Chapter 16
State of Energy Estimation of Li-Ion Batteries Using Deep Neural Network and Support Vector Regression

Pradeep Kumar, Yasser Rafat, Paolo Cicconi, and Mohammad Saad Alam

Abstract Efficient management of the power and energy output of a high voltage battery pack requires a precise estimation of the State of Energy (SOE). For the accurate estimation of SOE, this work presents two data-driven methods as Deep Neural Network (DNN) and a regression model, i.e. Support Vector Regression (SVR). The effectiveness of the SOE estimation was compared, analysed, and studied through these models under similar conditions. For performance enhancement of estimation, a modified algorithm based on the grid search of optimized hyperparameters was proposed and evaluated in both the models. For training of the model at subsequent thermal ranges, two case studies were performed using US06, UDDS, LA92, and HWFET drive cycles and at four different temperature levels (−10, 0, 10, and 25 °C), for each cycle. The results indicate that the DNN method has provided enhanced performance for State of Energy Estimation as compared to the regression models of ML, i.e. SVR. This work highlights the prevailing challenges in the industry and proposes the potential recommendation for Battery Management System (BMS) development and SOE estimation in next-generation EV applications.

Keywords The State of Energy · Deep neural network · Support vector regression · Hyperparameters optimization · Regression

P. Kumar (✉)
Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada
e-mail: kumar84@uwindsor.ca

Y. Rafat
Sustainable Energy and Acoustics Research Laboratory, SEARL, Mechanical Engineering Department, Aligarh Muslim University, Aligarh 202002, India

P. Cicconi
Università Degli Studi Roma Tre, Rome, Italy

Y. Rafat · M. S. Alam
Centre of Advanced Research in Electrified Transportation, CARET, Aligarh Muslim University, Aligarh, India

I. Ali et al. (eds.), *Computational Modelling in Industry 4.0*,
https://doi.org/10.1007/978-981-16-7723-6_16

1 Introduction

With the growing problems such as the depletion of energy resources and the problem of global warming caused by use of internal conventional engines-based vehicles, electrical vehicles (EVs) have attracted the very high attention of people (Tie and Tan 2013; Xia et al. 2017). In recent years, a lot of development has taken place in the electrochemical energy storage system. Different energy storage systems have been proposed for the use in EVs such as Nickel/Metal Hydride (NiMH) battery, Lithium ion (Li-ion) battery, Fuel cells, ultra-capacitors, etc. (Iclodean et al. 2017). With the advancement and improvement in the properties of Li-ion batteries such as energy density, low self-discharge rate, long cycle life, and safety performance, they become widely applicable in EVs, electronics, mobile devices, etc. (Lu et al. 2013). A Li-ion battery module consists of an array of Li-ion cells. Temperature management is one of the issues of the Li-ion batteries which affect performance and safety (Gandoman et al. 2019; Kumar et al. 2020). The literature shows several solutions to improve the cooling effects by optimized battery layout (Qian et al. 2019), Heat Pipe (Lu et al. 2020), Phase Change Materials (Jaguemont et al. 2018), modular and mixed solutions (Cicconi et al. 2020). This paper aims to estimate the level of State of Energy (SOE) considering the achieved cell temperature with the operation parameters, i.e. voltage and current. The estimation of the cell state parameters such as State of Charge (SOC), Open Circuit Voltage (OCV), State of Energy (SOE), etc., is very important for the accurate functioning of the system and for the long life of batteries. These state parameters are also important in the estimation of driving range of electric vehicles which is also one of the complex issues as studied by Ronan German et al. (2020). However, the correct estimation of these state parameters is quite complex and difficult due to the non-linear behaviour of the electrochemical processes in Li-ion cells (Hafsaoui and Sellier 2010).

Traditionally, SOC estimation is performed for the identification of residual energy and the protection of batteries from being overcharged and discharged. From the recent literature review, it is well shown that different researchers had used a wide variety of techniques for the accurate estimation of SOC such as current integral method (Ng et al. 2009), proportional-integral (PI) method (Xu et al. 2014), electrical model-based method (Plett 2004b, c; He et al. 2013; Zhong et al. 2014), Sliding-mode observer based (Kim 2006, 2010), Kalman filter-based algorithms (Plett 2004a, b, c; Xu et al. 2012; Xiong et al. 2014), and neural network model (Kang et al. 2014) methods. With the increasing demand of Li-ion batteries for different applications, there is increased demand of use of Battery Management System (BMS) as well. Therefore, it is more and more important to accurately analyse the battery states for the proper functioning of batteries. But Liu et al. (2014) reviewed several disadvantages in the use of SOC for the estimation of residual energy. Because SOC is defined as the ratio of available capacity to the maximum stored charge in the battery, i.e. nominal capacity, from which it can be seen that there is no representation of the State of Energy. SOC clearly gives the information only about the residual

capacity, not about the energy. That's why there is the need to estimate SOE independently. SOE is defined as the available energy to the maximum stored energy in Li-ion battery. Some researchers (Shen 2007; Hausmann and Depcik 2013; Waag and Sauer 2013; Zheng et al. 2013) used residual available capacity instead of SOC for the estimation of SOE. Waag and Sauer (2013) used the estimation of battery electromotive force for the identification of the battery capacity and SOC. While Shen et al. (2007) defined state of available capacity of battery, instead of SOC for the estimation of battery residual capacity. These are some of the few works which used the capacity instead of SOC in the study. It is also important to study the effect of discharge current and temperature, since at the same SOC; SOE can be different because discharge efficiency is dependent on discharge current and temperature (Liu et al. 2014). Few researchers worked in this direction to understand the working of Li-ion batteries, as Wang et al. (2014a, b) show the study on the estimation of electronic conductivity of LiFePO4 cells at different temperatures to understand the low-temperature electrochemical performance at carbon-coated and uncoated cathodes. Yi et al. (2013) developed a model for the study of behaviour of Li-ion battery of temperature dependence on discharge in low ambient temperature. These studies prove that for the accurate estimation of SOE, it is necessary to consider the effect of discharge current and temperature.

In recent years, different researchers developed different techniques for the estimation of SOC and also proved well with the results. But SOC is different from SOE because SOE is the product of the residual battery capacity per OCV. The trend of variation of SOE is different from SOC. Similar to the estimation of SOC, few researchers developed systematic methods for the SOE measurement (Mamadou et al. 2012, 2019; Liu et al. 2014). In these studies, direct evaluation techniques were shown for the residual energy of battery with the consideration of battery discharge states. Some other researchers such as Stockar et al. (2011) and Kermani et al. (2011) presented the method of power integral for the estimation of SOE. But these studies were not found very accurate for the SOE measurement due to the measurement noises in the current and voltages of battery. Wang et al. (2014a, b) proposed the joint estimation technique of SOC and SOE to minimize the negative effect of power integral method. But in this method SOE estimation accuracy depends on the SOC estimation accuracy. To overcome this issue, Zhang et al. (2015) proposed the method of model-based joint estimation of SOC and SOE.

From the literature reviews, it can be seen that there is still a lot of possibility of research for the estimation of SOE. Since in this area, research has not been done as far as compared to the SOC estimation. Also, it was found out that SOE estimation is somewhat hard to estimate using mathematical relations and equations. So, there is a need to develop a method which is less complex and computationally efficient to be employed in real BMS condition. Therefore, this paper presents the two data-driven approaches for the estimation of SOE such as a regression model, i.e. SVR and DNN. A modified DNN-based method is shown for the accurate estimation of SOE. The modified method consists of the two simultaneous processes, i.e. first the optimization of hyperparameters and then the DNN model development which will be used for the prediction. In the neural networks, the critical task is the optimization

of hyperparameters which is accurately done using a grid-based approach. In the proposed grid-based approach, grids were developed based on the Root Mean Square Error (RMSE), training time, and hyperparameters such as the number of neurons and number of layers. Optimized values of the number of neurons and number of hidden layers were chosen when minimum RMSE with suitable training time was recorded. The same process was used in the development of regression model using SVR for the estimation of SOE. Grid-based approach was used in the modified algorithm of SVR model for the searching of optimized hyperparameters such as regularization parameter (C) and gamma (γ).

