An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries

Article · January 2010

CITATIONS
90

READS
1,050

5 authors, including:

Abhinav Saxena
GE Global Research
116 PUBLICATIONS **3,479** CITATIONS
SEE PROFILE

Kai Goebel
Palo Alto Research Center
411 PUBLICATIONS **8,437** CITATIONS
SEE PROFILE

Bhaskar Saha
NIO USA
65 PUBLICATIONS **3,298** CITATIONS
SEE PROFILE

An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries

Jie Liu[1], Abhinav Saxena[2], Kai Goebel[3], Bhaskar Saha[4], and Wilson Wang[5]

[1] *Dept. of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON, K1S 5B6, Canada*
jliu@mae.carleton.ca

[2] *Stinger Ghaffarian Technologies at NASA Ames Research Center, Moffett Field, CA 94035, USA*
abhinav.saxena@nasa.gov

[3] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
kai.goebel@nasa.gov

[4] *Mission Critical Technologies at NASA Ames Research Center, Moffett Field, CA 94035, USA*
bhaskar.saha@nasa.gov

[5] *Dept. of Mechanical Engineering, Lakehead University, 955 Oliver Road, Thunder Bay, ON, P7B 5E1, Canada*
wilson.wang@lakeheadu.ca

ABSTRACT

Prognostics is an emerging science of predicting the health condition of a system (or its components) based upon current and previous system states. A reliable predictor is very useful to a wide array of industries to predict the future states of the system such that the maintenance service could be scheduled in advance when needed. In this paper, an adaptive recurrent neural network (ARNN) is proposed for system dynamic state forecasting. The developed ARNN is constructed based on the adaptive/recurrent neural network architecture and the network weights are adaptively optimized using the recursive Levenberg-Marquardt (RLM) method. The effectiveness of the proposed ARNN is demonstrated via an application in remaining useful life prediction of lithium-ion batteries.

1. INTRODUCTION

Prognostics and health management (PHM) is an enabling discipline of technologies and methods with the potential capability of solving reliability problems that have been manifested due to complexities in design, manufacturing, environmental and operational conditions (Pecht and Jaai, 2010). Although a key component of PHM is prognostics, it is also the least mature element under development (Goebel *et al.*, 2008). Prognostics entails the use of the current and previous system (or component) states to predict the future states of a dynamic system, and the forecast information can be used for: 1) condition monitoring to provide a reliable alarm before a fault reaches critical levels, so as to prevent machinery performance degradation, malfunction, or even catastrophic failures (Liu *et al.*, 2010); and 2) schedule of repairs and maintenance in advance (Jardine *et al.*, 2006).

Traditionally, prognostics can be implemented using either model-based approaches or data-driven approaches (Schwabacher and Goebel, 2007). Model-based approaches typically involve building mathematical functions to describe the physics of the system and failure modes and thus incorporate physical understanding of the system into the estimation of remaining useful life (RUL) (Saha and Goebel, 2009; Orchard *et al.*, 2005; Oppenheimer and Loparo, 2002; Adams, 2002; Chelidze and Cusumano, 2004). However, an accurate analytical model is usually difficult to derive for a complex dynamic system, especially when the system operates under noisy and/or uncertain environments (e.g., most real-world industrial applications). Data-driven approaches use pattern recognition and machine learning to establish the connection between operational data and system health, thereby enabling changes in system states to be detected (Tse and Atherton, 1999; Gupta and Ray, 2007; Goebel *et al.*, 2008). Classical data-driven

approaches for nonlinear system prediction also include the use of stochastic models such as the autoregressive (AR) model (Groot and Wurtz, 1991), the threshold AR model (Tong and Lim, 1980), the bilinear model (Subba Rao, 1981), the projection pursuit (Friedman and Stuetzle, 1981), the multivariate adaptive regression splines (Friedman, 1991), and the Volterra series expansion (Brillinger, 1970). These approaches rely on past patterns of the degradation of similar systems to project future system states and are more appropriate when the understanding of first principles of system operation is not comprehensive and/or when the system is sufficiently complex that developing an accurate model is prohibitively expensive. Since the last decade, more research interests in system state forecasting have shifted to the use of data-driven flexible models such as neural networks (Tse and Atherton, 1999; Atiya *et al*., 1999) and neural fuzzy systems (Jang, 1993; Korbicz, 2004).

