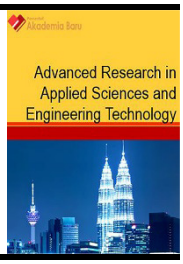
Penerbit Akademia Baru

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: www.akademiabaru.com/araset.html
ISSN: 2462-1943



Modeling battery state of charge in wireless sensor networks based on structured multi-layer perceptron

Open Access

Amzar Omairi [1], Z. H. Ismail [1,*]

[1] Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, Kuala Lumpur, 54100, Malaysia

ARTICLE INFO

Article history:
Received 31 May 2016
Received in revised form 30 June 2016
Accepted 15 August 2016
Available online 19 December 2016

ABSTRACT

Energy consumption of Wireless Sensor Networks (WSN) is an important aspect in the design requirement. This is especially true in a situation where WSN is being operated in isolated areas and thus relying on batteries due to unavailability of power infrastructure. Since energy efficiency is the main concern in the deployment of WSN, the sensor node must keep track of the charge that is left in the battery, commonly referred as the State of Charge (SoC). To prevent the discontinuation of the operation of the sensor node from power cut off, it is important to find an analytic model for the battery's state of charge. In this paper, an optimized structure of Multi-Layer Perceptron (MLP) is utilized to obtain a model of the battery state-of-charge in wireless sensor nodes. Results show the suitability of the method that produces accurate and simple models, capable of being implemented even in low cost and very constrained real motes.

Keywords:
Wireless sensor networks, State-of-charge, Multi-layer perceptron, Energy consumption

1. Introduction

A wireless sensor network (WSN) is composed of a number of sensor nodes that communicate with each other through a wireless network where the data of each node are integrated. WSN has numerous significant applications in the recent years, such as remote environmental monitoring. This has allowed the deployment of WSN lately, to have sensors that are smaller, cheaper, and reliable. Wireless interfaces are equipped in these WSN thus allowing communication with one another to form a network.

WSN comprise of a set of nodes (motes) capable of sensing, processing, storing, and communicating [1]. The critical issue to combat in developing WSN is the limited amount of energy that are available in each of the motes. An application may take weeks to drain the battery of the

* *Corresponding author.*
E-mail address: zool@utm.my (Z. H. Ismail)

sensor node in other cases consume it in a matter of days depending on what types of sensors and communication means the network is using.

In most cases, energy consumption is an important aspect in the design requirement. This is especially true in a situation where WSN is being operated in isolated areas and thus relying on batteries due to unavailability of power infrastructure. Regrettably, the energy density of the batteries did not follow the same trend in the advancement in WSN, and energy harvesting systems can power only a limited sorts of devices, usually with restricted capabilities [2]. Additionally, these sensor nodes must comply with a very strict power allowances to fulfil their duties while providing a prolonged lifetime.

Alongside the development of wireless sensor networks, research in battery technology has attracted more and more attention worldwide to further improve the lifetime and operability of such power source. As one of the key parameters of a battery, the state of charge (SoC) is the most crucial factor to initiate the study on battery management system. SoC estimation is especially critical as over-discharging batteries could result in an irreversible damage or reduced service time [3]. Over The SoC is defined as percentage of remaining available capacity at a given time in an operation [4].

Estimation of SOC has become extensive area of research with companies in search of improve the performance of their portable products' battery life, a number of approaches are presented in the literature to monitor SOC [5,8]. In this study, the neural network method of MLP is going to be implemented and they are able to generate predictive results for many of the processes [9] in WSN. This is mainly to the nature of neural network that can learn and update their internal structure to adapt to a dynamic input. Moreover, neural network is effective in data processing due to its parallelism in computation [10]. Neural networks are data-driven in nature and able to build a system model without detailed physical knowledge of a system [11].

Thus, the main objective of this paper is to review and study a benchmark method [12], for obtaining simple battery models that aims to estimate precisely the state of charge level based on multilayer perceptron. The rest of the paper is organized as follows: Section 2 describes the data collection. In Section 3, an overview of MLP architecture is explained while numerical simulation results are provided in Section 4, evaluating the performance of the optimized structure MLP. Finally, concluding remarks are presented in Section 5.