The main aim of this study is to develop an efficient model for the estimation of SOE and to overcome the shortcomings of regression model. For the implementation of such task, in this study two data-driven methods were used for the development of SOE estimation model, DNN and SVR. The results obtained from the DNN model supports that it is more efficient in the estimation as compared to the SVR. The results shown in Sect. 4 clearly state that the DNN model is fully efficient in overcoming the shortcoming of regression model. DNN model shows significant results in the prediction of SOE at different drive cycles and at different temperatures as well.

2 Proposed Methodology for SOE Estimation

The SOE of battery can be defined by Eq. (1) as the ratio of residual energy to the maximum available energy, where residual energy is denoted by $E_{res,k}$, maximum energy by E_M, battery energy efficiency as η_e, battery terminal voltage and current at kth time instant as $V_{t,k} and I_k$, T_s is the time sampling period. The methodological approach for the estimation of SOE is described in Fig. 1. The approach is focused on the accurate estimation of SOE followed by the selection of optimized hyperparameters for the respective estimation learning models. As shown in Fig. 1, raw data was collected from the publicly available dataset (Kollmeyer et al. 2020). The data for SOE was processed from this dataset using the Eq. (1). Data cleansing was performed in the next step such as resampling of the dataset to 1 Hz frequency and the removal of errors in data (such as Voltage, Current spikes). After the data cleansing, the further dataset was developed according to the respective case studies of this research. In this study, two estimation learning model was selected, i.e. DNN and SVR. Optimization of hyperparameters was the main step after the choosing of the training dataset, validation dataset, and the selection of model. Hyperparameters are responsible for estimation accuracy, computation time, and computational speed. The processes of estimation in both models are shown in detail in Sect. 2.1 and 2.2. The performance evaluation was done on the basis of RMSE and MAE. The evaluation accuracy is discussed in the result section, i.e. Sect. 4.

$$SOE_{k+1} = \begin{cases} SOE_k - \frac{E_{res,k}}{E_M} \\ SOE_k - \frac{\eta_e V_{t,k} I_k T_s}{E_M} \end{cases} \tag{1}$$

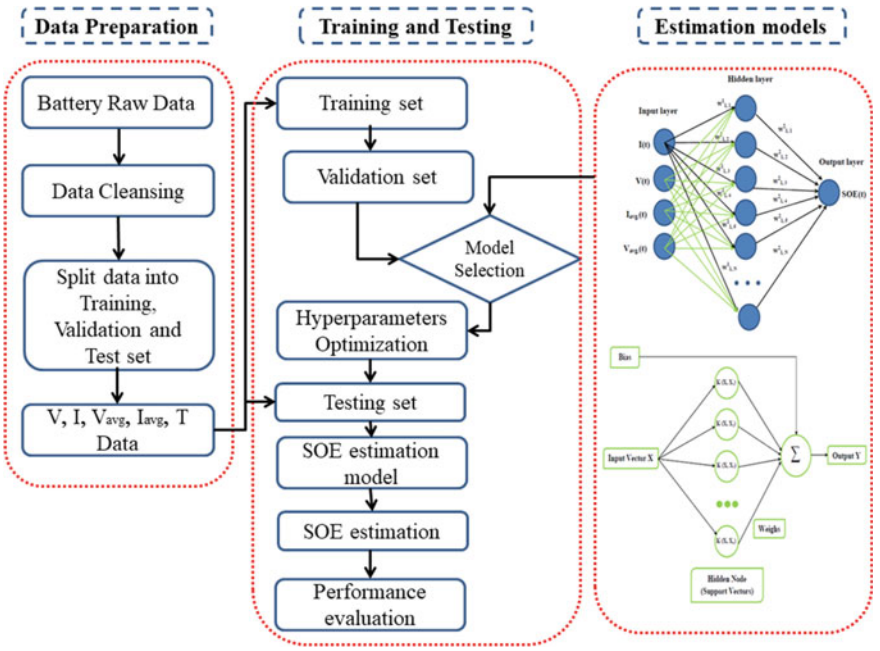


Fig. 1 The proposed methodological approach for SOE estimation

2.1 DNN Architecture

DNN is part of the family of machine learning methods based on artificial neural networks (ANN). In the present time, DNN has wide applications, there are many examples available where DNN architecture shows good response over the conventional algorithms such as DNN showed the level of accuracy more the human level such as in 2015 Microsoft Research's deep neural algorithm won the ImageNet challenge with the error of about 3.57% (He et al. 2016).

DNN is considered fast and efficient in comparison to the conventional methods for the estimation of SOE, since in conventional method expertise is required in battery chemistry and to mathematical model the battery behaviours. On the other hand, DNN doesn't involve such complexity for the SOE estimation. DNN consists of three layers, i.e. input layer, hidden layer, and output layer. The input and output layer remains fixed while the hidden layer can be varied to find out the optimum value where minimum error can obtain. Along with the layers, nodes or it can be said as neurons are also present in the respective layers to capture the non-linearity between the system input and output. The optimum value of nodes is also required to be found with the hidden layers for the development of the efficient and computational fast model.

DNN learns the battery functioning by mapping the input parameters with the output parameters. In this study for the creation and validation of DNN model a

publicly available dataset was used (Kollmeyer et al. 2020). The dataset consists of battery parameters of LG 18650HG2 Li-ion battery at four different drive cycles such as US06, UDDS, HWFET, LA92, and all the data of drive cycles are available at four different temperature set, i.e. 10, 0, 10, and 25 °C. Battery parameters available in the dataset are current, voltage, temperature, and SOE with time step of 1 s. For the development of the DNN model, vector of battery parameters was provided to the input layer where parameters were current, voltage, temperature, average current, and average voltage, while the output layer was having the single parameter, i.e. SOE. The model was created in such a way that it can effectively map the variation in input parameters with the variation in output, i.e. SOE and by using this learning method, it will be able to predict the SOE for any new input parameters value. Mathematically the input vector can be represented as—$X(t) = [I(t), V(t), T(t), I_{avg}(t), V_{avg}(t)]$ and the output can be represented as—$Y(t) = SOE(t)$, where I(t), V(t), T(t), $I_{avg}(t)$, $V_{avg}(t)$, and SOE(t) denote the current, voltage, temperature, average current, average voltage, and estimated SOE at time step t. The input vector X(t) fed to the input layer to map with the SOE(t) values at the output layer. Figure 2 illustrates the architecture of deep neural network with input, output, and hidden layers along with the neurons in each layer. It is showing the process of mapping of observables with the desired output. The computational speed of offline training of DNN is fast because to capitulate the output Y(t), input vector X(t) perform certain matrix multiplication whereas other strategies are not so computational fast since it involves partial differential equations. The hidden layer activation is represented by Eq. (2). Since DNN is matrix-based,

