IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2021, accepted April 27, 2021, date of publication May 3, 2021, date of current version May 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077192

Long Short-Term Memory Networks for Facility Infrastructure Failure and Remaining Useful Life Prediction

RODNEY KIZITO[1], PHILLIP SCRUGGS[1], XUEPING LI[1], (Member, IEEE), MICHAEL DEVINNEY[2], JOSEPH JANSEN[3], AND REID KRESS[3]

[1]Department of Industrial and Systems Engineering, The University of Tennessee, Knoxville, TN 37996, USA
[2]Consolidated Nuclear Security, LLC, Oak Ridge, TN 37830, USA
[3]Electric Power Research Institute, Oak Ridge, TN 37832, USA

Corresponding author: Xueping Li (Xueping.Li@utk.edu)

This research was supported in part by the Maintenance Advance Technology Initiative (MATI) and the Chad/Ann Blair Holliday Fellowship.

ABSTRACT Sensors attached to an asset acquiring vibration patterns during both operational and failure states have been used to diagnose fault conditions and to predict future failures of the components being monitored. In this research, we investigate Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), for failure diagnosis and remaining useful life (RUL) prognosis of such deteriorating components. LSTM networks' long-term dependency capability, which allows LSTM's to recall information for long term sequence lengths, can also be used to predict the probability of failure within a specified time frame. In this paper, we develop and apply a stylized LSTM model to a motor degradation dataset for the purposes of diagnosing failure and predicting the probability of failure within a specified time frame, as well as predicting RUL. We developed the dataset by acquiring automated sensor measurements from an induction motor attached to a destructive test platform. The performance of the LSTM model on the developed dataset is compared to that of the Random Forest (RF) algorithm as RF is reputably known for classification and regression. The results demonstrate that the LSTM provides quality predictions of motor failure, failure probability and RUL on the developed dataset. When compared to the RF approach, the LSTM performs comparably well in failure classification and outperforms the RF in RUL prediction.

INDEX TERMS Predictive maintenance, deep learning, long short-term memory, condition monitoring, rotor bar failure, Industrial Internet of Things, prognostic health management, RUL prediction.

I. INTRODUCTION

The field of reliability and maintenance engineering has incorporated new technologies, including artificial neural networks and other Machine Learning (ML) based algorithms, in order to improve the predictability of diagnosing machinery faults and failure times. The expansion from physics based models, which depend on scientifically proven feature analysis by subject matter experts towards the ML methodologies presented in recent literature, is mainly driven by the need to automate and scale such systems. ML models are based on observations of data streams rather than physical-based models which solely rely on classical mechanics for each respective use case [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Nagarajan Raghavan.

Even with a multitude of applications utilizing feature learning based approaches for vibration data, there are several reasons why a feature engineering based system may not be ideal for increasingly complex and scaled maintenance operations [2]–[4]. Feature engineering and physics based approaches require a subject matter expert with knowledge of the specific system or a statistical based approach utilizing calculated features that may not describe the characteristics of a dynamic signal needed to accurately classify the fault. Some faults have not been characterised, resulting in an inability for a statistical based approach to identify such faults which may exist during the operation of an asset. Once these unidentified faults become recognizable as a fault by a subject matter expert, the pattern must be duplicated and then implemented into the expert system manually.

In order to prevent the issues when scaling a subject matter expert based solution, feature learning or ML methods can

be utilized. Unlike feature engineering where a subject matter expert or human must select relevant features, feature learning is able to extract useful features from the raw data signal. An algorithm would optimally learn relevant features from the raw data without human input and scale autonomously unlike manually engineered features [2]. Research has also been successful in identifying patterns on disparate datasets that come from assets with varying amounts of unknown physical degradation [5]. Data driven approaches have also been utilized for identifying specific maintenance actions that were matched to various assets at higher frequencies than others and indicating different future schedules for maintenance items based on the past maintenance history of each respective asset as well as successfully modeling the changes of the asset from the long memory of states in order to predict the current running status [6], [7].

Such data driven methodologies of fault prediction have also shown to be more accurate in many circumstances compared to thus driving additional research into further development [8]. Several methods that have gained popularity for predicting remaining useful life (RUL) and trends in performance of assets include those utilizing neural networks and deep learning, such as Recurrent Neural Networks (RNNs). Even with the successful application of RNNs and other deep learning techniques to vibration datasets and other popular condition monitoring sensor datasets, such methodologies lack the ability to capture long memory states of monitored assets. Long Short-Term Memory (LSTM) networks present an opportunity to improve on feature learning due to their ability to learn long term dependencies and capability to contain four neural network layers opposed to only one in RNNs [9]. LSTMs are simply capable of capturing this variation in a way that other methods cannot due to the ability to incorporate the most recent changes in the dataset as having increased importance towards the ultimate state of the operating asset.

The LSTM methodology has been presented in several articles as it relates to predictive maintenance (PdM) procedures and RUL estimation. In [5], a new LSTM architecture was proposed for predicting RUL given short sequences of monitored observations with random initial wear amounts. The research proposed a new objective function that was suitable for RUL estimation and proposed a new target generation approach for training the LSTM networks. The new target generation approach required fewer assumptions about the actual physical degradation of a system or asset in order to make an accurate prediction.

In this research, we will develop a stylized LSTM model under the context of PdM to predict equipment failures, estimate the probability of failure and RUL. We apply our model to a motor degradation dataset we developed by collected sensor measurements from an induction motor attached to a destructive motor test platform. Previous research performed on accelerated aging motor degradation test stands focused primarily on only diagnosing bearing failures [10]–[12] and did not incorporate other potential failures such as rotor bar failure which is a common cause of induction motor failure [13]. Additionally, previous research on rotor bar failure focused primarily on the monitoring of changes in stator current spectra at different load levels and not on the vibration of a motor [14]. Research presenting methods for fault diagnosis based on current signature analysis have proven successful for detecting motor faults and states [11], [12], [15]; however the detection of additional faults utilizing additional sensors still remain understudied. Datasets consisting of acquired vibration signals and additional other sensors may yield different results on different assets due to sensor arrangement and environmental interference. Such arrangements may lead to different diagnostic results, as experimental results demonstrate that some approaches can effectively identify the machine running conditions and significantly outperform other fusion methods for diagnosing failure [16].