2. Data collection

Multilayer perceptron (MLP) neural network is being used in this study since it is going to be based on the model that has been previously used for WSN application for the battery's state of charge estimation [12].

This section discusses the experimental data traces we use for the benchmark. The method done prior the modelling stage is the stress test bench. Where the system applies consecutives cycles of charge and discharge automatically. The intention is to degrade the batteries that we use during the measurement phase presented above, applying a certain number of charge and discharge cycles.

A control system opens and closes the switches, and it monitors the battery voltage. The control system begins detecting the battery status: charged or uncharged, then it applies a charge or discharge cycle. When the battery voltage is rated lower than 2.7 V, the battery is considered uncharged, and when the battery voltage is above 3.7 V.

Data traces are processed to generate the SoC model by applying a filter in the first stage to reduce noise. Derived parameters are then calculated by using the original traces by introducing

measurement conditions. To capture the behaviour of the battery several traces of the charging and discharging processes are recorded. A single trace consists of voltage and drain current measurements of a charging or discharging process. In the discharging case several traces at different load levels are recorded.

3. Structured MLP architecture

MLP is a feedforward neural network that composed of three main layers which are the input, hidden and output layers.

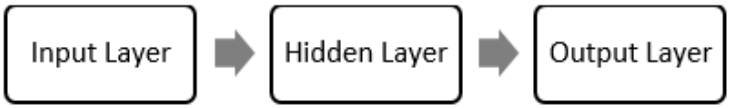


Fig. 1. MLP division

The first layer consists of one node for each of the components of the input data. The nodes in the input layer have a transfer function of unity which means their only function is to distribute the inputs to the nodes in the following layer.

The outputs of the input layer are connected to the inputs of the second layer which also where the hidden layer(s) starts. The outputs of the hidden layer are, in turn, connected to the inputs of the third layer, and so on. The final layer will eventually generate the output values of the MLP network. In this arrangement, the layers that sit between the first and last are not visible from outside the network, and is henceforth referred to as 'hidden layer'.

For every connections carrying the outputs of a layer to the inputs of the next layer have a weight assigned to them which includes the input layer. The node outputs are multiplied by these weights prior reaching the inputs of the following layer. Moreover, in MLP there are learning algorithms to set the values of the weights and the same basic structure (with different weight values) would enable it to perform many tasks.

Multilayer Perceptron delivers a powerful base-learner, with advantages such as noise tolerance and nonlinear mapping where the method is increasingly used in Data Mining due to its good behaviour in terms of predictive knowledge [13].

3.1. Hidden layers

In, MLP the nature of decision boundaries differs with the network topology. There are several conditions to consider:

- Single layer: have the ability to place a hyperplane in the input space (SLP).
- Two layers (single hidden layer): have the ability to define a decision boundary surrounding a single convex regression of input space. Sufficient to make universal approximator out of MLP.
- Three layers (dual hidden layers): have the ability to generate random decision boundaries.

Decision boundary can be estimated closely by a two layer network with sigmoidal activation functions when there are insufficient hidden units [14]. Fig. 2 shows a fully connected neural network with only one hidden layer [10].

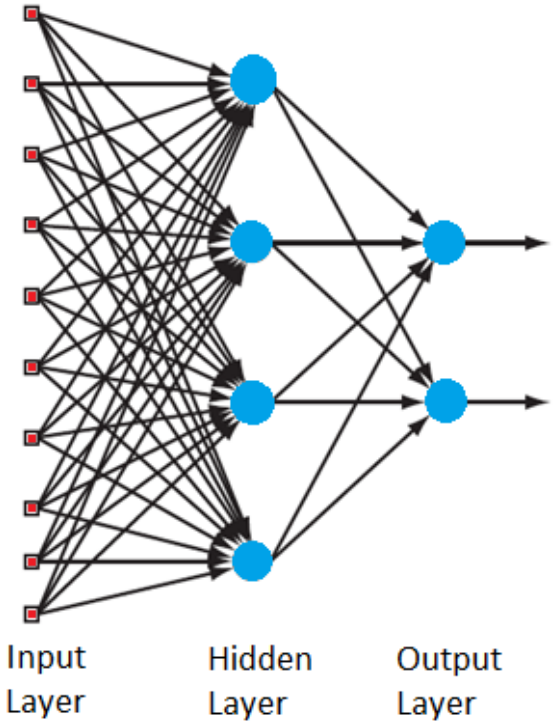


Fig. 2. Fully connected one hidden layer neural network

3.2. Nodes/Neurons

Every nodes are assigned by weights and output signals which are a function of the summation of the inputs to the node altered by a simple nonlinear transfer, or activation, function [15]. The number of output nodes is usually decided by the amount of output classes, while the number of input nodes is determined by the amount of input dimensions.

