Remaining Useful Life Estimation Using Long Short-Term Memory Neural Networks and Deep Fusion

YANG ZHANG[1], **(Member, IEEE), PAUL HUTCHINSON**[2],
NICHOLAS A. J. LIEVEN[3], **AND JOSE NUNEZ-YANEZ**[1]
[1]Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1TH, U.K.
[2]Beran Instruments Ltd., Torrington EX38 7HP, U.K.
[3]Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TH, U.K.

Corresponding author: Jose Nunez-Yanez (j.l.nunez-yanez@bristol.ac.uk)

ABSTRACT Estimation of Remaining Useful Life (RUL) is a crucial task in Prognostics and Health Management (PHM) for condition-based maintenance of machinery. In order to transmit and store the sensor data for archiving and long term analysis, data compression techniques are regularly used to reduce the requirements of bandwidth, energy and storage in modern remote PHM systems. In these systems the challenge arises of how the compressed sensor data affects the RUL estimation algorithms. A main drawback of conventional statistical modeling approaches is that they require expert prior knowledge and a significant number of assumptions. Alternative regression based approaches and deep neural networks are known to have issues when modeling long-term dependencies in the sequential data. Recently Long Short-Term Memory (LSTM) neural networks have been proposed to overcome these issues and in this paper we create a LSTM network and data fusion approach that can estimate the RUL with compressed (distorted) data. The experimental results indicate that the proposed method is able to estimate RUL reliably with narrower error bands compared to other state-of-the-art approaches. Moreover, the proposed method is able to predict RUL from both the raw and compressed datasets with comparable accuracy.

INDEX TERMS Machine health monitoring, remaining useful life (RUL), long-short term memory, recurrent neural network, data compression.

I. INTRODUCTION

Prognostics and engineering maintenance of complex machinery, such as aero-engines, gearboxes or power generation systems, are crucial and expensive tasks. Failures in machine structural integrity can be a significant contributor to increased running costs, unplanned outages and even catastrophic events in machines. With increasing system complexity, conventional scheduled preventive maintenance processes are becoming less capable of meeting the increasing industrial efficiency demands. Condition-based monitoring using intelligent prognostic and health management (PHM) technologies is capable of reducing risks and maintenance costs and increasing the reliability, availability and efficiency of machines. The estimation of remaining useful life (RUL) in machines is essential in PHM systems to improve the efficiency of maintenance schedules, avoid engineering failures and implement cost savings.

In general, the major task of RUL estimation is to predict the time left before the machinery losses its operational ability based on the information of the historical time-series sensors data which is obtained by the condition monitoring system [1], [2]. The existing methods of RUL estimation can be categorized into four groups: physics model-based approaches, statistical-based approaches, machine learning based approaches and hybrid approaches. In order to facilitate

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

these approaches, continuous streams of raw data are routinely captured by the condition monitoring systems with high precision requirements. The storage of high precision floating point data is costly, and a high bandwidth data throughput is required to transfer the data from the sensor to the central monitoring station. This is currently addressed by applying algorithms that produce low bandwidth condition indicators in the process discarding raw data. However, failure understanding and RUL estimation resulting from transient events in retrospect becomes more difficult if the reduced sampling frequency means that valuable information has been lost. An alternative approach is to apply efficient data compression algorithms [3] that can adapt its acquisition conditions and compression ratio to the machine state without loss of critical information.

This paper proposes a deep learning end-to-end Long-short Term Memory network with data fusion (LSTM-Fusion) method for RUL estimation in prognostics. Raw sensor measurements with normalization are directly used as inputs to the neural network. This means that no prior expertise in prognostics, physics model assumptions or signal processing is required and this facilitates the industrial application of the proposed method. By utilizing variable time window sizes, data normalization, LSTM and fusion techniques it is possible to capture the latent hierarchical structure in the time-series sequence by encoding the temporal dependencies with different timescales using a novel fusion update mechanism. The proposed method is expected to obtain better (higher accuracy) and reliable (lower error-band) RUL estimation not only at the end of the machine life but also during the healthy status of the machine. Comprehensive analysis of the proposed approach and comparisons with existing methods are presented in this study.

The main contributions of this paper are as follows:

- A novel RUL estimation method is proposed that utilizes LSTM-Fusion techniques. Experimental results demonstrate the superiority of the proposed network after comparison with seven state-of-the-art methods using the same datasets.
- An in depth evaluation that explores the number of layers of the proposed neural network which influence the performance of the proposed method. The performance is analyzed from both a reliability and computational complexity point of view.
- A thorough evaluation is performed on how the performance of the proposed LSTM-Fusion method is influenced by compressed/distorted sensor data using the publicly available C-MAPSS dataset. Experimental results indicate the optimum compression/distortion levels which can be tolerated by the proposed method.

This paper starts with an overview of the related work in RUL estimation methods in Section II, along with brief introductions of Long-short Term Memory (LSTM). Section III presents the proposed LSTM-Fusion network architecture in detail. Section IV presents descriptions of the experi-

mental dataset, while Section V presents the effectiveness and superiority of the proposed method after comparisons with other methods and different testing conditions. Finally, Section VI concludes this study and suggests potential future work.

II. RELATED WORK

In the past ten years, several papers and standards have reviewed RUL prediction approaches and categorized them into four groups according to their techniques and methodologies: physics model-based approaches, statistical-based approaches, hybrid approaches and machine learning based approaches [2], [4]–[6]. In the following subsections, we review state-of-the-art RUL estimation approaches based on conventional physics/statistical model approaches and machine learning based approaches.

A. PHYSICS/STATISTICAL MODEL-BASED APPROACHES

Physical and statistical model-based approaches utilize empirical physical models and mathematical modeling approaches of failure mechanisms to interpret machine damage and degradation processes [5], [7].

Physics model-based approaches are correlated to material characteristics and stress levels, which are described by experiments, finite element analysis or empirical models. The famous physics model of fatigue crack growth, named the Paris-Erdogan (PE) law, has been first proposed in [8] for RUL estimation of machinery. Over the last few years, many different versions of PE models were developed and utilized in the field of prognostics and RUL prediction of machinery [9]–[13]. Other physics models for RUL estimation of machinery include the Forman crack growth law [14] that has been utilized to estimate the RUL of cracked rotor shafts [15]. Moreover, particle filtering has been employed, in conjunction with the Norton law to characterize the turbines creep evolution and RUL estimation [16], [17]. In order to obtain accurate estimation of RUL, the physics models are developed with prior knowledge of the failure mechanisms and empirical estimation of model parameters. The drawback of physics model-based approach is that it is difficult to understand and model the physics of damage for complex mechanical systems and parameters such as material inconsistencies, imperfect boundary conditions, loads and uncertainty are generally disregarded.

Statistical data driven approaches work by building statistical models based on available past observed data in a probabilistic way [18]. The RUL estimation models have been constructed by utilizing the available data to fit a probabilistic model without relying on any physics or engineering assumptions. Statistical model-based approaches are effective in describing the uncertainty of the degradation process and RUL estimation. Therefore statistical data driven approaches have become the most popular among the four categories (including physics-, statistical-, hybrid- and machine learning-approaches).

Particularly effective are statistical data driven approaches for RUL estimation based on condition monitoring data (CM) modeling in conjunction with the estimation of the distance between the CM data and predefined threshold levels of the model. Previous papers have reviewed statistical model-based approaches systematically [2], [18], [19].