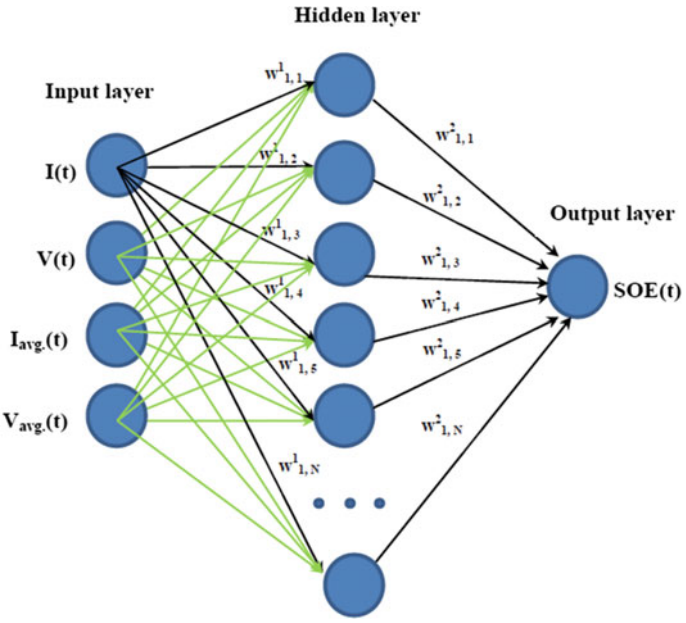


Fig. 2 Architecture of 2-layer DNN with representation of input, hidden and output layer

there are certain variables which need to be defined such as $w^{L}_{j,k}$ (which denotes the weights connection between neuron j in layer L-1 and neuron k in layer l), b^{L}_{k} (which denotes the bias), and h^{L}_{k} (which denotes the activation, respectively, of neuron k in layer L).

$$h_k^L = \sigma\left(\sum_k \left(w_{j,k}^L h_k^{L-1}(t) + b_k^L\right)\right) \tag{2}$$

Reading Eq. (2), σ is the activation function. Due to simplicity in training and testing, the non-linearity or activation function used for hidden layer is Rectified Linear Unit (ReLU) which can be given by Eq. (3).

$$\sigma(\mathrm{x}) = \begin{cases} 0 for x < 0 \\ x for x \geq 0 \end{cases} \tag{3}$$

Similarly, at the output layer, SOE can be calculated using Eq. (4).

$$SOE(t) = \eta\left(\sum_k \left(w_{j,k}^L h_k^{L-1}(t) + b_k^L\right)\right) \tag{4}$$

where L is the last hidden layer of network and η is the activation function for output layer which was chosen as Sigmoid function and can be represented by Eq. (5).

$$\eta(\mathrm{x}) = \frac{1}{(1 + \exp(-\mathrm{x}))} \tag{5}$$

2.1.1 DNN Hyperparameters and Its Optimization

Effective modelling of DNN requires selection of optimum values of hyperparameters because the computational speed and performance of model depends upon these hyperparameters. There are number of hyperparameters available such as hidden layers, neurons, learning rate, optimization algorithm, activation functions, etc. But the optimization of all the hyperparameters at the same time will not be a time-effective approach; therefore the most important hyperparameters were optimized in this study which was number of hidden layers and number of neurons in hidden layer. Rest of the hyperparameters were selected based on the literature survey such as learning rate was kept 0.001, optimization algorithm was chosen as Adam due to its good performance, ReLU activation function was used for hidden layers and Sigmoid function for output layer. Optimization algorithm developed in this study for the hidden layers and neurons selection is shown in Sect. 2.1.2. Grid-based approach was used for developing the optimization algorithm and US06 drive cycle at 25

°C was used for the validation in this optimization technique. Range of values was chosen both for hidden layers and neurons, to search the combination where root mean square error (RMSE) minima will occur. The range of values checked for hidden layers was from 2 to 8, while the neurons set of values were (2, 4, 8, 16, and 32). Table 1 shows the values of RMSE and training time for the different combinations of hidden layer and neurons. From Table 1, it can be seen that the minimum RMSE occur at the combination of 3 hidden layers and 16 neurons with training time as 775.7467 s. Therefore, keeping in mind about the computational efficient concept, combination of 3 hidden layers and 16 neurons in each hidden layer was chosen for the further study. Table 2 shows the list of all the optimized hyperparameters used in this study.

Table 1 Grids showing the values of RMSE and training time for different combinations of hidden layers and neurons

Layers Neurons	2	3	4	5	6	7	8
Root mean Square Error (RMSE)							
2	0.173742	0.161347	0.286389	0.143908	0.145942	0.286164	0.287129
4	0.098428	0.110402	0.285939	0.027262	0.08073	0.170202	0.051682
8	0.089132	0.038056	0.030829	0.022372	0.017867	0.016731	0.015806
16	0.083765	**0.014997**	0.019165	0.017265	0.018367	0.016447	0.015747
32	0.040811	0.036811	0.031811	0.028821	0.021734	0.0199	0.017876
Training Time (seconds)							
2	640.9791	694.9241	719.4053	841.239	895.1712	939.1342	951.2096
4	568.7269	751.9329	582.862	770.4256	831.8109	864.1268	831.5526
8	582.5267	774.7212	592.3911	769.8025	893.7469	890.4075	913.9425
16	586.9675	**775.7467**	583.1441	657.4654	815.7815	871.2201	855.9458
32	590.2175	800.4176	608.9079	686.7024	976.5569	1007.908	852.5247

Table 2 Optimized values selected for the hyperparameters of DNN

Hyperparameters	Optimized values
Number of hidden layers	3
Number of neurons	16
Optimization algorithm	Adam
Activation function for hidden layers	ReLU
Activation function for output layer	Linear
Learning rate	0.001

2.1.2 DNN Training Strategy

This section gives the information regarding the strategy used for the training of DNN, with the insight about incorporation of optimization technique used for hidden layer and neurons along with the training. In this study, before starting the training of the model optimized value for number of hidden layers and neurons were found and the selection procedure is shown in Sect. 2.1.1. Appendix A.1 shows the modified algorithm used in this study for the development of effective DNN model. Algorithm starts with the importing of dataset used for the training. Dataset used in this study were at the different drive cycles such as US06, UDDS, HWFET and LA92. These drive cycles have the values for different parameters such as Current, Voltage, Temperature, Average Current, Average Voltage, and SOE at four different temperature set, i.e. −10, 0, 10, and 25 °C. Input parameters that will be fed into the input layers were put in a vector X and output parameter, i.e. SOE in Y. Since neural network have 3 layers, input, output, and hidden layers, input and output layers were ready as X and Y but number of hidden layers and number of neurons in each layers were still needed to be found and optimized. Therefore for finding the optimized values, range of values for hidden layer and neurons too were decided. RMSE was calculated for the different combinations of these values using the grid-based approach. Table 1 shows the RMSE and training time values for the different combinations. Finally, after the selection of optimized values of number of hidden layers and neurons the modelling of DNN begins. Optimization of these hyperparameters are utmost important because performance and efficiency of model depends on these parameters. In this study, the activation function of hidden layer was chosen to be ReLU, since from the literature study it was found to be most efficient among the other alternatives. The model used in this study updates the weights with the Adam optimization technique with learning rate 0.001. Mean Squared Error (MSE) loss function and Mean Absolute Error (MAE) metrics was chosen during the training of model. Performance analysis was done using the RMSE, as it can evaluate the percentage deviation of predicted values from original values. Training was done using the computer system with specifications of 4th generation core i5 processor and intel HD graphics. In all the cases of this study, model was trained for 1000 epochs to minimize error and to improve the model efficiency. Models were developed for 2 different cases, as in Case 1 model was trained with 3 drive cycles, i.e. UDDS, HWFET, and LA92 at all temperatures set and then performance evaluation was done using US06 drive cycle at different temperatures. In Case 2, the accuracy of model was analysed in mapping the temperature effect. All drive cycles were used for training but considering only three temperature set, i.e. −10, 0, and 25 °C. In this case evaluation was done using three drive cycles as UDDS, LA92, and HWFET at 10°C. The main purpose of considering these case studies were to evaluate the performance of model at some new temperature and new drive cycle which was not considered during the training. Results and comparison of these 2 cases are shown in Sect. 3. The performance of all the models was evaluated on the basis of MAE and RMSE error metrics.