Although data-driven flexible-model approaches have demonstrated some superior properties to other classical forecasting tools, advanced research still needs to be conducted before it can be applied to real-world industrial applications (Wang, 2007): 1) improving the convergence properties, especially for multi-step predictions; and 2) enhancing the adaptive capability to accommodate time-varying system conditions. Consequently, the aim of this paper is to develop an adaptive recurrent neural network (ARNN) for prognostic applications. The developed ARNN predictor is new in the following aspects: 1) a multi-layer state adaptive and recurrent paradigm is proposed for multi-step forecasting, such that the information from the previous steps could be properly utilized to improve the forecasting accuracy; 2) a recursive Levenberg-Marquardt (RLM) algorithm is adopted to adaptively optimize the network parameters to accommodate time-varying system conditions; 3) the proposed ARNN is implemented for the RUL prediction of lithium-ion batteries. The useful life of a battery is defined by the time until which a battery is able to maintain a minimum charge capacity when fully charged. In this experiment data were collected at fixed time intervals (every four weeks), therefore, RUL is calculated as the time remaining until the battery charge retention capacity drops to 70% of its full capacity observed at the beginning.

The organization of this paper is as follows. The developed ARNN predictor is described in Section 2. The effectiveness of this ARNN predictor is demonstrated in Section 3 via an application in battery RUL prediction. Some conclusions and remarks are included in Section 4.

2. THE ADAPTIVE RECURRENT NEURAL NETWORK

The developed ARNN predictor includes two main components: the adaptive and recurrent neural network architecture and the network parameter optimization using a RLM method, which will be described in the following discussions.

2.1 The ARNN Network Architecture

The network architecture of the proposed ARNN predictor is schematically shown in Figure 1. It is a hybrid model of both direct, adaptive, and recursive predictions; that is, the ARNN predictor is constructed based on a feed-forward multi-layer neural network with adaptive and recurrent feedback links from user-selected nodes. The connection topology may vary from one application to another, which can be conveniently redefined by users. The adaptive feedback links represent temporal information spatially while the recurrent feedback links deal with time explicitly. The feedback units copy the activations of the nodes from the previous time step, and allow the network to memorize the clues from the past, which forms a reasoning base for current processing.

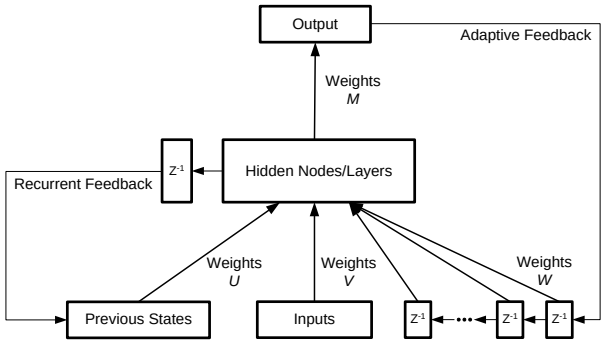

Figure 1: The network architecture of the developed ARNN predictor.

In the developed ARNN predictor, the sigmoid activation functions (AF) are used for the nodes in the hidden layers and the linear AF is chosen for the nodes in the output layer; correspondingly the inputs $x_i(n)$, the previous states from the hidden layers $y_h(n-1)$, and the previous outputs $y_o(n-l)$ should be properly normalized to [0 1] or [-1 1] based on the type of the sigmoid AFs employed. For n-th sample, the output of the j-th node in the first hidden layer is given by

$$y_j(n) = f(net_j), \tag{1}$$

$$net_j(n) = \sum_h u_{jh} y_h(n-1) + \sum_i v_{ji} x_i(n) + \sum_l w_{jl} y_o(n-l) + \theta_j, \tag{2}$$

where θ_j is the bias; U, V and W are the network weight matrices that correspond to the previous states, the inputs and the previous outputs, respectively; l = 1, 2, …, L, where L denotes the maximum output feedback depth; $f(\cdot)$ is the chosen AF. The connections between the first hidden layer and the network output can take any form of multi-layer perceptron (MLP), denoted by

$$y_o(n) = g(y(n)), \tag{3}$$

where $g(\cdot)$ represents the nonlinear mapping of this MLP. The proposed state adaptive and recurrent feedbacks include temporal aspects into network such that the useful information from the previous steps could be properly utilized for a more accurate estimation on the future system states. The network weights are optimized by the developed RLM method (Liu *et al.*, 2009) as discussed in the following section.

2.2 The ARNN Network Training

The ARNN predictor as developed in Section 2.1 should be properly trained to generate an optimal input/output mapping. For offline training, representative training data should cover all the possible application conditions (Liu *et al.*, 2009). Such requirement is difficult to achieve in real-world applications because most machines/systems operate in noisy and/or uncertain environments. Usually, the classical forecasting schemes are employed for time-invariant systems or systems with slowly varying model parameters. However, system's dynamic characteristics may change suddenly, for example, just after repairs or regular maintenance. In this section, a RLM method is adopted to adaptively optimize the ARNN weights to accommodate time-varying system conditions.