Regression-based methods are commonly used for RUL estimation due to their simplicity. Autoregressive models assume that the future state value of machines depends linearly on its previous observations and on a stochastic term [6]. In [20], the autoregressive model is combined with a particle filter algorithm to predict the RUL of machinery. An autoregressive integrated moving average and polynomial regression method are employed in [21] for RUL estimation in the wavelet domain. The limitation of the regression-based method is that the model mainly depends on the trend of historical observations, which could lead to an unreliable RUL estimation over time. A random coefficient regression method is proposed for RUL estimation in [22], which characterizes the stochastic variability of degradation processes by adding random coefficients. The proposed method used a nonlinear mixed-effects model to characterize the machinery degradation processes and estimated the RUL probability density functions by utilizing Monte-Carlo simulation. Other variants of the random coefficient models combined with Bayesian filtering algorithm for machinery RUL estimation have been proposed in [23]–[25]. The random coefficient regression models can work out probability density function of the machinery for RUL estimation. However, the models assume a Gaussian distribution for the random coefficients that can only achieve accurate results in some special cases and cannot model the temporal variability of RUL estimation [26].

Wiener processes have been proposed for machinery degradation process modeling, which are appropriate for the case that the degradation process varies bi-directionally with Gaussian noises over time [27]. Wiener processes for characterizing the machinery degradation the distribution of the first passage time (FPT) can be formulated as the inverse Gaussian distribution [27]. In [28], a time-varying degradation drift of Wiener process models is transformed into a constant degradation drift by a time-scale transformation method, where the RUL estimation can be represented by the FPT crossing the certain thresholds. The advantage of Wiener process models is that they can represent the temporal variability of the degradation processes. However, the drawback of these previous research works is that the Wiener processes are types of regression models, which are based on the assumption that the future state only depends on the current state and it is independent of the past state as it is the case with the Markov property. In order to solve this drawback, Si et al. in [29], [30] employed a particle filter to smooth out the non-Gaussian degradation state and random-effect parameters from previous measurements. The experimental results show that this approach can improve the model fitting and the accuracy of the RUL estimation.

Gamma process models work under the assumption that machine degradation is monotonic and evolving only in one direction. In this case the increments of the degradation processes are modelled with independent random variables with a gamma distribution [31]. In [32], stochastic filtering with the Gibbs algorithm is proposed to find the hidden degradation state of the gamma process. The RUL can be estimated based on gamma process probability distribution with a predefined failure threshold. The shortcoming of the Gamma process models is that they are also restricted to the assumption of the Markov property which has been mentioned in Wiener processes.

Markov models underlying assumptions mean that the machine degradation processes should evolve in a finite state-space following the principle of the Markov property. Kharoufeh et al. proposed a hybrid RUL estimation method based on Markov models and stochastic failure models to compute the RUL distributions numerically along with their moments status. The number of the states in a Markovian model was estimated by a *K*-means clustering analysis which implied that there must be sufficient data on failures and degradation to estimate the model parameters [33]. The drawback of the Markov models is that they cannot model the hidden health states of the degradation processes. In order to overcome this, the Hidden Markov Model (HMM) has been employed for machinery prognostics [34]–[36]. Furthermore, hidden semi-Markov model was employed to improve the flexibility of the HMMs for representing complicated state transition processes in machinery degradation processes for prognostics and RUL estimation [37]–[39]. It is worth noting that the Markov models based methods suffer from the Markov property limitation, which may lead to an approximation for the true process.

Covariate-based hazard models have been employed for lifetime modeling to describe the degradation processes caused by one or more factors in the conditioning monitoring (CM) information. The Proportional Hazards Model (PHM) was proposed by Cox in [40], and has been widely used to relate the system's CM variables and external factors to the failure of a system. The PHM assumes that the hazard rate of a system is composed of two multiplicative factors, a baseline hazard function and a function of covariates. The PHM with the Weibull distribution was employed in conjunction with the condition monitoring process for RUL estimation of machinery in [41]. Tran et al. proposed a hybrid system including autoregressive moving average model, PHM and support vector machine to assess the machine health degradation and predict the RUL of the machine [42]. The limitation of the PHM is that it requires sufficient failure event data and associated CM information simultaneously. However, these two types of data may not always be available in practice.

B. MACHINE LEARNING BASED APPROACHES
In the past years, machine learning techniques, especially deep neural network-based approaches have been developed to model the highly nonlinear, complex and

multi-dimensional systems without prior knowledge of the system and machinery physical behaviors [43]. Machine learning approaches attempt to learn the machine degradation characteristics using optimization techniques from sensor raw data instead of building physics- and statistical-models.

Artificial Neural Network (ANN), and Multi-layer Perceptron (MLP) are employed for modeling the RUL of the bearings in [44], [45]. The proposed neural network takes the multiple CM measurement values at the present and past observations as the inputs and RUL as the output. Zhang et al. proposed a multi-objective deep belief networks (DBN) ensemble method for RUL estimation [46]. An evolutionary algorithm is integrated with the conventional DBN training method to evolve multiple DBNs, which are combined to establish an ensemble model for NASA's turbofan engine RUL estimation [47]. Bektas et al. proposed a neural network based method by learning a similarity model that is fed by the use of data normalisation and filtering methods for operational trajectories of complex systems for RUL estimation [48].

In the last few years, Deep Neural Networks (DNNs) are emerging as a highly effective neural network architecture for object recognition, classification tasks [49], speech recognition [50] and other applications for image quality assessment [51]. All of these tasks demonstrate that the performance of DNNs can overcome conventional methods significantly. DNNs utilize a stack of multiple layers of nonlinear processing units for feature extraction and transformation. The ''deep'' in ''deep neural network'' refers to the number of layers through which the data is transformed. Each successive layer uses the output from the previous layer as input to fully capture the representative information from raw input data [52]. The performance improvement may benefit from high-level abstractions of input data which can be modeled with the help of the complex deeper structures, leading to more efficient feature extraction compared to shallower networks.

In order to investigate the capability of DNNs for prognostics and RUL estimation for machinery, Babu et al. proposed a Convolutional Neural Network (CNN) to estimate the RUL of system based on normalized variate time series signals from sensors [53]. Two convolutional layers, two average pooling layers and one fully connected layer are adopted in the proposed DNN architecture. The experimental results indicate that the proposed CNN based method can be used to outperform three other regression methods, including the Multi-Layer Perceptron (MLP), Support Vector Regression (SVR) and Relevance Vector Regression (RVR) on publicly available datasets. Li et al. proposed a deep CNN architecture for RUL estimation by using the raw collected data with time window approach for sample preparation in order to obtain better feature extraction by the CNN [54]. However, since the collected CM signals have complex multimodalities and high heterogeneity, the CNN-based methods have difficulties extracting heterogeneous features representing the variation from running until the point of machine

failure. Li et al. proposed an additional work by using Short-time Fourier transform for time-frequency transformation and applied multi-scale feature extraction with CNN to enhance the network learning ability [55]. Li et al. in [56] proposed a cross-domain fault diagnosis method for rolling element bearing, where a deep generative model has been employed for generating artificial fault signal in target domain based on healthy signal. Then the generated artificial fault signal has been used for domain adaptation. da Costa et al. [57] proposed LSTM based domain adversarial neural network approach to performing unsupervised domain adaptation for RUL prediction, where a gradient reversal layer has been employed to perform adversarial learning during training and induce a domain-invariant representation.

Recurrent Neural Networks (RNNs) were proposed for sequence learning, which have been employed in speech recognition, machine translation, natural language processing etc. and achieved state-of-the-art performance [58]. Particularly, unlike conventional multi-layer perceptron networks that can only learn the representation from input data to target vectors, RNNs are not limited to discrete internal states and they allow for continuous, distributed sequence representations. This means that RNNs have the capability to learn the latent representation from the entire history of previous inputs to target vectors by building connections between units from a directed cycle. The benefits of RNNs allow a memory of previous states to be kept in the network's internal state. Malhi et al. proposed RNN based learning approach for long-term prognostics of machine health status monitoring [59]. The Continuous Wavelet Transform (CWT) can be employed for preprocessing of the vibration signals from a rolling bearing. Gugulothu *et al.* [60] proposed a RNN based method for RUL estimation, which embedded time series data using RNNs as an encoder and drew a health index curve with it. The RUL is predicted by comparing the latter curve with the normal health index curve.