The main contribution of the paper is the development of a general framework and the application of a stylized LSTM model to the dataset we generated from the destruction of rotor bars on the motor within the testing experiment. The LSTM model's performance is then compared to that of a Random Forest (RF) approach, which is a reputably strong prediction algorithm.

II. LSTM METHODOLOGY

The basic unit of an LSTM hidden layer is called the *memory block* and contains one or more *memory cells*. LSTMs perform learning by enforcing constant error flow through recurrently self-connected linear units called *constant error carousels*, known as CECs, within cell units [9]; an activated CEC is known as the *cell state*. Through the learning process, multiplicative gate units are able to open and close access to the cells. These input and output gates protect the CEC from forward flowing activation and backward flowing error [9]. If a gate is closed, meaning the activation is near zero, noise and irrelevant inputs are prevented from entering the cell, and the cell state does not affect the rest of the LSTM network.

As described in [17], let c_j denote the j^{th} memory cell. A cell is updated based on three sources: net_{c_j} - the input to the cell itself, out_j - a multiplicative unit input to output gate, and in_j - a multiplicative unit input to the input gate. Additionally, let in_j and out_j's activation at a time t be donated by $y^{in_j}(t)$ and $y^{out_j}(t)$, respectively. Let u denote the input units, gate units, memory cells, or conventional hidden units if any are present. Let w denote the weight on any unit and f denote a logistic sigmoid with range [0,1]. Time is represented by t, where $t = 1..T$. At each time step $t = 1..T$, a forward pass (update of all units) and a backward pass (computation of error signals for all weights) occurs [9]. Thus, input gate (y^{in}) and output gate (y^{out}) activation are described as follows:

$$y^{out_j}(t) = f_{out_j}(net_{out_j}(t)), \tag{1}$$

and

$$y^{in_j}(t) = f_{in_j}(net_{in_j}(t)), \tag{2}$$

where

$$net_{out_j}(t) = \sum_u w_{out_j u} y^u(t-1), \tag{3}$$

$$net_{in_j}(t) = \sum_u w_{in_j u} y^u(t-1), \tag{4}$$

$$net_{c_j}(t) = \sum_u w_{c_j u} y^u(t-1). \tag{5}$$

Furthermore, let the internal state of the cell be denoted as s_{c_j}, let a differentiable function that squashes net_{c_j} be denoted as g and a differentiable function that scales the memory cell outputs from the internal state s_{c_j} be denoted as h. Then, at a time t, c_j's output can be denoted as $y^{c_j}(t)$ where

$$y^{c_j}(t) = y^{out_j}(t) h(s_{c_j}(t)). \tag{6}$$

When $t > 0$, the internal state $s_{c_j}(t)$ can be described as

$$s_{c_j}(0) = 0, s_{c_j}(t-1) + y^{in_j}(t) g((net_{c_j}(t)). \tag{7}$$

The general architecture for a memory cell c_j is displayed in Figure 1, where the outside box represents c_j and its gate units in_j and out_j. The recurrent self-connection is depicted and indicates feedback with a delay of 1 time step [17]. A key describing the various components of the diagram is also included.

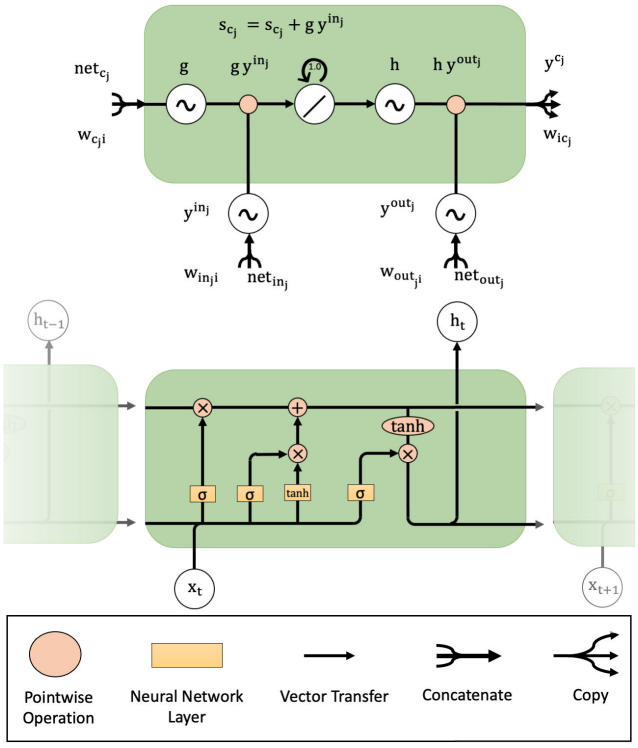


FIGURE 1. The top image describes the architecture of memory cell c_j displaying the activation, input and output gates. The bottom image is the general LSTM cell state displaying the connections, branches and components of the cell; the connections, branches and components of the cell described in the key.

Figure 2 depicts the full LSTM architecture, combined with the equations that relate the flow of information through the LSTM model; full explanations of the equations can be found at [18].

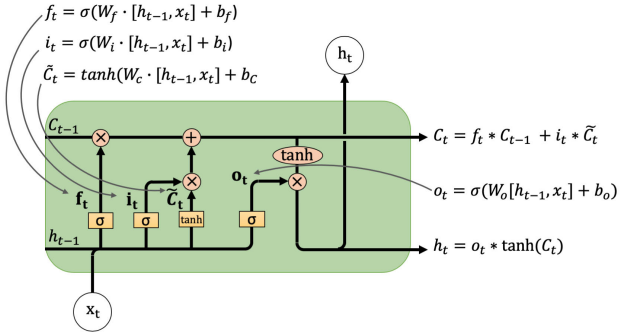


FIGURE 2. Complete LSTM architecture with equations showing how information moves through the cell; equations are explained in full at [18].

III. DATASET DEVELOPMENT AND TESTBED DESIGN

In order to facilitate and support the research project a reliable data stream that met the appropriate engineering standards as defined by literature and organizational requirements was acquired. These requirements are shown in Table 1. The datasets targeted included data streams from sensors including vibration and temperature from the assets shown in Figure 3. Through collaboration with the laboratory staff at Electric Power Research Institute (EPRI) a test frame was constructed and assets were set up. The installation of the appropriate sensors and data acquisition devices was then able to occur in order to facilitate the ability for the defined data streams to be sent to the data repository.

