Remaining Useful Life Estimation for Systems with Abrupt Failures

Wei Huang[1], Hamed Khorasgani[2], Chetan Gupta[3], Ahmed Farahat[4], and Shuai Zheng[5]

[1,2,3,4,5] *Industrial AI Laboratory, Hitachi America Ltd., Santa Clara, CA, USA*
wei.huang@hal.hitachi.com
hamed.khorasgani@hal.hitachi.com
chetan.gupta@hal.hitachi.com
ahmed.farahat@hal.hitachi.com
shuai.zheng@hal.hitachi.coms

ABSTRACT

Data-driven Remaining Useful Life (RUL) estimation for systems with abrupt failures is a very challenging problem. In these systems, the degradation starts close to the failure time and accelerates rapidly. Normal data with no sign of degradation can act as noise in the training step, and prevent RUL estimator model from learning the degradation patterns. This can degrade RUL estimation performance significantly. Therefore, it is critical to identify degradation mode during the training step. Moreover, in the application step, predicting RUL when the system is in normal mode and is not showing any sign of degradation can generate inaccurate estimations, and reduce faith in the model. In this paper, we propose a new RUL estimation method that incorporates an early degradation mode detection step to automatically identify the earliest point of time at which the degradation starts to happen. When the degradation mode is detected, a Long Short Term Memory (LSTM) neural network is applied to predict system RUL. As a case study, we apply the proposed method for RUL estimation in 2018 PHM Data Challenge. The case study demonstrates that our solution achieves more accurate RUL estimation compared to several baseline methods.

1. INTRODUCTION

Remaining Useful Life (RUL) is the remaining time that a component or system can function in with the required performance (Si et al., 2011). RUL estimation is a crucial element in condition-based maintenance, prognostics and health management systems which can improve product quality, reduce cost and downtime in different industries such as manufacturing processes, transportation systems, and power generation plants. In prognostics, the degradation of a component or system is typically a non-linear function of several parameters such as operation environment, and work load. Normally, a system or component is in a healthy operating condition in early stage of use. System performance starts to degrade after operating for a period of time. The degradation process accelerates over time till a complete breakdown occurs. Degradation curves usually represent transitions from a roughly constant value to a linearly decreasing curve towards the end of life. The knee in the curve reflects the point when the degradation starts (Heimes, 2008).

Generally, there are two broad categories of approaches for RUL estimation: 1) physics-based (model-based) approaches and 2) data-driven approaches. Physics-based approaches use domain knowledge and basic principles of physics to model degradation processes such as fatigue crack growth, battery degradation, and corrosion. Sensor measurements are typically used to estimate the degradation model parameters and update the model overtime (Khorasgani et al., 2013). Model-based RUL estimation results are easy to understand and justified. However, for complex systems with several components, it is often difficult and expensive to generate degradation models. For these systems, data-driven methods, which rely purely on available past observed data and statistical models, can be used as an alternative solution.

Among data-driven methods, deep neural networks are widely applied for sequence classification and prediction problems. C. Zhang et al. (2017) used a multi-layer neural network for RUL estimation on a real-world dataset with multiple sensor measurements. LSTM networks are a special kind of Recurrent Neural Network (RNN) which can maintain long-term memories. Due to their ability to learn long term dependencies, LSTM networks have been widely demonstrated to be useful for learning sequences with longer term patterns. Multiple research groups have used LSTM for RUL estimation (Hsu & Jiang, 2018; Zheng et al., 2017; Yuan et al., 2016; J. Zhang et al., 2018). Zheng et al. (2017) proposed an LSTM networks integrated with fully connected neural networks (NNs) and their results show their model outperforms

other deep learning methods in RUL estimation on multiple datasets.

For the abrupt failures, the degradation mode starts very close to the failure. RUL estimation is very challenging when we have abrupt failures. In the learning step, the large amount of normal data can hide the degradation pattern in the dataset. In the implementation step, making RUL estimation while the system is still in a healthy operating mode and the sensors show no sign of degradation can lead to meaningless estimations and damage the credibility of the RUL estimator model. In this work, we design an early fault detection module to detect system degradations in the early stages. When the degradation modes are detected, we apply the LSTM networks structure proposed by Zheng et al. (2017) to estimate the system RUL.

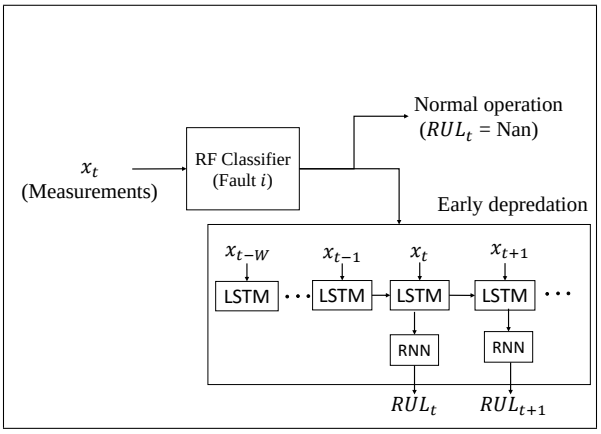

Figure 1. RUL estimation approach.