RNNs can be trained via backpropagation through time for supervised tasks with sequential input data and target outputs. However, due to gradient exploding and vanishing problems during backpropagation of model training, training a deep RNN is difficult and may not capture the long-term dependencies from the sequential signal [61], [62]. Long Short-Term Memory (LSTM) is a type of RNN, which was designed to prevent backpropagated errors causing gradient exploding and vanishing issues in RNNs [63]. LSTM employ input gates, forget gates and output gates, where forget gates were introduced in LSTM to avoid long-term dependency problems. Considering that the LSTM networks can capture long period temporal dependencies and nonlinear dynamics in sequential signal, LSTMs have achieved great success on applications of speech recognition and machine translation [64].

The characteristics of the LSTM make it a natural choice for machinery RUL estimation due to the considerable time lag between inputs and their corresponding outputs. A simple LSTM is employed in [65] and a bidirectional LSTM

is proposed in [66] for RUL estimation. A hybrid deep learning model combining CNN and LSTM is demonstrated for machine health monitoring in [67], where a CNN is employed for local features extraction and bi-directional LSTM [68] is demonstrated and built on CNN outputs for the temporal information encoding and representation learning. Al-Dulaimi *et al.* [69] and Li *et al.* [70] proposed similar approaches, where instead of a series connection which proposed in [67], a hybrid network architecture by combining CNN and LSTM network in parallel manner for RUL estimation has been presented respectively. Particularly, in [70], a directed acyclic graph network architecture is employed in this hybrid design. The outputs from the LSTM and CNN network are summed by elements-wise and fed into a second LSTM network, then forwarded to a fully connected layer for RUL prediction. However, these hybrid network architectures could require a significant large number of network parameters for tuning and result in a large model, which may be difficult for practical applications.

Malhotra et al. proposed a LSTM-based encoder-decoder structure in [71], which is utilized to transform a multivariate input raw sequence to a fixed-length vector, then the vector is used to produce the target sequence by the decoder. The model learns to capture the behavior of a machine by learning to reconstruct multivariate time series corresponding to normal behavior in an unsupervised manner. Then, the reconstruction error is used as a health index to estimate the degradation, and in turn estimate the RUL of the machine.

In this paper, a LSTM-Fusion network for RUL estimation is proposed. Furthermore, industrial CM systems use multiple sensors and generate continuous streams of raw signals for machine health monitoring. Signal compression techniques have increasingly significant roles in modern CM systems. The effects of compression on the RUL estimation have not been previously investigated although they could have negative effects on the accuracy and robustness of the machinery degradation modeling and RUL prediction. To bridge this gap, we investigate the performance of the proposed LSTM-Fusion network by using compressed/distorted sensors data with different compression levels.

III. PROPOSED LSTM-FUSION NETWORK FOR RUL ESTIMATION

In this section, the proposed LSTM-Fusion network for RUL estimation is presented, including an overview of the standard LSTM network, and the proposed hierarchical multiscale fusion approach for better capturing the temporal dependencies in the time series sequence signal. The implementation details of our proposed method are presented in Section III-B.

A. LSTM NETWORK

LSTM as a special RNN structure has proven stable and powerful for modeling long-range dependencies for sequence modeling [62], [63]. The major novelty of LSTM is the innovation of the memory 'cell', c_t, which essentially acts as an accumulator of the state information in the sequence signal.
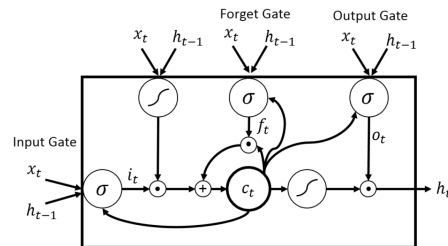

FIGURE 1. Long short-term memory cell.

The memory cell is accessed, written and cleared by several self-parameterized controlling gates. The information will be accumulated to the cell if the *'input gate'*, i_t, is activated when a new input comes. Then, the past cell status c_{t-1} could be ''forgotten'' in this process if the *'forget gate'*, f_t, is on. Whether the latest cell output c_t will be propagated to the final state h_t is further controlled by the *'output gate'*, o_t. In order to avoid the vanishing and exploding problem for the traditional RNN model, the memory cell and gates have been designed to control information flow so that the gradient will be trapped in the cell and be prevented from vanishing too quickly (also known as constant error carousels) [62], [63]. In this paper, we follow the formulation of the Fully Connected LSTM (FC-LSTM) as in [72]. The key equations to define the hidden layer function, $\mathbb{H}$, are listed in (1) below:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o)$$
$$h_t = o_t \circ \tanh(c_t) \tag{1}$$

where σ is the logistic sigmoid activeation function, 'o' denotes the Hadamard product, model parameters including $W \in \mathbb{R}^{d \times k}$ and $b \in \mathbb{R}^d$ are shared by all time steps and learned during model training, d and k is the number of sensors in each time step and dimension of hyperparameter of hidden vectors, h. Multiple LSTMs can be stacked and temporally concatenated to form more complex structures. The output at the terminal time step is used to predict the output by a linear regression layer, as shown in the equation below:

$$\bar{y}_i = W^r h_i^T \tag{2}$$

where $W^r \in \mathbb{R}^{k \times z}$ and z is the dimension of the output. The Euclidean loss has been employed as the model loss function in this task. We use the following shorthand notation to denote the recurrent LSTM operation as below:

$$(h_t, c_t) = \text{LSTM}(x_t, h_{t-1}, c_{t-1}) \tag{3}$$

The following section describe in detail the proposed LSTM-Fusion network.

B. PROPOSED LSTM-FUSION NETWORK

In order to estimate the RUL of machinery, an algorithm must learn to predict the future given only a partial temporal
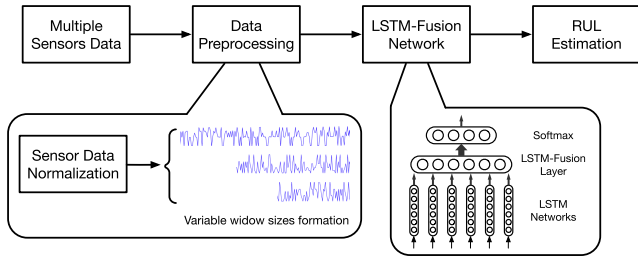
FIGURE 2. System overview.

FIGURE 3. Proposed LSTM-Fusion network structure for RUL estimation.

context. This makes the RUL estimation challenging and also differentiates it from fault detection of machinery by utilizing discriminative classifiers. The proposed network models the RUL estimation with a recurrent architecture which unfolds through time. This allows us to train a single model that learns to handle partial temporal context of varying time steps. Furthermore, machinery RUL estimation is challenging because the contextual information can come from multiple sensors with different data modalities. In such applications the way information from different sensors is fused is critical to the application's final performance. In this section, an end-to-end deep learning architecture which jointly learns to model and fuse information from different sensory signals with variable time window size is described. Figure 2 shows the overview of the proposed RUL estimation method.

Particularly, for training an RUL estimation model, we observe the time-series sequences and the labeled remaining useful life $\{(x_1, x_2, ..., x_t), y\}$. The target is to train a network which predicts the RUL given partial temporal observation samples of the time-series sequence, $\{(x_1, x_2, ..., x_t) \mid t \leq T\}$, in a sequence-to-target prediction manner, where T denotes the variable time window size. In this experiment, T ranges between [10, 100] with 20 step size. Given training examples, $\{(x_1, x_2, ..., x_t)_j, y_j\}_{j=1}^N$, where j denotes the jth time window size in N, we train the LSTM neural network to map the sequence of observations of the sensors data, $(x_1, x_2, ..., x_t)$ to the sequence of the labeled RUL, $(y_1, y_2, ..., y_t)$, such that $y_t = y, \forall t$. Trained in this manner, the neural network will attempt to map all sequences of partial observations $(x_1, x_2, ..., x_t)_j, \forall t \leq T$ to the RUL label y_t. In this way, our model explicitly learns the characteristics of the latent RUL representation with variable lengths of the observation samples, (h_t^T, c_t^T). In the next step we describe how the proposed LSTM deep fusion method fuses the latent representations.