2.2 SVR Architecture

In machine learning, Support Vector Machines (SVM) is the supervised learning models which analyse data for the classification and regression analysis. This study focuses on the SVR analysis technique. The functioning of SVR is in the way that it makes a complex relationship between input parameters and single output by optimizing the objective function, whose values get minimized when the predicted value from model approaches the target or original output value. For this case also the input parameters remain same as it was used for DNN, i.e. current, voltage, temperature, average current and average voltage. The major task of SVR is to correctly and precisely map the input parameters with the output value, i.e. SOE. For this study, it is not possible to have a linear relationship between the input and target values. Therefore, for this type of circumstance input values get linearly mapped with the output values in higher dimensional plane and the functions used for these transformations are known as Kernels (Müller et al. 2001). The available kernels functions in the SVR technique are sigmoid function, polynomial function, Radial Basis function (RBF), etc. Among the other available kernel function and based on the nature of the relationship between input and output parameters, RBF kernel was the most suitable for this study (Schölkopf et al. 1997). The architecture of the SVR is shown in Fig. 3, where an insight about the mapping of input and output parameters can be understood easily. From Fig. 3, it is shown that $K(x, x_n)$ is the output of the nth hidden node for the input X, also it is the mapping of input vector x and support vectors by precisely choosing kernel function (Chen and Yu 2007).

In the SVM technique, the two regression methods are available, i.e. ε—SVR and υ—SVR. Both methods differ from each other in a way that in ε – SVR methods

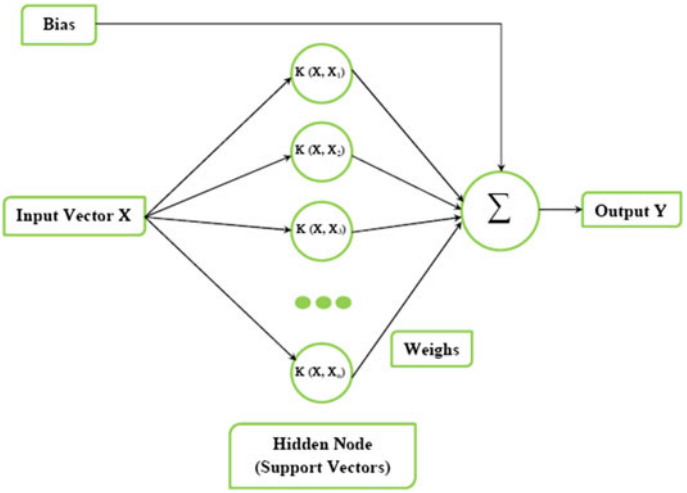


Fig. 3 υ—SVR architecture

there is no control on the number of data vectors from input dataset that become support vectors. While in υ—SVR method, there is the control on the limit of error tolerance where υ represents the upper bound on the fraction of error from training dataset and the lower bound on the fraction of support vectors. This control on limit of error tolerance can regulate by the use of regularization parameter C. For this study, authors find υ-SVR method more suitable since error tolerance in the prediction can be controlled.

2.2.1 SVR Hyperparameters and its Optimization

Similar to DNN modelling, the SVR modelling also involves hyperparameters handling and its optimization. In the SVR analysis the main hyperparameters which affect the performance and efficiency of model are gamma (γ), C, kernel function, and υ. The γ parameter shows the range of influence of input training data samples. Simply, it can be defined as the inverse of radius of influence of support vectors selected by model from the dataset. C is the regularization parameter which controls the trade-off between achieving the low training and low testing error. Kernel function is responsible for making the required relationship between input and output parameters. Due to the nonlinear relationship between the input and target parameters of this study, RBF was chosen as the kernel function. The use and functioning of υ is already explained in Sect. 2.2, it should be in the interval of (0, 1]. As the optimization of all the hyperparameters at the same time cannot be the efficient way to do the required study, therefore the default value of υ was chosen as 0.5, RBF kept as the kernel function, and then the optimum value was selected for the γ and C using the optimization technique similar to the DNN case. The grid-based approach similar to the one shown in Sect. 2.1.1 for the optimization of DNN hyperparameters was used for the optimization of C and γ. The range of values decided for the regularization parameter, C was –0.01, 0.1, 1, 10 and 100. While three values were chosen for gamma parameter for iterating with the combination in C, i.e. 0.1, 1, and 10. Table 3 shows the values of RMSE and training time for the different combinations of C and γ, validation drive cycle, i.e. US06 at 25°C was used for the testing. From these grids one combination of C and γ was chosen at which RMSE was found to be minimum and the training time was very optimum. The combination which satisfied these criteria was C as 0.1 and γ as 10. The RMSE obtained at this combination was 0.016497, which was minimum among rest of the grid combination and training time was 698.3137 s. All the parameters which were used in this modelling and their respective selected optimum values are shown in Table 4.

2.2.2 SVR Training Strategy

One of the most important parts of modelling in machine learning is the training strategy. This section will give the detail insight to the reader about the modified υ – SVR algorithm with the simultaneous optimization of hyperparameters. The

Table 3 Grids showing the values of RMSE and training time for different combinations of C and γ

C Gamma (γ)	0.01	0.1	1	10	100
Root Mean Squared Error (RMSE)					
0.1	0.099553	0.07286	0.065908	0.049061	0.040832
1	0.049851	0.03734	0.024061	0.017731	0.018968
10	0.031577	**0.016497**	0.017534	0.017223	0.007121
Training Time (seconds)					
0.1	667.7578	655.6799	671.5883	432.3391	1165.202
1	631.336	651.5541	407.5546	1106.363	7541.776
10	616.2116	**698.3137**	1557.31	11363.25	39771.38

Table 4 Optimized values selected for the hyperparameters of SVR

Hyperparameters	Optimized values
C	0.1
γ	10
υ	0.5
Kernel function	RBF

modified algorithm used for this method is shown in Appendix A.2. The details of the optimization of C and γ are already shown in Sect. 2.2.1. The datasets used for the training in this case are also similar with the DNN training. In this technique also 2 cases were studied such as in first case, three drive cycles, i.e. UDDS, LA92, and HWFET at temperature set of −10, 0, 10, and 25 °C was used for training. In this case the performance was evaluated by using the US06 drive cycle at all the same temperatures from −10 to 25 °C. In case study 2 as that of DNN training explained in Sect. 2.1.2, such that all the four cycles were fed for the training at three temperatures, i.e. −10, 0, and 25 °C. The validation was done using the three drive cycles as UDDS, LA92, and HWFET at 10°C. All these trainings were done using the same kernel function as RBF and nu (υ) as 0.5. Since in the SVR technique there is no involvement of epochs, therefore its computation time fully depends on the selection of these hyperparameters. Training was done using the same computer system with specifications of 4th generation core i5 processor and intel HD graphics. The performance of all the cases was studied on the basis of MAE and RMSE error metrics. The detailed comparison among the results of these cases is shown in Sect. 3.