For hidden nodes, the number of nodes are depending on which factor would affect the MLP as a whole the most, as insufficient nodes would make the network to not model complex decision boundaries. In the other hand, a large number of nodes would cause the network to have poor generalization.

MLP are described as being totally connected, with every node connected to every other node in the next and previous layer [16].

3.3. Sigmoid function

A sigmoid function produces a curve with an “S” shape which is a form of logistic function. This function is a differentiable and real valued. The function is generally used in MLP to introduce non-linearity in the model. The sigmoid function is can be represented in Fig. 3 where the function assigns new values between 0 and 1.

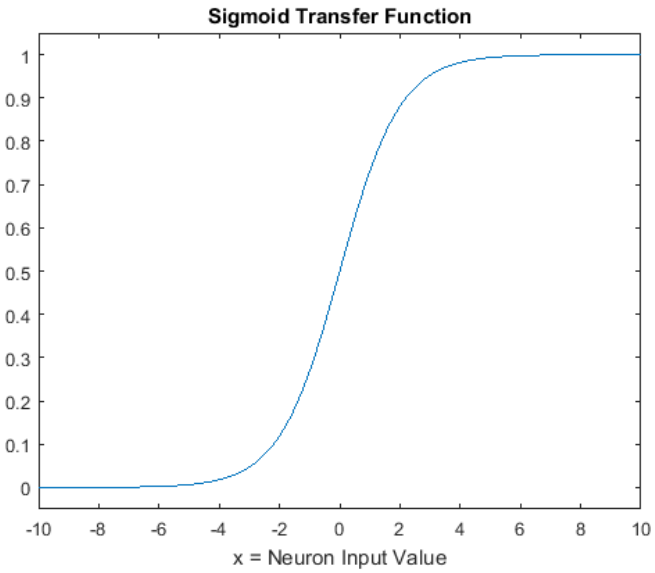


Fig. 3. Sigmoid transfer function

There are two main advantages of employing sigmoid function in MLP for predicting SOC, first, a squashing function ensures that the values in the network are always within a reasonable range. Second, a non-linear transfer function is crucial to allow a feed-forward networks to approximate any arbitrary equation, in which, to map any input to a desired output [17] which in this case the SOC reading based on the battery's parameters. The sigmoid activation function can be expressed by (1).

$$f = sig(s) = \frac{1}{1+e^{-P_i}} \quad (1)$$

where, p_i is the pre-activation function.

Pre-activation function can be defined as the output of a neuron where each input is multiplied by previously established weight, these products are then summed together.

4. Modelling IF SOC IN WSN based on MLP

The output, which also represents the pre-activation function (inputs multiplied by weights) can be express by (2).

$$p_i = \sum_{i=1}^{n} (x_i \cdot k_{ij}) + th_i \quad (2)$$

where, x_i, is the neuron input values, k_{ij}, is the assigned weights, th_i, is the threshold value of each node which is the bias (assumed=1) multiplied by the bias's weight. The pre-activation value n is sent through the activation function itself, f, which is the sigmoid in this case. Then, the resulting output, can be considered as input for the pre-activation function in the following layer which can be expressed as,

$$n_i = f(p_i) \quad (3)$$

In the case of two layer MLP (single hidden layer) being adapted by the benchmark model, the following pre-activation value can be considered as the estimation of the SOC.