The proposed LSTM-Fusion architecture must fuse multiple learned latent representations from training samples: $S_i = \{(x_1, ..., x_t)\}_j$. An obvious way to fuse sensory data is by concatenating the time series signal as input to the network. However, we found that this simple concatenation performs poorly. The proposed method is to add and train an additional sensory fusion layer which combines the high-level representations of sensor data. In particular, the proposed architecture first passes the sensor signals S_i with variable window size independently through a separate LSTM sub-networks
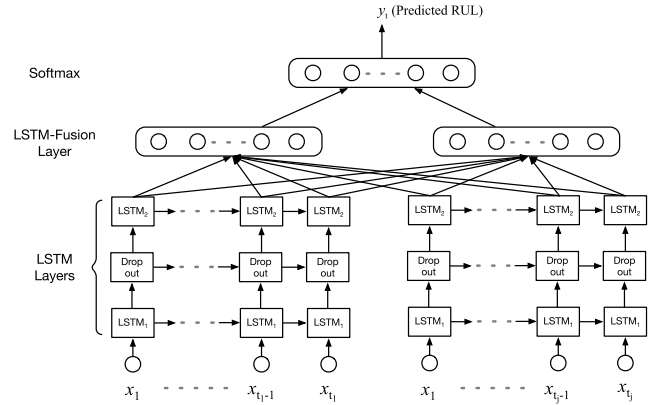
defined as in (4). The high level representations from the LSTM sub-networks $\{(h_1^x, ..., h_t^x)\}_j$, are then concatenated at each time step t and passed through a fully connected fusion layer which fuses the high level representations output from the sub-networks, the fusion layer can be obtained by (5). The proposed LSTM-Fusion architecture is shown in Figure 3. The output representation of the fusion layer, $\mathfrak{F}_t$, is then passed to the softmax layer for RUL estimation by (6).

$$\{(h_t^x, c_t^x)\}_j = \{\text{LSTM}_x(x_t, h_{t-1}^x, c_{t-1}^x)\}_j \quad (4)$$

LSTM-Fusion:

$$\mathfrak{F}_t = \tanh(W_f[\{h_t^x\}_j] + b_f) \quad (5)$$
$$y_t = \text{softmax}(W_y \mathfrak{F}_t + b_y) \quad (6)$$

where W_* and b_* are the learned model parameters, $\{\text{LSTM}_x*\}_j$ denotes the sensor signals with variable window size T and the above operations are performed from $t = [1, T]$. Particularly, for the neural network in our proposed LSTM-Fusion architecture we use multiple LSTMs of size 100 with *sigmoid* gate activation and *tanh* activation for hidden representation. The fully connected fusion layer uses *tanh* activation and outputs a 100 dimensional vector. Dropout is also applied after each LSTM layer to control overfitting (rate = 0.2). The final layer is a dense output layer with a single unit and linear activation for RUL prediction. This architecture can be extended to handle additional time series signals from multiple sensors.

The main purpose of the training process is to optimize the learned parameters, including weights and biases, such that the loss function, L, is minimized. The loss function can be obtained as follows:

$$L = \frac{1}{T} \sum_{t=1}^{T} \|\text{RUL}_{\text{Est}}^t - \text{RUL}_{\text{True}}^t\|_2^2 \quad (7)$$

where $\text{RUL}_{\text{True}}^t$ and $\text{RUL}_{\text{Est}}^t$ denotes the labeled RUL and estimated RUL at time t respectively. Particularly, the $\text{RUL}_{\text{True}}^t$ is calculated as the difference between the current time t and the end of useful time. The "*RMSprop*" optimizer has been utilized in the training process, and dropout regularization

has been employed to avoid model overfitting. The early stopping in the training process has been employed when the learned model does not show performance improvements on the validation dataset.

A solution must be found for the issue that the multiple raw sensor signals are in different units and magnitudes and the corresponding data does not have equal contributions to the neural network. Therefore, data normalization is necessary to convert all of the sensors signals to a normalized space. The Z-score standardizing method [73] has been employed to solve this problem. The Z-score measures the distance of the feature point from the mean in terms of the standard deviation, so called standardization of data.

C. PERFORMANCE EVALUATION

In order to compare this work with state-of-the-art methods, we employ two objective performance metrics, including *Scoring* function and Root Mean Square Error (RMSE) [47], [54], which are described in detail in the next paragraph.

The *Scoring* function is employed extensively for evaluating the performance of the RUL estimation methods. The metric can be described as follows:

$$S = \begin{cases} \sum_{i=1}^{N} (e^{-\frac{h_i}{13}} - 1), & \text{for } h_i < 0 \\ \sum_{i=1}^{N} (e^{\frac{h_i}{10}} - 1), & \text{for } h_i \geq 0 \end{cases} \tag{8}$$

where S is the estimated *Score*, N is the number of engines in the test dataset and h is the difference between the predicted RUL and the ground truth. The advantage of the *scoring* function is that it penalizes late prediction more than early predictions. This is beneficial since late RUL predictions are more dangerous than early predictions since they could result in delays of the maintenance operations needed by the machine. However, the drawback is that a single outlier estimation (such as a very late prediction) would dominate the overall performance score. Therefore, a single evaluation score cannot fully represent the performance of the algorithm.

Additionally, the Root Mean Square Error (RMSE) of the estimated RUL to the ground truth has been utilized to evaluate the performance of the algorithm. The RMSE has equal weight to both early and late prediction of the RUL. Therefore, the RMSE produces a better overall evaluation of the performance of the RUL estimation algorithm. The RMSE can be obtained as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} h_i^2}{N}} \tag{9}$$

IV. C-MAPSS DATASET

The NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset has been employed to evaluate the performance of the proposed method. The C-MAPSS dataset contains 4 sub-datasets as listed in Table 1. These datasets are widely used benchmark data and contain simulated data produced using a model based Turbofan engine degradation simulation program, C-MAPSS, developed by NASA [47]. The datasets include sensor data with a

TABLE 1. C-MAPSS dataset details.

Dataset	FD001	FD002	FD003	FD004
Training trajectories	100	260	100	249
Test trajectories	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

different number of operating conditions and fault conditions, as shown in Table 1. Each of these datasets have been further divided into training and testing subsets. In each dataset, the engine number, operational cycle number and operating settings are listed corresponding to the sampled data from the 21 sensors. More detail of the data structure of the datasets can be found in [47]. The engine is operating normally at the beginning of each time series, and develops a fault at some point in time which is unknown. In the training set, the fault grows until a system failure. In contrast, the data in the test set are provided up to some time prior to the system failure. The objective is to estimate the number of remaining operational cycles before the system failure based on the provided data in the test dataset.

V. EXPERIMENTAL RESULTS

In this section, we have performed extensive experiments to evaluate the proposed method. Firstly, a comparison is performed of the proposed LSTM-Fusion method with seven state-of-the-art methods, including Multi-layer Perceptron (MLP) [74], Support Vector Regression (SVR) [75] and Relevance Vector Regression (RVR) [76], CNN based regression method [53], LSTM based RNN [77], deep convolution neural network based method [54], on the raw C-MAPSS dataset. Secondly, we evaluate our proposed method with multiple LSTM layers, ranging from 1-6 LSTM layers, to show how by increasing the number of LSTM layers it is possible to extract better the high level representation. This experiment is also designed to explore the computational complexity versus performance of the proposed LSTM-Fusion architecture. Thirdly, the evaluation of the proposed LSTM-Fusion model is performed on the distorted sensors data with different compression levels.