Table 5 LG 18650HG2 cell parameters

Cell parameter	Specification
Chemistry	$Li[NiMnCo]O_2$ (H-NMC) / Graphite + SiO
Nominal Voltage	3.6 V
Charge	1.5A,4.2,50 mA End-Current (CC-CV) Normal
	4A, 4.2 V,100 mA End-Current (CC-CV) Fast
Discharge	2 V End Voltage, 20A Max Continuous Current
Nominal capacity	3.0 Ah
Energy density	240 Wh/Kg

3 Processing of Data for Training and Validation

3.1 Data Collection

In this study for the development of data-driven models and for the evaluation, the open source data was used available from Macmaster University (Kollmeyer et al. 2020). LG 18650HG2 battery was used in the testing for the gathering of this data as mentioned in the technical instruction of the open source data. The detailed information of battery used for the testing is shown in Table 5. The data are available for four different drive cycles, i.e. US06, UDDS, LA92, and HWFET tested at four different ambient temperature conditions as −10, 0, 10, and 25 °C. For this study all the required data for these drive cycles were gathered from this source (Kollmeyer et al. 2020) and then further data were prepared according to the cases of this particular study. The training and validation dataset prepared according to the cases as shown in Sect. 3.2.

3.2 Training and Validation Dataset

In this study, four drive cycles were used for the training in different combination as shown in Table 6. Simulation of battery cells by the use of these driving cycles is very time efficient as well as cost efficient. Use of these drive cycles in battery simulation makes it easier to monitor the behaviour of battery cells during real-world driving patterns without doing any extra experimentation on real physical driving vehicle. Driving cycles used in this study were Supplemental Federal Test Procedure

Table 6 Training and Validation dataset used in the study

Cases	Testing dataset	Validation dataset
Case 1	UDDS, LA92, and HWFET drive cycle at -10, 0, 10, and 25 °C	US06 drive cycle at -10, 0, 10, and 25 °C
Case 2	US06, UDDS, LA92, and HWFET drive cycle at -10, 0, and 25°C	UDDS, LA92, and HWFET drive cycle at 10 °C

(US06), Urban Dynamometer Driving Schedule (UDDS), Unified Driving Schedule (LA92), and the Highway Fuel Economy Driving Schedule (HWFET). These cycles are described in Appendix B. As an example, Fig. 4 shows the behaviour of the battery parameters, i.e. current, voltage, and temperature of US06 drive cycle at all ranges of temperature from −10 to 25 °C. Training and validation dataset used in

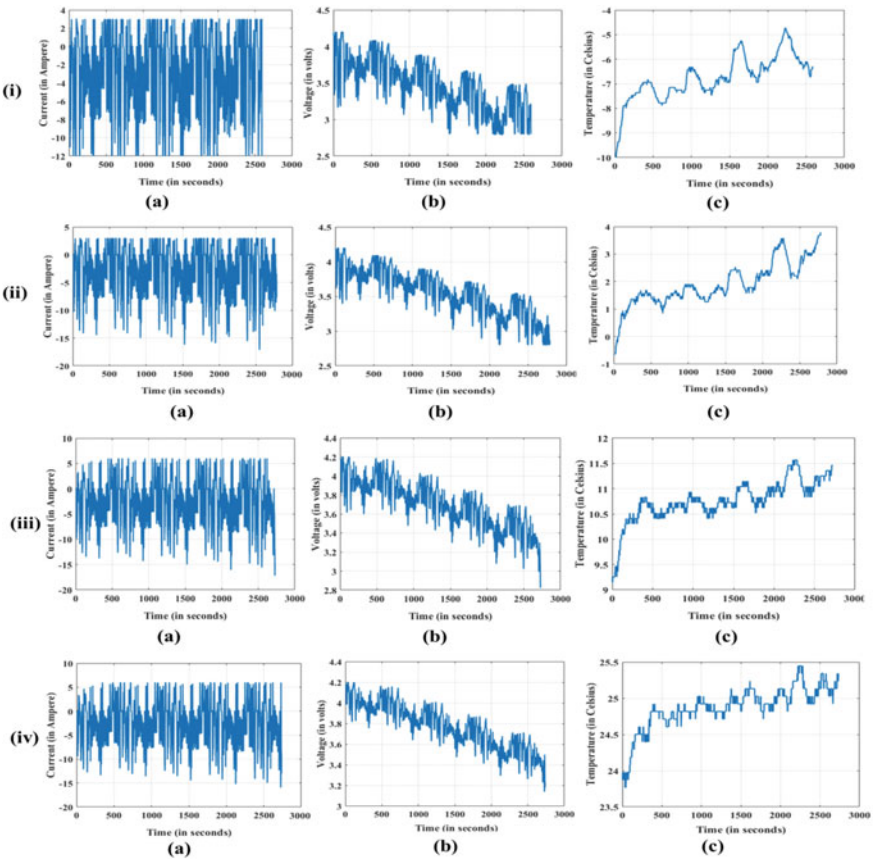


Fig. 4 The report of the US06 drive cycle in terms of: **a** Current, **b** Voltage, **c** Cell Temperature. This test was done at four conditions of room temperature: (i) −10 °C, (ii) 0 °C, (iii) 10 °C, (iv) 25 °C

different cases for the study in this research are shown in Table 6. Two cases were studied using DNN and then results of DNN were compared by another Machine Learning technique, i.e. SVR. These two cases are already explained in Sect. 2.1.2 and 2.2.2 according to the respective DNN and SVR training.

All the input vectors in each case were normalized before feeding for the training. Normalization technique was used to improve the convergence rate and to remove the negative influence. In this study min–max normalization was used since it retains the original data distribution pattern, only scaled the dataset in the range of [0, 1] as shown in Eq. (6) (Jain et al. 2005).

$$z_i^k = \frac{x_i^k - \min(x)}{\max(x) - \min(x)} i \in \{1, 2, \ldots\ldots, n\} \tag{6}$$

where x is the input vectors such as I(t), V(t), T(t), $I_{avg}(t)$, $V_{avg}(t)$, and n represents the total number of samples of data in the different drive cycles.

3.3 Evaluation Metrics

For the study of performance of trained models, some evaluation metrics were used such as MAE and RMSE. These evaluation or error metrics made the way of the comparison of different machine learning methods and different case study done in these methods in this research paper. These evaluation or error metrics can be represented by the Eqs. (7), (8), and (9).

$$\text{MAE} = \frac{1}{N}\sum_{k=1}^{N}\left(\left|\text{SOE}_k - \text{SOE}_k^*\right|\right) \tag{7}$$

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left(\left|\text{SOE}_k - \text{SOE}_k^*\right|\right)^2 \tag{8}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\left|\text{SOE}_k - \text{SOE}_k^*\right|\right)^2} \tag{9}$$

where SOE_k is the predicted value by model, SOE_k^* is the actual value obtained from battery testing at time step k and N is the total number of training samples.

4 Results and Discussion

As mentioned in earlier section, the input vector fed in to the DNN and SVR can be represented as $X = [I(t), V(t), T(t), I_{avg}(t), V_{avg}(t)]$ and the output can be represented as $Y = SOE(t)$, where $I(t)$, $V(t)$, $T(t)$, $I_{avg}(t)$, $V_{avg}(t)$ and $SOE(t)$ denotes the current, voltage, average current, average voltage and estimated SOE at time step t. The drive cycles used for the training and validation are mentioned in Sect. 3.2 and were recorded at sampling frequency of 1 Hz. The following subSects. 4.1 and 4.2 will show the information about the results obtained after the training and validation according to different cases used in this study. Comparative analysis is shown in these sections for the DNN and SVR modelling for the prediction of SOE.

4.1 Case Study 1

In this case, as described earlier that three drive cycles namely UDDS, LA92, and HWFET were fed for training at all temperature sets as −10, 0, 10, and 25 °C. For the evaluation of the trained model US06 drive was used at all the temperature ranges. Comparative study is shown here between DNN and SVR technique of machine learning. Estimation performance of DNN and SVR was compared on the basis of error metrics, i.e. RMSE and MAE. Figure 5 depicts the prediction of SOE using both data-driven methods, i.e. DNN and SVR. It shows the comparative representation of SOE between actual SOE, predicted SOE using DNN, and predicted SOE using SVR. Figure 5a–d show the prediction of SOE using US06 drive cycle at four different temperature ranges, i.e. −10 °C, 0 °C, 10 °C, and 25 °C, respectively. Table 7 shows the values of RMSE and MAE for both the technique of machine learning, i.e. DNN and SVR.

