



## State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach

Ephrem Chemali<sup>a,\*</sup>, Phillip J. Kollmeyer<sup>a</sup>, Matthias Preindl<sup>b</sup>, Ali Emadi<sup>a</sup>

<sup>a</sup> Department of Electrical and Computer Engineering, McMaster Institute for Automotive Research and Technology, McMaster University, Hamilton, ON, Canada

<sup>b</sup> Department of Electrical Engineering, Columbia University in the City of New York, New York, NY, USA



### HIGHLIGHTS

- Deep neural network used to map battery signals directly to SOC.
- Deep neural network self-learns network weights.
- Neural network SOC estimator is shown to be computationally efficient.
- Increased SOC estimation accuracy and robustness by adding noise to training data.
- One deep neural network learns to estimate SOC over many ambient temperatures.

### ARTICLE INFO

#### Keywords:

Battery management systems  
Deep neural networks  
Energy storage system  
Li-ion batteries  
Machine learning  
State of charge estimation

### ABSTRACT

Accurate State of Charge (SOC) estimation is crucial to ensure the safe and reliable operation of Li-ion batteries, which are increasingly being used in Electric Vehicles (EV), grid-tied load-leveling applications as well as manned and unmanned aerial vehicles to name a few applications. In this paper, a novel approach using Deep Feedforward Neural Networks (DNN) is used for battery SOC estimation where battery measurements are directly mapped to SOC. Training data is generated in the lab by applying drive cycle loads at various ambient temperatures to a Li-ion battery so that the battery is exposed to variable dynamics. The DNN's ability to encode the dependencies in time into the network weights and in the process provide accurate estimates of SOC is presented. Moreover, data recorded at ambient temperatures lying between –20 °C and 25 °C are fed into the DNN during training. Once trained, this single DNN is able to estimate SOC at various ambient temperature conditions. The DNN is validated over many different datasets and achieves a Mean Absolute Error (MAE) of 1.10% over a 25 °C dataset as well as an MAE of 2.17% over a –20 °C dataset.

### 1. Introduction

Li-ion batteries are not only heavily used in most portable electronics and Electric Vehicles (EV) but are also used in smart-grid technology for load levelling as well as in newer technologies like Unmanned Aerial Vehicles and passenger drones aimed for medium to short range distances [1]. This can be attributed to many advantages that Li-ion batteries offer over other batteries. These include a high specific energy and energy density which allows electrified vehicles longer electric-only driving range, high cycle life, high Coulombic efficiency (up to 98%) and low self-discharge [2,3]. In 2015, 50% of all nitrogen oxide air pollutants in the world, corresponding to 53 million tonnes of airborne nitrogen oxide emissions, can be attributed to the

transportation sector. Furthermore, half of the overall health-related economic cost, estimated to be \$865 billion in 2010, is credited to air pollution [4]. Nowadays, some countries are taking proportionate action to counteract these negative effects by banning new petrol and diesel powered vehicles by 2030 or as early as 2025, in the case of Norway. Due to the advantages of the Li-ion batteries outlined above, electrified vehicles powered by Li-ion batteries are currently one of the best ways to mitigate these issues.

A reliable state of charge estimation is required to ensure an accurate gauge of a vehicle's remaining driving range as well as proper balancing of the battery pack [3,5,6]. Due to unpredictable driving habits and the repeated acceleration and deceleration of a vehicle, the battery can be exposed to highly dynamic load demands. As a result of

\* Corresponding author. Department of Electrical and Computer Engineering, McMaster Institute for Automotive Research and Technology, McMaster University, Hamilton, ON, Canada.

E-mail address: [chemale@mcmaster.ca](mailto:chemale@mcmaster.ca) (E. Chemali).

these dynamic load demands, SOC estimation is a tedious task. SOC is not an observable quantity, therefore its accurate estimation becomes essential for reliable and safe operation of the vehicle [5,7].

SOC is defined as the remaining charge within the battery and is defined as the ratio of the residual capacity of the battery to its nominal capacity [3]. The relationship between the battery's observable signals to the estimated SOC is a highly non-linear one, varying with temperature and discharge/charge currents [8,9]. Traditionally, the two main estimation methods have used *open circuit voltage* based techniques and *coulomb counting* [7,10]. These methods are known to have their limitations and have been generally displaced by more sophisticated methods. They include Luenberger observer [7,11], adaptive observer [7,12], sliding mode observer [7,13,14], and Kalman filters [15–17]. Typically, in observer methods, the parameters of an equivalent circuit model like resistances and open circuit voltage are fit to observed battery current and voltage data. An estimate is issued by mapping these parameters to SOC. In Kalman filter-based algorithms, it is typically required to linearize around an operating point which can significantly increase computational load. The measured current, voltage and the previously estimated SOC are provided to the algorithm and the filter issues an estimate of SOC at the next time step. These techniques are often tied to some battery model, like a lumped parameter model or an equivalent circuit model which require arduous model identification to adequately represent the non-linear behavior of a battery. In addition, they often require large numbers of parameters or different versions of the model to perform SOC estimation at varying ambient conditions.

Strategies involving classic machine learning algorithms have also been used in the past. The benefit of these types of techniques is that they can be trained with real world data and self-learn SOC estimation without the need for hand-engineered models. However, when neural network were solely used, the results were typically not accurate enough, and therefore required the additional use of Kalman filters or other inference mechanisms to achieve sufficient estimation accuracy. Although some works have used Kalman filters in conjunction with combined battery models or equivalent circuit battery models [18], many other works have also used them in conjunction with NN battery models. In Ref. [19], a trained 2-layer Neural Network (NN) with 30 neurons in the hidden layer estimates terminal voltage within a 4% Root Mean Square (RMS) error. However, to estimate SOC and to further reduce the RMS error to 2%, the NN is used as a battery model in an Extended Kalman Filter (EKF). In Ref. [20], an Extreme Learning Machine is used at a constant ambient temperature of 25 °C. An SOC estimation error of under 1.5% is claimed however this is only achieved in conjunction with a Kalman filter as well. Furthermore, the extreme learning machine is trained on constant discharge pulses hence their performance in transient load demand, experienced in real world scenarios, is unknown. In Ref. [21], a SVM is used with a moving window to increase computational efficiency when modeling the battery; a Mean Absolute Error (MAE) of less than 2% is achieved. However, as is the case for the above works, it achieves this MAE in conjunction with an EKF. In Ref. [22], a load classifying neural network is trained on 12 US06 drive cycles however different neural networks are used for idling, charging and discharging operation. The method achieves an average estimation error of 3.8% or 2.6% when additional filtering is performed. Furthermore, validation is performed on pulse discharge tests hence the method's performance in real world applications is unknown.

More recently, additional works have utilized model-based and machine learning-based approaches for battery SOC estimation. One such approach uses a moving average estimation with a reduced electrochemical model which is able to perform estimation without linearization error and allows for constraints on states like the internal resistance state and Li-ion concentration [23]. In Ref. [24], a fuzzy C-means and subtractive clustering method is used along with a SVM for SOC estimation. The work performed in Ref. [25] builds on the latter fuzzy-SVM approach by using a genetic algorithm-based fuzzy C-means clustering technique with a backpropagation algorithm to estimate SOC and is claimed to outperform classical fuzzy modeling techniques.

Advancements in modern machine learning techniques are accelerating faster than ever before due to constantly improving computing power and increased access to vast pools of data. Nowadays, machine learning algorithms have become deeply entrenched in our lives. They are now the dominant algorithms used for object recognition in images and video sequences, natural language processing on smartphones and predictive analytics in many industries, to name a few [26].

This work showcases how a machine learning technique like Feedforward Neural Networks (FNN) as well as Deep Feedforward Neural Networks (DNN), can accurately estimate SOC without the help of Kalman filters or any other inference methods. Specifically, this work contributes the following novelties. (1)A DNN can map observable signals from the battery like voltage, current and temperature directly to the battery SOC, avoiding additional filters and estimation algorithms like Kalman filters used in traditional systems. (2)The DNN can self-learn its own weights by using learning algorithms like gradient descent. This is markedly different than incumbent techniques like lumped parameter models, equivalent circuit or electrochemical models which require a great deal of time to hand-engineer and parameterize. (3)It will be shown that one DNN can learn to estimate SOC at different ambient temperature conditions. This is beneficial since traditional estimation techniques must use different models or different look-up tables for estimation at different ambient temperatures.

After a brief introduction, the second section will discuss the deep neural networks constructed in this work. In the third section, the experimental apparatus for the battery testing and data logging is described. In the fourth section, the performance of the DNN is tested with many validation datasets recorded at constant and at varying ambient temperatures.