A. EXPERIMENT 1: PERFORMANCE OF THE PROPOSED METHOD

This experiment is designed to perform a comprehensive evaluation of RUL estimation using seven different methods. The results are listed in Table 2. We report the experimental results with the same settings on the four sub-datasets. The two stacked LSTM layer structure has been employed in this experiment with 100 and 50 nodes in the first and second LSTM layer respectively. We performed 10-fold cross validation on the training and validation datasets to estimate performance of the learned model and the best model has been applied on the testing dataset. The train-test evaluations have been shuffled 10 times and we report the averaged results listed in Table 2.

TABLE 2. Performance comparisons of the proposed method and state-of-the-art methods on the C-MAPSS dataset.

| Method	RMSE				Scoring function			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
MLP [74]	37.56	80.03	37.39	77.37	1.7972×10^4	7.8028×10^6	1.7409×10^4	5.6166×10^6
SVR [75]	20.96	41.99	21.05	45.35	1.3815×10^3	5.8990×10^5	1.5983×10^3	3.7114×10^5
RVR [76]	23.79	31.29	22.37	34.34	1.5029×10^3	1.7423×10^4	1.4316×10^3	2.6509×10^4
CNN [53]	18.45	30.29	19.82	29.16	1.2867×10^3	1.3570×10^4	1.5962×10^3	7.8864×10^3
LSTM [77]	16.14	24.49	16.18	28.17	3.38×10^2	4.45×10^3	8.52×10^2	5.55×10^3
DCNN [54]	12.61	22.36	12.64	23.31	2.737×10^2	1.0412×10^4	2.841×10^2	1.2466×10^4
Li-DAG [70]	11.96	20.34	12.46	22.43	$\mathbf{2.29 \times 10^2}$	2.73×10^3	5.35×10^2	$\mathbf{3.37 \times 10^3}$
LSTM-Fusion	**11.18**	**16.12**	**10.24**	**21.97**	2.5546×10^2	$\mathbf{1.3988 \times 10^3}$	$\mathbf{2.1117 \times 10^2}$	7.7269×10^3
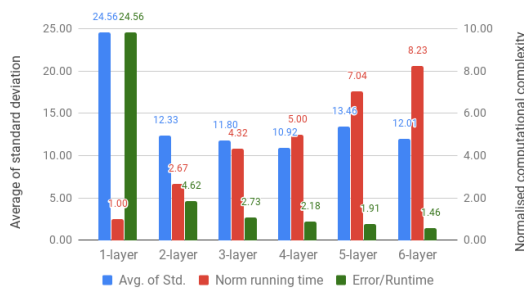
FIGURE 4. Averaged standard deviation of RUL prediction error (blue bar), Normalized computational complexity (red bar) and error to runtime ratio (green bar).
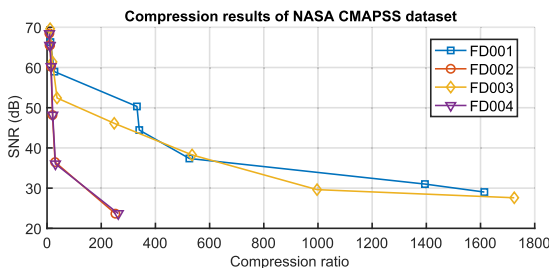

FIGURE 5. The rate-distortion performance (Compression Ratio (CR) vs. Signal to Noise Ratio (SNR)) of four datasets. The distortion levels have been generated by different bitdepth representation of the dataset, e.g. higher bitdepth representation achieves better SNR with lower CR, but lower bitdepth (4-bitdepth) achieves higher CR with lower SNR.

FIGURE 6. The average prediction error with variable bitdepth compression (solid lines) and uncompressed raw data (dash lines) of four datasets respectively.

FIGURE 7. The absolute difference of the RUL estimation on compressed data and raw data.

Table 2 shows the results of RMSE and *Score*. We compare the proposed LSTM-Fusion results with other seven state-of-the-art methods. The experimental results indicate that the proposed LSTM-Fusion outperforms all other approaches under RMSE metric evaluation, and comparable results of the *Score* value are obtained compare to the state-of-the-art methods. The experimental results show that our method outperforms slightly the method proposed in [70], this suggests that our core idea of fusing the extracted features which have variable time window sizes has merit performance for capturing the local and global characteristics of the signal for the RUL prediction.

B. EXPERIMENT 2: INFLUENCE OF THE NUMBER OF THE LSTM LAYERS

As shown in the literature, deep stacked LSTMs often obtain better prediction accuracy than shallower models. However,
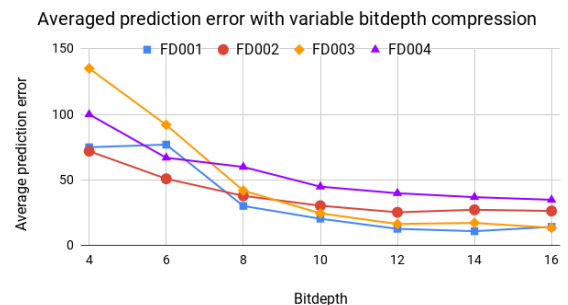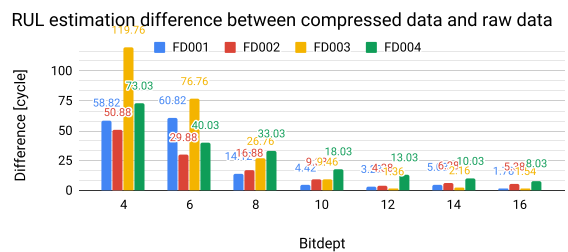
simply stacking more layers of LSTM works only to a certain number of layers, beyond which the network becomes slow and difficult to train, likely due to the exploding and vanishing gradient problem [78]. This experiment is designed to illustrate the relationship among the performance, the computational complexity and the number of layers in our proposed deep LSTM-Fusion architecture for RUL estimation.