TABLE 1. Data acquisition hardware requirements.

Engineering Requirement	
Motor Nameplate Information	1.5 hp, 3-phase, 2-pole, 208V
Vibration Signal Acquisition Rate	10240 Samples / Sec
Length of Files (Duration)	5 seconds / file
Time of File Acquisition	Schedule up to 20 occurrences / day

The testbed consisted of a platform for monitoring three phase induction motor assets during accelerated failure testing. The experimental test setup was constructed in order to perform destructive testing on a variety of components and is based on previous research. To evaluate the reliability of the proposed method, a total of 6 data sets are collected from the test motor and shaft with a variety of rotor bar faults [14], [19]. Each data set was collected by the data acquisition equipment at 10240 Hz for 2 seconds. Such a sample rate is consistent with other literature [14]. The sensors and equipment attached to each asset are described in Figure 3. The data acquisition chosen was selected for its ability to be utilized both in a research and industrial setting. The National Instruments cRIO data acquisition devices and modules utilized for data acquisition were calibrated to ensure that signals were recorded accurately. The vibration input

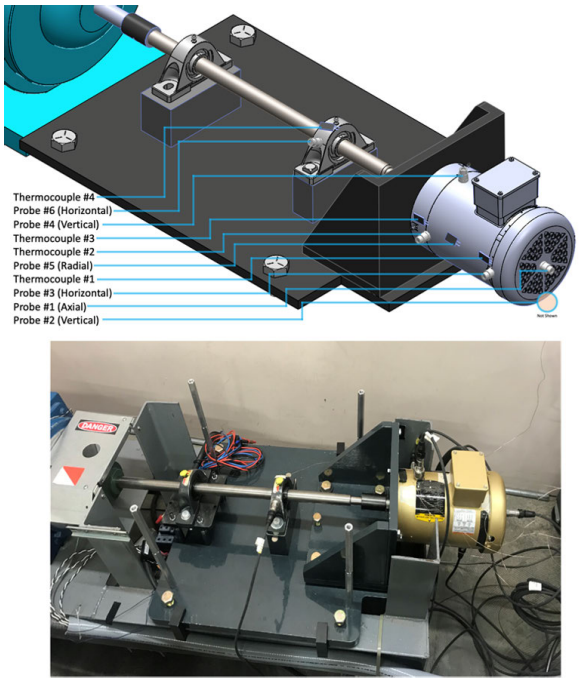


FIGURE 3. UT and EPRI Test destructive test stand where experiments are performed and datasets collected.

module incorporates software-selectable AC/DC coupling, IEPE open/short detection, and IEPE signal conditioning. The input channels simultaneously measure signals. Each channel also has built-in anti-aliasing filters that automatically adjust to the selected sample rate. Prior to performing experiments, structural resonance testing was performed on the motor test bed to ensure that the structural vibrational motion did not overlap with the mechanical vibrations created by the misalignment and drilled rotor bar experiments as well as normal motor operation. The vibration sensors utilized were PCB Sensor accelerometer model 603C01 and the thermocouples used were k-type thermocouples. One motor and a total of six accelerometers positioned as shown in Figure 3 were used for vibration data collection in the experiment, and data was collected over a total of 34 days.

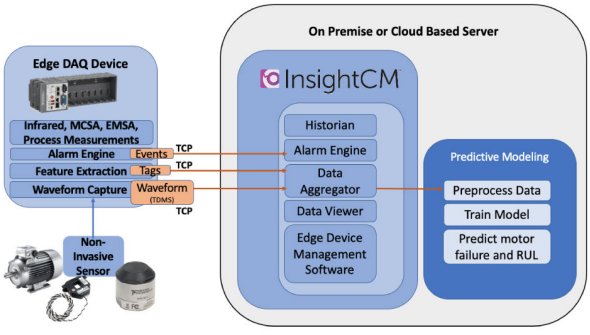


FIGURE 4. Diagram depicting the architecture of this research from data acquisition to predictive modelling.

Figure 4 depicts the architecture of this research, from the data acquisition using the sensors and edge device, to the predictive modelling. The software utilized for data acquisition and collection was National Instruments' InsightCM on a Windows Server 2016 Machine. For the predictive modelling, we utilized the tensor flow package in Python and ran the model on a Tesla V100 G32 GPU server with 256 GB RAM.

IV. MODEL DEVELOPMENT AND APPLICATION

Our study develops and applies an LSTM model to achieve three PdM goals: (1) diagnose a motor's failure; (2) predict the probability of a motor failing 24 hours in advance of the actual failure occurring; (3) predict the RUL of a motor in terms of the percentage of health remaining until a defined failure point is reached. This section details how we developed and applied our LSTM model to the dataset acquired from our testbed. This section also describes the data preprocessing for the RF model that we use for comparison purposes to the LSTM model.

A. LSTM DATA PRE-PROCESSING

Using the data acquired from our testbed, we determined that there were no correlations among the variable inputs and thus all six sensors were used as model inputs for the LSTM. We visualized the data using plots and identified that the actual failure occurred on day 28 of the 34-day experimental test. To account for noisy data, we used a one-minute frequency division to get well separated values in the finalized dataset (24*60 = 1,440 minutes in a day), which leaves our dataset with around 1,440 observations for each day.

The sampling frequency was not the same for all 27 healthy days we collected data from the motor. These changes of frequency were caused by field engineers on the testbed, which were not intentional. Since samples were collected using four different combinations of sampling frequency and duration, we divided the dataset into four sub-datasets (in numerical order of the days) with each sub-dataset having the same sampling frequency. This allowed us to choose the step for the separation of the indices.

B. RF DATA PRE-PROCESSING

We compare the LSTM model's performance to a RF approach. RF models use ensemble learning on the bagging method of decision trees, which are constructed using samples and a majority vote is taken for the prediction [20]. RF models are commonly known as strong performers for classification tasks (i.e., motor failure classification) [21], but also perform well with regression based predictions (i.e., motor RUL) [22]. The inputs of the RF model differs from those of the LSTM. The LSTM model uses just the six sensors from the motor as model inputs and is capable of capturing the time dependency in the data and generating features. For the RF, we apply time-domain statistical measures (e.g., mean, standard deviation, 25%, 50% and 75% quartiles) and statistical tests (e.g., trend, homogeneity, kurtosis, skewness) to each of the six sensors to generate features to serve as inputs for the RF model. The use of time-domain statistics

helps to capture the time dependency in the dataset for the RF model.

