Proceedings of the Canadian Society for Mechanical Engineering International Congress 2020
CSME Congress 2020
June 21-24, 2020, Charlottetown, PE, Canada

A Recurrent Neural Networks Approach for Estimating the Core Temperature in Lithium-Ion Batteries

Olaoluwa Ojo[1], Xianke Lin[1]*, Haoxiang Lang[1], Youngki Kim[2]
[1]Department of Automotive, Mechanical and Manufacturing Engineering, Ontario Tech University, Oshawa, ON, Canada
[2]Department of Mechanical Engineering, University of Michigan Dearborn, Michigan, USA

Abstract— The safety and reliability of Lithium-ion batteries are increasingly critical, especially as more products on the market are powered by them. The core temperature of batteries is one of the important factors to consider when improving safety, longevity, and performance. To overcome the inability to practically obtain direct core temperature measurements, this paper proposes a neural network-based estimation method using a gated recurrent unit. This approach can estimate the core temperature to a high level of accuracy using commonly measured signals such as voltage, current, state of charge, ambient and surface temperatures. Experimental results demonstrate excellent estimation performances over cycling and between different batteries of the same type. The proposed method does not require a strenuous parameter tuning operation, model derivation and simplification, or a deep understanding of the electrochemical processes in the battery. It should be also highlighted that, compared to the other available options in literature, this technique has the advantage of easy implementation.

Keywords- core temperature estimation, lithium-ion, recurrent neural networks, GRU

I. INTRODUCTION

The modern awakening to the consequences of climate change is causing an increase in demand for sustainable products as seen with electric vehicles (EVs) and renewable energy sources. The most popular energy storage system for this application is the Lithium-ion (Li-ion) battery due to its relatively higher specific power density and energy density when compared to other chemistries [1, 2]. Li-ion batteries, however, have drawbacks that depend on temperature. Temperature outside the optimal operating range has been found to affect battery safety, longevity [3–7] and performance [8–10]. The effects of operating a battery outside these limits could develop into thermal runaway, reduction in capacity and cycle life, as well as internal impedance variation. This poses a problem with electric vehicles since they often require high discharge or charge currents that generate heat.

In order to prevent these consequences, an accurate observation of battery temperature is crucial. A popular method of measuring battery temperature is with surface-mounted temperature sensors such as thermocouples, thermistors, or resistance temperature detectors (RTDs). The issue, however, arises with the realization that the core temperature of a cell can differ from the surface temperature at high C-rates [11] which renders the surface temperature measurement imprecise to what it actually is in the cell. This awareness has piqued interests and attention in core temperature estimation and predictions. Some attempts involve the use of adaptive observers [2, 12–14] to estimate the core temperature based on data and models. Richardson *et al* proposed an electrothermal-based algorithm that associates electrochemical impedance spectroscopy (EIS) with surface temperature measurements for core temperature estimation [15]. Other literatures have also proposed predicting the temperature distribution with high fidelity models [16, 17]. Although these methods solve the problem of core temperature estimation, they can be quite involved for researchers in the field due to the intense parameter estimation requirement and complexity of such an approach, as well as the need for an in-depth understanding of electrochemistry.

This paper introduces an easy to implement neural network approach that removes this barrier to entry, thereby, allowing researchers and engineers alike to focus on improving safety and reliability in Li-ion batteries and the products they power. The proposed technique is a data-driven approach that uses a recurrent neural network (RNN) known as the gated recurrent unit (GRU). This neural network (NN) model receives commonly measured battery signals such as current, voltage, SOC, ambient and surface temperatures and delivers highly accurate estimations of the core temperature in real time. The use of the GRU for estimation does not involve any thermal models, their derivations or simplification, nor does it require an expertise in electrochemistry. Estimations are accomplished by training the NN to learn the non-linear relationship between the variables, and then testing the network on measurements from processes like constant current constant voltage (CC-CV) cycles or drive cycle current profiles. This NN approach is able to:

- Accurately learn the non-linear thermal relationship between the voltage, current, state of charge, surface temperature and the core temperature
- Maintain a high level of accuracy over cycling
- Extend this estimation experience to other batteries of the same form-factor and chemistry

To summarize the key contributions of this study, this approach enables its users to easily set up and estimate core temperature without an in-depth understanding of the intricacies of the battery nor the need for model derivation, model simplification or a strenuous parameter estimation process. It is a neural network model able to learn the non-linear relationship between a multivariate input and the core temperature. It has the potential of scale since only one model needs to be trained with data from one battery in a pack and beyond.

The remainder of this paper is organized as follow. First, we introduce the neural network model for estimating core temperature as well as the training process. Then, a detailed description of the experiment set up and data collection process is introduced. Next, several test cases are investigated, and the results and observations are discussed. Finally, we conclude the paper.

II. NEURAL NETWORK MODEL FOR CORE TEMPERATURE ESTIMATION

A. The Gated Recurrent Unit (GRU)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks that are used for time-varying or sequential data predictions. The main purpose of an RNN is to allow data to persist, thereby recognizing patterns from past data and using those patterns to make estimations or future predictions. An RNN is a subset of the supervised learning family because of its ability to learn the relationship between its inputs and the required output(s). A supervised neural network learns this relationship by minimizing a loss function with respect to the model weights. Therefore, as the model learns, the loss decreases towards a local or absolute minimum following a gradient and updating the weights. Although a vanilla RNN is able to learn these sequential relationships, it is unable to maintain dependencies or intuitions with data further in the past. This limitation is caused by two issues known as the exploding and vanishing gradient problems [18–20]. These problems occur when training with the backpropagation through time (BPTT) method. The exploding gradient drives the weights responsible for reflecting long-term dependencies to oscillate, while the vanishing gradient drives these same weights to a norm of zero (not substantially changing with each new epoch), making learning incredibly time-consuming or impossible. Gated recurrent units and long short-term memory neural networks are variations of the vanilla RNN because they solve the problem of the exploding and vanishing gradients. They are, hence, better suited for capturing long-term contexts. The similarity between both is their ability to remember features from further in the past without being affected by the gradient problems mentioned earlier. This found immunity is due to their memory feature and gates which allow them to easily control the flow of data from the past. The difference between both, however, is that the GRU trades off a better memory for faster training due to the reduced number of weights. Unlike language translation, for example, the superior memory of the LSTM over GRU is not as important for core temperature estimation since the context needed for its estimation need not extend as far into the past.

RNNs have seen rapid growth in research for predicting sequential data. They can be found in natural language processing, language translation, music compositions and many more. RNNs are also experiencing growth in battery research. Some applications are seen in the prediction of capacity fade [21, 22] and state of charge (SOC) estimation [23, 24]. These examples are proof that this method is indeed accepted with confidence in research.

The structure of the GRU neural network is shown in Figure 1. The GRU takes in a number of samples (n) that are reformatted from the dataset. These samples are formed by applying a sliding-window across the sequential data obtaining one new datapoint in every sample. Each sample ($X_k \in \mathbb{R}^{mx4}$, $where\ 1 \leq k \leq n$) is a matrix made up a number of timesteps (m) consisting of previous and present voltage ($\vec{V}$), current ($\vec{I}$), ambient temperature ($\vec{T_a}$) and the surface temperature ($\vec{T_s}$) inputs represented as column vectors by the arrows above the terms. These inputs along with the previous hidden state or previous output (h_{k-1}). are used to estimate the current hidden state (h_k) which is either sent to another GRU layer or a fully connected layer as the output. The memory of the GRU layer is kept in the hidden state and is propagated to subsequent timesteps.