Figure 8 and 9 shows a sample result depending on the number of the layers stacked in our proposed architecture, ranging between [1 6]. The x-axis is the actual RUL of the test engine, y-axis is the predicted RUL. The green line indicates the ground truth RUL values of the example test dataset FD001 of the test engine ID: 49. The blue curve indicates the averaged RUL prediction values by proposed LSTM-Fusion architecture. The light blue region indicates the standard deviation of all RUL predictions at each time stamp. The orange curve indicates the difference between
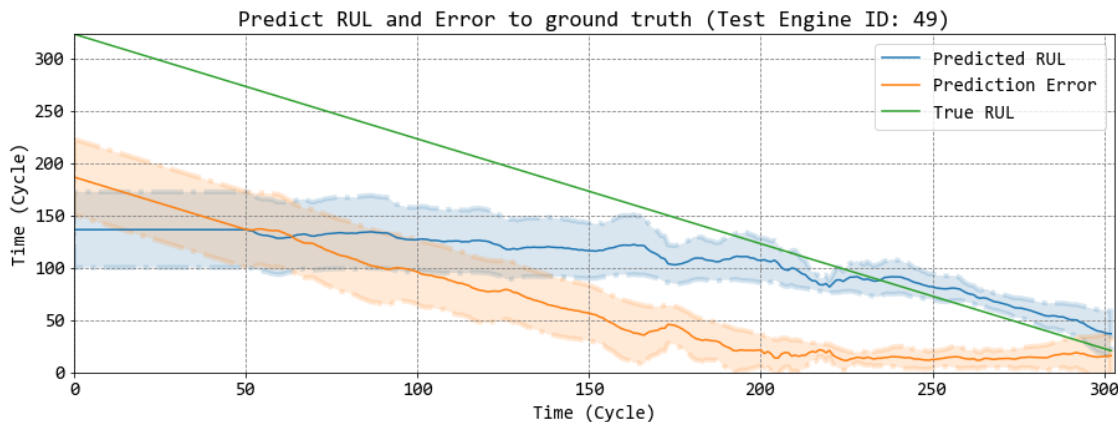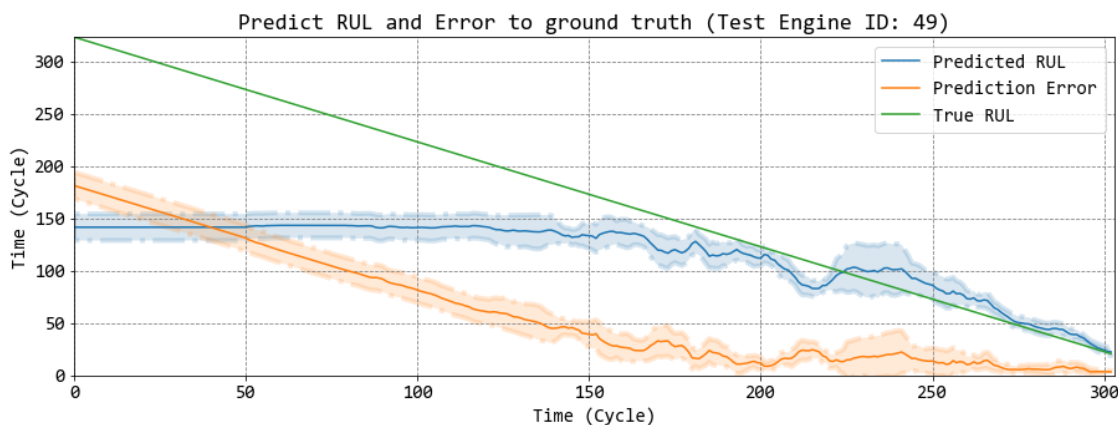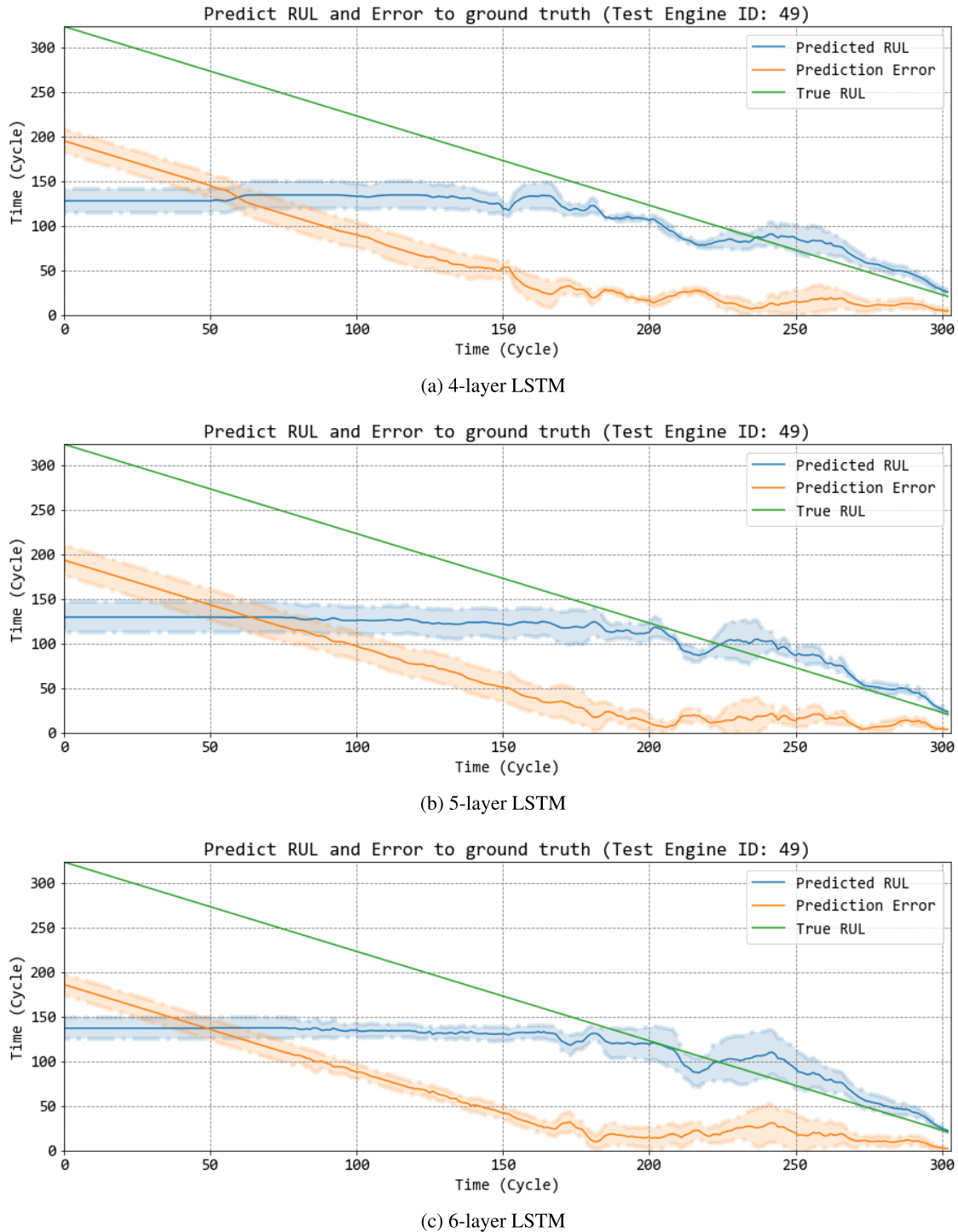
(a) 1-layer LSTM
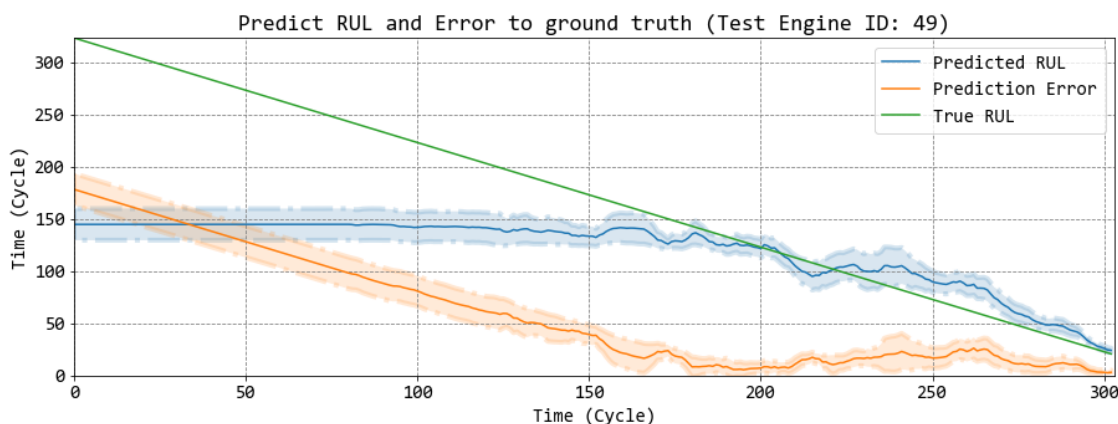
(b) 2-layer LSTM

(c) 3-layer LSTM

FIGURE 8. Performance evaluations of stacking LSTMs from 1-3 layers with proposed LSTM-Fusion architecture, including averaged RUL prediction (blue line), averaged prediction error (yellow line) and ground truth (green line). Transparent regions are error band accordingly.

predicted RUL values (blue curve) and ground truth (green line). The light orange region shows the standard deviation of the predicted errors.