For a training data pair, $\left\{x^{(n)} \quad y_d^{(n)}\right\}^T$, $x^{(n)}$ is a vector that contains the network inputs as well as the previous states and outputs; n = 1, 2, …, N; N is the total number of training data sets; $y_d^{(n)}$ is the desired output. The objective function for all the N training data sets is defined as

$$E = \frac{1}{2}\sum_{n=1}^{N} \lambda^{N-n}\left(y_o^{(n)} - y_d^{(n)}\right)^2 = \frac{1}{2}\sum_{n=1}^{N} \lambda^{N-n}\varepsilon(n,\Theta)^2 \tag{4}$$

where $\lambda \in (0 \quad 1]$ is the forgetting factor, and $\lambda \in (0.9 \quad 1]$ is usually used to avoid possible convergence instability; Θ denotes all the network weights and $\Theta \in R^{q\times 1}$; ε is the point forecasting error.

The computation of the inverse Hessian matrix in the classical LM is time-consuming and impractical for real-time applications (Jang *et al.*, 1997). In this case, a remedy is to apply the matrix inversion lemma to avoid the direct inversion of Hessian matrix (Ljung and Soderstrom, 1983). Instead of adding the $q \times q$ matrix $\mu(n)I$ at each step, only one diagonal element is added at a time. As a result, the weights of the ARNN are recursively updated by

$$\begin{aligned}\hat{\Theta}(n) &= \hat{\Theta}(n-1) \\ &\quad - R(n-1)\varphi(n,\hat{\Theta}(n-1))^T \varepsilon(n,\hat{\Theta}(n-1))\end{aligned} \tag{5}$$

$$\begin{aligned}R(n) = \frac{1}{\lambda}\{&R(n-1) \\ &- R(n-1)\varphi^*(n,\hat{\Theta}(n-1))^T[\lambda\Lambda^{*-1}(n) \\ &+ \varphi^*(n,\hat{\Theta}(n-1))R(n-1)\varphi^*(n,\hat{\Theta}(n-1))^T]^{-1} \\ &\varphi^*(n,\hat{\Theta}(n-1))R(n-1)\}\end{aligned}$$

where $\hat{\Theta}(n)$ is the estimate of $\Theta(n)$ at sample instant n; $\varphi = d\varepsilon/d\Theta$ is the Jacobian matrix; R is the inverse of an approximated Hessian matrix, and $R(0)$ is chosen as an identity matrix $\alpha_N I$ with a constant $\alpha_N \in [10^3 \quad 10^5]$; φ^* and Λ^* are, respectively, given as

$$\varphi^*(n,\hat{\Theta}(n-1))^T = \begin{bmatrix} & 0 \\ & \vdots \\ & 0 \\ \varphi(n,\hat{\Theta}(n-1))^T & 1 \\ & 0 \\ & \vdots \\ & 0 \end{bmatrix} \leftarrow \text{position } n \text{ (mod } q) +1, \tag{7}$$

$$\Lambda^*(n) = \begin{bmatrix} 1 & 0 \\ 0 & q\mu(n) \end{bmatrix}. \tag{8}$$

The RLM parameter $\mu(n)$ conversely affects the trust region radius (Jang *et al.*, 1997), which should be adaptively modified based on the performance of the updated weights. The following strategy is applied in this work: the initial value $\mu(1)$ is set to 0.1; if the objective function in Eq. (4) decreases as the weights are updated, reduce $\mu(n)$ by $\mu(n)/k$; otherwise, if the objective function increases as the weights are updated, enlarge the $\mu(n)$ by $k\mu(n)$, where k is a design parameter. In determining these LM parameters, a series of simulation tests have been conducted in terms

of training accuracy and convergence speed. $\mu(1) \in [0.01, 1]$ have been tested and $\mu(1) = 0.1$ is used as an initial value which will be updated in training. On the other hand, $k \in [1.01, 2.00]$ was explored in this work; a larger value of the updating parameter k may lead to training instability, whereas a smaller k slows down the training process. $k = 1.15$ is used in this work.

3. PERFORMANCE EVALUATION

The effectiveness of the proposed ARNN predictor is shown for a battery health management application for RUL prediction. We compare its performance with the classical recurrent neural network (RNN) and the recurrent neural fuzzy system (RNF) (Liu *et al.*, 2009) in more details in addition to making comparisons with other data driven algorithms that have been applied to the battery health dataset in earlier works.

3.1 Application: Battery RUL Prediction

Batteries are widely used in various engineering and household systems. An effective prognostic tool is critically needed to predict the future capacity of a battery and reliably estimate its RUL. The forecast RUL information can be used not only for battery fault detection to prevent performance degradation of the related equipment, but also for scheduling battery recharge which is critical in many applications such as the emerging electric vehicles and aerospace industries. Both applications differ in the way they define RUL. For one, RUL is computed by tracking its charge retention capacity that tends to diminish with every recharge cycle, and for the other RUL is defined by the expected discharge of a battery within a charge cycle. While, both are critical applications, the experimental data used in this paper is suitable for prediction battery capacity only.