From the benchmark model [12], a two layer with two neuron network is used in the MLP configuration with 500 epochs and using sigmoid as the activation function. The neural network is represented in Fig. 5 with the default learning rate and momentum values of 0.3 and 0.2 respectively. The first benchmark model with only one input can be expressed as,

$$SoC(norm) = th_2 + k_2 \cdot \frac{1}{1+e^{-(k_1 \cdot V_{low} + th_1)}} \quad (4)$$

where the input weight (k_{ij}) and threshold (th_i) coefficient obtained was, k_1=-11.336; th_1=6.056; k_2=-2.100; th_2=1.191.

If we compare the equation (4) and (3) it would fit the expression in equation (4) perfectly as the resulting output of the final node, y is represented by the SoC attribute in equation (4).
The pre-activation equation can be expressed by,

$$\mathrm{p_1} = \sum_{i=1}^{n}(x_1 \cdot \mathrm{k_{1\,0}}) + \mathrm{th_1} = \mathrm{V_{low}} \cdot \mathrm{k_1} + \mathrm{th_1} \quad (5)$$

Since there is only one hidden layer, the output of the hidden node can be equated by,

$$\mathrm{n_1} = \mathrm{f(p_1)} = \mathrm{f}\frac{1}{1+e^{-p_1}} = \frac{1}{1+e^{-(V_{low} \cdot k_1 + th_1)}} \quad (6)$$

Finally, the output of the final node located in the output layer can be equated by,

$$n_2 = SoC(norm) = f(p_1) \cdot k2 + th2 = \mathrm{th_2} + \mathrm{k_2} \cdot \frac{1}{1+e^{-(k_1 V_{low} + th_1)}} \quad (7)$$

The same comparison can be made with the second benchmark model which uses 4 nodes, two layers (one hidden layer), with four different inputs namely, cycles (Ciclos), temperature (Temp), relationship between voltages and currents (R), and slope (Vpend) where the expression at the output node is given by,

$$\mathrm{n_4} = \mathrm{SoC(norm)} = \sum_{i=1}^{n}(\mathrm{n_i} \cdot \mathrm{k_{ij}}) + \mathrm{th_4} = \mathrm{th4} + \mathrm{k41} \cdot \mathrm{n1} + \mathrm{k42} \cdot \mathrm{n2} + \mathrm{k43} \cdot \mathrm{n3} \quad (8)$$

Since there are three hidden nodes,

$$n_1 = \frac{1}{1+e^{-(k11 \cdot R + k12 \cdot V_{low} + k13 \cdot T + k14 \cdot S + k15 \cdot \log(C) + th1)}}$$

$$n_2 = \frac{1}{1+e^{-(k21 \cdot R + k22 \cdot V_{low} + k23 \cdot T + k24 \cdot S + k25 \cdot \log(C) + th2)}}$$

$$n_1 = \frac{1}{1+e^{-(k31 \cdot R + k32 \cdot V_{low} + k33 \cdot T + k34 \cdot S + k35 \cdot \log(C) + th3)}}$$

The subsequent hidden nodes can be expressed by,

$$n_{i+1} = f(p_{i+1}) = f\frac{1}{1+e^{-p_{i+1}}} \quad (9)$$

One of the most well addressed problems that arises in neural network is the problem of multiple local minim [18]. This can be tackled by distributing the input data for training, validating and testing subsets. The training set is going to be used in computing the values of gradient and updating the overall network weights and biases. The validation set is used to avoid overfitting where the error value on the validation set is monitored hand in hand during the training process. Finally, the test set is used to compare different models where the error value is not used during training.

Optimization wise, in the process of global search, it may stop the convergence to a non-optimal solution and determine the optimum number of hidden layers of three ANN. Recently, some studies in the optimization architecture problems such as using the radial basis function in neural network have been introduced in order to determine the final neural networks parameters [19],[20].

Back propagation is a gradient descent search algorithm which is based on the minimization of the total mean square error between actual output to the desired output.
The subsequent hidden nodes can be expressed by,

$$n_{i+1} = f(p_{i+1}) = f\frac{1}{1+e^{-p_{i+1}}} \tag{10}$$

5. Simulation results and discussion

Multilayer perceptron (MLP) neural network is being used in this study since it is going to be based on the model that has been previously used for WSN application for the battery's state of charge estimation [12].