The time-domain statistical measures and tests are computed for each of the six sensors by using a 60 minute period of data. This creates nine features (mean, standard deviation, 25%, 50% and 75% quartiles, trend, homogeneity, kurtosis and skewness) for each of the six sensors. We also took into consideration the relationship between periods, which allows the RF to see what was happening in the previous periods. To do this, we generated another nine features per sensor using a 120-minute period which includes the original 60-minute period plus the 60-minute period prior. For example, if we are to predict the value 121, we take the period [60:120] as the principle period to generate statistical measures and tests. We then take the period [0:120] as the large period, to take into account the period before, and generate nine more statistical measures and tests on this large period as well. This allows us to take the periods as one data point for the RF, and the features are used as the inputs for the RF model.

C. LSTM MODEL ARCHITECTURE AND PARAMETERS

The LSTM model is developed using two layers and an output layer. The first layer contains 64 neurons and has a 20% dropout, while the second layer contains 32 neurons and also has a 20% dropout. The number of layers and neurons in each layer heavily affects the prediction performance and run-time of an LSTM. Thus we settled on 64 and 32 neurons for the LSTM layers because such a model provided strong performance within reasonable run-time. The neuron dropout rate is used to control overfitting in an LSTM, especially when a small dataset is being used as in the case of this research. Researchers commonly use a dropout rate of 20%, thus we set our dropout rate to 20%.

A dense layer, which performs the operation on the input layers and returns the output prediction, is then applied. Dense layers contain activation functions which transform the weighted inputs to the outputs. A regression-based LSTM model can use a *tanh* or *relu* activation function. Tanh provides values in $[-1, 1]$. Since we predict RUL as a percentage (i.e., [0, 1]), we can use *tanh* or *relu* as the activation function and we chose to use *relu*. For the classification-based LSTM model, a *sigmoid* activation function is used for binary classification of motor failure.

D. PREDICTING FAILURE AND RUL WITH LSTM

The inputs for the LSTM model are the six sensor variables, while the outputs are the healthy/failed (0/1) response column for the classification-based LSTM and the RUL response column for the regression-based LSTM. We set the sequence length of data (memory) to 60. Thus, the LSTM uses 60 minutes of data, as an input, to predict the next value. More data can be taken as input, but computation time can increase as well. The quality of the LSTM model's prediction will increase the more data (larger sequence length) we take in, but the quality of prediction reaches a maximum limit to where it will not improve even if more data is taken.

We fine tune the learning rate, which determines how fast and how good the training process will be. We use 20 iterations, and for each iteration we change the learning rate to see how the error is affected. Ultimately, we select the learning rate with the lowest or minimal error. We then balance the dataset. This can be done using the oversampling method to the minority class (failed observations) which will add more rows to the dataset that resemble the already recorded minority class observations, or by adding weights to the loss function, which then penalizes the imbalance in the dataset. The LSTM's application is summarized in Algorithm 1.

Algorithm 1 The LSTM Prediction Algorithm

Input: Data from six sensors on motor
Output: Prediction of motor failure or RUL
Step 1: Split the dataset into train and test data
Step 2: Normalize the data as [0,1] and balance data
Step 3: Generate sequence as 3-D array (d, s, f)
  * d = training samples
  * s = sequence length
  * f = number of features
Step 4: Select a learning rate for training
Step 5: Build LSTM network using [2, a, b, 1] dimensions; 2 input layers, *a* neurons in first, layer, *b* neurons in second layer, 1 output layer
Step 6: Train the LSTM network
  for each *i* in range(epochs) **do**
    model.fit(train_inputs, train_outputs)
  end
Step 7: Predict motor failure or RUL
  * Failure - *model.predict_classes(train_inputs)*
  * RUL - *model.predict(train_inputs)*
Step 8: Validate trained model on test data
Step 9: Evaluate prediction performance

1) FAILURE PREDICTION

We generate a binary response variable for failure prediction, where "1" signifies failure and "0" signifies healthy. We assign a "1" for the binary response variable of all observations on days 28-34 since day 28 is where failure occurs. To predict failure 24 hours in advance of failure, we assign a "1" for the binary response variable of all observations on days 27-34 as 27 is one day prior to the actual failure on day 28. The six-sensor variable inputs are normalized and then split into training and testing datasets. For the failure classification LSTM model, we use days 6, 10, 17, 30 and 33 as the test dataset and the remaining days as the train dataset; the train dataset has a total of 36,335 observations, while the test dataset has 7,255 observations (failure classification dataset contains data from all 34 days of the experiment).

The classification-based LSTM model is fit by training the input data to the binary classification column. We use

100 epochs and 5% of the training data is used for validation in the training process. We then predict if a motor will fail and the probability of a motor failing 24 hours in advance of the actual failure occurring. For evaluation of the LSTM model's performance, we use the following metrics: (1) *accuracy*, which measures the fraction of samples predicted correctly; (2) *precision*, which measures how accurate your model predicts failures - percentage of failure predictions that are actually failures; (3) *recall*, which measures how well the model captures those failures - percentage of true failures the model captures; (4) *F1-Score*, which measures test accuracy and is a balance between *precision* and *recall* where a value of 1 is the best result; and (5) *confusion matrix*, which is an $N \times N$ matrix ($N = \#$ of target classes) comparing the actual target values with those predicted by the model.

2) RUL PREDICTION

Predicting RUL with too much data in each state of the motor's degradation (due to a high sampling frequency) will provide large RUL values, making it harder for the LSTM model to be able to predict the RUL well. We use another approach by predicting the percentage of RUL. Predicting the percent of RUL also helps provide stable weights within the LSTM model. We generate the RUL response variable (as a percentage of remaining health) column, where failure is assumed to be the last observation of day 27; RUL should be 0.0 for the last observation of the dataset. The input variables are then normalized and split into a testing and training dataset. For the RUL model, we use days 2, 8, 15, 26 as the test dataset and the remaining days as the train dataset. The train dataset has a total of 28,981 observations, while the test dataset has 5,832 observations (RUL dataset contains data from days 1-27 of the 34 day experiment and the first point from day 28 when the actual failure occurs).

