	**0.0051**

The performance shows that the proposed model outperforms the gated-RNNs as the average MAE is 0.0051 while that of the GRU, LSTM and the BiLSTM are 0.1016, 0.0532 and 0.1095 respectively. Fig. 10a shows the performance comparison of the ensemble and the gated-RNNs at 40°C. Fig. 10b also shows the ensemble was able to converge better than the gated-RNNs. Both figures also show the commonly reported negative characteristics of the gated-RNNs as it fluctuates around the true SOC which is not seen in the ensemble model.

V. CONCLUSION

The proposed ensemble model proved to be very accurate and robust under various aggressive drive cycles input with an average steady state MAE performance 0.0043 (0.4%) on US06, 0.0051 (0.5%) on FUDS and 0.0056 (0.56%) on US06-FUDS drive cycle. The model is very robust and stable as it stayed within a 1% error bound at steady state, and under extreme dynamic temperatures. The proposed algorithm was also able to meet the convergence requirement of a superfast charging EV by converging in about 2 minutes on average with an average MAE Performance of 0.0074 (0.7%) when initialized half-way from the true SOC. The ensemble model is also seen to outperform the conventional gated-RNNs model under the limited training data constraints.

The proposed parallel design model is also expected to be relatively fast in comparison to series hybrid SOC estimators when implemented on multi-core BMS processor.

To further improve the performance of the proposed model, we suggest including another adaptive SOC estimator to the ensemble at an increased computational cost. We also suggest the design of rich drive cycle training data that encompasses erroneous initialization conditions to improve the convergence performance of the NARX.

REFERENCES

[1] Z. Rezvani, J. Jansson, and J. Bodin, "Advances in consumer electric vehicle adoption research: A review and research agenda," *Transp. Res. Part D, Transp. Environ.*, vol. 34, pp. 122–136, Jan. 2015, doi: 10.1016/j.trd.2014.10.010.

[2] M. Muratori, M. Alexander, D. Arent, M. Bazilian, P. Cazzola, E. M. Dede, J. Farrell, C. Gearhart, D. Greene, A. Jenn, M. Keyser, T. Lipman, S. Narumanchi, A. Pesaran, R. Sioshansi, E. Suomalainen, G. Tal, K. Walkowicz, and J. Ward, "The rise of electric vehicles–2020 status and future expectations," *Prog. Energy*, vol. 3, no. 2, Mar. 2021, Art. no. 022002, doi: 10.1088/2516-1083/abe0ad.

[3] D. Pevec, J. Babic, A. Carvalho, Y. Ghiassi-Farrokhfal, W. Ketter, and V. Podobnik, "Electric vehicle range anxiety: An obstacle for the personal transportation (R)evolution?" in *Proc. 4th Int. Conf. Smart Sustain. Technol. (SpliTech)*, Jun. 2019, pp. 1–8, doi: 10.23919/SPLITECH.2019.8783178.

[4] O. T. T. Kim, N. H. Tran, V. Nguyen, S. M. Kang, and C. S. Hong, "Cooperative between V2C and V2V charging: Less range anxiety and more charged EVs," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 679–683, doi: 10.1109/ICOIN.2018.8343205.

[5] R. A. Hanifah, S. F. Toha, and S. Ahmad, "Electric vehicle battery modelling and performance comparison in relation to range anxiety," *Proc. Comput. Sci.*, vol. 76, pp. 250–256, Jan. 2015, doi: 10.1016/j.procs.2015.12.350.

[6] L. Zhao, W. Yao, Y. Wang, and J. Hu, "Machine learning-based method for remaining range prediction of electric vehicles," *IEEE Access*, vol. 8, pp. 212423–212441, 2020, doi: 10.1109/ACCESS.2020.3039815.

[7] S. Scheubner, A. T. Thorgeirsson, M. Vaillant, and F. Gauterin, "A stochastic range estimation algorithm for electric vehicles using traffic phase classification," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6414–6428, Jul. 2019, doi: 10.1109/TVT.2019.2918544.

[8] E. A. ElGhanam, M. S. Hassan, and A. H. Osman, "Deployment optimization of dynamic wireless electric vehicle charging systems: A review," in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS)*, Sep. 2020, pp. 1–7, doi: 10.1109/IEMTRONICS51293.2020.9216415.