Dynamic models have been built for PHM of lithium ion batteries (Gao *et al.*, 2002). These models take into consideration nonlinear equilibrium potentials, rate and temperature dependencies, thermal effects, and transient power response. However, it still remains difficult to accurately predict the RUL of a battery using a model-based approach when environmental and load conditions changes; furthermore, modeling a lithium-ion battery from the first principles of the internal electrochemical reactions can be very tedious and computationally intractable. Developing data-driven methods that use sensor measurements to make these predictions provides alternative ways to make predictions in absence of good system model. In this work, we have implemented the developed data-driven ARNN predictor for the RUL prediction of lithium-ion batteries. The battery RUL is estimated in terms of capacity degradation (Goebel *et al.*, 2008; Saxena *et al.*, 2009). The failure threshold is generally defined by the manufacturer for a specific application. In this case, a lithium-ion battery is deemed to fail when its capacity $C/1$ fades by 30% of the rated value. The batteries' capacity is usually inaccessible to be gathered though sensor measurements; therefore, indirect measurements are employed to assess battery health. A lumped parameter model can be used that models the inside chemistry of the batteries though a simple electrical circuit and the change in resistive components of this circuit explains the reduction in battery capacity. Features extracted from sensor data of voltage, current, power, impedance, frequency, and temperature readings are used to estimate the internal parameters in the lumped-parameters battery model. The values of these internal parameters change with various aging and fault processes like plate sulfation, passivation, and corrosion (Goebel *et al.*, 2008). Specifically, parameters R_E and R_{CT} are tracked and employed for battery RUL prediction, where R_E denotes the electrolyte resistance and R_{CT} denotes the charge transfer resistance in the lumped-parameter model (Saha *et al.*, 2009). $(R_E + R_{CT})$ is typically inversely proportional to the capacity $C/1$ and can be estimated through the electrochemical impedance spectroscopy test. The data used in this study were from the second generation, Gen 2, 18650-size lithium-ion cells that were cycle-life tested at 60% state-of-charge (SOC) and temperature (25°C and 45°C) (Christophersen *et al.,* 2006).

3.2 Results and Discussion

The prediction algorithms developed in this study namely; the ARNN predictor, the classical RNN predictor and the RNF predictor are trained using the history $(R_E + R_{CT})$ data collected from multiple batteries of the same type at 25°C and 45°C, and are then employed to predict an unknown $(R_E + R_{CT})$ trajectory of the test battery collected at 45°C. Figure 2 shows the $(R_E + R_{CT})$ trajectories that were used for network training and testing. As can be seen from the figure below, these batteries take more than a year to age. Running such systems to failure in a lab environment could be a lengthy and rather costly process. Therefore, data from multiple batteries being aged in parallel was collected on a monthly basis. However, the $(R_E + R_{CT})$ data available in this work for network training are quite limited in the sense it has very few points (one per month or less than twenty per aging battery) to let the network train properly and hence could not be directly used for early RUL

predictions; thus data interpolation was applied to generate more data points for a proper network weights optimization.

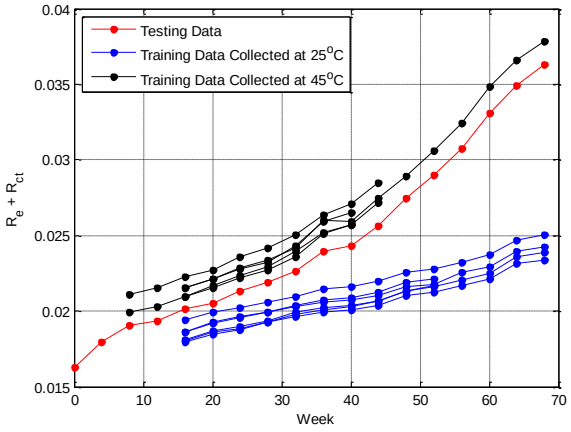

Figure 2: The battery parameter data used for network training and testing.

The learning process on the degradation patterns carried out from the history $(R_E + R_{CT})$ data is somewhat complicated. As seen in Figure 2, various trajectories obtained from experimental data are shifted that results in different target values for a given time input during learning phase. Therefore, the same input data (i.e., the previous degradation states), the training targets (i.e., the one-step-ahead state) may be significantly different, which confuses the learning of the predictors and thus necessitate more inputs and finer training scenarios. The initialization settings for the parameters associated with the proposed RLM learning algorithm are based on the literature, which is a common practice for most of learning algorithms (e.g., the initial settings on weights and learning rate in steepest gradient). During our testing phase, we found that the RLM algorithm is very stable in terms of convergence and repeatability when the proper ranges of learning parameters are specified and therefore impart higher degree of confidence in the prediction results.