4.2 Case Study 2

In case 1 the performance of model was analysed on a new drive cycle which was not fed for the training. In this case, model was evaluated using the drive cycles at temperatures set other than which fed for training. In this case all the drive cycles were used for the training but only at three temperature values, i.e. −10, 0, and 25 °C. Drive cycles at 10 °C were kept for the testing of developed model accuracy. Three drive cycles at 10 °C, i.e. UDDS, LA92, and HWFET was used for the validation

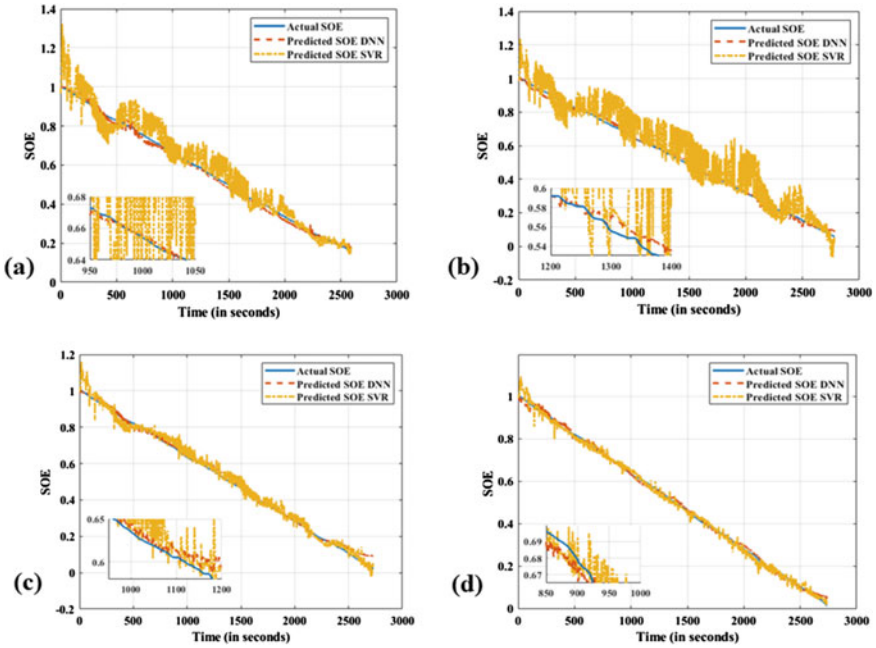


Fig. 5 Prediction of SOE using DNN and SVR technique on US06 drive cycle at following temperature range **a** at −10 °C temperature **b** at 0 °C temperature **c** at 10 °C temperature **d** at 25 °C temperature

Table 7 Evaluation results on the individual drive cycles

	DNN		SVR	
Testing Drive Cycle	RMSE	MAE	RMSE	MAE
US06 at −10°C	0.0148	0.0125	0.0602	0.0442
US06 at 0°C	0.0149	0.0118	0.0898	0.0672
US06 at 10°C	0.0178	0.0128	0.0287	0.0218
US06 at 25°C	0.012	0.010	0.0164	0.0122

Table 8 Evaluation results on the combination of drive cycles

	DNN		SVR	
Training Drive Cycle	RMSE	MAE	RMSE	MAE
UDDS at 10 °C	0.0154	0.0120	0.0207	0.0156
LA92 at 10 °C	0.0189	0.0162	0.0257	0.0177
HWFET at 10 °C	0.0118	0.0093	0.0339	0.0235

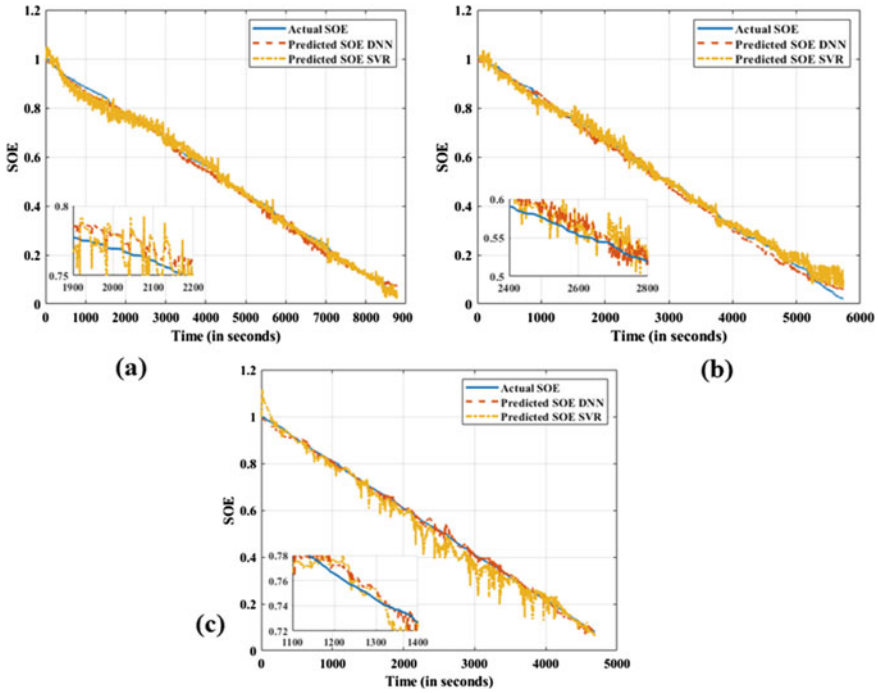


Fig. 6 Prediction of SOE using DNN and SVR technique on following drive cycle at 10 °C temperature **a** UDDS drive cycle **b** LA92 drive cycle **c** HWFET drive cycle

purpose. This same procedure was followed in both ML techniques. Figure 6 shows the predicted SOE behaviour for the different combinations of drive cycles in graphical way. In this case also, it was seen that the error in the prediction of SOE was high for SVR method in comparison to DNN. The RMSE and MAE error values are shown in Table 8 for both methods and it can be easily seen that DNN performance is significantly better as compared to SVR.

5 Conclusion

This work offers a unique contribution to the accurate estimation of SOE. In this study modified algorithm for DNN and SVR with optimization techniques are implemented. The effectiveness of DNN and SVR methods were compared for the prediction of SOE. Instead of using any random hyperparameters, this study proposed the approach to optimize the hyperparameters and then used the optimized values of hyperparameters in the subsequent prediction. During the optimization, it was noticed that the error in the prediction reduces as the hidden layers and number of neurons increases in the case of DNN but higher values of them make the model computationally less efficient. Similarly in the case of SVR technique, if the value of regularization parameter (C) and gamma (γ) increases then the training time increases significantly, this makes the model less time efficient. This work shows the unique method for the optimization of hyperparameters of both techniques, i.e. DNN and SVR. The prediction result of this work shows that the DNN model is more efficient in the prediction of SOE as compared to SVR. The results obtained in the different case studies show that the SOE prediction by DNN model at a drive cycle different from the one used in training is quite good. In the same way, DNN model is able to predict the SOE at different thermal ranges other than those used in training. The results of the case studies evidently prove that the prediction from DNN is far better than SVR. This work also suggests that the conventional regression model used for the estimation of SOE can be upgraded using the DNN model in the future. This work highlights the prevailing challenges in the industry and proposes the potential recommendation for BMS development and SOE estimation in next-generation EV applications.