In a GRU layer, there are 2 gates: the update gate and the reset gate. The gates are represented by the sigmoid function (σ). These functions squash elements in vectors to a range between 0 and 1 thereby allowing or preventing data from flowing further on. The update gate is responsible for updating entries in the hidden state, it chooses what part of the hidden state to replace with the new concatenated inputs. The reset gate is responsible

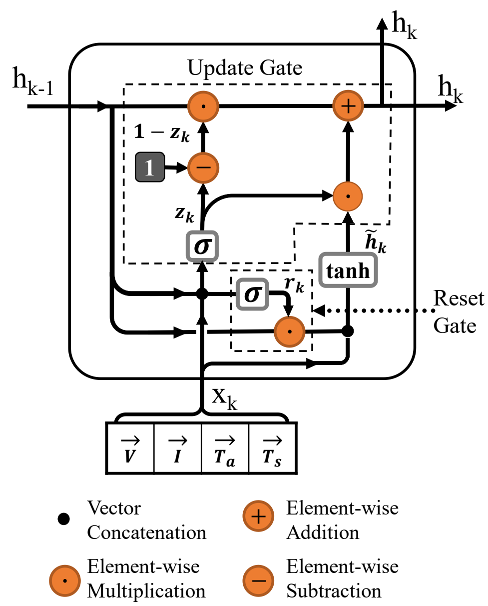

Figure 1. A GRU architecture unfolded in time

for resetting or forgetting irrelevant features in the network. The mechanics of a GRU cell can be described by the following set of equations:

$$
\begin{aligned}
r_k &= \sigma(U_r h_{k-1} + W_r x_k + b_r) \\
z_k &= \sigma(U_z h_{k-1} + W_z x_k + b_z) \\
\tilde{h}_k &= tanh(U_{\tilde{h}}(r_k \circ h_{k-1}) + W_{\tilde{h}} x_k + b_{\tilde{h}}) \\
h_k &= ((1 - z_k) h_{k-1} + z_k \tilde{h}_k)
\end{aligned} \tag{1}
$$

where r and z are the reset and update activations, while h and $\tilde{h}$ are the hidden state and its candidates for each sample k. W is the set of weights for input (x_k) and U is the set of weights for the previous hidden state (h_{k-1}), while the hyperparameter b is the bias applied to the results from each gate. The tanh function is similar to the sigmoid activation function except it squashes the vector element between -1 and +1. The neural network (NN) model used in this paper uses two GRU layers and a fully-connected layer that receives the hidden state of the last GRU layer and reduces it to one neuron which becomes the estimated core temperature (T_c) for that timestep. Two layers are used here to allow the NN to easily extract some abstract relationships among the inputs. The numbers of neurons for the two GRU layers are 256 and 128 respectively. Figure 2 illustrates the NN model layered next to each other.

III. DATA COLLECTION AND TEST CASES

A. Specifications and experiment Setup

Two 2.5Ah lithium iron phosphate ($LiFePO_4$ or LFP) battery cells from *A123* (Batt-A and Batt-B) were chosen for this experiment. The batteries were put through both driving cycle current profiles and constant charge/discharge cycles. Data acquisition was carried out on an in-house test system shown in Figure 3(A). The test system recorded voltage, current, state of charge, ambient temperature, surface temperature, and core temperature data. Temperature measurement was accomplished by attaching and inserting two T-type thermocouples onto the surface and into a perforation created at the centre of the positive terminal of the battery. The test equipment consisted of one programmable power supply capable of 80V and 60A, an electronic DC load capable of sinking up to 40A, voltage, current, and temperature sensing units, a relay module for switching between the power supply and the electronic load. A computer is also included for control.