Figure 1 illustrate our RUL estimation methodology for systems with abrupt failures. In the first step, we use a Random Forest (RF) classifier to identify the system operating mode as 1) normal or 2) degrading. When the system is in normal operation, it is not feasible to predict RUL. Therefore, we do not report a numerical value for RUL. When the system is degrading, we use the LSTM networks with fully connected NNs to estimate the system RUL. The rest of this paper is organized as follows. Section 2 presents the definitions and the problem formulation. Our early fault detection method is presented in Section 3. Section 4 presents our RUL estimation methodology. Section 5 presents the experimental results on an ion milling chamber system dataset. Section 6 presents the conclusions of the paper.

2. PROBLEM FORMULATION

Generally, systems are designed for an specific purpouse, e.g., a battery is designed to store a pre-specified amount of electrical power. We define the system useful life, the period of time the system can function with an acceptable performance.

Definition 2.1 ***System useful life*** *is a period of time during which the system is usable for the purpose it was designed.*

In practice, system performance degrades after a period of time. We define the degradation process as

Definition 2.2 ***Degradation process*** *is a period of time during which the system performance declines.*

Gradual decrease in the capacitance of an electric battery to store power is an example of a system in the degradation mode. The degradation mode eventually leads to the system complete failure. The time of failure is called system End of Life (EOL). For a given current time point, t, the system RUL is the time interval between t and the EOL. In this paper, we call a failure mode abrupt failure when the degradation mode is significantly shorter than the system useful life.

Definition 2.3 ***Systems with abrupt failures*** *are systems with degradation mode much shorter than the system useful life.*

As we mentioned earlier, RUL estimation is not feasible before the degradation mode. For systems with abrupt failures, the relatively short period of degradation mode leaves limited time for operators to plan the rest of the mission and take required actions before the EOL, e.g., driving to a repair shop before the car dies. Therefore, accurate RUL estimation during the degradation mode is even more critical for these systems.

3. EARLY FAULT DETECTION

Detecting system degradation in early stages is a crucial step for accurate RUL estimation in our proposed method. However, early fault detection is not a trivial task. It is challenging to extract features sensitive to the faults. Moreover, we usually do not have access to sufficient fault data for training. In this section we will address these problems.

3.1. Features

Feature extraction is the most critical step in designing a diagnostic algorithm. We can categorize the features for fault detection and isolation into three main groups (Khorasgani et al., 2018):

- *Sensor measurements:* primarily, sensor data can be used as the set of features for fault detection and isolation.
- *Domain knowledge features:* domain experts can identify important features for detecting and isolating each fault. Moreover, they can help to define new features by providing critical information about nominal behavior of each measurement with respect to others.
- *Physics-based residuals:* the set of system equations can be used to generate residuals which represent analytical redundancy relations among measurements during nominal operation. When a fault occurs, residuals can capture inconsistency among the measurements.

A combination of sensor measurements, knowledge-based features, and physics-based residuals can be used as the set of features for fault diagnosis. In the case study, we use sensor measurements with a physics-based residual for early fault detection.

3.2. Training

RF classifiers, like most other classifiers, are designed to minimize the overall error rate, and therefore, their performance degrades when the training data is imbalanced. For imbalanced datasets, the classifiers tend to focus on the prediction accuracy of the majority classes, which often leads to poor results for the minority classes. This is a huge problem for early fault detection and isolation because even though the goal is to detect faults as soon as possible, the majority of training data are normal data. There are different approaches in the data level and algorithm level to address this problem. At the algorithm level, typically the cost functions are modified to represent higher penalties for misclassification of minority class data points. At the data level, up-sampling of minority classes or down-sampling of majority classes are the two main solutions in the literature.

Unlike algorithm level methods, the data level solutions are independent from classification algorithms and can be used with any classifier. By down-sampling the majority classes to make the number of samples in these classes close to the rarest classes in the dataset, we can lose significant amount of information. On the other hand, up-sampling the minority classes increases the computational complexity and can lead to overfitting. In this paper, we apply a hybrid approach. We first up-sample the fault data points, we then randomly select a subset of normal points equal to the number of oversampled fault samples.