WEKA is an acronym for Waikato Environment for Knowledge Analysis. It is a prevalent suite of machine learning software written in Java, developed at the University of Waikato. WEKA is available under the GNU General Public License [21].WEKA software is used to simulate and classify the raw data with attributes like temperature (Temp), battery cycle (Ciclos), low voltage (Vl), low current (Il), slope of low voltage (Vpend) and relation between current and voltage (R).

The MLP in WEKA is a classifier that uses backpropagation to classify instances. This network can be built by hand and/or created by an algorithm. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units).

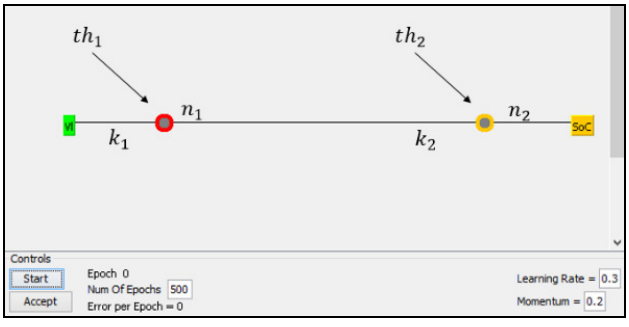


Fig. 4. Neural network representation of Equation 4 (Two Layers, Two Neurons)

Figure 4 shows the neural network representation of equation (4) while TABLE I presents all the data obtained from adding another input to equation (4).

Table 1
Simulation result with inclusion of low-current value (Il)

Cycles	Input	Correlation	Mean Abs. Error	Root Mean Square Error	Relative Absolute Error (%)	Root Relative Squared Error (%)
223	Vl	0.9993	0.9842	1.0192	1.3733	4.0768
223	Vl, Il	0.9994	1.0865	1.3975	4.3458	4.8408
220	Vl	0.9994	1.3157	1.4607	5.2628	5.0598
220	Vl, Il	0.9994	1.3656	1.5039	5.4622	5.2096
33	Vl	0.9993	0.8299	1.0529	3.3193	3.6471
33	Vl, Il	0.9994	0.8575	1.0669	3.4298	3.6957
31	Vl	0.9994	1.3332	1.5489	5.3326	5.3654
31	Vl, Il	0.9995	1.2948	1.5153	5.1789	5.2491
26	Vl	0.9993	0.7859	1.072	3.1436	3.7135
26	Vl, Il	0.9993	0.7602	1.0567	3.0409	3.6606

Table 1 shows the performance of equation (4) with another input current draw, Il, included, the first cycle shows different data trace being used for different charge-discharged cycle the battery has went through. The correlation tells how much the desired output value to the estimated value is associated, where a value near to 1 reflects a very strong linear relation. From Table 1 results, only two out of five data sets with different cycles shows improvement if another input which is the low current, Il, is included in the network, this are shown by cycles 26 and 31. The result shows the performance of equation (4) with another input current draw is included for the training phase. Only two out of five data sets with different cycles shows improvement if another input which is the low current, Il, is included in the network, this is shown by cycles 26 and 31. This further affirms the black box nature of a neural network machine learning and is highly dependent on the training phase where a more complicated topology of the network will not always yield better classification result as shown in the 2-3 neuron configuration network as shown in Table 2. Table 2 shows the performance of the MLP employing data trace from 223 charge-discharge cycle is employed with four results from different layer and neuron count is configured in the network. Only a single input is used which is the voltage reading, Vl.

Table 2
Simulation result from single input variable with different layers and neuron count

Cycles	Input	Hidden Layer(s)	Neuron Configuration in each layer	Correlation	Mean Abs. Error	Root Mean Square Error	Relative Absolute Error (%)	Root Relative Squared Error (%)
223	Vl	1	1	0.9994	0.7649	1.0869	3.0515	3.7578
223	Vl,	2	1-2	0.9994	0.8585	1.1702	3.4247	4.0461
223	Vl	2	1-3	0.9996	0.5704	0.934	2.2755	3.2293
223	Vl	2	2-2	0.9998	0.4502	0.6154	1.7961	2.1276
223	Vl	2	2-3	0.9996	0.5700	0.8883	2.2739	3.0712
223	Vl	2	3-3	0.9999	0.2284	0.2837	0.9113	0.981

Figure 5 depicts the classifier output data that is represented by (4) taken directly from Weka which shows how the errors, threshold values, th and weights, k are collected. The figure also highlights where each of the coefficients used in representing the MLP models is located in the classifier output window in WEKA. In addition, the results show the performance of the model by the

mean absolute error, root mean squared error, relative absolute error and root relative squared error.