The regression-based LSTM model is fit by training the input data to the RUL response column. 5% of the training data can be used for validation in the training process; if training dataset is small, then testing data can be substituted for validation as well. The output layer applies a *relu* activation corresponding to the regression based RUL prediction and we use 150 epochs. We evaluate the LSTM model's performance by using the following metrics: (1) *mean absolute error (MAE)*, which measures the difference between the actual and predicted values by averaging the absolute difference over the data set; (2) *mean square error (MSE)*, which measures the difference between the actual and predicted values by squaring the average difference over the data set; and (3) *R-squared*, which represents the coefficient of how well the values fit compared to the actual values. A summary of the steps taken to develop and apply the model from this research are displayed in Figure 5.

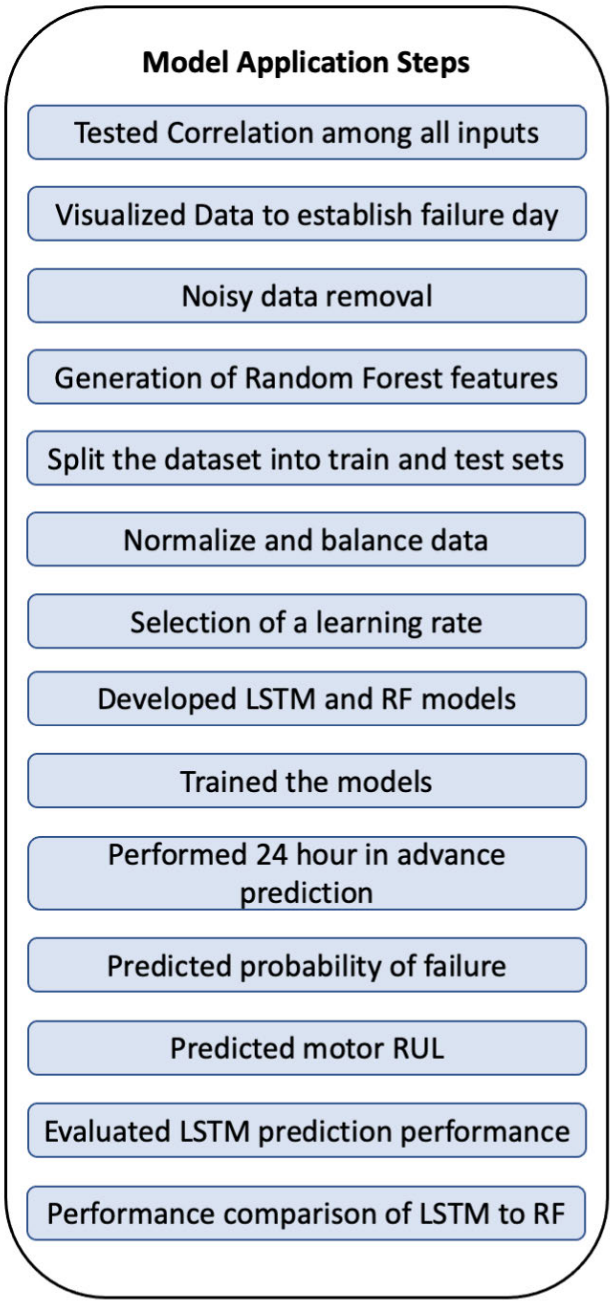


FIGURE 5. Summary of the steps taken to develop and apply the model.

V. RESULTS

Results from the analysis are provided in two parts: (1) results of the failure prediction, which depict how well the LSTM model diagnosed motor failure and predicted a motor failure 24 hours in advance of the actual failure; (2) results for the RUL prediction, which depict how well the LSTM model performed when predicting the RUL of a motor. The LSTM's performance results are compared to that of an RF approach.

A. FAILURE PREDICTION RESULTS

For evaluation of the failed versus healthy diagnosis, a confusion matrix is used to depict how well the LSTM model performed. A confusion matrix is a two-dimensional matrix with one dimension expressing the true class of an object and the second dimension expressing the class that the

TABLE 2. Failure Prediction Results. For the confusion matrix: (1,1) = true positives; (1,2) = false positives; (2,1) = false negative; (2,2) = true negative.

Hours (Days) in Advance	Metrics	LSTM		RF	
24 hours (1 Day)	Accuracy	0.934		0.904	
	Precision	1.000		0.840	
	Recall	0.833		1.000	
	F1-Score	0.909		0.913	
	Confusion Matrix	4189	0	3534	835
		462	2304	0	4369

classifier assigns, ultimately summarizing classification performance [23]. We also provide accuracy, precision, recall and F1-Score as performance metrics for the LSTM failure prediction model. These results are presented in Table 2 and compared to the performance of the RF approach.

The results show that our developed LSTM model can diagnose failure and predict a motor failing, 24 hours in advance of the actual failure occurring, with an accuracy of about 93%. In terms of the confusion matrices, the model classified all 4189 healthy observations accurately, which provides a precision score of 1.0. For the F1-Score, which provides us a balance between the precision and recall and where a value of 1.0 is optimal, the results show a strong F1-Score of 0.91. When compared to the RF approach, the results show that the LSTM model performs just as well when classifying motor failure 24 hours in advance as the RF model, which is generally known as a strong classifier. The LSTM model had a higher accuracy (0.934) and precision (1.000) than the RF model, while the RF model had a higher recall (1.000) and F1-score (0.913).

It is also important to note that the dataset used for the RF contains time-domain statistical measures (e.g., mean, standard deviation, quartiles) and statistical tests (e.g., trend, homogeneity, kurtosis, skewness) as features that were developed from each sensor's data to improve the RF's ability to capture the time dependency in the dataset, whereas the LSTM dataset contains only the 6 sensors' data as inputs; this helped increase the performance of the RF when compared to the LSTM. LSTMs are capable of capturing the time dependency in the data and thus the LSTM model does not need such features as model inputs in order to perform well. Additionally, the LSTM is a deep learning based model which requires a large amount of data for peak performance [24]. The LSTM's performance was hindered due to our small dataset, whereas an RF's performance is not as negatively affected by small datasets.