As shown in Figure 8 and 9, the experimental results indicate that stacking more LSTM layers can achieve better prediction accuracy (smaller standard deviation values of light orange region), however, the vanishing returns can be observed when the number of LSTM layers stacked is higher than 4 layers. Specifically, Figure 4 shows that the relationship between the averaged standard deviation of

(a) 4-layer LSTM

(b) 5-layer LSTM

(c) 6-layer LSTM

FIGURE 9. Performance evaluations of stacking LSTMs from 4-6 layers with proposed LSTM-Fusion architecture, including averaged RUL prediction (blue line), averaged prediction error (yellow line) and ground truth (green line). Transparent regions are error band accoringly.

the prediction errors (blue bars) and normalized running time (red bars) for each LSTM layer setting. The green bars are the ratio of the corresponding blue bars and red bars to indicate the vanishing return effect of the stacking of LSTM layers. In Figure 4, the blue bars show that the accuracy of the RUL prediction can not be improved further when the number of LSTM layers is higher than 4, and the computational complexity increases linearly according to the number of the layers stacked in the proposed architecture.

In summary, the experimental results indicate that the optimal number of the LSTM layers for the proposed architecture is 4 and this number achieves a good balance between the RUL prediction accuracy and running time. It is difficult to obtain benefits by stacking more than 4 layers in the proposed method.

C. EXPERIMENT 3: INFLUENCE OF THE COMPRESSION/DISTORTION TO THE PROPOSED METHOD

This experiment is designed to evaluate the reliability and robustness of the proposed LSTM-Fusion architecture in a modern practical scenario that deploys lossy data compression - the bandwidth and storage requirements are reduced using the state-of-the-art vibration signal encoder proposed in [3] to compress the raw vibration signal. There are seven distortion levels created by compressing the vibration signal with different bitdepth representations, from 4-bitdepth (higher distortion, coarse accuracy) to 16-bitdepth (low distortion, high accuracy). All of the compression parameters and conditions have been set as originally presented in [3].

Figure 5 shows the compression ratio (CR) results and the corresponding value of signal-to-noise-ratio (SNR) for the four evaluated datasets respectively. As shown in the figure, the higher bitdepth representation (16-bitdepth) obtains better precision in the vibration signal, thus it achieves approximately 80 dB reconstruction accuracy, but it obtains lower compression ratio. On the other hand, the 4-bitdepth compressor achieves less than 30 dB reconstruction quality, but compresses the data significantly, achieving up to 1607:1 and 1721:1 compression ratio on *FD001* and *FD003* datasets respectively.

In order to evaluate the influence of compression/distortion on the proposed method, the averaged prediction errors (RMSE) of the last 50 cycles of the test datasets have been calculated corresponding to variable bitdepth representations, as shown in Figure 6. In general, the averaged prediction error decreases according to the bitdepth increment. The prediction error can be reduced significantly when the bitdepth increases from 4-bitdepth to 10-bitdepth. However, the prediction error is largely unaffected when bitdepths are higher than 10-bitdepth. This means that the proposed method can model the health status and estimate the RUL of the machine with a 10-bitdepth compressed/distorted sensor data accurately. We can also observe that the compressability of the datasets is not constant. Accuracy RUL estimation with the 10-bitdepth is calculated using on distorted/compressed data that has approximately 60.24 dB signal-to-noise-ratio (SNR) and 14.83:1 compression ration on the FD002 and FD004 datasets. For the FD001 and FD003 dataset, the proposed method can estimate the RUL accurately at 44.45 dB and 46.12 dB SNR with 340.4:1 and 247.9:1 compression ratios respectively. This may be cause by the higher complexity of the data in FD002 and FD004 (see Table 1), which contain six operating conditions while only one operating condition is present in FD001 and FD004. In addition,

Figure 7 illustrates the absolute difference of the RUL estimation evaluated with compressed and raw data. A similar effect to the previous result can be observed in Figure 7 where the difference of the estimated RUL on compressed and raw data is reduced as the precision of the compression process increases. Due the larger number of operating conditions in FD002 and FD004 datasets, the result of 16 bitdepth in Figure 7 shows higher difference on these two datasets compared to others. The experimental results indicate that the proposed approach is able to robustly predict the RUL under challenging situations in which data has been dramatically compressed/distorted due to limited bandwidth and storage requirements in the condition monitoring system.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a deep neural network approach for machinery health status modeling to perform remaining useful life (RUL) estimation. The proposed LSTM-Fusion neural network architecture addresses open challenges in the area of condition-based maintenance and RUL estimation for rotational machines. The contributions of this paper can be summarized in the following four points:

- Firstly, the proposed LSTM-Fusion architecture can fuse multi-sensors data with variable time window sizes, which allows the neural network to be able to capture the short-range (local) and long-range (global) characteristics of the data from multiple sensors. The experimental results show that the proposed LSTM-Fusion architecture achieves the best RUL estimation performance and outperforms other seven state-of-the-art methods.

- Secondly, in order to explore the optimum number of layers of the proposed LSTM-Fusion architecture, the *Exp. 2* has been designed. The experimental results indicate that stacking more LSTM layers can achieve better RUL prediction accuracy, however the computational complexity increases linearly. Moreover, the vanishing returns of the RUL estimation performance can be observed when the number of layers is higher than 4-layer.

- Thirdly, the proposed method is capable of achieving robust health status modeling and RUL estimation on the raw and compressed dataset. Particularly, the experimental results presented in Section V-C indicate that the proposed method can accurately estimate the RUL with distorted/compressed data which has up to 340.4:1 compression ratio with 44.45 dB signal-to-noise-ratio on the FD001 dataset. This enables remote prognostic and condition monitoring applications to be deployed with lower bandwidth and storage requirements.

- Fourthly, the proposed novel end-to-end deep LSTM-Fusion architecture for RUL estimation does not require prior knowledge of the condition-based health monitoring and does not rely on any empirical assumptions, such as "piece-wise linear RUL target function" as it was required in [53], [77].

Future work involves the integration of the proposed method in a power generation plant and its evaluation with streaming multi-channel time series signals. To ensure that the research done in this paper can be reproduced and encourage further work we have made the models available at https://github.com/eejlny/LSTM_FUSION

REFERENCES

[1] K. Worden, T. Baldacchino, J. Rowson, and E. J. Cross, "Some recent developments in SHM based on nonstationary time series analysis," *Proc. IEEE*, vol. 104, no. 8, pp. 1589–1603, Aug. 2016.

[2] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, May 2018.

[3] Y. Zhang, P. Hutchinson, N. A. J. Lieven, and J. Nunez-Yanez, "Optimal compression of vibration data with lifting wavelet transform and context-based arithmetic coding," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017.

[4] C. R. Farrar and K. Worden, *Structural Health Monitoring: A Machine Learning Perspective*. Hoboken, NJ, USA: Wiley, Nov. 2012.

[5] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006.

[6] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mech. Syst. Signal Process.*, vol. 25, no. 5, pp. 1803–1836, Jul. 2011.

[7] H. Hanachi, J. Liu, A. Banerjee, Y. Chen, and A. Koul, "A physics-based modeling approach for performance monitoring in gas turbine engines," *IEEE Trans. Rel.*, vol. 64, no. 1, pp. 197–205, Mar. 2015.

[8] P. Paris and F. Erdogan, "A critical analysis of crack propagation laws," *J. Basic Eng.*, vol. 85, no. 4, pp. 528–533, Dec. 1963.

[9] Y. Lei, N. Li, S. Gontarz, J. Lin, S. Radkowski, and J. Dybala, "A model-based method for remaining useful life prediction of machinery," *IEEE Trans. Rel.*, vol. 65, no. 3, pp. 1314–1326, Sep. 2016.