4. RUL ESTIMATION

Normally, a system or component is operating at its healthy condition in early stages of operation. In the systems with abrupt failures, the system performance starts to degrade when a fault occurs in the system. For these systems, it is unfeasible to make precise RUL estimation from the beginning. To address this problem, we used RF classifiers to design an early degradation detection module in the previous step. In this section, we present a deep learning model for RUL estimation in a reasonable time frame before the end of life. This section includes the basic structure of our deep neural networks module, data preparation, and model evaluation.

4.1. LSTM Networks for RUL Estimation

After degradation mode detection, we apply a deep learning method using the LSTM structure. We assume the dataset includes a set of devices with the same type of sensor measurements, where each device can have multiple run-to-fault data sequences. Consider one sequence of run-to-fault data which is a multivariate time series in the matrix form of $X = [X^1, X^2, ..., X^t, ..., X^T]$, where T is the time of the fault and each data sample is a D-dimensional vector $X^t = [{x_1}^t, {x_2}^t, ...,{x_D}^t]$ where $t = 1, 2, ..., T$. Our LSTM network structure with one sequence of run-to-fault sensor data is shown in Fig. 2. The sensor data sequences are the input variables for the model to predict the RUL at each sample time.

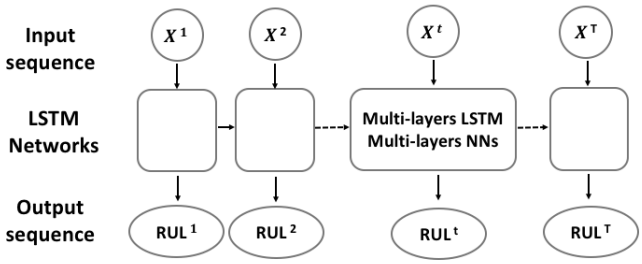

Figure 2. LSTM model structure for RUL estimation

The model is a composition of multiple layers of LSTMs followed by fully connected multiple layers of NNs. It uses this complex deep learning network to learn the long-term temporal dependencies vertically and complex relationship between different sensor measurements horizontally for different fault modes and possibly degradation modes. Dropout is a recently developed regularization technique (Hinton et al., 2012). Because of the complexity of the deep learning model, we include dropout (Srivastava et al., 2014) and L2 Regularization during the training to prevent overfitting. The combination of dropout and regularization can reduce generalization error significantly (Srivastava et al., 2014). The key idea for dropout is to randomly mask units (along with their connections) from the network so that these units won't influence the propagation during model training. In our model, we apply dropout to the input connections with the LSTM nodes for each LSTM layer. The dropout on the input means that the data on the input connection to each LSTM cell/block will be excluded, at a given probability, from node activation and weight updates.

In model training, the estimated RUL value (RUL_{Est}) is compared with the ground truth RUL value (RUL_{GT}) to calculate the Mean Square Error (MSE) as the model objective function.

$$MSE = \frac{1}{n}\sum(RUL_{Est} - RUL_{GT})^2 \qquad (1)$$

where n is the number of measurements in each sequence sample. We use RMSprop optimizer, an adaptive learning rate method, to train the model. RMSprop divides the learning rate by an exponentially decaying average of squared gradients. We also use early stopping to stop the training process

when there is no improvement on the validation data set.

4.2. Data Preparation for LSTM Model Training

Although LSTM is good at learning long-term dependencies. The natural encoder-decoder architecture for LSTM network, which uses a fixed-length internal representation for the input sequence, imposes a limitation in learning very long sequences. For datasets with high sampling rates, we can make RUL estimation in a moderately long and reasonable time frame by downsampling the dataset in the pre-processing step to summarize and shorten the long sequences without losing all the long term history information.

In the downsampling step, we select a sample point of every $Sample_R$ samples. We start the sampling by selecting the end sample (where fault happened) of the original long run-to-fault sequence and backtrack until the number of selected samples reaches a given maximum length, max_L. Then we slide the starting point from the end of the original sequence to one sample before the end and follow the same process to form another summarization of the original long run-to-fault sequence. Again, we repeat the process N times by sliding the starting point one sample before the previous one.

Iteratively, a set of shortened sequences are constructed which represent different summarization of the original long run-to-fault sequence. Compared with random sampling in a fixed window, our sampling approach keeps a fixed gap and is able to maintain similar performance degradation level between two adjacent samples in the sampled run-to-fault sequence. Meanwhile, this sampling approach may also be used as a type of data augmentation scheme in order to create many possible different input sequences from the original input sequence. This method improves the robustness of our model when available run-to-fault sequences for the training are limited.