Overall, the results show us that our LSTM model can diagnose failure and predict the probability of failure 24 hours in advance well. Figure 6 and 7 display the LSTM and RF model predicted diagnosis versus the actual diagnosis from the test dataset respectively, while Figure 8 shows the predicted probability of failure versus the actual failure of

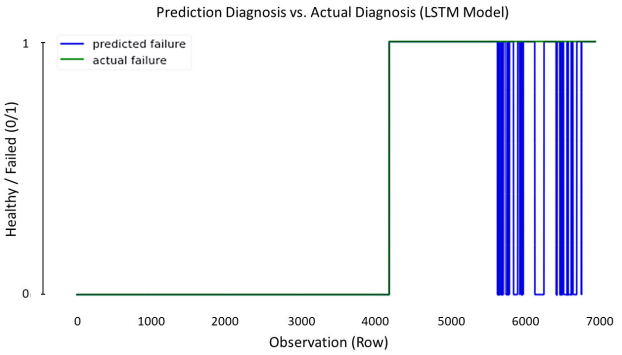


FIGURE 6. Predicted Failures vs. Actual Failures graph for the 24 hours in advance LSTM model. Ideally, the desire is to see only one line on the chart which means that all failed/healthy observations were diagnosed accurately. The blue vertical lines represent the falsely diagnosed observations.

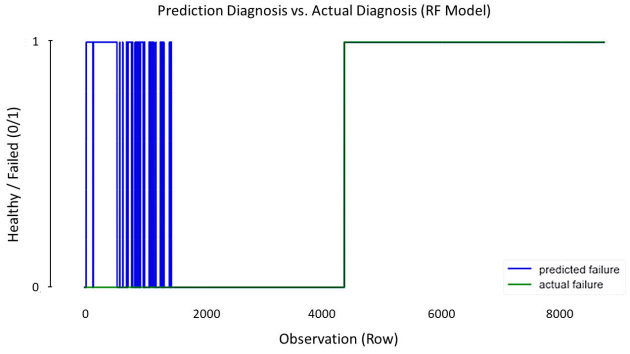


FIGURE 7. Predicted Failures vs. Actual Failures graph for the 24 hours in advance RF model. Ideally, the desire is to see only one line on the chart which means that all failed/healthy observations were diagnosed accurately. The blue vertical lines represent the falsely diagnosed observations.

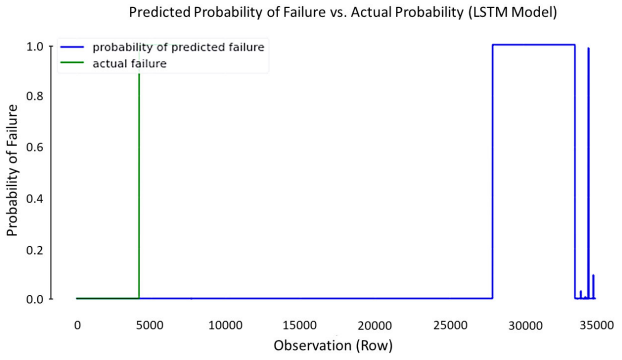


FIGURE 8. Predicted Probability of Failures vs. Actual Probability of Failures for the 24 hours in advance LSTM model; probability of >= 0.5 signifies failure and < 0.5 signifies health. Ideally, the desire is to see only one line on the chart which means that all predicted failure probabilities match the actual probabilities of 0.0 and 1.0. The blue vertical lines represent probabilities of the inaccurately predicted observations.

the LSTM model. In Figure 6, the blue vertical lines after observation 4189 represent the 462 observations (see confusion matrix in Table 2) that were falsely diagnosed by the model; the same blue vertical lines in Figure 7 represent the 835 falsely diagnosed observations of the RF model. Figure 8 shows the predicted probability of failure for each

observation where a probability of >=0.5 signifies failure and <0.5 signifies healthy. The goal for both figures is to see only one line, which would mean that all failed/healthy observations were diagnosed accurately and that all predicted failure probabilities match the actual failures.

TABLE 3. RUL Results.

Metric	LSTM	RF
MAE	0.075	0.091
MSE	0.008	0.009
R^2	0.92	0.92

B. RUL PREDICTION RESULTS

The RUL prediction results are displayed in Table 3. The first evaluation metric used is the mean absolute error (MAE), which measures the mean of the absolute values of each prediction error - difference between actual RUL values and predicted RUL values [25]. The testing MAE of the RUL prediction model was determined to be 0.075, which is low and signifies a quality prediction model. The second evaluation metric is the mean square error (MSE), which is the average squared difference between the predicted RUL values and the actual RUL values [25]; the desire is to have as low of an MSE as possible. The testing MSE of the RUL prediction model was determined to be 0.008. Such a result is considered low and could be even lower if the dataset was larger. We also computed the coefficient of determination (R^2) of the model. R^2 is a measure of how much variation within the response variable is explained by the model, where a measure of 0.00 (0%) signifies that the model does not explain any of the variability of the response variable around its mean and a measure of 1.00 (100%) signifies that the model explains all the variability of the response variable around its mean [26]. The results produced a testing R^2 of 0.92, which signifies a good model by general standards of R^2. Overall, the results show that our RUL prediction model generalizes well to unseen data and is a quality prediction model for RUL of a motor.

When compared to the RF approach, the LSTM model outperforms the RF approach with a lower *MAE* and *MSE* when predicting RUL. The difference between the models is small and is shown with an equivalent R^2 of 0.92 for both the LSTM and RF. As stated previously, the LSTM is a deep learning based model that requires a large amount of data for peak performance, and thus would have performed even better had we generated a larger dataset from the destructive motor testbed; RF approaches are not as negatively affected performance-wise with smaller datasets.

The loss function (the objective function of the LSTM network that seeks to minimize the model's error) is displayed in Figure 9. The plot shows comparable loss as the epochs increase on the training and validation, which is desired and shows that the 150 epochs utilized to fit the model are adequate. Figure 10 and 11 display the predicted RUL values versus the actual RUL values for the LSTM and RF respectively, and it can be seen that the LSTM has a slightly better matching of predicted and actual than the RF.