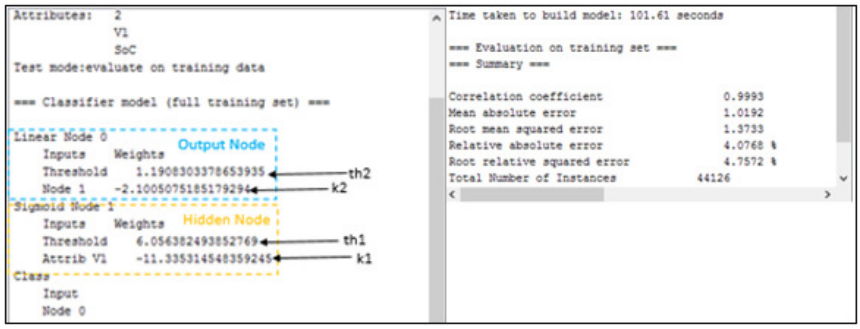


Fig. 5. Classifier model for Equation 4 (Two Layers, Two Neurons)

From the weight coefficients and threshold values obtained shown in Fig. 5, the WEKA classifier is able to give the final weight coefficient in order to produce the best correlation and error reading for the given input data set relative to the optimum SoC output data which is linear.

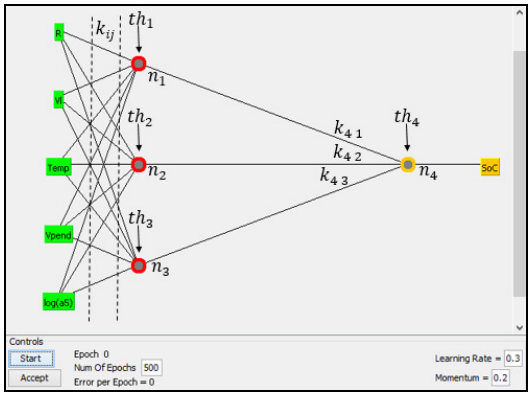


Fig. 6. Neural network representation of Equation 4 (Two Layers, Two Neurons)

Figure 6 shows the configuration of benchmark model equation (8) is presented in neural network architecture. Fig. 7 shows how all the errors, threshold values, th and weights, k are gathered. In this case, a single hidden layer MLP structure is used where there are 18 weight coefficients obtained for the four input variables to the three hidden nodes and for the output of the hidden layer to the output node of the network, which is illustrated in Fig. 6.

It can be observed that the difference between the given coefficients in (4) model and the simulation using data set of cycle 223 is very small as shown in Fig. 5 and Table 1. This shows that the expression in the benchmark models is derived from the root equation of neural network with multiple layers and nodes. Various weight coefficients are collected from WEKA with respect to all of the four nodes. For this simulation the values of coefficient are dissimilar from the benchmark due to the larger number of nodes which has a greater probability of obtaining unique sets of weights and threshold values using data set of cycle 223 [22].

6. Conclusion

This paper reviews on how two models are constructed and validated using multilayer perceptron neural network in classifying attributes being used in this paper. The effects of using

additional variable and different network topology were also demonstrated. The simulation done in this study is to show that the models from MLP can be obtained from tools like WEKA in rather a short amount of time with the configuration described and represented by (7) and (8).

Acknowledgment
This work was supported in part by the Ministry of Higher Education, Malaysia and Universiti Teknologi Malaysia under Grant no. Q.K130000.2543.13H80.