4.3. Model Evaluation

We use the Root Mean Square Error (RMSE) and a Symmetric mean absolute percentage error (SMAPE) of estimation results defined in (2) to evaluate the performance of our estimation model.

$$SMAPE = 100\%\frac{1}{n}\sum\frac{|RUL_{Est} - RUL_{GT}|}{|RUL_{Est}| + |RUL_{GT}|} \qquad (2)$$

where n is the number of sequences in the data set. RMSE is a widely used evaluation metric for RUL estimation models which gives even penalties to the estimation errors no matter how close the estimations are to the fault. Mean absolute percentage error (MAPE) gives higher penalty when the estimation is close to the end of life. In real applications, accurate RUL estimation is typically more critical when the system is close to failure. Therefore, MAPE is a more reasonable metric to measure RUL estimation performance than RMSE. However, MAPE is not defined when the ground truth is zero. Therefore, we use SMAPE to overcome this issue.

5. CASE STUDY: 2018 PHM DATA CHALLENGE

In this section, we apply our RUL estimation method to the 2018 PHM Data Challenge[1].

5.1. Ion Milling System

The dataset includes 20 ion milling machine operating data. The Ion Milling process is a common approach for designing microwave circuits. The process uses an ion beam to remove excess materials from the wafers and creates desired patterns. A rotating fixture is used to rotate the wafer at different angles facing the ion beams. The wafer can also be shielded from the ion beams using a shutter mechanism. A Particle Beam Neutralizer (PBN) system is used to control the ion beam shape and ion distribution as it travels to the wafer surface. Each recipe can have different set of steps, and each step may require different configuration settings such as rotation speed, angles, beam current/voltages, etc. At each step, the system processes the wafer for a set amount of time.

The wafers are cooled by a helium/water system called flow-cool. The cooling system passes helium gas behind the wafer at a specified flow rate. The helium gas is indirectly cooled by a water system. The wafer and fixture o-ring separates the flow-cool gas from the ion mill vacuum chamber. Different types of failure can occur in the ion milling cooling system. In this paper, we focus on the following faults.

- *Fault mode 1* occurs when flow-cool pressure drops.
- *Fault mode 2* occurs when flow-cool pressure becomes too high.
- *Fault mode 3* represents flow-cool leakage.

5.2. System variables

In addition to time and runnum (the number of times a machine has been run), there are three types of variables for each machine.

1. *Categorical variables* such as recipe, and recipe step describe machine settings during the process. The number of combinations of different setting parameters is too large and therefore, it is not practical to learn a model for each operational setting. For example, the recipe parameter has over 500 different categorical values. Moreover, using these variables as inputs to our model can lead to overfitting. Among the categorical variables, we only use fixture shutter position in early fault detection and RUL estimation. Fixture shutter position values are $[0, 1, 2, 3, 255]$ in the dataset. We apply one-hot encoder to represent this categorical variable as binary vectors.

[1] See https://www.phmsociety.org/events/conference/phm/18/data-challenge

2. *Sensor variables* such as flow-cool flow rate and flow-cool pressure represent sensor measurements during system operation. Sensor variables can have a wide range of amplitudes. For example, the temperature values are typically much larger than the flow rate measurements. This can bias machine algorithms. To avoid this problem, variable standardization is typically the first pre-processing step. However, the sensor variables in our dataset have been anonymized and normalized. Therefore, we don't have to perform data normalization.
3. *Operating time variables* such as etch source usage and actual step duration measure the number of time different parts in the system have been used or the time duration for a particular step in the process. These variables represent system operation life and play an important role in RUL estimation.

5.3. Data pre-processing

As the first pre-processing step, we remove the samples with missing values. There are some periods of time when the machines were not operating. However, since the provided ground truth of the RUL is not based on operating time, we can see sudden jumps in RUL between system operations. In model training, we transfer the ground truth from time to fault, to the number of samples to fault. This removes the jumps between system operations. Since the sampling rate is roughly 4 seconds when system is operating, the number of samples to fault is an acceptable representation of system RUL. We split the raw sequence for each equipment into multiple run-to-fault sequences for each fault. Some run-to-fault sequences are very short with few measurements and some contain a large time gap which implies a large number of missing samples. We apply the following criteria to filter invalid run-to-fault data sequences.

1. We truncate each run-to-fault sequence with a large time gap.

2. The remaining sequences with more than M_{min} historical measurements are considered to be valid sequences for model training.

5.4. Using LSTM for RUL Estimation

In this section, we apply the LSTM model for RUL estimation for the 2018 PHM Data Challenge training data set. In the 2018 PHM Data Challenge training data set, we have multivariate time series data of 20 similar ion and wafer mill etch equipments. We assume that the three fault modes are independent and train three LSTM based RUL estimation models for each fault respectively. We use the sensor variables, operating time variables and fixture shutter position variable in the categorical variables as the variables for model training. After converting the fixture shutter position to five binary vectors with one-hot encoder, we have 21 parameters for each measurement.