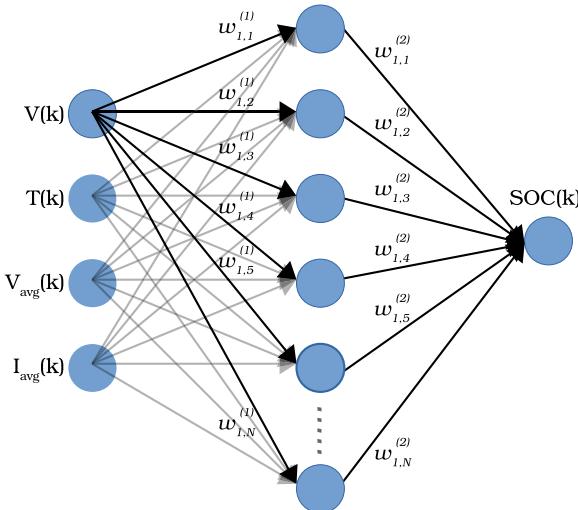
## 2. Deep neural networks for SOC estimation

There are many examples where deep learning architectures have made significant improvements over conventional algorithms. In 2012, AlexNet, a deep convolutional neural network won the ImageNet competition where teams are tasked with classifying over 1 million high resolution images in 1000 different categories. AlexNet achieved a top-5 error rate of 15.3% compared to a more traditional model taking second place with a top-5 error of 26.2% [27]. Recently, Microsoft Research's deep learning algorithm, called a deep residual network, won the 2015 ImageNet challenge with an error rate of 3.57% which even surpasses human level accuracy valued at 5.1% [28].

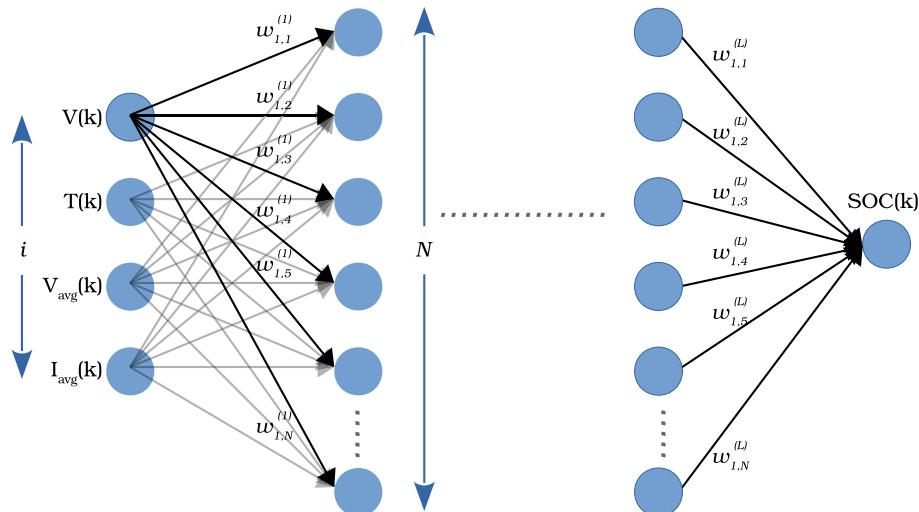
Traditional machine learning techniques contain no more than one or two layers of non-linear and linear transformations [29]. With the advent of faster computational power and an abundance of available real world data, deeper architectures were investigated which, in many cases, allowed researchers to make striking improvements in many applications [27,30–33].

Feedforward neural networks, whose 2-layer and multi-layer DNN architectures are shown in Fig. 1, can, in principle, model any non-linear system by mapping the observables to a desired output. Once trained off-line, FNN and DNN offer fast computational speeds online since they are composed of a series of matrix multiplications, as opposed to other strategies which can contain computationally intensive calculations like partial differential equations. When FNN and DNN are applied for SOC estimation, a typical dataset that is used to train the networks is defined by  $D = \{(\psi(1), SOC(1)^*), (\psi(2), SOC(2)^*), \dots, (\psi(\tau), SOC(\tau)^*)\}$ , where  $SOC(t)^*$  and  $\psi(t)$  are the ideal state-of-charge value and the vector of inputs at time step  $t$ , respectively. The current measurement used to determine the ideal  $SOC(t)^*$  is described in more detail in the next section of this paper. The vector of inputs is defined as  $\psi(t) = [V(t), T(t), I_{avg}(t), V_{avg}(t)]$  where  $V(t)$ ,  $T(t)$ ,  $I_{avg}(t)$  and  $V_{avg}(t)$  represent the voltage, temperature, average current and average voltage of the battery at time step  $t$ . The average current and voltage are both calculated over  $\xi$  precedent time steps, which ranged from 50 to 400 time steps. This is not to be confused with the total dataset time span defined by  $\tau$ , where  $\xi < \tau$ . Many different types of inputs

## 2-Layer Network



## L-Layer Network



**Fig. 1.** Architecture of 2-layer neural network (top) and architecture of Deep Neural Network (DNN) (bottom). The input data is given by  $\psi(t) = [V(t), T(t), I_{avg}(t), V_{avg}(t)]$  where  $V(t)$ ,  $T(t)$ ,  $I_{avg}(t)$  and  $V_{avg}(t)$  represent the voltage, temperature, average current and average voltage of the battery at time step  $t$ . The output of the DNN is the estimated SOC at every time step.

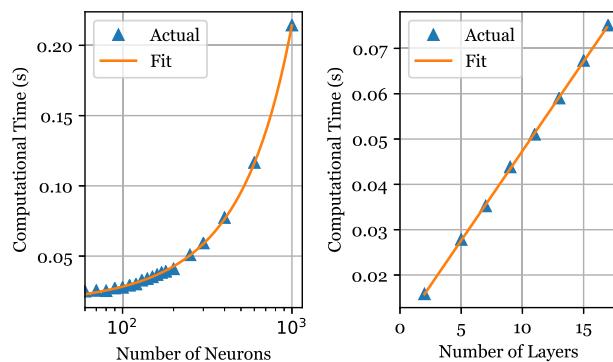
were examined and it was found that  $I_{avg}$ ,  $V_{avg}$  are better values to feed into the network rather than feeding many antecedent values of current and voltage. The number of weights in the first layer of the network increases proportionally to the number of additional inputs. For example, in a network which has 10 neurons in the first hidden layer, each additional input will require 10 additional weights for a fully-connected layer. Therefore, better computational efficiency and lower memory requirements are achieved by capturing the system dynamics in time through an averaging of current and voltage.

In this paper, TensorFlow [34], a machine learning framework, is used in conjunction with two NVIDIA Graphical Processing Units (GPU); a TITAN X and a GeForce GTX 1080 TI GPU. The TensorFlow framework provides the ability to quickly prototype and test different network architectures as a result of its ability to automatically compute backpropagation. The latter describes the process by which the network weights can be updated at the end of every training epoch and will be further discussed below. Although training is performed on a GPU to

take advantage of parallel computing and to accelerate the training process, when performing real time estimation after training is complete, the algorithms discussed in his paper can be flashed on a standard microprocessor. In this work, typical training times required to train a DNN can be anywhere from a few hours to 40–50 hours, depending on whether the DNN is trained over a single ambient temperature or over 5 different ambient temperatures.

Feedforward neural networks are matrix-based and can be represented by the below composite function. A few important variables are first defined. Let  $w_{j,k}^l$  denote the weight connection between neuron  $j$  in layer  $l - 1$  and neuron  $k$  in layer  $l$ . Let  $b_k^l$  and  $h_k^l$  be the bias and the activation function, respectively, of neuron  $k$  in layer  $l$ . The activation function is computed as follows;

$$h_k^l(t) = \eta \left( \sum_k (w_{j,k}^l h_k^{l-1}(t) + b_k^l) \right) \quad (1)$$



**Fig. 2.** Experimentally verified relationships between computational time and number of neurons as well as number of layers. (right) Semi-log plot of computational time vs. the number of neurons. The number of layers in the DNN is fixed to 4 layers. (left) Computational time as a function of the number of layers. The number of neurons in each layer of the DNN is fixed to 50 neurons.

**Table 1**  
Computational time of a DNN and of an extended Kalman filter.

SOC Estimation Technique	Computational Speed (seconds)
DNN (2 layers, 256 neurons)	0.07
Combined Model + EKF	0.66

Results obtained in MATLAB using Intel Xeon E5630, 2.53 GHz and 32 Gb of RAM.