[9] R. Collin, Y. Miao, A. Yokochi, P. Enjeti, and A. von Jouanne, "Advanced electric vehicle fast-charging technologies," *Energies*, vol. 12, no. 10, p. 1839, May 2019, doi: 10.3390/en12101839.

[10] K. S. R. Mawonou, A. Eddahech, D. Dumur, E. Godoy, D. Beauvois, and M. Mensler, "Li-ion battery pack SoC estimation for electric vehicles," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2018, pp. 4968–4973, doi: 10.1109/IECON.2018.8591187.

[11] J. Tian, C. Chen, W. Shen, F. Sun, and R. Xiong, "Deep learning framework for lithium-ion battery state of charge estimation: Recent advances and future perspectives," *Energy Storage Mater.*, vol. 61, Aug. 2023, Art. no. 102883, doi: 10.1016/j.ensm.2023.102883.

[12] S. Guo and L. Ma, "A comparative study of different deep learning algorithms for lithium-ion batteries on state-of-charge estimation," *Energy*, vol. 263, Jan. 2023, Art. no. 125872, doi: 10.1016/j.energy.2022.125872.

[13] F. Yang, S. Zhang, W. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries using LSTM and UKF," *Energy*, vol. 201, Jun. 2020, Art. no. 117664, doi: 10.1016/j.energy.2020.117664.

[14] X. Song, F. Yang, D. Wang, and K.-L. Tsui, "Combined CNN-LSTM network for state-of-charge estimation of lithium-ion batteries," *IEEE Access*, vol. 7, pp. 88894–88902, 2019, doi: 10.1109/ACCESS.2019.2926517.

[15] C. Vidal, P. Kollmeyer, E. Chemali, and A. Emadi, "Li-ion battery state of charge estimation using long short-term memory recurrent neural network with transfer learning," in *Proc. IEEE Transp. Electrific. Conf. Expo. (ITEC)*. MI, USA: IEEE, Jun. 2019, pp. 1–6, doi: 10.1109/ITEC.2019.8790543.

[16] P. K. Terala, A. S. Ogundana, S. Y. Foo, M. Y. Amarasinghe, and H. Zang, "State of charge estimation of lithium-ion batteries using stacked encoder–decoder bi-directional LSTM for EV and HEV applications," *Micromachines*, vol. 13, no. 9, p. 1397, Aug. 2022, doi: 10.3390/mi13091397.

[17] J. Chen, Y. Zhang, J. Wu, W. Cheng, and Q. Zhu, "SOC estimation for lithium-ion battery using the LSTM-RNN with extended input and constrained output," *Energy*, vol. 262, Jan. 2023, Art. no. 125375, doi: 10.1016/j.energy.2022.125375.

[18] X. Fan, W. Zhang, C. Zhang, A. Chen, and F. An, "SOC estimation of Li-ion battery using convolutional neural network with U-Net architecture," *Energy*, vol. 256, Oct. 2022, Art. no. 124612, doi: 10.1016/j.energy.2022.124612.

[19] Z. Wang, A. Fotouhi, and D. J. Auger, "State of charge estimation in lithium-sulfur cells using LSTM recurrent neural networks," in *Proc. Eur. Control Conf. (ECC)*. Petersburg, Russia: IEEE, May 2020, pp. 1079–1085, doi: 10.23919/ECC51009.2020.9143926.

[20] X. Chen, W. Shen, M. Dai, Z. Cao, J. Jin, and A. Kapoor, "Robust adaptive sliding-mode observer using RBF neural network for lithium-ion battery state of charge estimation in electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 1936–1947, Apr. 2016, doi: 10.1109/TVT.2015.2427659.

[21] G. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packsPart 1. Background," *J. Power Sources*, vol. 134, no. 2, pp. 277–292, Jun. 2004, doi: 10.1016/S0378-7753(04)00359-3.

[22] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs," *J. Power Sources*, vol. 134, no. 2, pp. 262–276, Aug. 2004, doi: 10.1016/j.jpowSources2004.02.032.

[23] G. L. Plett, "Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs," *J. Power Sources*, vol. 161, no. 2, pp. 1369–1384, Oct. 2006.

[24] C. Hu, B. D. Youn, and J. Chung, "A multiscale framework with extended Kalman filter for lithium-ion battery SOC and capacity estimation," *Appl. Energy*, vol. 92, pp. 694–704, Apr. 2012, doi: 10.1016/j.apenergy.2011.08.002.