Table 1. Number of sequences in model training and testing

Fault modes	Training	Testing
Fmode1	199	39
Fmode2	23	4
Fmode3	44	10

Table 2. Comparison of LSTM and other approaches for Fault mode 1 on Testing set.

Models	RMSE (seconds)	SMAPE
RFR	5294	29.27%
MLP	5004	28.10%
LSTM	1877	13.90%

Zheng et al. (2017) developed a network with two LSTM layers to learn the hidden relations between the measurement variables and discover the degradation patterns as the main indicators of RUL. For the 2018 PHM Data Challenge data set, we construct same RUL estimation model architecture for the three faults. The original model network has 4 hidden layers with 32 nodes in the first LSTM layer, 64 nodes in the second LSTM layer, 8 nodes in the third and 8 nodes in the forth layer. The models for the three faults are trained and tested separately. After training the three faults separately, we doubled the node number in two LSTM layers in the models associated with Fault mode 1 and Fault mode 2 for better performance.

Since it is impossible to make RUL estimation for very long time periods, the value of max_L can not be too large. Here, we set $max_L = 300$, $N = 15$ and $Sample_R = 15$ (roughly 1 sample/4 minutes). Therefore, $max_L \times Sample_R + N = 4515$ historical measurements (which covers roughly 5 hours) before the fault happened are taken for model training. It is a reasonable time frame that can represent the performance degradation period. We consider $M_{min} = 3000$. Roughly 80% of the run-to-fault sequences generated from the train data set are randomly selected for model training, and the remaining 20% is used for model testing. The number of sequences in training and testing data set generated from the 2018 PHM Data Challenge train data set is summarized in Table 1.

Different equipment have different number of failures. Some equipment have very few failures. If the number of failures is less than 4 times for an equipment, then all the failures are used for model training. In the training step, 90% of the run-to-fault sequences of the training set are randomly selected for model training and the remaining 10% are used for validation. The model for each fault is trained 10 times and the validation split is applied in each repetition. The training process is a non-convex optimization problem. For each fault

Table 3. Comparison of LSTM and other approaches for Fault mode 2 on Testing set.

Models	RMSE (seconds)	SMAPE
RFR	5567	30.16%
MLP	4113	23.78%
LSTM	2557	16.94%

Table 4. Comparison of LSTM and other approaches for Fault mode 3 on Testing set.

Models	RMSE (seconds)	SMAPE
RFR	5476	29.92%
MLP	5194	29.11%
LSTM	1469	11.74%

data set, we run the LSTM 10 times and record the parameters of the model with best estimation performance on the validation set.

We compare our LSTM method with other approaches including Multi-Layer Perceptron (MLP) and Random Forest Regression (RFR). We constructed the MLP network with three hidden layers. The number of nodes and other parameters are varied when training and the model with best performance is recorded. For RFR model, we implemented random search and grid search to find the optimal parameter set including the number of trees in the forest, the maximum depth of the tree, the number of features to consider when looking for the best split, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node and whether bootstrap samples are used for building the trees. The models are applied to the testing set. We show the RMSE and SMAPE of the estimation results on the testing set for different methods in Table 2, Table 3, and Table 4.

Two random examples of RUL estimations from the testing set for the three faults are shown in Fig. 3, Fig. 4 and Fig. 5. As shown in these figures, the absolute error of the predicted RUL are generally below 30 minutes throughout the 5 hours RUL monitoring. We notice that the LSTM method outperform the other RUL estimation methods. Moreover, we can see that the absolute error at the beginning is usually higher. The possible reason is that the historical data is limited at the beginning. We can also notice that the predicted RUL curve generated by our LSTM method for fault mode 2 (flow-cool pressure too high) is different from the predicted curves for the two other faults. The predicted RUL curves shown in Fig. 3 and Fig. 5 are smoothly tracking the real RUL curve. While the predicted curve for fault mode 2 as it is shown in Fig. 4 decrease slowly at the beginning and suddenly drops down. This indicates that the equipment performance degradation modes for different faults could vary and it is possible that the fault mode 2 is more abrupt than the other two faults.