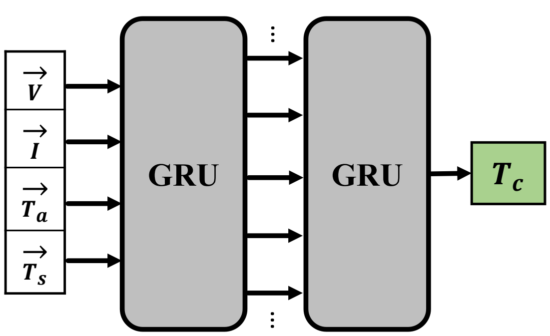

Figure 2. The neural network (NN) model

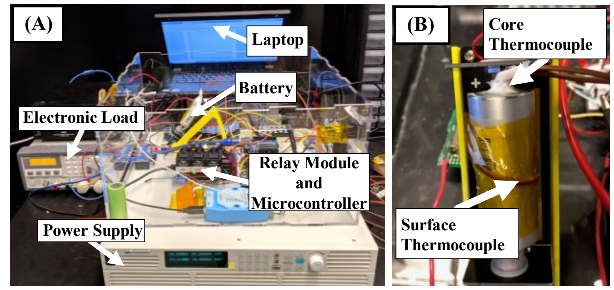

Figure 3. (A) Battery test system; (B) thermocouple placements

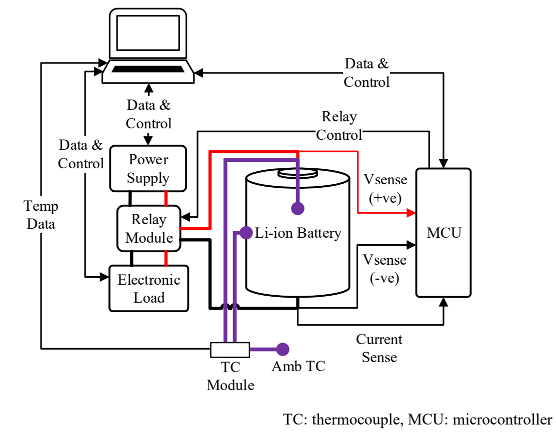

Figure 4. Connection Diagram of the Battery Test System.

An illustration of the connection of the experimental set up is presented in Figure 4.

B. Data collection process

Data collection for training and validation is comprised of multiple cycles of constant charge and discharge while data for testing was obtained from current profiles based on popular driving cycles. The training data collection process starts with charging the battery to maximum capacity, and waiting until the core and surface temperatures converge ($T_s \approx T_c$). A discharge-charge cycle is then applied to bring the SOC from 100% to a target SOC and back to 100%. After each charge and discharge step, the battery is allowed to cool down until $T_s \approx T_c$. The target SOCs differ in each cycle - they range from 10% to 90% in 10% increments. This charge-discharge-cooldown cycle for all target SOCs is regarded here as a set. Six sets are carried out for a complete dataset. Each set is run at a different C-rate ranging from 1C – 6C.

The testing data is comprised of data collected from running 5 current profiles derived from the EUDC, HWFET, LA92, UDDS, and US06 driving cycles. All profiles applied to the battery are applied starting at 80% SOC and within 1C and 6C. The data collection procedure for the testing data is similar to the procedure for training and validation data collection. The battery is initially charged to full, then discharged to 80% SOC, the system waits to allow the core and surface temperature values to converge then begins running the current profiles. Each profile run is also followed by a cooldown period, and then a recharge to 80% SOC. All five profiles begin at 80% SOC because EV

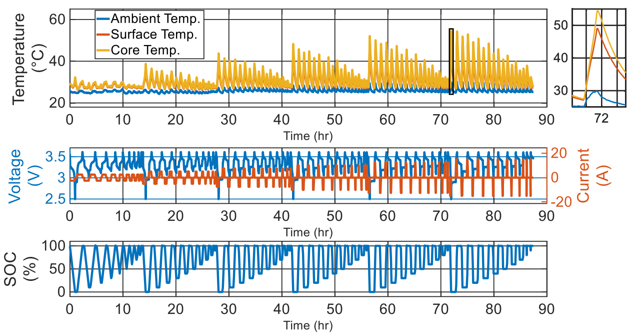

Figure 5. Batt-A train dataset

manufacturers typically set a maximum of 80-90% SOC to prolong battery life.