References

[1] Dâmaso, Antônio, Davi Freitas, Nelson Rosa, Bruno Silva, and Paulo Maciel. "Evaluating the power consumption of wireless sensor network applications using models." *Sensors* 13, no. 3 (2013): 3473-3500.
[2] Orfei, Francesco, Riccardo Mincigrucci, Igor Neri, Flavio Travasso, Helios Vocca, and Luca Gammaitoni. "Hybrid autonomous transceivers." In *Proc. EDERC2012: 5th European DSP Education and Research Conference, Amsterdam*. 2012.
[3] Cope, Richard C., and Yury Podrazhansky. "The art of battery charging." In *Battery Conference on Applications and Advances, 1999. The Fourteenth Annual*, pp. 233-235. IEEE, 1999.
[4] Sepasi, Saeed, Reza Ghorbani, and Bor Yann Liaw. "Inline state of health estimation of lithium-ion batteries using state of charge calculation." *Journal of Power Sources* 299 (2015): 246-254.
[5] Santhanagopalan, Shriram, and Ralph E. White. "State of charge estimation using an unscented filter for high power lithium ion cells." *International Journal of Energy Research* 34, no. 2 (2010): 152-163.
[6] Piller, Sabine, Marion Perrin, and Andreas Jossen. "Methods for state-of-charge determination and their applications." *Journal of power sources* 96, no. 1 (2001): 113-120.
[7] Pang, Shuo, Jay Farrell, Jie Du, and Matthew Barth. "Battery state-of-charge estimation." In *American Control Conference, 2001. Proceedings of the 2001*, vol. 2, pp. 1644-1649. IEEE, 2001.
[8] Anbuky, Adnan H., and Phillip E. Pascoe. "VRLA battery state-of-charge estimation in telecommunication power systems." *IEEE Transactions on Industrial Electronics* 47, no. 3 (2000): 565-573.
[9] Araujo, P., G. Astray, J. A. Ferrerio-Lage, J. C. Mejuto, J. A. Rodriguez-Suarez, and B. Soto. "Multilayer perceptron neural network for flow prediction." *Journal of Environmental Monitoring* 13, no. 1 (2011): 35-41.
[10] He, Wei, Nicholas Williard, Chaochao Chen, and Michael Pecht. "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation." *International Journal of Electrical Power & Energy Systems* 62 (2014): 783-791.
[11] Basheer, I. A., and M. Hajmeer. "Artificial neural networks: fundamentals, computing, design, and application." *Journal of microbiological methods* 43, no. 1 (2000): 3-31.
[12] Lajara, Rafael J., Juan J. Perez-Solano, and Jose Pelegri-Sebastia. "A method for modeling the battery state of charge in wireless sensor networks." *IEEE Sensors Journal* 15, no. 2 (2015): 1186-1197.
[13] Mitra, Sushmita, Sankar K. Pal, and Pabitra Mitra. "Data mining in soft computing framework: a survey." *IEEE transactions on neural networks* 13, no. 1 (2002): 3-14.
[14] Phil Woodland, Module 4F10 Statistical Pattern Processing Statistical, Cambridge University Engineering Department, 2012.
[15] Bijeni, Z., Manger, R., Pulji, K., Multilayer perceptrons data compression, 26 (2007) 45–62.
[16] Gardner, Matt W., and S. R. Dorling. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences." *Atmospheric environment* 32, no. 14 (1998): 2627-2636.
[17] Kros, John F., Mike Lin, and Marvin L. Brown. "Effects of the neural network s-Sigmoid function on KDD in the presence of imprecise data." *Computers & operations research* 33, no. 11 (2006): 3136-3149.
[18] Bianchini, Monica, Marco Gori, and Marco Maggini. "On the problem of local minima in recurrent neural networks." *IEEE Transactions on Neural Networks* 5, no. 2 (1994): 167-177.
[19] Li, Minqiang, Jin Tian, and Fuzan Chen. "Improving multiclass pattern recognition with a co-evolutionary RBFNN." *Pattern Recognition Letters* 29, no. 4 (2008): 392-406.
[20] Chang, Gary W., Cheng-I. Chen, and Yu-Feng Teng. "Radial-basis-function-based neural network for harmonic detection." *IEEE Transactions on Industrial Electronics* 57, no. 6 (2010): 2171-2179.
[21] American Institute of CPAs, Business Intelligence, AICPA Website, 2015.
[22] Likas, Aristidis. "Probability density estimation using artificial neural networks." *Computer physics communications* 135, no. 2 (2001): 167-175.