In our implementation, the temperature was used as an additional network input. As indicated earlier that the experimental data was available from tests conducted at different temperatures and that the operational temperature significantly affects the capacity life of a battery, a prediction should take into account such operational environmental conditions. This facilitates prediction of battery capacity at different temperatures. Several initial runs are taken to examine the complexity (or nonlinearity) of the data so as to help define the structure of the related NN-based predictors. It must be noticed that designing a suitable network is usually considered an art with few guidelines available from the literature. For a fair comparison, the ARNN and RNN predictors were built with the same number of layers and nodes, that is, six nodes for the first hidden layer, eight nodes for the second hidden layer, and four nodes for the third hidden layer. The maximum adaptive feedback depth was set to four for the ARNN and RNN predictors and was limited to three for the RNF predictor (constrained by the structure, Liu *et al.*, 2009). The predictors are trained using the history $(R_E + R_{CT})$ data of the battery of the same type so as to capture the patterns on how the temperature $x_1(n)$, the current battery state $y_o(n)$, and the previous battery states $y_o(n-4)$, $y_o(n-3)$, $y_o(n-2)$, $y_o(n-1)$ affect the one-step-ahead battery state $y_o(n+1)$. After the learning step the trained network is used to predict the $(R_E + R_{CT})$ degradation trend on an unknown test trajectory. Multiple predictions were generated starting from week 20 to week 64 at an every four weeks interval. The predictor was run fifty times to obtain 50 $(R_E + R_{CT})$ trajectories to gather statistical significance that were then averaged to compute the final predicted trajectory. Using a linear mapping from $(R_E + R_{CT})$ to battery capacity (Saha and Goebel, 2008), the battery capacity from the predicted trajectories was derived. As mentioned earlier, the threshold for determining the RUL was chosen at 70% of the rated capacity.

To quantify the prognostic performance, the α-λ prognostic metric, as defined in (Saxena *et al.*, 2010), is employed. We first compare the predictions from these three predictors with the four algorithms that were previously used for the same dataset (Goebel *et al.*, 2008; Saha *et al.*, 2009; Saxena *et al.*, 2009). Figure 3 shows the predicted means from all seven different approaches that include the ARNN predictor, the classical RNN predictor, the RNF predictor, a Particle .Filter based predictor, Gaussian process regression (GPR), relevance vector machine (RVM), and a simple polynomial regression approach. Since the previous works compared only the point predictions, a comparison using the point predictions is made here. Comparing these approaches in Figure 3, following observations can be made: 1) the proposed ARNN predictor, the classical RNN predictor and the particle filtering (PF) based predictor provide a better prognostic performance than the other four predictors in terms of prediction accuracy; 2) the performance of the RNF predictor is not as good as the ARNN and RNN predictor; this may be attributed to the fact that the RNF predictor is constrained by the number of network inputs (usually less than five), and the complexity of the RNF structure (e.g., prognostic rules and consequent parameters) increases exponentially as

the number of inputs rises; based on a limited number of training data, the neural network based techniques could be more suitable in this prognostic application than the RNF to capture the common/hidden characteristics between the training data and testing data; 3) the proposed ARNN predictor outperforms all the other predictors in this test and performs within 90% accuracy of the true RUL since early predictions; 4) the test results also indicate that the prognostic performance improves when adaptive/recurrent feedbacks from the previous system states are properly utilized, 5) the implemented RLM approach shows a better learning capability than the classical training algorithms.

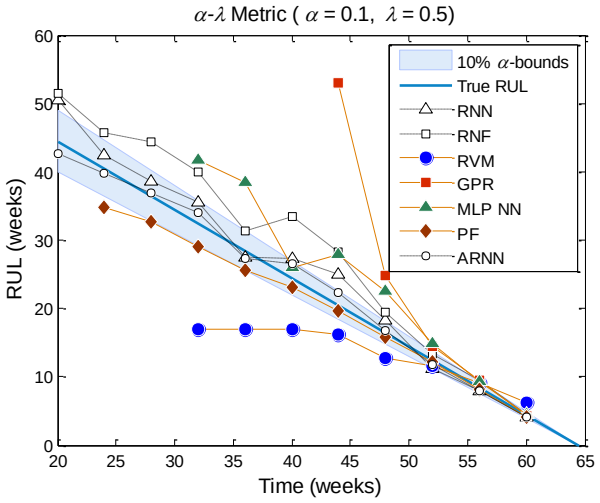

Figure 3: The battery RUL prediction comparison between seven different prediction methods.

To better understand the prediction variability (uncertainty), distributions of the RUL predictions over 50 program runs were analyzed using the enhanced prognostic metrics that were developed in (Saxena *et al.*, 2010). These results are plotted in Figures 4-6.