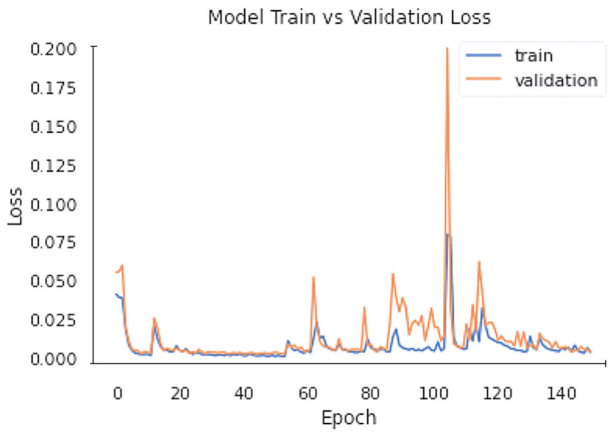


FIGURE 9. Training and validation loss vs. Epochs graph for RUL prediction model. 150 epochs were used and the plot shows comparable loss as the epochs increase, which is desired.

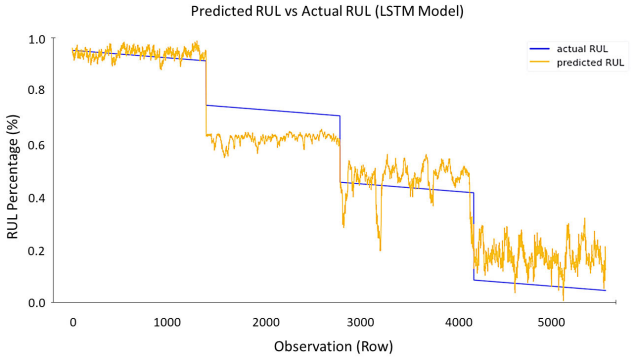


FIGURE 10. Predicted RUL percentage vs. Actual RUL Percentage graph for the LSTM model. The four splits on the chart represent the four days of data (days 2, 8, 15 and 26) assigned to the testing dataset for evaluation of the trained model.

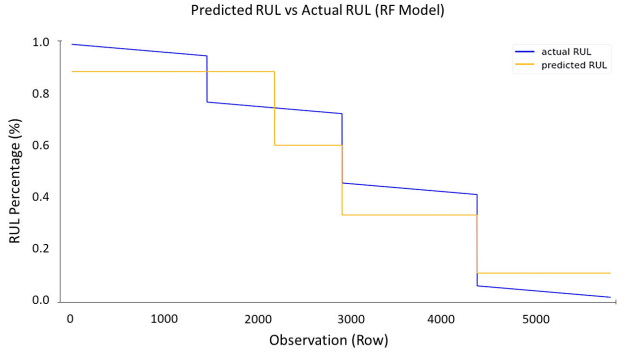


FIGURE 11. Predicted RUL percentage vs. Actual RUL Percentage graph for the RF model. The four splits on the chart represent the four days of data (days 2, 8, 15 and 26) assigned to the testing dataset for evaluation of the trained model.

Recall that our RUL values are expressed as percentages thus providing a percentage of health remaining until the defined failure point is reached. The predicted RUL percentages (orange) fall within range of the actual RUL percentages (blue) across all observations, providing a visual portrayal of the low MSE test results of the model.

In Figure 10 and 11, the four splits on the chart represent the model's evaluation on the four days (day 2, 8, 15 and 26) used as the testing dataset. The vertical line jumps in-between the four horizontal lines represent the missing days of data in the testing dataset; those missing days are the days we assigned to the training dataset when we split the data.

VI. CONCLUSION AND DISCUSSIONS

This paper presents a stylized LSTM model and applies it to a developed motor degradation dataset for PdM purposes. Specifically, the LSTM model is used to diagnose failure, predict the probability of failure within a specified future time frame and remaining useful life (RUL) of a motor. For the failure diagnosis and probability of failure results, the LSTM model performs well when diagnosing failure and predicting the probability of failure 24 hours in advance of the actual failure occurring; this is shown by high accuracy and F1-score results. Thus, usage of the LSTM model can help provide a maintenance team a 24 hour intervention period on the monitored motor. The LSTM performed just as well when classifying failure as the reputably strong classifier RF approach. In terms of RUL prediction, the LSTM outperformed the RF approach by a small margin. The small dataset we developed hindered the performance of the LSTM model as LSTM model's are deep learning based and typically require a large amount of data for peak performance. With a larger dataset, the LSTM model could be trained to perform even better. For the RUL prediction, the mean absolute error (MAE) and mean squared error (MSE) are low and the R^2 is high, when the LSTM model is applied to the unseen testing data. This shows the model has major predictive value for a motor's RUL, and a low testing MSE shows that the model generalizes well to other data. Thus, using the LSTM model, a maintenance team can know how much life a motor has remaining and better plan maintenance actions. Future research directions and extensions include deploying trained LSTM models on edge devices, updating LSTM networks in an online fashion and training LSTM models on larger datasets to improve their performance when predicting motor failure and RUL.

REFERENCES

[1] S. O. Erikstad, "Design patterns for digital twin solutions in marine systems design and operations," in *Proc. UIST*, May 2018, pp. 354–363.

[2] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke, "Deep learning for infrared thermal image based machine health monitoring," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 151–159, Feb. 2018.

[3] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 812–820, Jun. 2015.

[4] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. Jun, and J. W. Sutherland, "Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data," *Procedia CIRP*, vol. 80, pp. 506–511, May 2019.

[5] A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking LSTM for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, Jan. 2019.

[6] J. Gardner, D. Koutra, J. Mroueh, V. Pang, A. Farahi, S. Krassenstein, and J. Webb, "Driving with data: Modeling and forecasting vehicle fleet maintenance in Detroit," 2017, *arXiv:1710.06839*. [Online]. Available: http://arxiv.org/abs/1710.06839

[7] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu, and S. Yang, "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23551–23560, 2018.

[8] M. Baptista, E. M. P. Henriques, I. P. de Medeiros, J. P. Malere, C. L. Nascimento, and H. Prendinger, "Remaining useful life estimation in aeronautics: Combining data-driven and Kalman filtering," *Rel. Eng. Syst. Saf.*, vol. 184, pp. 228–239, Apr. 2019.

[9] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[10] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mech. Syst. Signal Process.*, vols. 64–65, pp. 100–131, Dec. 2015.

[11] R. N. Toma, A. E. Prosvirin, and J.-M. Kim, "Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers," *Sensors*, vol. 20, no. 7, p. 1884, Mar. 2020.