C. Training Process

The GRU neural network was trained on two *Nvidia Tesla P100* GPUs supplied through a cluster on the *Compute Canada (CC)* system. The neural network was modelled on a python-based neural networks library built on *Tensorflow* called *Keras* for training and inference. A mean square error loss function ($MSE = \frac{1}{z}\sum_{i=1}^{z}(\widehat{Y_i} - Y_i)^2$) was used in conjunction with an optimizer known as *RMSprop* that works on the following principle:

$$E[\nabla^2]_k = \gamma E[\nabla^2]_{k-1} + 0.1\nabla^2{}_k$$
$$\theta_{k+1} = \theta_k - \frac{\eta}{\sqrt{E[\nabla^2]_k + \epsilon}}\nabla_k \tag{2}$$

where γ is the momentum term (recommended value of 0.9), η is the learning rate (0.0001 in this study), ∇_k is the gradient of the loss function (*MSE*) with respect to the weights at time step k, ϵ is a smoothing term that prevents division by zero, $E[\nabla^2]_k$ is called the moving average of the squared gradient and finally, θ is the set of hyperparameters including weights and biases. The optimizer is what performs the training by minimizing the loss.

As previously mentioned, each dataset imported into the NN is sampled with a sliding window, 60 timesteps deep, where the last timestep is the current timestep to be estimated. In order to conserve memory, 128 batches of these samples were fed in each time to the NN for training. The training dataset was split into an 80%-20% grouping for the training and validation data respectively. During inference (testing), however, a single batch was fed in to induce real-time estimation. The neural network was trained for 200 epochs during which a contingency for overfitting was implemented simultaneously. In order to prevent overfitting, the weights of the epoch with least validation loss were saved and used.

D. Test Cases

1) *Core temperature estimation performance*: Since the goal of this study is to accurately estimate the core temperature, the first test case evaluates the performance of the network. The neural network's performance is quantified by the maximum absolute error (MAE) and the maximum error (MAX). After validating the performance on the test dataset collected from Batt-A (dataset-A1), a number of other test cases are considered to further prove the capability of this approach.
2) *Estimation performance over cycling*: The test data and result referred to earlier for Batt-A is used here as a benchmark before cycling. Therefore, after collecting the first iteration of the test dataset for Batt-A (dataset-A1) and verifying it on the NN model, the result is compared to the result after cycling. After dataset-A1, Batt-A is cycled 100 times at 10 A and then subjected to a second iteration of data collection (dataset-A2). The ability of the NN model to accurately estimate core temperature is then verified on dataset-A2. In each iteration, a MAE of less than 0.1ºC is expected.
3) *Estimation performance on a second cell of the same type*: In this test case, a second battery (Batt-B) of the same type and from the same manufacturer is tested. This battery is also put through the same driving cycle current profiles and the resulting test dataset is used on the NN model trained on Batt-A. The purpose of this is to examine the performance of a model trained once and used on other cells of the same type in a battery pack.

IV. Results and Discussion

A. Core temperature estimation performance

The neural network model based on the gated recurrent unit (GRU) is trained on the training dataset from Batt-A. Training and testing the network for 200 epochs lasted for less than 5 hours. The chosen number of epochs, as with many other NN parameters, was an iterative process to ensure the best training and validation losses without overfitting too long or requiring a longer training duration. As mentioned earlier, the trained model weights with the least validation loss were chosen in order to enable the model to generalize well on unseen data. Immediately after training the NN model, the model is used for testing datasets one sample at a time (batch size = 1). The first iteration of the test dataset from Batt-A (dataset-A1) is fed through the NN model and the result is shown in Figure 6. In the figure, the residual or absolute error is plotted first, while the estimated and measured core temperatures are plotted below. In this illustration, the NN model is able to produce a MAE of 0.066ºC and MAX of 0.275ºC. The MAE attained here outperforms the expected result of 0.1ºC which signifies that the neural network was able to learn the relationship between the inputs and the core temperature while maintaining its ability to generalize on unseen data. The reason for the strong estimation performance is due to the presence of the surface temperature. The model is able to primarily take the current, SOC and surface temperature and infer the core temperature from them. This is what is meant when the model is said to have learned the thermal relationship. Without the surface temperature, the NN model will be unable to learn this thermal relationship.