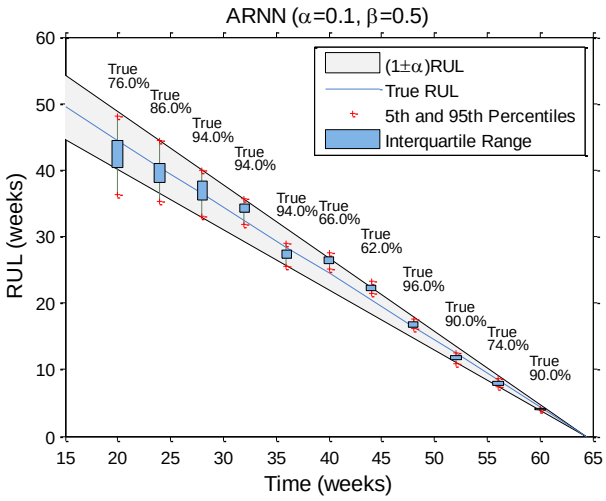

Figure 4: The distribution of the RUL predictions from the proposed ARNN predictor.
Also indicated on the plots are the percentages of RUL distribution overlap with the 10% accuracy cone and whether it satisfies the α-λ metric at all times according to the β-criterion with β =.05.It can be seen that the confidence interval of the RUL prediction from the proposed ARNN predictor is markedly smaller than the classical RNN predictor and the RNF predictor, which indicates that the ARNN predictor can provide a more reliable RUL prediction on this dataset.

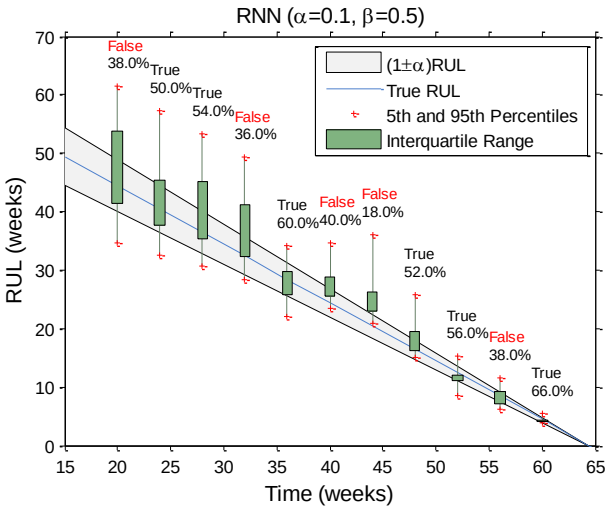

Figure 5: The distribution of the RUL predictions from the classical RNN predictor.

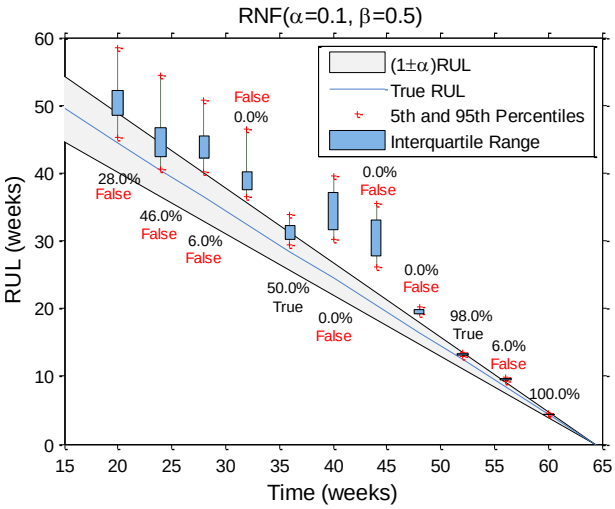

Figure 6: The distribution of the RUL predictions from the RNF predictor.

4. CONCLUSION

In this paper, an adaptive recurrent neural network (ARNN) has been developed for remaining useful life (RUL) prediction of lithium-ion batteries. This ARNN

predictor can enhance the prediction accuracy by properly utilizing the previous system states through adaptive/recurrent feedbacks. A recursive Levenberg-Marquardt (RLM) technique is implemented to adaptively optimize the ARNN weights so as to accommodate time-varying system conditions. The developed ARNN predictor was applied to predict the remaining useful life (RUL) of lithium-ion batteries. The results of this investigation have shown that the ARNN technique can effectively learn system states from a limited number of measurements to update the data-driven nonlinear prediction model. It outperforms the classical RNN and the RNF in battery RUL prediction.

ACKNOWLEDGMENT

This project was financially supported by Mission Critical Technologies (MCT) at NASA Ames Research Center (NARC), the Natural Sciences and Engineering Research Council of Canada (NSERC), and the University of California at Berkeley.