Appendix A

A.1 Modified DNN Algorithm With the Optimization Technique

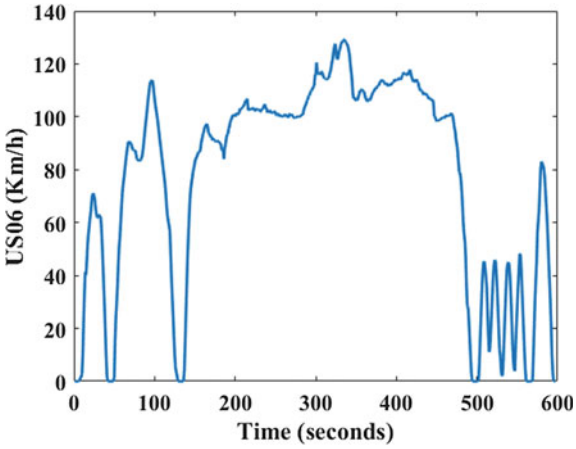
```
Import dataset  // Import the dataset required for the training
X = ['Current', 'Voltage', 'Average Current', 'Average Voltage'] // Input parameters
Y = ['SOE']  // Output parameters
Define range of values of hidden layers // range of values, example- 2, 3, 4……
Define range of values of neurons // range of neurons values, example- 1, 2, 4 ……
// Initializing optimization loop for hidden layers and neurons
For i in range of hidden layers values  // loop for the range of hidden layers values
    For j in range of neurons values // loop for the range of neurons values
    find RMSE  // rmse for the combination of hidden layer and neurons values
Hidden layers and neurons = (L, N) // Optimum values on the basis of minimum of rmse
Import model  // import the model of DNN
From layers import Dense  // import dense layer that will be add in model
model. add(Dense(units = N , input_dim = no. of input parameters,  activation = 'relu'))
// add the dense layer, where N is optimum value of neuron, input_dim is the no, of input
parameters and activation function is ReLU. More layers can be add in the similar manner
model.add(Dense(units = 1, activation = 'linear')) // Output layer with Linear activation
model.compile(optimizer = Adam(learning_rate = 0.01), loss = 'MSE', metrics = ['mae'])
// Compile the model with Adam optimizer, learning rate fixed to be 0.01, loss function be
MSE and metrics is MAE
model.fit(X, Y, epochs = 10000)  // fitting of model where epochs is the no. of iterations
Import testing dataset // import the dataset for the testing of performance of developed
model
Define X_test and Y_test  // define the X_test and Y_test in the similar manner as done for
training X and Y
Y_predict = model.predict(X_test)  // Calculate the predicted output
RMSE(Y_predict, Y_test)  // Finally calculate the rmse for predicted and actual output
```

A.2 Modified υ—SVR Algorithm With the Optimization Technique

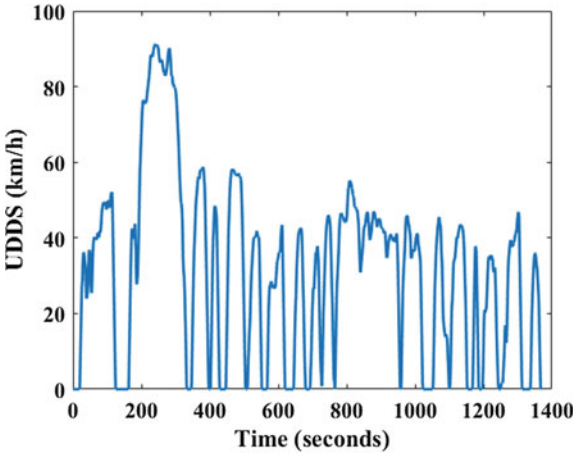
```
Import dataset  // Import the dataset required for the training
X = ['Current', 'Voltage', 'Average Current', 'Average Voltage'] // Input parameters
Y = ['SOE']  // Output parameters
Define range of values of C // range of values selected were 0.01, 0.1, 1, 10, 100
Define range of values of γ // range of γ values selected 0.1, 1, 10
// Initializing optimization loop for hidden layers and neurons
For i in range of γ values  // loop for the range of γ values
    For j in range of C values // loop for the range of C values
    find RMSE  // rmse for the combination of C and γ values
C and γ = (c, gp) // Optimum values on the basis of minimum of rmse
Import model  // import the υ-SVR model of SVM
model = NuSVR(kernel = 'rbf', C = c, gamma = gp, nu = 0.5)  // fitting of model
Import testing dataset // import the dataset for the testing of performance of developed model
Define X_test and Y_test  // define the X_test and Y_test in the similar manner as done for
training X and Y
Y_predict = model.predict(X_test)  // Calculate the predicted output
RMSE(Y_predict, Y_test)  // Finally calculate the rmse for predicted and actual output
```

Appendix B

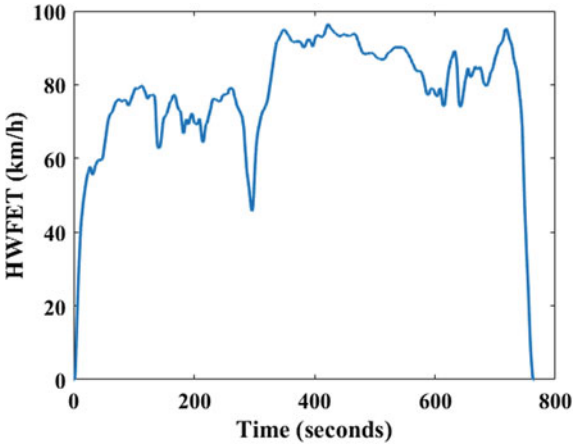
B.1 US06 Drive Cycle Velocity Profile



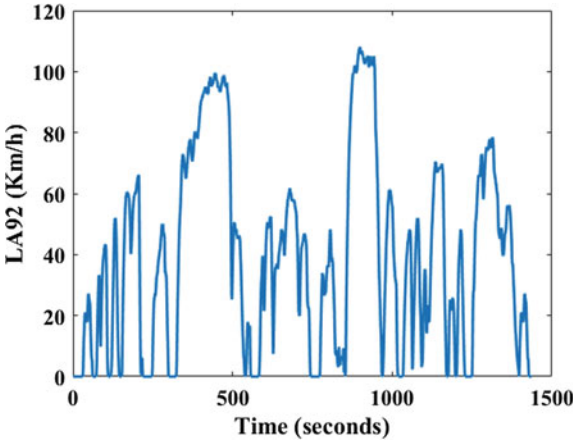
B.2 UDDS Drive Cycle Velocity Profile



B.3 HWFET Drive Cycle Velocity Profile



B.4 LA92 Drive Cycle Velocity Profile



References

Chen ST, Yu PS (2007) Pruning of support vector networks on flood forecasting. J Hydrol, Elsevier 347(1–2):67–78. https://doi.org/10.1016/j.jhydrol.2007.08.029

Cicconi P, Kumar P, Varshney P (2020) A support approach for the modular design of li-ion batteries: a test case with PCM. J Energy Storage, Elsevier Ltd 31:101684. https://doi.org/10.1016/j.est.2020.101684

Gandoman FH et al (2019) Concept of reliability and safety assessment of lithium-ion batteries in electric vehicles: Basics, progress, and challenges. Appl Energy. https://doi.org/10.1016/j.apenergy.2019.113343

German R et al (2020) Dynamical coupling of a battery electro-thermal model and the traction model of an EV for driving range simulation. IEEE Trans Veh Technol. https://doi.org/10.1109/TVT.2019.2955856