[25] F. Yang, Y. Xing, D. Wang, and K.-L. Tsui, "A comparative study of three model-based algorithms for estimating state-of-charge of lithium-ion batteries under a new combined dynamic loading profile," *Appl. Energy*, vol. 164, pp. 387–399, Feb. 2016, doi: 10.1016/j.apenergy.2015.11.072.

[26] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation," *Int. J. Electr. Power Energy Syst.*, vol. 62, pp. 783–791, Nov. 2014, doi: 10.1016/j.ijepes.2014.04.059.

[27] J. Tian, R. Xiong, J. Lu, C. Chen, and W. Shen, "Battery state-of-charge estimation amid dynamic usage with physics-informed deep learning," *Energy Storage Mater.*, vol. 50, pp. 718–729, Sep. 2022, doi: 10.1016/j.ensm.2022.06.007.

[28] X. Qin, M. Gao, Z. He, and Y. Liu, "State of charge estimation for lithium-ion batteries based on NARX neural network and UKF," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, vol. 1. Finland: IEEE, Jul. 2019, pp. 1706–1711, doi: 10.1109/INDIN41052.2019.8972319.

[29] Y. Yang, *Temporal Data Mining via Unsupervised Ensemble Learni*. Amsterdam, The Netherlands: Elsevier, 2016, doi: 10.1016/b978-0-12-811654-8.00002-6.

[30] *Battery Data | Center for Advanced Life Cycle Engineering*. Accessed: Jun. 06, 2023. [Online]. Available: https://calce.umd.edu/battery-data

[31] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Appl. Energy*, vol. 113, pp. 106–115, Jan. 2014, doi: 10.1016/j.apenergy.2013.07.008.

[32] N. Somakettarin and T. Funaki, "Study on factors for accurate open circuit voltage characterizations in mn-type Li-ion batteries," *Batteries*, vol. 3, no. 4, p. 8, Mar. 2017, doi: 10.3390/batteries3010008.

[33] C. Zhang, J. Jiang, L. Zhang, S. Liu, L. Wang, and P. Loh, "A generalized SOC-OCV model for lithium-ion batteries and the SOC estimation for LNMCO battery," *Energies*, vol. 9, no. 11, p. 900, Nov. 2016, doi: 10.3390/en9110900.

[34] R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, W. Zheng, H. Wang, W. Wang, and M. Wang, "A study on the open circuit voltage and state of charge characterization of high capacity lithium-ion battery under different temperature," *Energies*, vol. 11, no. 9, p. 2408, Sep. 2018, doi: 10.3390/en11092408.

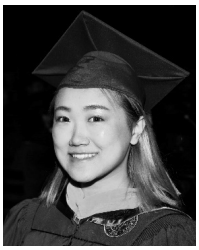
[35] *USABC Electric Vehicle Battery Test Procedures Manual. Revision 2*, document DOE/ID–10479-Rev.2, Lockheed Idaho Technol. Co., Jan. 1996.
[36] O. U.S. EPA. *Dynamometer Drive Schedules*. Accessed: Jun. 04, 2023. [Online]. Available: https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules
[37] O. U.S. EPA. *EPA US06 or Supplemental Federal Test Procedure (SFTP)*. Accessed: Jun. 04, 2023. [Online]. Available: https://www.epa.gov/emission-standards-reference-guide/epa-us06-or-supplemental-federal-test-procedure-sftp
[38] M. S. H. Lipu, M. A. Hannan, A. Hussain, M. H. M. Saad, A. Ayob, and F. Blaabjerg, "State of charge estimation for lithium-ion battery using recurrent NARX neural network model based lighting search algorithm," *IEEE Access*, vol. 6, pp. 28150–28161, 2018, doi: 10.1109/ACCESS.2018.2837156.
[39] A. Wunsch, T. Liesch, and S. Broda, "Groundwater level forecasting with artificial neural networks: A comparison of long short-term memory (LSTM), convolutional neural networks (CNNs), and non-linear autoregressive networks with exogenous input (NARX)," *Hydrol. Earth Syst. Sci.*, vol. 25, no. 3, pp. 1671–1687, Apr. 2021, doi: 10.5194/hess-25-1671-2021.
[40] M. S. H. Lipu, A. Hussain, M. H. M. Saad, A. Ayob, and M. A. Hannan, "Improved recurrent NARX neural network model for state of charge estimation of lithium-ion battery using pso algorithm," in *Proc. IEEE Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Apr. 2018, pp. 354–359, doi: 10.1109/ISCAIE.2018.8405498.
[41] I. Mukherjee and S. Routroy, "Comparing the performance of neural networks developed by using levenberg–marquardt and quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2397–2407, Feb. 2012, doi: 10.1016/j.eswa.2011.08.087.
[42] R. van der Merwe and E. Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2003, doi: 10.1109/ICASSP.2003.1201778.
[43] A. S. Paul and E. A. Wan, "RSSI-based indoor localization and tracking using sigma-point Kalman smoothers," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 860–873, Oct. 2009, doi: 10.1109/JSTSP.2009.2032309.
[44] R. van der Merwe, E. Wan, and S. Julier, "Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Aug. 2004, p. 8, doi: 10.2514/6.2004-5120.
[45] J. Zhu, N. Zheng, Z. Yuan, Q. Zhang, X. Zhang, and Y. He, "A SLAM algorithm based on the central difference Kalman filter," in *Proc. IEEE Intell. Vehicles Symp.*, China: IEEE, Jun. 2009, pp. 123–128, doi: 10.1109/IVS.2009.5164264.
[46] M. U. Ali, A. Zafar, S. H. Nengroo, S. Hussain, M. J. Alvi, and H.-J. Kim, "Towards a smarter battery management system for electric vehicle applications: A critical review of lithium-ion battery state of charge estimation," *Energies*, vol. 12, no. 3, p. 446, Jan. 2019, doi: 10.3390/en12030446.
[47] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158, doi: 10.1109/ASSPCC.2000.882463.