B. Estimation performance after cycling

After training the neural network model on the training dataset collected from Batt-A and testing the model on dataset-A1, Batt-A was put through 100 cycles of charge-discharge cycles. After cycling, another test dataset was gathered (dataset-

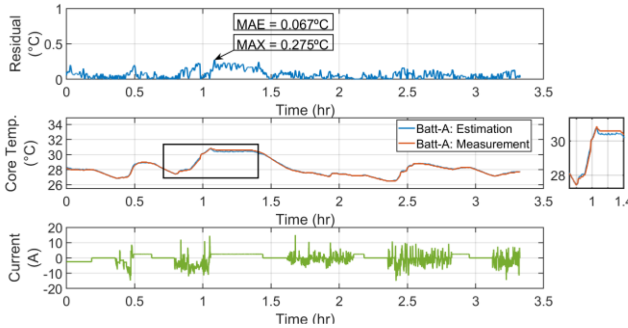

Figure 6. Core estimation result trained and tested on Batt-A

A2) and validated on the NN model. The resulting plot is shown in Figure 7. The performance of the NN model on dataset-A2 is very close to that on dataset-A1. The MAE and MAX values of 0.074ºC and 0.365ºC respectively are only a fraction less accurate than those seen in the first test case despite 100 cycles in-between them. Nevertheless, the result is within the expected maximum absolute error (MAE) of 0.1ºC. The experimental results demonstrate the performance of the NN model over cycling.

A way to potentially improve the capability of the model to accurately estimate the core temperature irrespective of the cycle life spent is to include the dataset from cycling during training, in other words, using both the regular training data and the data from cycling 100 or more times to train the model. This will enable the model to learn the long-term relationship between the change in voltage and core temperature. Therefore, as the impedance of the battery increases with longer cycles, and the voltage variation changes, the model is able to adapt and accurately estimate core temperature.

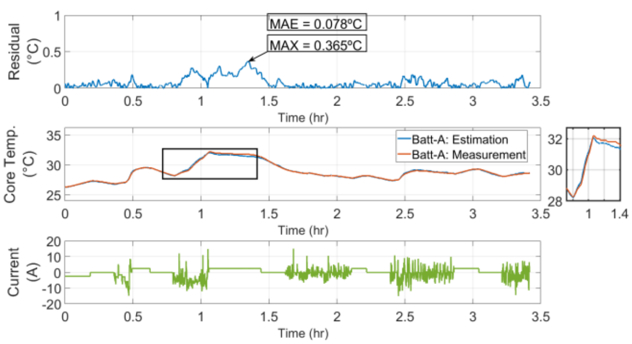

Figure 7. Batt-A core estimation result after 100 cycles

C. *Estimation performance on a second battery of the same type*

The purpose of this subsection is to test the capability of the model on other batteries of the same type. Successfully accomplishing this means that data from only one battery is needed to train the model in practice. To test this hypothesis, data collected from Batt-B is put through the NN model trained earlier (on data from Batt-A). The result is shown in Figure 8. It is evident from the plot that the NN model is still able to accurately estimate the core temperature of another battery achieving a MAE of 0.063ºC and MAX of 0.297ºC. This outcome confirms the ability of the NN model to generalize, thereby performing well on different datasets. The good performance is, however, limited to datasets from batteries of the same type. The reason for this is the similar elements batteries of the same type share such as dimensions, materials, assembly, and manufacturing process. These physical attributes add to the thermal properties of the batteries such as heat capacity, therefore, enabling them to consistently produce similar temperature curves given the same input.