NOMENCLATURE

- n sample index (represents time for time series)
- i input index
- j node index in a network layer
- h layer index in a network
- l index to denote output feedback depth (look back distance in time)
- θ_j bias for the j^{th} node
- U weight matrix (u_{jh}) for previous states
- V weight matrix (v_{ji}) for network inputs
- W weight matrix (w_{jl}) for previous outputs
- $f(\cdot)$ activation function (AF)
- $g(\cdot)$ a nonlinear mapping between a hidden layer and the network output of a MLP
- $x_i(n)$ network inputs
- $y_h(n\text{-}1)$ previous states from hidden layer h
- $y_o(n\text{-}l)$ previous output looking back l time instants
- $x^{(n)}$ vector of network inputs and previous states and outputs
- $y_d^{(n)}$ desired output (target) for the network
- λ forgetting factor to weigh previous inputs differently $\lambda \in (0 \quad 1]$
- Θ all network weights, $\Theta \in R^{q\times 1}$
- ε point forecasting error
- $\hat{\Theta}(n)$ is the estimate of $\Theta(n)$ at sample instant n
- φ is the Jacobian given by $d\varepsilon/d\Theta$
- R is the inverse of an approximated Hessian matrix
- α_N is a constant to initialize R; $R(0) = \alpha_N I$
- $\mu(n)$ RLM parameter inversely proportional to trust region radius
- k a design parameter to scale μ
- R_E denotes the electrolyte resistance of the battery
- R_{CT} denotes the charge transfer resistance of the battery
- C/1 battery capacity
- α-λ accuracy bounds to satisfy while evaluating prediction performance at specific time instant, $\lambda(EoL\text{-}t_p)$, relative to ailing life of the components
- β required probability specified for RULs to fall within α bounds

REFERENCES

M. Pecht and R. Jaai (2010). A prognostics and health management roadmap for information and electronics-rich systems, *Microelectronics Reliability*, vol. 50, pp. 317-323, 2010.

D. E. Adams (2002). Nonlinear damage models for diagnosis and prognosis in structural dynamic systems, in *Proceedings of SPIE 4733*, pp. 180-191, 2002.

A. Atiya, S. El-Shoura, S. Shaheen, M. El-Sherif (1999). A comparison between neural-network forecasting techniques-case study: river flow forecasting, *IEEE Transactions on Neural Networks*, vol. 10, pp. 402-409, 1999.

D. R. Brillinger (1970). The identification of polynomial systems by means of higher order spectra, *Journal of Sound and Vibration*, vol. 12, pp. 301-313, 1970.

D. Chelidze and J. P. Cusumano (2004). A dynamical systems approach to failure prognosis, *Journal of Vibration and Acoustics*, vol. 126, pp. 2-8, 2004.

J. P. Christophersen, I. Bloom, J. P. Thomas, K. L. Gering, G., L. Henriksen, V. S. Battaglia, and D. Howell (2006). Advanced Technology Development Program for Lithium-Ion Batteries: Gen 2 Performance Evaluation Final Report, Information Bridge: DOE Scientific and Technical Information, [Online] Available INL/EXT-05-0091, 3 July 2006.

J. H. Friedman (1991). Multivariate adaptive regression splines, *Annals of Statistics*, vol. 19, pp. 1-141, 1991.

J. H. Friedman and W. Stuetzle (1981). Projection pursuit regression, *Journal of the American Statistical Association*, vol. 76, pp. 817-823, 1981.

L. Gao, S. Liu, and R. A. Dougal (2002). Dynamic lithium-ion battery model for system simulation, *IEEE Transactions on Components and Packaging Technologies*, vol. 25, no. 3, pp. 495-505, 2002.

K. Goebel, B. Saha, A. Saxena, J. R. Celaya, and J. P. Christophersen (2008). Prognostics in battery health management, *IEEE Instrumentation &*

Measurement Magazine, pp. 33-40, August 2008.

K. Goebel, B. Saha, and A. Saxena (2008). A comparison of three data-driven techniques for prognostics, in *Proceedings of the 62nd Meeting of the Society for Machinery Failure Prevention Technology (MFPT)*, pp. 119-131, Virginia Beach, VA, May 2008.

C. Groot and D. Wurtz (1991). Analysis of univariate time series with connectionist nets: A case study of two classical examples, *Neurocomputing*, vol. 3, pp. 177-192, 1991.

S. Gupta and A. Ray (2007). Real-time fatigue life estimation in mechanical structures, *Measurement Science and Technology*, vol. 18, pp. 1947-1957, 2007.

J. Jang (1993). ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 23, pp. 665-685, 1993.

J. Jang, C. T. Sun, and E. Mizutani (1997). Neuro-Fuzzy and Soft Computing: A computational approach to learning and machine intelligence, Prentice Hall, pp. 129-193, 1997.

J. Jang, C. Sun, and E. Mizutani (1997). *Neuro-fuzzy soft computing*, Upper Saddle River, NJ: Prentice-Hall, 1997.

A. Jardine, D. Lin, D. Banjevic (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing*, vol. 20, pp. 1483-510, 2006.

J. Korbicz (2004). *Fault Diagnosis: Models, Artificial Intelligence, Applications*, Springer, Berlin, 2004.

J. Liu, W. Wang, and F. Golnaraghi (2009). A multi-step predictor with a variable input pattern for system state forecasting, *Mechanical Systems and Signal Processing*, vol. 23, pp. 1586-1599, 2009.

J. Liu, W. Wang, and F. Golnaraghi (2010). An enhanced diagnostic scheme for bearing condition monitoring, *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 309-321, 2010.