[12] C. Lessmeier, O. Enge-Rosenblatt, C. Bayer, and D. Zimmer, "Data acquisition and signal analysis from measured motor currents for defect detection in electromechanical drive systems," in *Proc. Eur. Conf. Prognostics Health Manage. Soc.*, 2014, pp. 8–10.

[13] P. F. Albrecht, J. C. Appiarius, R. M. McCoy, E. L. Owen, and D. K. Sharma, "Assessment of the reliability of motors in utility applications–updated," *IEEE Trans. Energy Convers.*, vol. EC-1, no. 1, pp. 39–46, Mar. 1986.

[14] B. Lu, Z. Gao, and J. Benzing, "Practical aspects of rotor cage fault detection for medium-voltage induction motors," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Oct. 2013, pp. 80–89.

[15] I.-H. Kao, W.-J. Wang, Y.-H. Lai, and J.-W. Perng, "Analysis of permanent magnet synchronous motor fault diagnosis based on learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 2, pp. 310–324, Feb. 2019.

[16] Z. Chen and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1693–1702, Jul. 2017.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[19] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications," *Mech. Syst. Signal Process.*, vol. 42, nos. 1–2, pp. 314–334, Jan. 2014.

[20] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[21] J. C. Quiroz, N. Mariun, M. R. Mehrjou, M. Izadi, N. Misron, and M. A. M. Radzi, "Fault detection of broken rotor bar in LS-PMSM using random forests," *Measurement*, vol. 116, pp. 273–280, Feb. 2018.

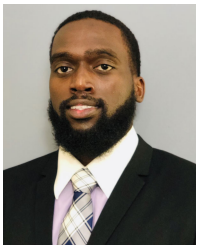
[22] P. Kundu, A. K. Darpe, and M. S. Kulkarni, "An ensemble decision tree methodology for remaining useful life prediction of spur gears under natural pitting progression," *Struct. Health Monitor.*, vol. 19, no. 3, pp. 854–872, May 2020.

[23] K. M. Ting, *Encyclopedia of Machine Learning and Data Mining*. Boston, MA, USA: Springer, 2017, p. 260.

[24] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, Dec. 2015.

[25] R. Pelánek, "A brief overview of metrics for evaluation of student models," in *Proc. CEUR Workshop*, 2014, vol. 1183, no. 2, pp. 151–152.

[26] J. Tellinghuisen and C. H. Bolster, "Using R^2 to compare least-squares fit models: When it must fail," *Chemometrics Intell. Lab. Syst.*, vol. 105, no. 2, pp. 220–222, 2011.



RODNEY KIZITO received the B.S. degree in industrial engineering from the University of Pittsburgh, Pittsburgh, PA, USA, in 2015, the M.S. degree in industrial engineering the University of Arkansas, Fayetteville, AR, USA, in 2017, and the Ph.D. degree in industrial engineering from The University of Tennessee, Knoxville, in 2021.

From 2015 to 2017, he was a Research Assistant with the Center for Excellence in Logistics and Distribution (CELDi), University of Arkansas. From 2017 to 2021, he was a Research Assistant with the Ideation Laboratory (iLab), The University of Tennessee, Knoxville. He is currently a Postdoctoral Fellow of the Solar Energy Technologies Office (SETO), U.S. Department of Energy. His research interests include energy and power system modeling, predictive analytics with machine learning, and simulation and optimization.

PHILLIP SCRUGGS received the B.S. degree in biomedical engineering from The University of Memphis, Memphis, in 2014, and the M.S. degree in biomedical engineering from The University of Tennessee, Knoxville, in 2016, where he is currently pursuing the Ph.D. degree in industrial and systems engineering.

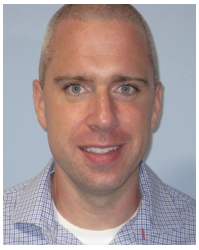
From 2015 to 2016, he was a Teaching Assistant for reliability and maintainability engineering. Since 2016, he has been a Research Assistant with The University of Tennessee. His research interests include innovation management, system modeling, predictive analytics, and connected devices (the IoT).

XUEPING LI (Member, IEEE) received the B.S. degree in automatic control and the M.S. degree in computer science from Nankai University, China, in 1996 and 1999, respectively, and the Ph.D. degree in industrial engineering from Arizona State University, in 2005.

He is currently a Professor with the Department of Industrial and Systems Engineering, The University of Tennessee, Knoxville (UTK). He is also the Co-Director of the Health Innovation Technology and Simulation Laboratory and the Director of the Ideation Laboratory (iLab) at UTK. His research interests include complex system modeling, simulation, and optimization with broad application in supply chain logistics, healthcare, and energy systems. He is a member of IISE and INFORMS. He is the President of the Modeling and Simulation Division, Institute of Industrial and Systems Engineering.

MICHAEL DEVINNEY received the B.S. degree in mechanical engineering and mathematics from Southern Illinois University, Carbondale, in 2011, and the M.S. degree in electrical engineering and applied physics from the University of Wisconsin, Madison, in 2014.

From 2013 to 2014, he was a Teaching Assistant and an Instructor with the Department of Mechanical Engineering, University of Wisconsin. He is currently a Research and Development Engineer with the U.S. Department of Energy.



JOSEPH JANSEN received the B.S. degree in mechanical engineering from The University of Tennessee at Knoxville (UTK), in 2011, and the M.S. degree in mechanical engineering from the University of South Carolina (USC), in 2017.

In 2018, he was with Westinghouse Electric Company. He is currently with the Electric Power Research Institute (EPRI), Knoxville, TN, USA. He is also a Technical Leader of EPRI. His research interests include electro-magnetic and machine design.

REID KRESS received the B.S., M.S., and Ph.D. degrees in mechanical engineering.

He has taught in the Mechanical, Electrical, and Industrial Engineering Departments at The University of Tennessee and has been a Robotics Research Engineer with the Oak Ridge National Laboratory. He is currently a Senior Technical Leader in the Power Delivery & Utilization Sector at the Electric Power Research Institute (EPRI), Knoxville, TN, USA, and teaches part-time for The University of Tennessee, East Tennessee State University, Lincoln Memorial University, or the Pellissippi State Technical Community College. His research interests include software development for specialized applications in national security, robotics, and information technology, simulation and systems modeling, dynamic system analysis (controls and vibrations), and applied mathematics.