In light of this, the NN approach has the potential to be scaled to battery modules and packs provided they are of the same type and supply the required data needed for inference. To implement on other battery chemistries or formfactors, it would be more appropriate to train the network on data from the new battery type.

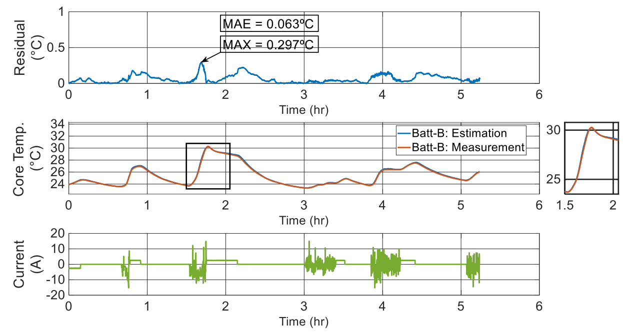

Figure 8. Estimation result trained on Batt-A and tested on Batt-B

V. CONCLUSION

The temperature of lithium-ion batteries is a critical factor affecting the safety and reliability in battery-powered devices. Since the internal or core temperature of a lithium-ion battery is often higher than that of the surface, the accurate estimation of core temperature is essential in battery management. This paper proposes a solution that uses a neural network model based on the gated recurrent unit to make precise estimations of the core temperature. This data-driven approach has the advantage of easy implementation because it does not involve any model derivation, reduction, or a strenuous parameter tuning operation, as demonstrated by experimental results. The neural network model showed the ability to learn the relationship between the surface and core temperature as well as the impact of the voltage, current, SOC and ambient temperature on its estimations. The neural network model was also found to perform well over cycling as well as on other batteries of the same type and build. This simple but effective structure highlights the possible extension of the proposed model to battery modules (and even packs). The results from these test cases are proof of the capability and viability of this method in research and production.

VI. FUTURE WORK

Although the neural network has a strong ability to estimate the core temperature of a lithium-ion cell, there are areas of improvement that will be explored in future work. One of which is the increased robustness to cycling over the entire life of a battery. One possible solution mentioned in the paper is to train the model on both the data allocated for training and the data collected during cycling. Training the model with these two will possibly help the model learn the relationship between voltage and core temperature better. Another possibility is to train and save multiple models based on the number of cycles exhausted by the battery. Hence, as the battery operated in practice is cycled, a model that has been trained for its specific cycle range is used. Another area of improvement is to test the NN model in different harsh environments such as hot and frigid conditions as well as a scaled-up test with multiple batteries in a battery pack.

REFERENCES

[1] X. Lin *et al.,* "Parameterization and observability analysis of scalable battery clusters for onboard thermal management," *Oil & Gas Science and Technology*, 2012.

[2] Y. Kim, S. Mohan, J. B. Siegel, A. G. Stefanopoulou, and Y. Ding, "The estimation of radial temperature distribution in cylindrical battery cells under unknown cooling conditions," in *52nd IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 5680–5685.

[3] D. Aurbach, "A short review of failure mechanisms of lithium metal and lithiated graphite anodes in liquid electrolyte solutions," *Solid State Ionics*, vol. 148, no. 3-4, pp. 405–416, 2002.

[4] X. Feng *et al.,* "Thermal runaway mechanism of lithium ion battery for electric vehicles: A review," *Energy Storage Materials*, vol. 10, pp. 246–267, 2018.

[5] X. Guodong, C. Lei, and B. Guanglong, "A review on battery thermal management in electric vehicle application," *Journal of Power Sources*, vol. 367, pp. 90–105, 2017.