**Table 2**  
Tested drive cycles and corresponding battery cell power characteristics.

Test	Use	Mean	RMS	Peak
		Power(W)	Power(W)	Power(W)
Cycle 1	Training	3	7	35
Cycle 2	Training	3	7	35
Cycle 3	Training	3	7	35
Cycle 4	Training	3	7	35
US06	Validation	6	13	34
HWFET	Validation	5	6	19
UDDS	Training	2	4	19
LA92	Training	2	7	35
NN	Training	3	10	39

where,

$$h_k^l(t) = SOC(t) \text{ for } l = L \quad (2)$$

$SOC(t)$  is the estimated state-of-charge at time step  $t$  calculated by the network. Due to its simplicity during training as well as during validation, the nonlinearity used in these networks is called Rectified Linear Units (ReLU) and is given by;

$$\eta = \max(0, h) \quad (3)$$

To understand how accurate the SOC estimate is compared to the ideal SOC value, an error signal is generated at the output of the network for each time step and is given by

$$e(t) = SOC(t) - SOC^*(t) \quad (4)$$

To gain a good understanding of the overall loss of the network, the following loss function is computed at the end of a forward pass;

$$\mathcal{L} = [\max(e)]^2 + \frac{1}{\tau} \sum_{t=0}^{\tau} (e(t))^2 \quad (5)$$

where  $\tau$  is the length of the sequence, as mentioned above. Both the average error and the maximum error are included in the overall error function so that the network expends its energy on minimizing both the

average error as well as the maximum error value and does not prioritize the minimization of one value over another. A forward pass starts when the training data is fed into the network and is complete when the SOC estimate for each time step is calculated as well as when the overall loss function is determined. A full training epoch,  $\epsilon$ , includes one forward pass and one backward pass; describing the process of sending the overall loss signal backward through the network to update the weights. To perform this backward pass, an optimization method called Adam [35] is used, which updates the network weights and biases based on the gradient of the loss function. This is defined by the following composite function;

$$\begin{aligned} m_\epsilon &= \beta_1 m_{\epsilon-1} \nabla \mathcal{L}(w_{\epsilon-1}) \\ r_\epsilon &= \beta_2 r_{\epsilon-1} \nabla \mathcal{L}(w_{\epsilon-1}) \\ \tilde{m}_\epsilon &= m_\epsilon / (1 - \beta_1^\epsilon) \\ \tilde{r}_\epsilon &= r_\epsilon / (1 - \beta_2^\epsilon) \\ w_\epsilon &= w_{\epsilon-1} - \alpha \frac{\tilde{m}_\epsilon}{\tilde{r}_\epsilon - \kappa}, \end{aligned} \quad (6)$$

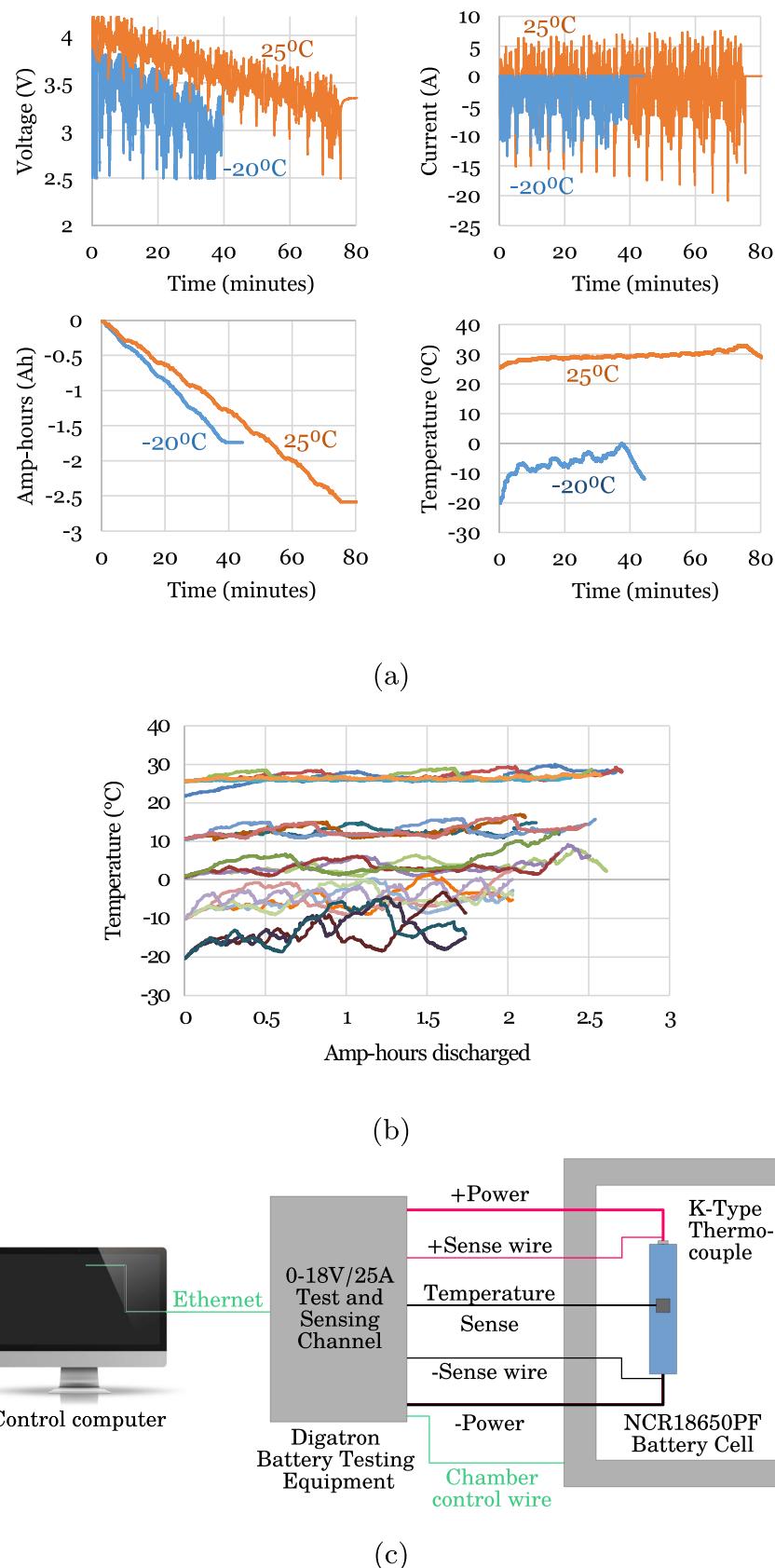
where  $\beta_1$  and  $\beta_2$  are decay rates set to 0.9 and 0.999, respectively,  $\alpha = 10^{-4}$  is the learning rate and  $\kappa$  is a constant term set to  $10^{-8}$ . The network parameters at the current training epoch are denoted by  $w_\epsilon$ . The network weights as well as the biases are initialized with a normally distributed random number generator having mean 0 and standard deviation of 0.05. It is only during training that a forward and backward pass are performed to iteratively update the network weights until a convergence criteria is met. With the backward pass, the network self-learns its weights and biases, offering significant improvements over traditional SOC estimation strategies which require the time-consuming construction and parameterization of hand-engineered battery models.

Training of the networks is performed offline and only when the convergence criteria is met and the overall loss function is minimized can the networks be applied online. During validation, which simulates online operation, we need only perform a forward pass to calculate the estimated SOC values at each time step and no backward pass is required since the network weights and biases have already been learned offline during training. FNNs and DNNs offer an advantage of lower computational overhead, once trained, since a forward pass is comprised mainly of consecutive matrix multiplications. This, in general, has higher computational efficiency than other algorithms, which might contain differential equations. This point is further quantified below through a comparison between the computational time of a DNN and that of an EKF model. In addition, a DNN, as will be shown in the results section of this paper, has the ability to encode the inherent response of a battery at various ambient temperatures thus reducing the memory required to store different parameters for different ambient temperatures as is typically done in traditional battery models. This offers great benefits when performing modelling and state estimation for large quantities of cells found in vehicle battery packs.

To evaluate the SOC estimation performance of our networks, a few different performance metrics are used. These include the Mean Absolute Error (MAE), the standard deviation of the errors (STDDEV) and finally the maximum error (MAX).

To gain a better understanding of the computational overhead discussed above, an analytical representation of the computational efficiency of these DNNs is given. Since most of the computational cost is due to floating-point operations, then it follows that the computational time of a forward pass is proportional to the number of multiplication and addition operations. Referring to Fig. 1, for each neuron of the DNN, there exists  $d$  multiplications and  $d - 1$  additions resulting, ideally, in  $2d - 1$  operations per neuron. Let  $N$  be the number of neurons in each layer,  $\tau$  be the total number of time steps in the sequential dataset and  $L$  be the number of layers in the DNN, then the number of floating-point operations is given by;

$$F(\tau, L, N) = \tau [N^2(2L - 4) + N(2d - L + 3) - 1] \quad (7)$$



**Fig. 3.** (a) US06 drive cycle applied on the same battery cell at an ambient temperature of 25 °C (orange) and –20 °C (blue). Recorded voltage (top right), current (top left), capacity (bottom left) and cell temperature (bottom left). (b) Cell temperature vs. Ah count for the different drive cycles applied on the same battery cell recorded at 5 different ambient temperatures. (c) Schematic of the battery test bench and data logging system. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

**Table 3**  
Panasonic 18650 PF cell parameters.

Nominal Open Circuit Voltage	3.6 V
Capacity	Min. 2.75 Ah/Typ. 2.9 Ah
Min/Max Voltage	2.5 V/4.2 V
Mass/Energy Storage	48 g/9.9 Wh
Minimum Charging Temperature	10 °C
Cycles to 80% Capacity	500 (100% DOD, 25 °C)

**Table 4**  
Test equipment specifications.

Cycler Manufacturer	Digatron Firing Circuits
Test Channel Used	25 A, 0–18 V channel
Voltage/Current Accuracy	± 0.1% Full Scale
Data Acquisition Rate Used	10 Hz
Thermal Chamber	Cincinnati Sub Zero ZP-8
Size	8 cu. Ft.
Accuracy	± 0.5 °C

Therefore, when letting  $\tau$  and  $L$  remain constant, it is immediately clear that the number of floating-point operations grows as the square of the number of neurons or in Big-O notation as  $\mathcal{O}(N^2)$ . On the other hand, if  $\tau$  and  $N$  remain constant, and the number of layers is allowed to vary, then it follows that the number of floating-point operations grows linearly with the number of layers,  $L$ , or in Big-O notation as  $\mathcal{O}(L)$ . Hence, it is typically preferred that networks grow in depth rather than in width to reduce computational time. This relationship is tested by timing the forward pass for varying numbers of neurons as well as for varying numbers of layers and are shown in Fig. 2.