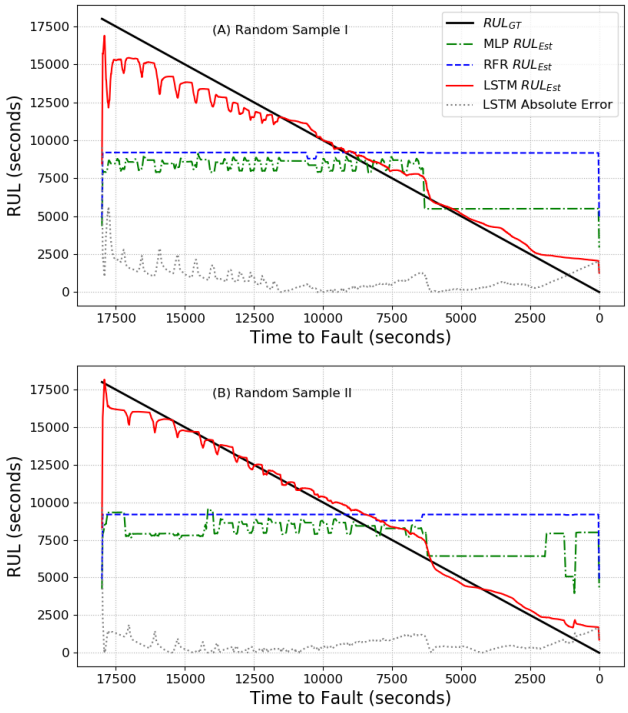

Figure 3. Two random samples of the RUL estimation results on testing data for Fault: flow-cool Pressure Too Low.

5.5. Early Degradation Detection

To apply our approach for RUL estimation in the entire dataset, we first develop an early fault detection module. We use the union of sensor measurements and the operating time variables as the primary set of features for early fault detection. Like the previous section, fixture shutter position is the only categorical variable that we use for fault detection. We apply one-hot encoding to convert this variable to binary vectors. In addition to the available variables in the dataset, we derive the following extra feature for early fault detection.

In general, the flow-cool flow rate (q) is a function of flow-cool pressure (p), $q = f(p)$. When a fault occurs this relationship can change. We use the normal data to learn the nonlinear relationship between flow-cool flow rate and flow-cool pressure.

$$q = \hat{f}(p), \tag{3}$$

We then use this model to estimate flow-cool flow rate as a function of flow-cool pressure at each operating point. The difference between the estimated flow-cool pressure and actual flow-cool pressure, r, is our additional feature for fault detection.

$$r = q - \hat{f}(p), \tag{4}$$

In the training step, we consider the data points with 5000s or less RUL as close to failure data points. Since there are

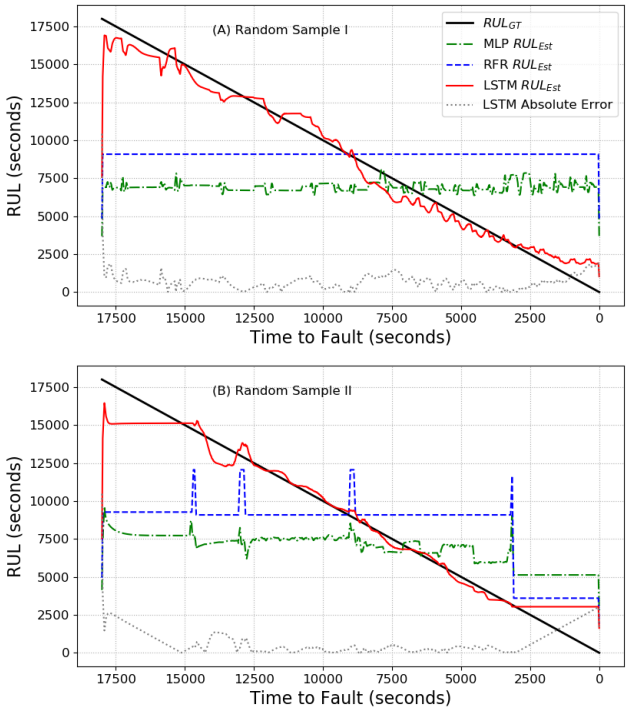

Figure 4. Two random samples of the RUL estimation results on testing data for Fault: flow-cool Pressure Too High.

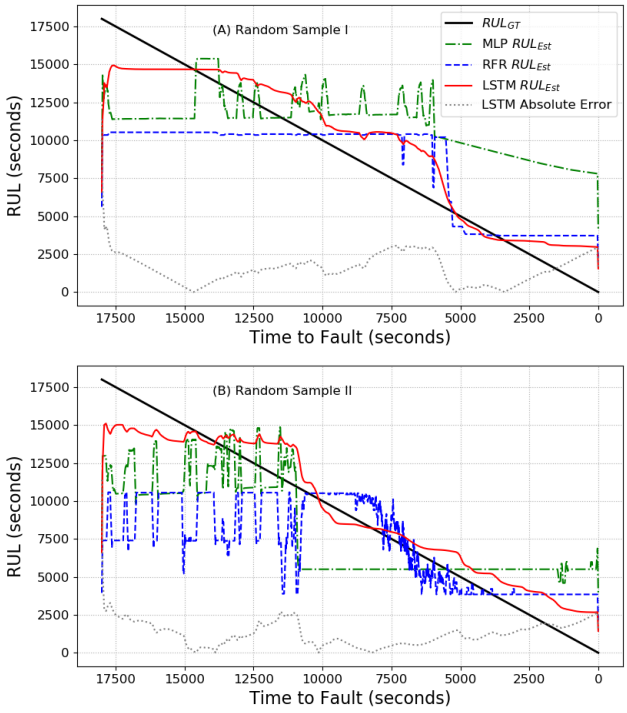

Figure 5. Two random samples of the RUL estimation results on testing data for Fault: flow-cool Leakage.