[6] J. Shim, R. Kostecki, T. Richardson, X. Song, and K.A. Striebel, "Electrochemical analysis for cycle performance and capacity fading of a lithium-ion battery cycled at elevated temperature," *Journal of Power Sources*, vol. 112, no. 1, pp. 222–230, 2002.

[7] S.S. Zhang, K. Xu, and T.R. Jow, "Low temperature performance of graphite electrode in Li-ion cells," *Electrochimica Acta*, vol. 48, no. 3, pp. 241–246, 2002.

[8] X. Hu, S. E. Li, and Y. Yang, "Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles," *IEEE Trans. Transp. Electrific.*, vol. 2, no. 2, pp. 140–149, 2016.

[9] X. Hu, H. Yuan, C. Zou, Z. Li, and L. Zhang, "Co-estimation of state of charge and state of health for lithium-ion batteries based on fractional-order calculus," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10319–10329, 2018.

[10] Q.-K. Wang, Y.-J. He, J.-N. Shen, X.-S. Hu, and Z.-F. Ma, "State of charge-dependent polynomial equivalent circuit modeling for electrochemical impedance spectroscopy of lithium-ion batteries," *IEEE Trans. Power Electron.*, vol. 33, no. 10, pp. 8449–8460, 2018.

[11] C. Forgez, D. Vinh Do, G. Friedrich, M. Morcrette, and C. Delacourt, "Thermal modeling of a cylindrical LiFePO4/graphite lithium-ion battery," *Journal of Power Sources*, vol. 195, no. 9, pp. 2961–2968, 2010.

[12] M. Debert, G. Colin, G. Bloch, and Y. Chamaillard, "An observer looks at the cell temperature in automotive battery packs," *Control Engineering Practice*, vol. 21, no. 8, pp. 1035–1042, 2013.

[13] X. Lin *et al.,* "Online parameterization of lumped thermal dynamics in cylindrical lithium ion batteries for core temperature estimation and health monitoring," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 5, pp. 1745–1755, 2013.

[14] J. Sun *et al.,* "Online internal temperature estimation for lithium-ion batteries based on Kalman filter," *Energies*, vol. 8, no. 5, pp. 4400–4415, 2015.

[15] R. R. Richardson, P. T. Ireland, and D. A. Howey, "Battery internal temperature estimation by combined impedance and surface temperature measurement," *Journal of Power Sources*, vol. 265, pp. 254–261, 2014.

[16] P. M. Gomadam, R. E. White, and J. W. Weidner, "Modeling heat conduction in spiral geometries," *J. Electrochem. Soc.*, vol. 150, no. 10, A1339, 2003.

[17] G.-H. Kim, A. Pesaran, and R. Spotnitz, "A three-dimensional thermal abuse model for lithium-ion cells," *Journal of Power Sources*, vol. 170, no. 2, pp. 476–489, 2007.

[18] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," (eng), *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[19] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *Int. J. Unc. Fuzz. Knowl. Based Syst.*, vol. 06, no. 02, pp. 107–116, 1998.

[20] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," Feb. 2013. [Online] Available: http://arxiv.org/pdf/1211.5063v2.

[21] Y. Zhang, R. Xiong, H. He, and Z. Liu, "A LSTM-RNN method for the lithuim-ion battery remaining useful life prediction," in *2017 Prognostics and System Health Management Conference (PHM-Harbin): Conference proceedings 21-23 October 2015, Vision Hotel, Beijing, China*, Harbin, China, 2017, pp. 1–4.

[22] M. Assefi, A. Hooshmand, H. Hosseini, and R. Sharma, "Battery degradation temporal modeling using lstm networks," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, Dec. 2018 - Dec. 2018, pp. 853–858.

[23] A. Y. Cheng, Y. Wang, Z. L. Cheng, X. S. Cheng, and F. Dei, "State of charge estimation for batteries using recurrent neural networks," in *2018 Chinese Automation Congress (CAC)*, Xi'an, China, 2018, pp. 390–395.

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