Although a good understanding of a DNN's computational complexity is important, it is also essential to compare the computational efficiency of the DNN with respect to other competing algorithms. To this end, the computational time of the DNN is compared to that of an EKF approach used in Ref. [18]. The goal of this test was to measure the time required by these two techniques in order to compute through 13000 data points. The results are shown in Table 1 and are based on an average of 20 separate executions of the two algorithms over the same dataset. Therefore, on average, the DNN is almost an order of magnitude faster than the EKF. This can be attributed to the fact that the EKF is required to linearize the non-linear system at every time step while the DNN mainly performs a series of matrix multiplications to achieve the desired output which is an inherently much simpler series of operations. The DNN outlined in Table 1 requires less than 3000 floating point operations per second (FLOPs), including the data preprocessing phase, which is within the realm of many conventional embedded processors able to handle up to  $10^9$  FLOPs.

### 3. Experimental data collection for training and validation of neural network SOC algorithm

#### 3.1. Drive cycles

A series of vehicle drive cycle and charging profiles were applied on a battery cell, and the resulting measured data [36] is used to train and validate the neural network SOC algorithm. The nine different drive cycles used are given in Table 2. The cycles were chosen to cover a range of mean, RMS, and peak power values for the battery, as would be experienced in a vehicle application, and to provide enough data to train the neural network. The tests were also performed at a range of temperatures, down to  $-20^{\circ}\text{C}$  where nonlinear resistance, battery self-heating, and diffusion effects make SOC estimation more tedious. Procedurally, the dynamometer drive schedules obtained from the United States Environmental Protection Agency (EPA) are provided to an electric vehicle model in the form of a speed profile and is used to

characterize vehicle fuel consumption [37]. A power profile is generated based on a large light duty passenger vehicle with a 35 kWh battery pack for each of these drive cycles. These are then scaled for a single battery cell in order to apply these drive cycles in the laboratory.

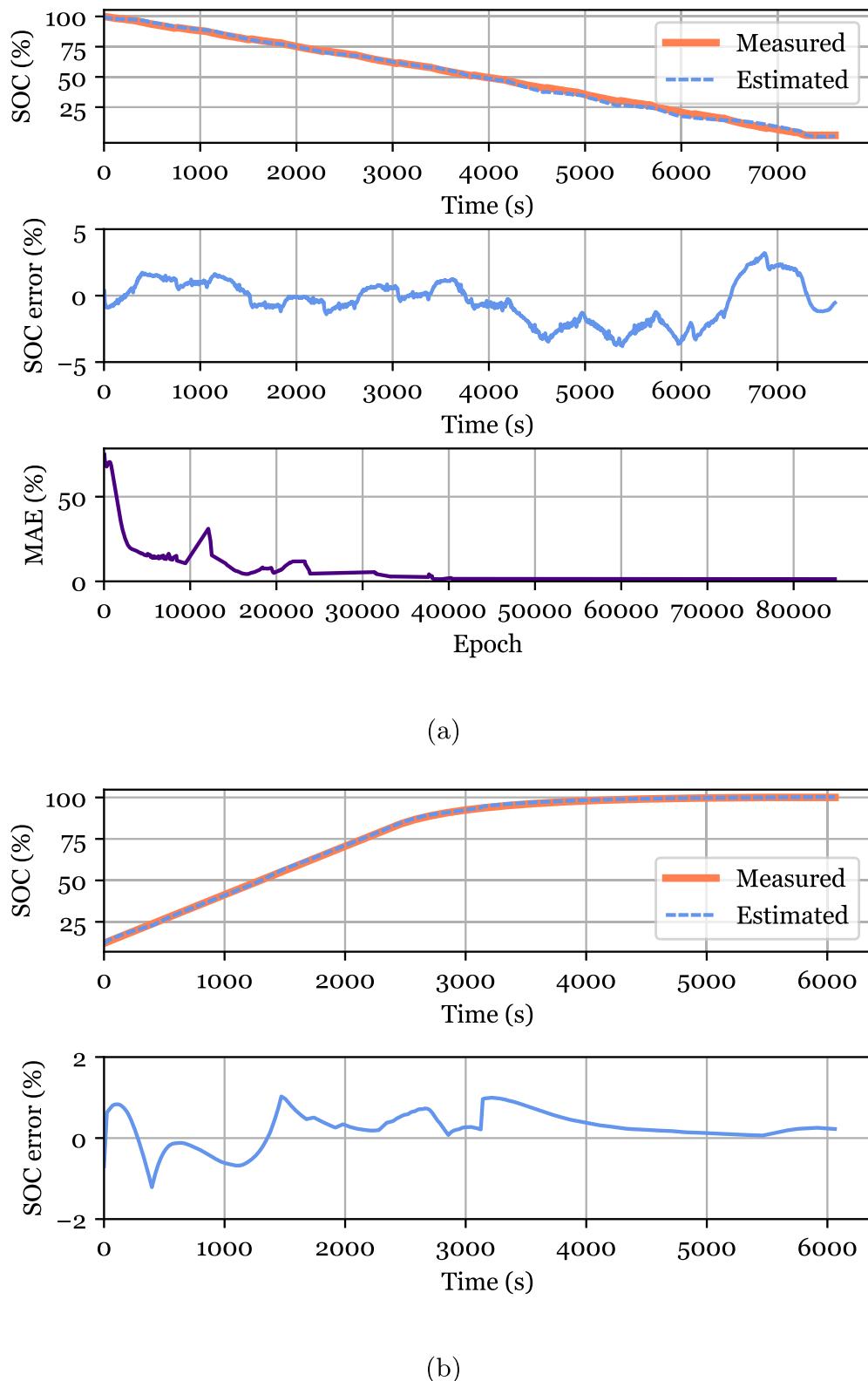
Four standard drive cycles are used in addition to a custom drive cycle which was built for this work. The standard drive cycles used are the Urban Dynamometer Driving Schedule (UDDS), the Highway Fuel Economy Driving Schedule (HWFET), the Unified Driving Schedule (LA92) and the Supplemental Federal Test Procedures or US06 [37]. The custom drive cycle, called the NN, was designed for this work and has additional dynamics useful for training the neural network. As an example, the voltage, current, capacity and cell temperature recorded from the same battery cell throughout the US06 drive cycle are shown in Fig. 3a). Fig. 3b) shows the cell temperature as a function of the amp-hour count for some of the different drive cycles applied to the same battery cell at the five different ambient temperatures. In compliance with the battery manufacturer's recommendations, regenerative braking below  $10^{\circ}\text{C}$  is not performed which is the reason for the lack of charging current at  $-20^{\circ}\text{C}$  in Fig. 3a). The other four cycles, cycles 1 through 4, are composed of a mixture of the four standard drive cycles performed in a random, repeating order.

The DNNs are validated using the US06 cycle, an aggressive and highly dynamic profile, and the HWFET cycle, a profile with moderate dynamics. The data for the remaining seven tests are used to train the neural network, as is indicated in Table 2.

#### 3.2. Test procedure, battery cell, and cycling equipment

The test procedure used is as follows: (1) set the thermal chamber temperature to  $25^{\circ}\text{C}$ , (2) wait 10 min or until the measured battery temperature is greater than  $10^{\circ}\text{C}$ , (3) perform a full charge, (4) set the chamber to the current test temperature, (5) wait for the battery temperature to stabilize, and (6) perform the drive cycle. The drive cycle is repeated until the battery voltage reaches 2.5 V for the  $25$  and  $10^{\circ}\text{C}$  tests. For the lower temperature tests, the battery resistance is much higher, causing the terminal voltage to hit 2.5 V periodically throughout the test. To accommodate the higher resistance, an amp-hour cut off 2.32Ah at  $0^{\circ}\text{C}$ , 2.03Ah at  $-10^{\circ}\text{C}$ , and 1.74Ah at  $-20^{\circ}\text{C}$  is used instead. The battery tester still has a lower voltage limit of 2.5 V, but the test is not terminated when this voltage is hit. The tester reduces current to prevent voltage from falling below the limit. The battery may hit the lower voltage limit several times when at higher currents and low SOC. The nine drive cycle tests were performed in the order given in Table 2, first at  $25^{\circ}\text{C}$ , and then at  $10$ ,  $0$ ,  $-10$ , and  $-20^{\circ}\text{C}$ . Following those tests, the nine drive cycles were also performed at a variable ambient temperature where the temperature was increased from  $10^{\circ}\text{C}$  to about  $25^{\circ}\text{C}$ . These varied tests are used to evaluate the SOC estimation algorithm for cases when the battery is warming up throughout the drive.