no clear boundaries for separating normal data from close to failure data points, we consider data points with $RUL > 50000s$ as normal data points. Therefore, the data points with $5000s \leq RUL \leq 50000s$ are considered as the boundary points and have not been used in training the RF classifier for that failure. To address imbalanced class distribution between normal and fault data, we first up-sample the fault data points by a factor of 1000, we then randomly select a subset of normal points equal to the number of over-sampled fault points.

5.6. Results on PHM18 Challenge Test Data Set

Table 5 and Table 6 represent the scores to evaluate 2018 PHM Data Challenge solutions. The first score is designed capture the RUL estimation performance close to the failure. The second score is designed to capture RUL estimation performance when the system is far from the end of life. The over all score is defined as the average of these two scores. A lower score is a better score.

Table 7, Table 8, and Table 9 demonstrate the scores for our method and several baseline solutions in estimating RUL for the first, second and third fault modes respectively. To train the baseline methods, we replaced NAN with maximum RUL in the data set for each fault. In the test step, we report NAN when a model predicts RUL higher than the maximum RUL in the training data. Compared to the baselines solutions, Linear Regression (LR), Support Vector Regression (SVR), Random Forest Regression (RFR), and Multilayer Perceptron (MLP) neural networks, our model (early fault detection + LSTM) performs far better.

6. CONCLUSIONS

In industrial systems, it is not feasible to make accurate RUL estimation before the beginning of the degradation process. Therefore, it is challenging to estimate RUL for the systems with abrupt failures. In this paper, we proposed a new solution for this hard problem. Our proposed solution incorporates an early degradation detection module to automatically detect the degradation mode. After the degradation mode detection, our solution applies an LSTM neural networks to estimate the RUL. We demonstrated the performance of our RUL estimation for the 2018 PHM Data Challenge.

REFERENCES

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*

Table 5. The first evaluation score for 2018 PHM Data Challenge.

Ground Truth RUL (GT)	Submission RUL (SUB)	Score
Number	Number	$\|GT - SUB\|e^{(-0.001GT)}$
NaN	Number	$\|SUB\|e^{(-0.001SUB)}$
Number	NaN	$\|GT\|e^{(-0.001GT)}$
NaN	NaN	0

Table 6. The second evaluation score for 2018 PHM Data Challenge.

Ground Truth RUL (GT)	Submission RUL (SUB)	Score
Number	Number	$0.1(GT - SUB)^2$
NaN	Number	$\frac{5}{\|SUB\|+3}$
Number	NaN	$20e^{\frac{-1}{\|GT\|+0.1}}$
NaN	NaN	0

Table 7. 2018 PHM Data Challenge scores for the first fault mode.

Method	Score 1	Score 2	Overall Score
LR	1,541.81	2.0×10^{12}	1.0×10^{12}
SVR	0.54	3.0×10^{11}	1.5×10^{11}
RFR	1,762.07	2.7×10^{12}	1.3×10^{12}
MLP	1,979.62	2.3×10^{12}	1.2×10^{12}
Our method	0.37	5.3×10^{8}	2.7×10^{8}

Table 8. 2018 PHM Data Challenge scores for the second fault mode.

Method	Score 1	Score 2	Overall Score
LR	742.37	1.1×10^{12}	5.6×10^{11}
SVR	0.079	1.5×10^{11}	7.7×10^{10}
RFR	573.15	1.2×10^{12}	5.9×10^{11}
MLP	50.36	4.0×10^{11}	2.0×10^{11}
Our method	0.079	9.8×10^{7}	4.9×10^{7}

Table 9. 2018 PHM Data Challenge scores for the third fault mode.

Method	Score 1	Score 2	Overall Score
LR	0.0461	2.2×10^{9}	1.1×10^{9}
SVR	0.0462	1.8×10^{10}	9.2×10^{9}
RFR	31.419	3.2×10^{10}	1.6×10^{10}
MLP	0.11	9.8×10^{9}	4.8×10^{9}
Our method	0.049	1.5×10^{8}	7.4×10^{7}

preprint arXiv:1207.0580.