L. Ljung and T. Soderstrom (1983). *Theory and practice of recursive identification. Cambridge*, MA: MIT Press, 1983.

C. H. Oppenheimer and K. A. Loparo (2002). Physically based diagnosis and prognosis of cracked rotor shafts, in *Proceedings of SPIE 4733*, pp. 122-132, 2002.

M. Orchard, B. Wu, and G. Vachtsevanos (2005), "A Particle Filter Framework for Failure Prognosis", Proceedings of World Tribology Congress III, Washington DC, Sept. 12-16, 2005.

B. Saha, K. Goebel, and J. Christophersen (2009). Prognostics methods for battery health monitoring using a Bayesian framework, *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 291-296, 2009.

B. Saha and K. Goebel (2009). Modeling Li-ion Battery Capacity Depletion in a Particle Filtering Framework, in *Annual Conference of the PHM Society*, San Diego CA, 2009.

B. Saha and K. Goebel (2008). Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques, in *Proceedings of IEEE Aerospace Conference*, Big Sky MT, 2008.

A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel (2009). Evaluating algorithm performance metrics tailored for prognostics, in *Proceedings of IEEE Aerospace Conference*, Big Sky MT, 2009.

A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel (2010). Metrics for Offline Evaluation of Prognostics Performance", International Journal of Prognostics and Health Management, Vol.1(1), pp. 20, 2010.

M. Schwabacher and K. Goebel (2007). A survey of artificial intelligence for prognostics, *AAAI Fall Symposium*, 2007.

T. Subba Rao (1981). On the theory of bilinear time series models, *Journal of the Royal Statistical Society*, vol. 43, pp. 244-255, 1981.

H. Tong and K. S. Lim (1980). Threshold autoregression, limited cycles and cyclical data, *Journal of the Royal Statistical Society*, vol. 42, pp. 245-292, 1980.

P. Tse and D. Atherton (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks, *Journal of Vibration and Acoustics*, vol. 121, pp. 355-362, 1999.

W. Wang (2007). An adaptive predictor for dynamic system forecasting, *Mechanical Systems and Signal Processing*, vol. 21, pp. 809-823, 2007.

Jie Liu received his B.Sc. in Mechanical Engineering (Precision Instruments) from Tianjin University, China, in 1998, his M.Sc. in Control Engineering from Lakehead University, Canada, in 2005, and his Ph.D. in Mechanical Engineering from the University of Waterloo, Canada, in 2008. He worked as a postdoctoral fellow in Mechanical Engineering at UC Berkeley for one and half years, before he took the current position as an Assistant Professor in the Department of Mechanical and Aerospace Engineering at Carleton University, Canada. His interdisciplinary research interests include vibration and system control, intelligent mechatronic systems, signal processing, and machinery condition monitoring.

Abhinav Saxena is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence NASA Ames Research Center, Moffett Field, CA. His research involves developing prognostic algorithms and methodologies to standardize prognostics that include performance evaluation and requirement specification for prognostics of engineering systems. He is a PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta. He earned his B.Tech in 2001 from Indian Institute of Technology (IIT) Delhi, and Masters Degree in 2003 from Georgia Tech. Abhinav has been a GM manufacturing scholar and is also a member of IEEE, AIAA, AAAI and ASME.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universität München, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is a senior scientist at NASA Ames Research Center where he leads the Diagnostics & Prognostics groups in the Intelligent Systems division. In addition, he directs the Prognostics Center of Excellence and he is the Associate Principal Investigator for Prognostics of NASA's Integrated Vehicle Health Management Program. He worked at General Electric's Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds eleven patents and has published more than 100 papers in the area of systems health management.

Bhaskar Saha received his Ph.D. from the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, USA in 2008. He received his M.S. also from the same school and his B. Tech. (Bachelor of Technology) degree from the Department of Electrical Engineering, Indian Institute of Technology, Kharagpur, India. He is currently a Research Scientist with Mission Critical Technologies at the Prognostics Center of Excellence, NASA Ames Research Center. His research is focused on applying various classification, regression and state estimation techniques for predicting remaining useful life of systems and their components, as well as developing hardware-in-the-loop testbeds and prognostic metrics to evaluate their performance. He has been an IEEE member since 2008 and has published several papers on these topics.

Wilson Wang received his M. Eng. in industrial engineering from the University of Toronto (Toronto, Ontario, Canada) in 1998 and Ph.D. in mechatronics engineering from the University of Waterloo (Waterloo, Ontario, Canada) in 2002, respectively. From 2002 to 2004, he was employed as a senior scientist at Mechworks Systems Inc. He joined the faculty of Lakehead University in 2004, and now is an associate professor in the Department of Mechanical Engineering. His research interests include signal processing, artificial intelligence, machinery condition monitoring, time series forecasting, intelligent control, and bioinformatics.