The tested battery cell is a Panasonic NCR18650PF, nickel cobalt aluminum (NCA) chemistry Li-ion battery, with a nominal capacity of 2.9Ah and other specifications as described in Table 3 [38,39]. The battery cell is tested with a 25 amp, 0–18 V rated channel of a Digatron Firing Circuits Universal Battery Tester, and placed in a Cincinnati Sub Zero thermal chamber, as is described in Table 4. The battery tester is very accurate; rated for less than 0.1% error, which is important because the measured battery amp-hours is used to calculate the reference or ideal SOC value. The maximum error of 0.1% translates to a 25 mA current error, which would result in a maximum of 25 mAh of error for a 1 h drive cycle test, a value small enough to not have a major effect on the results. The battery system and instrumentation is illustrated in Fig. 3c), which shows that the voltage sensing leads are connected directly to the battery terminals and that a thermocouple is connected to the center of the case.



**Fig. 4.** (a) DNN estimation accuracy, estimation error over the HWFET discharge validation dataset recorded at 25 °C and the mean absolute error as a function of training epochs. (b) Estimation accuracy and estimation error over validation charging profile also recorded at 25 °C. DNN is composed of 3 layers and 4 neurons in each layer.

### 3.3. Training data augmentation

To robustify the deep neural network, the training data is

augmented by injecting Gaussian noise into the battery measurement signals. Specifically, a normally distributed random number generator is used to generate noise for each of the battery's signals. We specify the

**Table 5**

SOC estimation accuracy of deep neural network trained on fixed ambient temperature data of 25 °C during validation.

Validation Dataset	MAE (%)	STDDEV (%)	MAX (%)
HWFET Dataset	1.35	0.94	3.80
US06 Dataset	1.85	1.20	5.14
Charging Dataset	0.39	0.26	1.21

mean and variance of this normally distributed random number generator based on typical noise levels observed in off-the-shelf measurement devices. In particular, Gaussian noise with 0 mean and a standard deviation of 2–4% is injected into the voltage, current and temperature measurements. In addition, to robustify against offsets and gains inherent in battery measurement devices, an offset is applied to all measurement signals and a gain is applied to the current measurement. Specifically, an offset of up to  $\pm 150$  mA and a gain of up to  $\pm 3\%$  is applied to the current measurements, an offset of up to  $\pm 5$  mV is applied to the voltage measurement and an offset of up to  $\pm 5$  °C is applied to the temperature measurement. New versions of the initial recorded training data is created with different noise, offset and gain levels. We create up to 20 new versions of the training data.

All the results showcased in this work are based on DNNs trained on data obtained from Panasonic 18650 PF cells. Nevertheless, the DNNs discussed in this work can be trained on any other type of battery cell. The network architectures will remain unchanged if used to perform SOC estimation on another battery however, the network might need to be retrained or better yet transfer learning can be used to simply retrain the last one or last few layers of the network. This would be less time-consuming than re-parameterizing a model for an alternative battery cell as would be typically performed for traditional models.

#### 4. State-of-charge estimation results

As mentioned earlier, the vector of inputs fed into the DNNs is given by  $\psi(t) = [V(t), T(t), I_{avg}(t), V_{avg}(t)]$  where  $V(t)$ ,  $T(t)$ ,  $I_{avg}(t)$  and  $V_{avg}(t)$  represent the voltage, temperature, average current and average voltage of the battery at time step  $t$ . The output of the DNN is the estimated SOC at  $t$ . The drive cycles used for training and validation are recorded at a sampling frequency of 1 Hz and are between 4000 and 13000 s long. The following two subsections examine the SOC estimation accuracy of the DNN at fixed ambient temperature and, thereafter, at variable ambient temperatures.

##### 4.1. SOC estimation at constant ambient temperature

In this section, the DNN is trained on up to seven complete discharge datasets which are augmented as described in subsection 3.3. To verify the DNN's performance in both fast and slow dynamics, validation is conducted on the US06 and HWFET datasets, respectively. These latter datasets as well as all other datasets used to obtain performance results are only used during validation and never during training. Regenerative braking is also allowed for the discharge datasets recorded at the higher temperatures however not at lower ambient temperatures since the battery is not rated for charging at temperatures lower than 10 °C. When regenerative braking occurs at the higher temperatures, the charging current is seen to spike up to 8 A. This is useful in order to ascertain the DNN's performance at higher charging currents even if momentary in nature. These higher charging spikes cannot be maintained for longer than a few seconds since this would also exceed the peak charge current rating of the battery. To evaluate the DNN's performance in charging scenarios for longer periods of time, the DNN is applied on a charging validation dataset.

The charge rate used in this dataset is 1C. There is interest in using a charge rate that is higher than this however 1C is already twice the

recommended charge rate and selecting an even higher rate is excessive and potentially dangerous.

Good performance is observed both in slow and fast dynamics as well as in the charging validation dataset, as can be seen in Fig. 4 and in Table 5, where an MAE of 1.35%, 1.85% and 0.39% is achieved over the HWFET, US06 and charging datasets, respectively. The DNN used to obtain these results is composed of 3 layers where the first two layers each contains 4 neurons and the last layer contains 1 neuron. The network is trained for up to 85000 epochs which culminates to 5 h of training time on the GPU. The MAE calculated over the HWFET validation dataset is plotted as a function of the training epochs and is also shown in Fig. 4a.

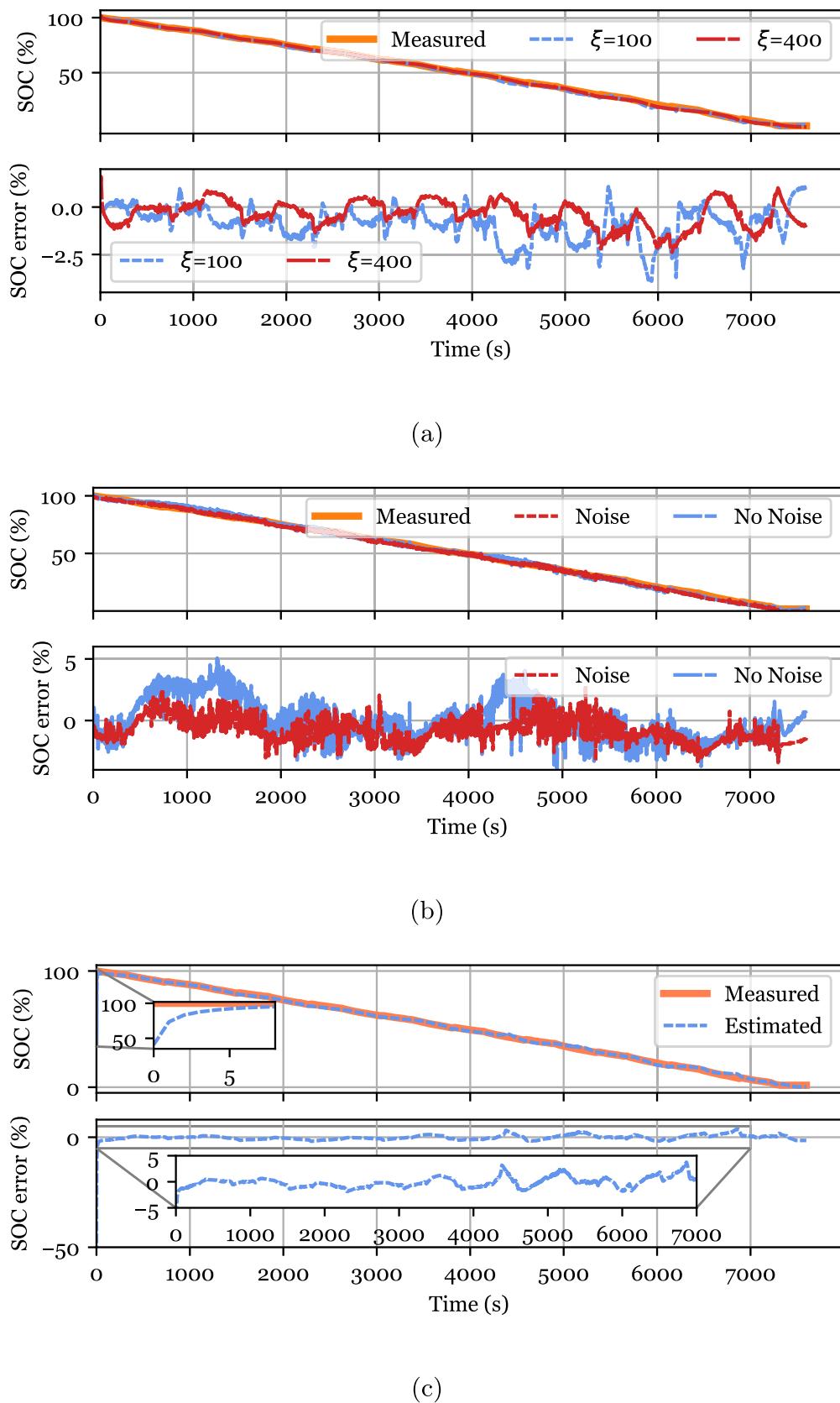
Different tests are performed to examine effects of various structural aspects of the DNN on SOC estimation accuracy. The first of such tests evaluates the influence of the  $\xi$  parameter, which is used when computing the average voltage and current of the input data, on estimation accuracy. In effect, this would examine how exposure to increased amounts of historical data could increase estimation accuracy of the DNN.