Hsu, C., & Jiang, J. (2018). Remaining useful life estimation using long short-term memory deep learning. In *2018 ieee international conference on applied system invention (icasi)* (pp. 58–61).

Khorasgani, H., Farahat, A., Ristovski, K., Gupta, C., & Biswas, G. (2018). A framework for unifying model-based and data-driven fault diagnosis. In *Annual conference of the prognostics and health management society (phm18).*

Khorasgani, H., Kulkarni, C., Biswas, G., Celaya, J. R., & Goebel, K. (2013). Degredation modeling and remaining useful life prediction of electrolytic capacitors under thermal overstress condition using particle filters. In *Annual conference of the prognostics and health management society (phm13).*

Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, *213*(1), 1–14.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958.

Yuan, M., Wu, Y., & Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *Aircraft utility systems (aus), ieee international conference on* (pp. 135–140).

Zhang, C., Hong, G. S., Xu, H., Tan, K. C., Zhou, J. H., Chan, H. L., & Li, H. (2017). A data-driven prognostics framework for tool remaining useful life estimation in tool condition monitoring. In *Emerging technologies and factory automation (etfa), 2017 22nd ieee international conference on* (pp. 1–8).

Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2018). Long short-term memory for machine remaining life prediction. *Journal of Manufacturing Systems*.

Zheng, S., Ristovski, K., & Farahat, C., A.and Gupta. (2017). Long short-term memory network for remaining useful life estimation. In *Prognostics and health management (icphm), 2017 ieee international conference on* (p. 88-95).

BIOGRAPHIES

Wei Huang received the B.E. degree in Optoelectronic Engineering from Beijing Institute of Technology (Beijing, China), in 2011, and Ph.D. degree in electrical and computer engineering from Northeastern University (Boston, U.S.), in 2017. During her PhD at the Laboratory for Ocean Acoustics and Ecosystem Sensing at Northeastern University, she has conducted research in passive ocean acoustic remote sensing. Her current research projects in Industrial AI Lab at Hitachi America R&D include developing signal decomposition approaches and acoustic sensor based failure detection.

Hamed Khorasgani received the B. Sc. degree in electronics and electrical engineering from Isfahan University of Technology, Isfahan, Iran, in 2009, the M. Sc. degree in mechatronics engineering from Amirkabir University of Technology, Tehran, Iran, in 2012, and the PhD in electrical engineering from the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA, in 2017. During his PhD at the Institute for Software Integrated Systems, he has conducted researches in analysis complex systems and their applications in diagnosis, prognostics, and fault tolerant control. His current research projects in Industrial AI Lab at Hitachi America R&D include developing hybrid methodologies and solutions for integrating model-based and data-driven diagnosis methods.

Chetan Gupta is the Chief Data Scientist & Architect, and manages the Industrial AI Lab at Hitachi America R&D. He has more than 15 years of experience in analytics, AI, big data, and related domains. Over his career he has worked both as a machine learning data scientist as well as in designing systems and architectures for big data applications. At Hitachi, he manages a large team of data scientists, architects and developers that is engaged in developing cutting edge solutions and opening new frontiers in the area of industrial analytics. His team builds fundamental horizontal technologies that are then used to build solutions for industry specific verticals. He has led efforts to build horizontal solutions in predictive maintenance, quality, operations monitoring and control, and for verticals such as mobility, mining, building energy management systems, etc. Over the years Chetan has led multiple research and development teams, and mentored young researchers. He has close to 50 patents either granted or under review and more than 40 publications in the area of data mining/machine learning, data stream systems, complex event processing, workload management, etc. Chetan has a Ph.D. in Mathematics and M.S. in Mathematical Computer Science and Chemical Engineering from University of Illinois, Chicago.

Ahmed K. Farahat holds a Ph.D. degree from the University of Waterloo in Canada and M.Sc. and B.Sc. degrees from Cairo University in Egypt, all in Computer Engineering. Dr. Farahat is currently a senior research scientist at the Industrial AI Laboratory at Hitachi America, Ltd. Previously, Dr. Farahat was a research scientist at the Big Data Laboratory at Hitachi America, Ltd. (2014-2017). He also worked as a postdoctoral fellow at the University of Waterloo (2013-2014), a research assistant at the University of Waterloo (2007-2012) and a research engineer at IBM Egypt (2005-2007). His research interests lie in the areas of machine learning and data mining, and their applications to industrial data analytics. Dr. Farahat is the recipient of the Best Paper Award Runner-Up at the 2013 IEEE International Conference on Data Mining (ICDM), and two best paper awards in specialized workshops at ICDM′09 and NIPS′13.

Shuai Zheng received the PhD degree in Computer Science from the University of Texas at Arlington in 2017. His main research interests are machine learning, data mining, and cloud computing.