Hafsaoui J, Sellier F (2010) Electrochemical model and its parameters identification tool for the follow up of batteries ageing. In: EVS 2010-sustainable mobility revolution: 25th world battery hybrid and fuel cell electric vehicle symposium and exhibition, vol 4, pp 386–395

Hausmann A, Depcik C (2013) Expanding the Peukert equation for battery capacity modeling through inclusion of a temperature dependency. J Power Sources, Elsevier 235:148–158. https://doi.org/10.1016/j.jpowsour.2013.01.174

He K et al (2016) Deep residual learning for image recognition. http://image-net.org/challenges/LSVRC/2015/. Accessed 30 June 2020

He Y et al (2013) A new model for state-of-charge (SOC) estimation for high-power Li-ion batteries. Appl Energy, Elsevier Ltd 101:808–814. https://doi.org/10.1016/j.apenergy.2012.08.031

Iclodean C et al (2017) Comparison of different battery types for electric vehicles. In: IOP conference series: materials science and engineering. https://doi.org/10.1088/1757-899X/252/1/012058

Jaguemont J et al (2018) Phase-change materials (PCM) for automotive applications: a review. Appl Therm Eng. https://doi.org/10.1016/j.applthermaleng.2017.12.097

Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern Recogn Pergamon 38(12):2270–2285. https://doi.org/10.1016/j.patcog.2005.01.012

Kang LW, Zhao X, Ma J (2014) A new neural network model for the state-of-charge estimation in the battery degradation process. Appl Energy, Elsevier Ltd 121:20–27. https://doi.org/10.1016/j.apenergy.2014.01.066

Kermani S et al (2011) PHIL implementation of energy management optimization for a parallel HEV on a predefined route. IEEE Trans Veh Technol 60(3):782–792. https://doi.org/10.1109/TVT.2011.2107534

Kim IS (2006) The novel state of charge estimation method for lithium battery using sliding mode observer. J Power Sources. https://doi.org/10.1016/j.jpowsour.2006.09.006

Kim IS (2010) A technique for estimating the state of health of lithium batteries through a dual-sliding-mode observer. IEEE Trans Power Electron. https://doi.org/10.1109/TPEL.2009.2034966

Kollmeyer P, et al (2020) LG 18650HG2 Li-ion battery data and example deep neural network xEV SOC estimator script. Mendeley 3.https://doi.org/10.17632/CP3473X7XV.3

Kumar P et al (2020) Critical review on battery thermal management and role of nanomaterial in heat transfer enhancement for electrical vehicle application. J Energy Storage 32:102003. https://doi.org/10.1016/j.est.2020.102003

Liu X et al (2014) 'A method for state of energy estimation of lithium-ion batteries at dynamic currents and temperatures. J Power Sources, Elsevier 270:151–157. https://doi.org/10.1016/j.jpowsour.2014.07.107

Lu L et al (2013) A review on the key issues for lithium-ion battery management in electric vehicles. J Power Sources. https://doi.org/10.1016/j.jpowsour.2012.10.060

Lu M et al (2020) Research progress on power battery cooling technology for electric vehicles. J Energy Storage. https://doi.org/10.1016/j.est.2019.101155

Mamadou K et al (2012) Definition of a state-of-energy indicator (SoE) for electrochemical storage devices: application for energetic availability forecasting. J Electrochem Soc, The Electrochemical Society 159(8):A1298–A1307. https://doi.org/10.1149/2.075208jes

Mamadou K et al (2019) The state-of-energy: a new criterion for the energetic performances evaluation of electrochemical storage devices. ECS Trans, the Electrochemical Society 25(35):105–112. https://doi.org/10.1149/1.3414008

Müller KR et al (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Netw:181–201. https://doi.org/10.1109/72.914517

Ng KS et al (2009) Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. Appl Energy, Elsevier Ltd 86(9):1506–1511. https://doi.org/10.1016/j.apenergy.2008.11.021

Plett GL (2004a) Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—part 1 Background. J Power Sources. https://doi.org/10.1016/j.jpowsour.2004.02.031

Plett GL (2004b) Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 2 modeling and Identification. J Power Sources. https://doi.org/10.1016/j.jpowsour.2004.02.032

Plett GL (2004c) Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—part 3 state and parameter estimation. J Power Sources. https://doi.org/10.1016/j.jpowsour.2004.02.033

Qian X et al (2019) Heat dissipation optimization of lithium-ion battery pack based on neural networks. Appl Therm Eng. https://doi.org/10.1016/j.applthermaleng.2019.114289

Schölkopf B et al (1997) Comparing support vector machines with gaussian kernels to radial basis function classifiers. IEEE Trans Signal Process 45(11):2758–2765. https://doi.org/10.1109/78.650102

Shen WX (2007) State of available capacity estimation for lead-acid batteries in electric vehicles using neural network. Energy Convers Manag Pergamon 48(2):433–442. https://doi.org/10.1016/j.enconman.2006.06.023

Stockar S et al (2011) Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles. IEEE Trans Veh Technol 60(7):2949–2962. https://doi.org/10.1109/TVT.2011.2158565

Tie SF, Tan CW (2013) A review of energy sources and energy management system in electric vehicles. Renew Sustain Energy Rev. https://doi.org/10.1016/j.rser.2012.11.077

Waag W, Sauer DU (2013) Adaptive estimation of the electromotive force of the lithium-ion battery after current interruption for an accurate state-of-charge and capacity determination. Appl Energy, Elsevier Ltd 111:416–427. https://doi.org/10.1016/j.apenergy.2013.05.001

Wang F et al (2014a) 'Low-temperature electrochemical performances of LiFePO4 cathode materials for lithium ion batteries', J Taiwan Instit Chem Eng. Taiwan Institute of Chemical Engineers 45(4):1321–1330. https://doi.org/10.1016/j.jtice.2014.02.013

Wang Y, Zhang C, Chen Z (2014b) A method for joint estimation of state-of-charge and available energy of LiFePO4 batteries. Appl Energy. https://doi.org/10.1016/j.apenergy.2014.08.081

Xia G, Cao L, Bi G (2017) A review on battery thermal management in electric vehicle application. J Power Sources. https://doi.org/10.1016/j.jpowsour.2017.09.046

Xiong R et al (2014) A data-driven multi-scale extended Kalman filtering based parameter and state estimation approach of lithium-ion olymer battery in electric vehicles. Appl Energy. https://doi.org/10.1016/j.apenergy.2013.07.061

Xu J et al (2014) The state of charge estimation of lithium-ion batteries based on a proportional-integral observer. IEEE Trans Veh Technol. https://doi.org/10.1109/TVT.2013.2287375

Xu L, Wang J, Chen Q (2012) Kalman filtering state of charge estimation for battery management system based on a stochastic fuzzy neural network battery model. Energy Convers Manage Pergamon 53(1):33–39. https://doi.org/10.1016/j.enconman.2011.06.003

Yi J et al (2013) Modeling the temperature dependence of the discharge behavior of a lithium-ion battery in low environmental temperature. J Power Sources, Elsevier B.V, 244:143–148. https://doi.org/10.1016/j.jpowsour.2013.02.085

Zhang W, Shi W, Ma Z (2015) Adaptive unscented Kalman filter based state of energy and power capability estimation approach for lithium-ion battery. J Power Sources. https://doi.org/10.1016/j.jpowsour.2015.04.148

Zheng Y et al (2013) LiFePO4 battery pack capacity estimation for electric vehicles based on charging cell voltage curve transformation. J Power Sources, Elsevier 226:33–41. https://doi.org/10.1016/j.jpowsour.2012.10.057

Zhong L et al (2014) A method for the estimation of the battery pack state of charge based on in-pack cells uniformity analysis. Appl Energy, Elsevier Ltd 113:558–564. https://doi.org/10.1016/j.apenergy.2013.08.008