AYODEJI S. OGUNDANA (Student Member, IEEE) received the B.S. degree in mechanical engineering from the Federal University of Technology Akure (FUTA), Nigeria, in 2012, and the M.S. degree in electrical engineering from Florida State University, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and control systems for electric-vehicle battery management systems.

PRANAYA K. TERALA (Student Member, IEEE) received the B.S. degree in electrical and communications engineering from the SRM Institute of Science and Technology, India, in 2016, and the M.S. degree in electrical engineering from Florida State University, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include energy storage technologies and machine learning for electric vehicle battery monitoring systems.

MIGARA Y. AMARASINGHE (Student Member, IEEE) received the B.S. degree in computer engineering from the joint Florida A&M University–Florida State University College of Engineering. He is currently pursuing the Ph.D. degree in electrical engineering with Florida State University, with a focus on machine learning and high-performance computing.

He was a Student Ambassador with Intel Corporation. His role enables him to engage in advanced research projects encompassing high-performance computing, machine learning, and generative AI. His research contributions have led to several conference participations and presentations. He is an active member of the American Society for Engineering Education (ASEE).



XUANCHEN XIANG received the B.S. degree in electrical engineering and automation from Huazhong University of Science and Technology (HUST), China, in 2017, and the M.S. degree in electrical engineering from Florida State University, in 2018, where she is currently pursuing the Ph.D. degree. Her research interest includes deep reinforcement learning control in power systems.

SIMON Y. FOO (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of South Carolina, in 1983, 1985, and 1988, respectively. In 1988, he joined Florida A&M University–Florida State University College of Engineering, where he is currently a tenured Professor with the Department of Electrical and Computer Engineering. He was formerly the ECE Department Chair, from 2010 to 2018. His research contributions have been in the areas of machine learning, particularly in deep learning as well as in high-efficiency multi-junction III–V compound, polymer, and quantum dot solar cells. He has authored or coauthored more than 100 refereed technical articles, was awarded a patent on high-efficiency multi-junction solar cells, in 2013, and contributed to two book chapters. He has graduated more than 50 M.S. and Ph.D. students. He is also the principal investigator of more than 20 funded research projects with a total funding of more than U.S. $3 million. His primary research sponsors include the Office of Naval Research, the National Security Agency, the National Science Foundation, the Air Force, Boeing Aircraft Company, and Florida Department of Transportation. He also served as a Technical Advisor with Airbus in the area of fire detection in aircraft cargo bays, in 2002. He has been a member of the Eta Kappa Nu Electrical Engineering Honor Society, since 1982. He has won several awards, including the Engineering Research Award from FAMU–FSU College of Engineering, in 2004 and 2008, the Engineering Honor Society Tau Beta Pi Teacher of the Year Award, and the best paper awards at international conferences.