A DNN is trained on input data with  $\xi = 100$  and another is trained on input data with  $\xi = 400$ . The results from these two networks are shown in Fig. 5a) and summarized in Table 6. Exposing the DNN to 400 historical data points as opposed to 100 offers good performance gains with a reduction in the MAE and MAX of approximately 30% and 30–40%, respectively. To ensure an unbiased comparison, training is stopped at 160,000 epochs.

The battery measurements performed in the lab are obtained in isolated and controlled environments to ensure good quality data. However, in the real world, battery measurements, like voltage, current and temperature measurements, can be of variable quality. A DNN can be taught to handle such noisy environments at training time. As a result, a comparison showing the estimation performance achieved on the augmented training dataset versus the unaugmented training dataset is shown in Table 7 and in Fig. 5b). From the latter table and figure, it becomes evident that using the augmented dataset for training achieves a significant reduction in MAE and MAX of 21–41% and 32–43%, respectively. By injecting Gaussian noise, offsets and gains on the measurement devices, not only is the accuracy of the DNN improved but the network is robustified for real world application.

Another important test is to verify the DNN's performance when it is initialized incorrectly. To that end, instead of providing the correct measured initial battery voltage of 4.2 V to the network, an incorrect initial value of 3.6 V is given. This is shown in Fig. 5c), where the DNN struggles to output a good estimate of SOC at the beginning of the dataset, where the error is over 50%, but quickly converges to good SOC estimation within the first 10 s of the dataset.

A large fraction of the performance results discussed above are obtained on DNNs with 3 layers; where the first two layers contain 4 neurons and the last fully-connected layer contains 1 neuron. Interestingly, this corresponds to 36 network weights which is relatively small in size and can be flashed on any embedded processor when operating in real time. To determine an optimal network architecture, the estimation accuracy is evaluated on networks with different numbers of layers and neurons. This is shown in Fig. 6a) and in Fig. 6b), respectively. For each test case shown in Fig. 6,  $\xi = 400$  and training is stopped at 100,000 epochs to ensure an unbiased comparison. It is clear that a deeper network can achieve better estimation accuracy as going from 2 layers to 4 offers a 10% and a 24% reduction in MAE and MAX error, respectively. In the same vein, going from 4 to 6 layers reduces the MAE and MAX error by 9% and 15%, respectively, however, increasing the depth up to 8 layers achieves no reduction in MAE or MAX. This could very well be a result of the gradients, defined in eq. (6), becoming much smaller as more layers are added to the network which leads to vanishing gradients and the inability to update the network weights during training. With respect to the number of neurons in the network, Fig. 6b) indicates that going from 1 neuron to 4 per layer



**Fig. 5.** (a) Performance of DNN trained over data with  $\xi = 100$  and over data with  $\xi = 400$ . (b) Performance of DNN trained over augmented training data and DNN trained over unaugmented training data. (c) Performance over incorrectly initialized network. DNNs have 3–8 layers and 4–8 neurons per layer. Validation is performed over 25 °C HWFET dataset.

**Table 6**SOC estimation accuracy using different averaging parameter  $\xi$  in training data.

Validation Dataset	MAE (%)	STDDEV (%)	MAX (%)
HWFET with $\xi = 100$	0.96	0.76	3.91
HWFET with $\xi = 400$	0.61	0.49	2.38
US06 with $\xi = 100$	1.16	0.86	4.54
US06 with $\xi = 400$	0.084	0.61	3.14

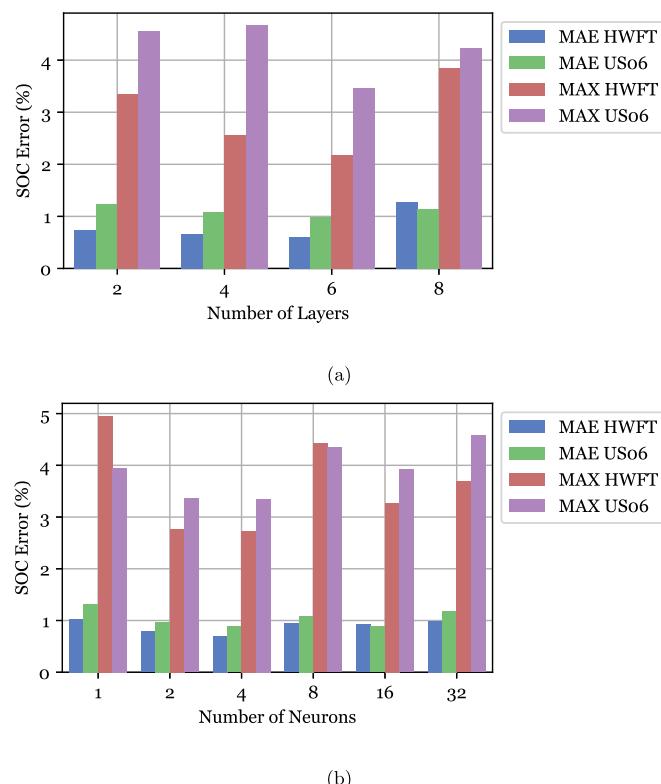
\*All validation results are obtained on 25 °C datasets.

**Table 7**

SOC estimation accuracy of DNN trained over augmented and unaugmented data.

Training Data Augmentation	Validation Dataset	MAE (%)	STDDEV (%)	MAX (%)
No	HWFET	1.35	0.94	5.04
Yes	HWFET	1.06	0.49	3.41
No	US06	2.67	2.08	15.14
Yes	US06	1.59	1.23	7.14

\*All validation results are obtained on 25 °C datasets.

**Fig. 6.** (a) Estimation accuracy measured during validation versus number of layers in DNN. The number of neurons per layer is fixed to 4 neurons. (b) Estimation accuracy measured during validation versus number of neurons. The number of layers in the DNN is fixed to 4 layers. All tests are performed over validation datasets recorded at 25 °C.

offers a reduction in MAE and MAX of 32–35% and 16–45%, respectively. However, 8 or more neurons offer no additional reduction in MAE or MAX error.

#### 4.2. SOC estimation at variable ambient temperature

In this section, two objectives are pursued. The first is to train the DNN on data taken at various ambient temperatures such that a single

DNN can estimate SOC at different ambient temperatures. The seven training datasets recorded at the 5 different ambient temperatures (25 °C, 10 °C, 0 °C, -10 °C and -20 °C), mentioned in [Table 2](#), are used for training. These datasets are augmented, as described in [subsection 3.3](#), to increase the robustness of the DNN which raises the training dataset size to 20 times the initial size. Two validation datasets from each of the 5 ambient temperatures are then used to evaluate the DNN's performance. The second objective is to investigate the DNN's capability to interpolate its ability to estimate SOC at ambient temperatures other than the ones one which it is trained. The DNN used in this section is 4 layers deep and has 8, 16, 32 and 1 neurons in the respective layers. The training data is composed of the battery voltage, temperature, average voltage and average current, as is used in the previous section, and the averaging parameter,  $\xi$  is set to 400. The time required to train this DNN is about 40 h using one of the GPUs discussed in [Section 2](#).

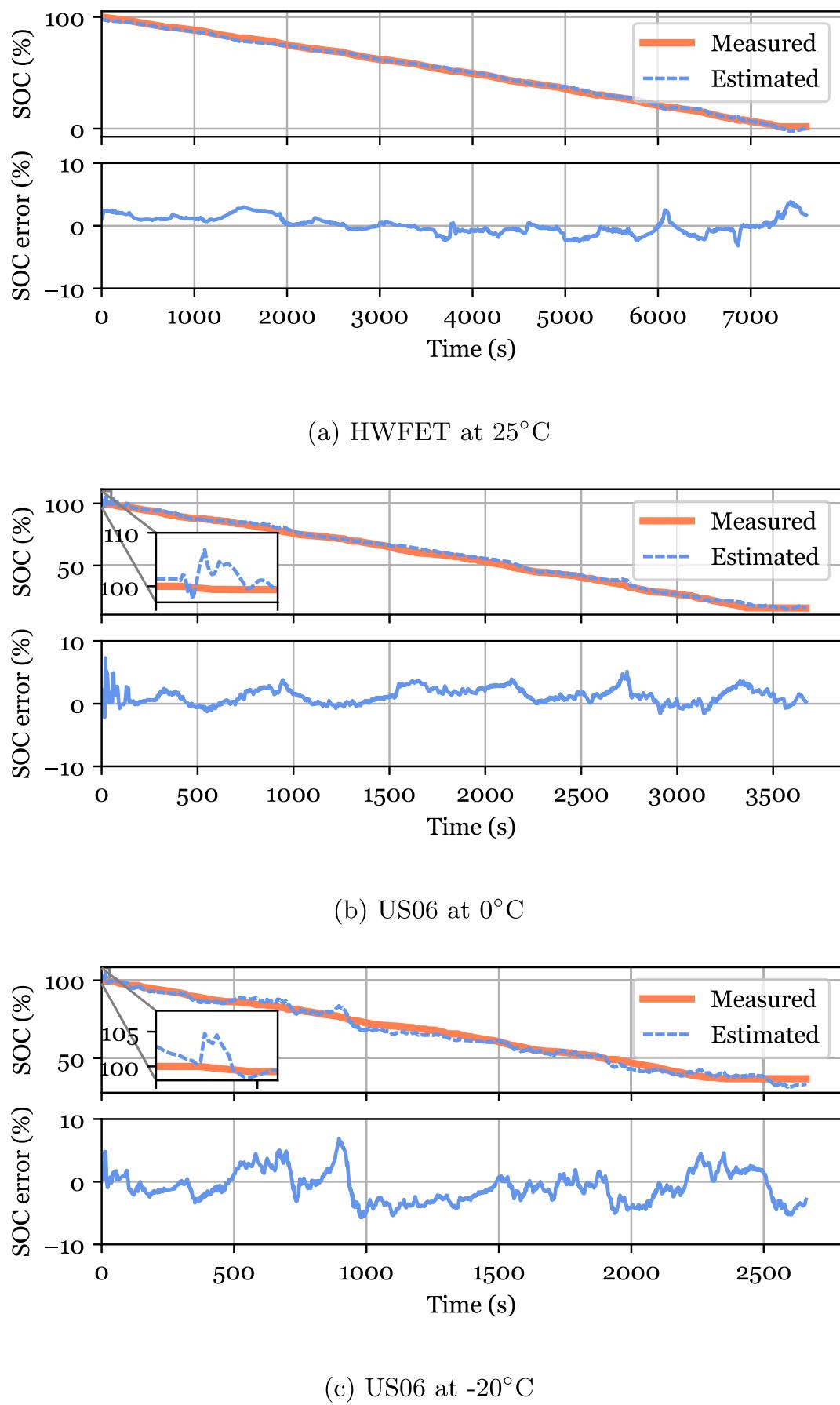
[Fig. 7](#) showcases the DNN's estimation performance over one validation dataset from 3 different ambient temperatures; 25 °C, 0 °C and -20 °C. In addition, [Fig. 8a](#) shows a bar graph outlining the DNN's performance over the two validation datasets for every one of the five ambient temperatures. Typically, higher estimation error is expected for the lower ambient temperatures since the disparity between the measured surface temperature of the battery and its internal temperature increases for lower ambient temperatures. Though, interestingly, the lowest MAE is achieved over a 0 °C dataset at 0.91%. This can be attributed to the added regenerative braking at the higher ambient temperatures which can add uncertainty. However, the MAE achieved at 25 °C is still quite low at 1.44%.

In most cases, the MAE over the HWFET datasets which are less dynamic in nature are lower than those obtained over the US06 datasets which are much more dynamic in nature. However, this phenomenon is not exhibited for the case of -20 °C. For the -20 °C ambient temperature, the MAE over the HWFET dataset is higher. This is likely because the average battery temperature for the HWFET test is -14 °C, which is much lower than the average temperature of -6.7 °C for the US06 test. While the US06 temperature is greater for all of the test cases, the temperature difference has more significance at these lower temperatures where battery resistance increases dramatically. The maximum error is not entirely representative of the performance of the DNN since a few outliers over thousands of estimated values do not represent the overall performance of the network, however MAX is provided in the results for completeness.

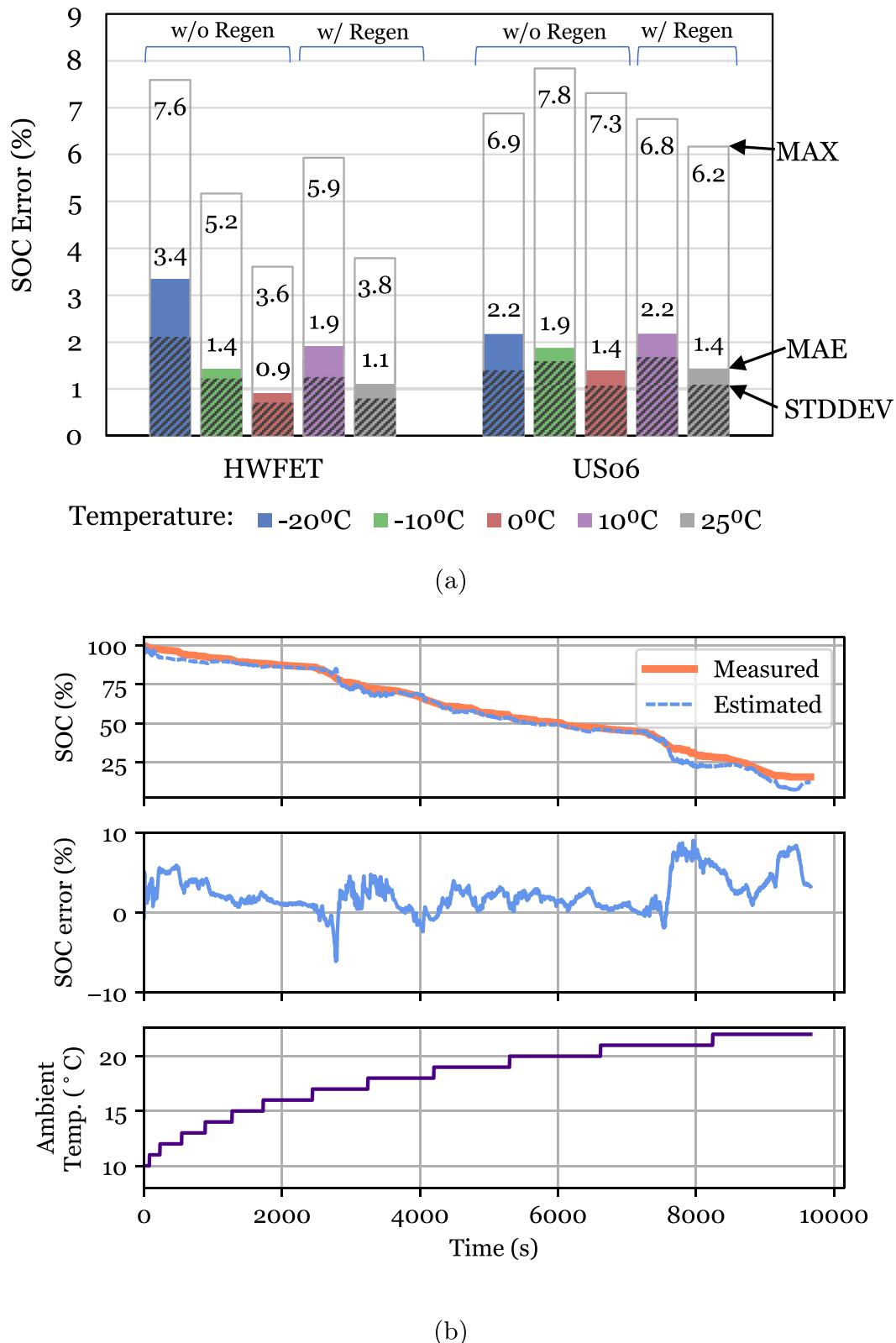
The DNN has good estimation performance when validated over constant ambient temperatures. However, depending on the geographical location within which a battery-powered vehicle may operate, a variation of 5 to 10 °C in ambient temperature is possible over the course of one day. Hence, a worthwhile exercise would be to validate the DNN over a dataset which has an ambient temperature that changes over time. Therefore, the DNN's estimation performance is tested over a validation cycle which increases from 10 °C to 25 °C over the course of the dataset, which is shown in [Fig. 8b](#). As can be seen, the DNN performs well over this validation dataset, maintaining good estimation accuracy even at ambient temperatures lying between 10 °C and 25 °C, on which the network is not trained. This interpolating ability can be a great advantage when training data is scarce.

#### 5. Conclusion

In the final analysis, this paper offers three different unique contributions. Firstly, on the modelling and estimation front, the DNNs used in this paper map the measured battery signals like voltage, current and temperature directly to SOC and achieves competitive estimation performance with MAEs below 1%. [Table 8](#) compares the DNN's estimation performance to other algorithms mentioned in literature for a more direct comparison. Secondly, on the parametrization front, the DNN self-learns all its weights, eliminating the need to hand-engineer



**Fig. 7.** Estimation performance of one DNN at 3 different ambient temperatures. DNN is composed of 4 layer, where 8, 16, 32 and 1 neurons are used in the respective layers.



**Fig. 8.** (a) SOC estimation accuracy of DNN trained on varying ambient temperature data during validation. (b) Estimation performance of DNN on validation set recorded at an increasing ambient temperature. From top to bottom, the plots shown are of the estimated SOC compared to the ideal SOC, the resulting errors at each time step and the ambient temperature. The validation dataset is composed of a mixture of HWFET, US06, LA92 and the NN drive cycle.

and parametrize traditional models, which is a very time-consuming and costly process. Thirdly, on the inference front, once trained, a DNN can operate online with relatively low computational time. In addition,

the DNNs presented in this work are robustified against measurement offsets, gains and noise such that they can retain great estimation performance regardless of the imperfections found in a vehicle's

**Table 8**

Comparison of SOC estimation error for six studies.

Method	Lowest Error	Temp.	Test Case	Li-ion Battery
Model Adaptive-Improved EKF [17]	< 1.5% MAE	25 °C	Two drive cycles, +2 to -1 A	1.1Ah A123 APR18650m1
AEKF w/ANNbattery model [19]	~3% RMS	20 °C (ambient)	± 1 A charging pulse profile	1.2Ah
AUKF w/extreme learning machine battery model [20]	~1.5% MAX	25 °C (ambient)	0.52 A 50% duty cycle pulse discharge	2.6Ah Samsung
Fuzzy NN w/genetic algorithm [40]	~0.9% APE	25 °C (ambient)	Constant resistance discharge (13 A)	10Ah MRL/ITRI
Radial Bias Function NN [41]	0.02% AAPE	Unspecified	Constant discharge rate (0.3C, 1C, 0.7C)	10Ah Lynd Power LYS347094S
DNN (estimator in this paper)	0.61% MAE 0.78% RMS 2.38% MAX (25 °C)	-20 to 25 °C (ambient)	Dynamic drive cycles, ± 18 A, Range of ambient temperatures	2.9Ah Panasonic NCR18650PF

measurement devices. Furthermore, the beauty of using this machine learning algorithm is that the intrinsic behavior of the battery at different ambient temperatures can be encoded into the weights of the DNN such that one single network can offer an accurate and robust estimation strategy at different ambient conditions. Finally, the results from extensive validation tests shown in this paper, illustrates that the DNN offers competitive estimation performance. As a result, it can be concluded that machine learning techniques are powerful tools when applied to Li-ion battery SOC estimation and potentially to other battery diagnostics. Specifically, for future work, applying these machine learning algorithms on State-of-Health estimation or State-of-Power estimation with different time horizons can be very beneficial. Other future works can also include the development of a more generalized machine learning algorithm not only able to estimate SOC at various ambient temperatures but also for different battery cells. Given the amount of data that is generated by a Li-ion battery pack is vast, it becomes instinctive to utilize data-driven approaches like machine learning models to perform state and parameter estimation. This work illustrates how these algorithms can self-learn their own weights and achieve competitive estimation performance over a large range of ambient temperatures.

## Acknowledgments

This work was supported in part by the Canada Excellence Research Chairs, Government of Canada in Hybrid Powertrain Program. Battery testing was performed at the Wisconsin Energy Institute at the University of Wisconsin-Madison. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

## References

- X. Hu, C. Zou, C. Zhang, Y. Li, Technological developments in batteries: a survey of principal roles, types, and management needs, *IEEE Power Energy Mag.* 15 (5) (2017) 20–31.
- A. Emadi, Advanced Electric Drive Vehicles, CRC Press, New York, 2015.
- E. Chemali, M. Preindl, P. Malysz, A. Emadi, Electrochemical and electrostatic energy storage and management systems for electric drive vehicles: state-of-the-art review and future trends, *IEEE Journal of Emerging and Selected Topics in Power Electronics* 4 (3) (2016) 1117–1134.
- World Energy Outlook, Special Report Energy and Air Pollution, International Energy Agency, 2016.
- P. Malysz, R. Gu, J. Ye, H. Yang, A. Emadi, State-of-charge and state-of-health estimation with state constraints and current sensor bias correction for electrified powertrain vehicle batteries, *IET Electr. Syst. Transp.* 6 (2) (2016) 136–144.
- L. McCurlie, M. Preindl, A. Emadi, Fast model predictive control for redistributive lithium-ion battery balancing, *IEEE Trans. Ind. Electron.* 64 (2) (2017) 1350–1357.
- W. Waag, C. Fleischer, D.U. Sauer, Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles, *J. Power Sources* 258 (2014) 321–339.
- R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, S. Habibi, Reduced-order electrochemical model parameters identification and SOC estimation for healthy and aged Li-ion batteries. Part I: parameterization model development for healthy batteries, *IEEE Journal of Emerging and Selected Topics in Power Electronics* 2 (3) (2014) 659–677.
- Z. Li, J. Huang, B.Y. Liaw, J. Zhang, On state-of-charge determination for lithium-ion batteries, *J. Power Sources* 348 (Supplement C) (2017) 281–301.
- M.A. Roscher, D.U. Sauer, Dynamic electric behavior and open-circuit-voltage modeling of LiFePO<sub>4</sub>-based lithium ion secondary batteries, *J. Power Sources* 196 (1) (2011) 331–336.
- J. Li, J.K. Barillas, C. Guenther, M.A. Danzer, A comparative study of state of charge estimation algorithms for lifepo<sub>4</sub> batteries used in electric vehicles, *J. Power Sources* 230 (2013) 244–250.
- L. Liu, L.Y. Wang, Z. Chen, C. Wang, F. Lin, H. Wang, Integrated system identification and state-of-charge estimation of battery systems, *IEEE Trans. Energy Convers.* 28 (1) (2013) 12–23.
- M. Gholizadeh, F. Salmasi, Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model, *IEEE Trans. Ind. Electron.* 61 (3) (2014) 1335–1344.
- X. Chen, W. Shen, Z. Cao, A. Kapoor, A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles, *J. Power Sources* 246 (2014) 667–678.
- C. Hu, B.D. Youn, J. Chung, A multiscale framework with extended kalman filter for lithium-ion battery {SOC} and capacity estimation, *Appl. Energy* 92 (2012) 694–704.
- F. Sun, X. Hu, Y. Zou, S. Li, Adaptive unscented kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles, *Inside Energy* 36 (5) (2011) 3531–3540.
- S. Sepasi, R. Ghorbani, B.Y. Liaw, Improved extended kalman filter for state of charge estimation of battery pack, *J. Power Sources* 255 (Supplement C) (2014) 368–376.
- W. Wang, D. Wang, X. Wang, T. Li, R. Ahmed, S. Habibi, A. Emadi, Comparison of kalman filter-based state of charge estimation strategies for li-ion batteries, *Transportation Electrification Conference and Expo (ITEC), 2016 IEEE, IEEE, 2016*, pp. 1–6.
- M. Charkhgard, M. Farrokhi, State-of-charge estimation for lithium-ion batteries using neural networks and ekf, *IEEE Transactions on Industrial Electronics* 57 (12) (2010) 4178–4187.
- J. Du, Z. Liu, Y. Wang, State of charge estimation for li-ion battery based on model from extreme learning machine, *Contr. Eng. Pract.* 26 (2014) 11–19.
- J. Meng, G. Luo, F. Gao, Lithium polymer battery state-of-charge estimation based on adaptive unscented kalman filter and support vector machine, *IEEE Trans. Power Electron.* 31 (3) (2016) 2226–2238.
- S. Tong, J.H. Lacap, J.W. Park, Battery state of charge estimation using a load-classifying neural network, *Journal of Energy Storage* 7 (Supplement C) (2016) 236–243.
- X. Hu, D. Cao, B. Egardt, Condition monitoring in advanced battery management systems: moving horizon estimation using a reduced electrochemical model, *IEEE ASME Trans. Mechatron.* 23 (1) (2018) 167–178.
- X. Hu, F. Sun, Fuzzy clustering based multi-model support vector regression state of charge estimator for lithium-ion battery of electric vehicle, *Intelligent Human-machine Systems and Cybernetics, 2009. IHMSC'09. International Conference on*, vol. 1, IEEE, 2009, pp. 392–396.
- X. Hu, S.E. Li, Y. Yang, Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles, *IEEE Transactions on Transportation electriification* 2 (2) (2016) 140–149.
- Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.* (2012) 1–9.
- K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, *Arxiv.Org* 7 (3) (2015) 171–180.
- L. Deng, Deep Learning: Methods and Applications, *Found. Trends® in Signal Process.* 7 (3–4) (2014) 197–387.
- D. CiresAn, U. Meier, J. Masci, J. Schmidhuber, Multi-column deep neural network for traffic sign classification, *Neural Network*. 32 (2012) 333–338.
- G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, *IEEE Signal Process. Mag.* 29 (6) (2012) 82–97.
- J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl, V. Svetnik, Deep neural nets as a method for quantitative structureactivity relationships, *J. Chem. Inf. Model.* 55 (2) (2015) 263–274.

- [33] D. Wang, A. Khosla, R. Gargeya, H. Irshad, A.H. Beck, Deep Learning for Identifying Metastatic Breast Cancer, ArXiv Preprint, (2016) 1–6arXiv:1591239.
- [34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale Machine Learning on Heterogeneous Systems, (2015) url <http://tensorflow.org>.
- [35] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, CoRR abs/1412.6980, <http://arxiv.org/abs/1412.6980>.
- [36] Phillip Kollmeyer, Panasonic 18650PF Li-ion Battery Data, Mendeley Data, v1, (2018) <https://doi.org/10.17632/wykht8y7tg.1>.
- [37] United States environmental protection agency, vehicle and fuel emissions testing, <https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules#vehicleDDS>, (2017).
- [38] Panasonic, Panasonic NCR18650PF Lithium-ion Battery Datasheet (June 2016), (2016).
- [39] Panasonic, Introduction of NCR18650PF, Panasonic, (2013).
- [40] Y.S. Lee, W.Y. Wang, T.Y. Kuo, Soft computing for battery state-of-charge (bsoc) estimation in battery string systems, IEEE Trans. Ind. Electron. 55 (1) (2008) 229–239.
- [41] W.-Y. Chang, Estimation of the state of charge for a lfp battery using a hybrid method that combines a rbf neural network, an ols algorithm and aga, Int. J. Electr. Power Energy Syst. 53 (2013) 603–611.