[10] M. A. Haile, J. C. Riddick, and A. H. Assefa, "Robust particle filters for fatigue crack growth estimation in rotorcraft structures," *IEEE Trans. Rel.*, vol. 65, no. 3, pp. 1438–1448, Sep. 2016.

[11] Y. Qian, R. Yan, and R. X. Gao, "A multi-time scale approach to remaining useful life prediction in rolling bearing," *Mech. Syst. Signal Process.*, vol. 83, pp. 549–567, Jan. 2017.

[12] M. Corbetta, C. Sbarufatti, A. Manes, and M. Giglio, "Real-time prognosis of crack growth evolution using sequential Monte Carlo methods and statistical model parameters," *IEEE Trans. Rel.*, vol. 64, no. 2, pp. 736–753, Jun. 2015.

[13] F. Zhao, Z. Tian, and Y. Zeng, "Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method," *IEEE Trans. Rel.*, vol. 62, no. 1, pp. 146–159, Mar. 2013.

[14] W. M. Ostachowicz and M. Krawczuk, "Coupled torsional and bending vibrations of a rotor with an open crack," *Arch. Appl. Mech.*, vol. 62, no. 3, pp. 191–201, Jan. 1992.

[15] C. H. Oppenheimer and K. A. Loparo, "Physically based diagnosis and prognosis of cracked rotor shafts," *Proc. SPIE*, vol. 4733, pp. 122–133, Jul. 2002.

[16] Y. Hu, P. Baraldi, F. D. Maio, and E. Zio, "Online performance assessment method for a model-based prognostic approach," *IEEE Trans. Rel.*, vol. 65, no. 2, pp. 718–735, Jun. 2016.

[17] P. Baraldi, F. Mangili, and E. Zio, "Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data," *Rel. Eng. Syst. Saf.*, vol. 112, pp. 94–108, Apr. 2013.

[18] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation—A review on the statistical data driven approaches," *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 1–14, Aug. 2011.

[19] Y. Zhang, P. Hutchinson, N. A. Lieven, and J. Nunez-Yanez, "Adaptive event-triggered anomaly detection in compressed vibration data," *Mech. Syst. Signal Process.*, vol. 122, pp. 480–501, May 2019.

[20] Y. Qian and R. Yan, "Remaining useful life prediction of rolling bearings using an enhanced particle filter," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 10, pp. 2696–2707, Oct. 2015.

[21] M. Ibrahim, N. Y. Steiner, S. Jemei, and D. Hissel, "Wavelet-based approach for online fuel cell remaining useful lifetime prediction," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5057–5068, Aug. 2016.

[22] C. J. Lu and W. O. Meeker, "Using degradation measures to estimate a time-to-failure distribution," *Technometrics*, vol. 35, no. 2, pp. 161–174, May 1993.

[23] M. Gašperin, A. Juričić, P. Boškoski, and J. Vižintin, "Model-based prognostics of gear health using stochastic dynamical models," *Mech. Syst. Signal Process.*, vol. 25, no. 2, pp. 537–548, Feb. 2011.

[24] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method," *J. Power Sources*, vol. 196, no. 23, pp. 10314–10321, Dec. 2011.

[25] E. Rabiei, E. L. Droguett, and M. Modarres, "A prognostics approach based on the evolution of damage precursors using dynamic Bayesian networks," *Adv. Mech. Eng.*, vol. 8, no. 9, Sep. 2016, Art. no. 168781401666674.

[26] M. D. Pandey, X.-X. Yuan, and J. M. Van Noortwijk, "The influence of temporal uncertainty of deterioration on life-cycle management of structures," *Struct. Infrastruct. Eng.*, vol. 5, no. 2, pp. 145–156, Apr. 2009.

[27] D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes*. Evanston, IL, USA: Routledge, 1965.

[28] G. A. Whitmore and F. Schenkelberg, "Modelling accelerated degradation data using wiener diffusion with a time scale transformation," *Lifetime Data Anal.*, vol. 3, no. 1, pp. 27–45, Jan. 1997.

[29] X. Si, T. Li, and Q. Zhang, "A general stochastic degradation modeling approach for prognostics of degrading systems with surviving and uncertain measurements," *IEEE Trans. Rel.*, vol. 68, no. 3, pp. 1080–1100, Sep. 2019.

[30] X. Si, T. Li, Q. Zhang, and C. Hu, "Prognostics for linear stochastic degrading systems with survival measurements," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3202–3215, Apr. 2020.

[31] H.-K. Wang, H.-Z. Huang, Y.-F. Li, and Y.-J. Yang, "Condition-based maintenance with scheduling threshold and maintenance threshold," *IEEE Trans. Rel.*, vol. 65, no. 2, pp. 513–524, Jun. 2016.

[32] K. Le Son, M. Fouladirad, A. Barros, E. Levrat, and B. Iung, "Remaining useful life estimation based on stochastic deterioration models: A comparative study," *Rel. Eng. Syst. Saf.*, vol. 112, pp. 165–175, Apr. 2013.

[33] J. P. Kharoufeh and S. M. Cox, "Stochastic models for degradation-based reliability," *IIE Trans.*, vol. 37, no. 6, pp. 533–542, Jun. 2005.

[34] C. Bunks, D. Mccarthy, and T. Al-Ani, "Condition-based maintenance of machines using hidden Markov models," *Mech. Syst. Signal Process.*, vol. 14, no. 4, pp. 597–612, Jul. 2000.

[35] F. Camci and R. B. Chinnam, "Health-state estimation and prognostics in machining processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 581–597, Jul. 2010.

[36] E. Ramasso and T. Denoeux, "Making use of partial knowledge about hidden states in HMMs: An approach based on belief functions," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 2, pp. 395–405, Apr. 2014.

[37] S.-Z. Yu, "Hidden semi-Markov models," *Artif. Intell.*, vol. 174, no. 2, pp. 215–243, Feb. 2010.

[38] R. Moghaddass and M. J. Zuo, "An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process," *Rel. Eng. Syst. Saf.*, vol. 124, pp. 92–104, Apr. 2014.

[39] Q. Liu, M. Dong, W. Lv, X. Geng, and Y. Li, "A novel method using adaptive hidden semi-Markov model for multi-sensor monitoring equipment health prognosis," *Mech. Syst. Signal Process.*, vols. 64–65, pp. 217–232, Dec. 2015.

[40] D. R. Cox, "Regression models and life-tables," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 34, no. 2, pp. 187–220, 1972.

[41] D. Banjevic and A. K. S. Jardine, "Calculation of reliability function and remaining useful life for a Markov failure time process," *IMA J. Manage. Math.*, vol. 17, no. 2, pp. 115–130, Apr. 2006.

[42] V. T. Tran, H. T. Pham, B.-S. Yang, and T. T. Nguyen, "Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine," *Mech. Syst. Signal Process.*, vol. 32, pp. 320–330, Oct. 2012.

[43] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mech. Syst. Signal Process.*, vol. 107, pp. 241–265, Jul. 2018.

[44] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," *Mech. Syst. Signal Process.*, vol. 21, no. 1, pp. 193–207, Jan. 2007.

[45] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *J. Intell. Manuf.*, vol. 23, no. 2, pp. 227–237, Apr. 2012.

[46] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017.

[47] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–9.

[48] O. Bektas, J. A. Jones, S. Sankararaman, I. Roychoudhury, and K. Goebel, "A neural network filtering approach for similarity-based remaining useful life estimation," *Int. J. Adv. Manuf. Technol.*, vol. 101, nos. 1–4, pp. 87–103, Mar. 2019.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[50] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[51] S. Jia and Y. Zhang, "Saliency-based deep convolutional neural network for no-reference image quality assessment," *Multimedia Tools Appl.*, vol. 77, no. 12, pp. 14859–14872, Jun. 2018.

[52] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[53] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Database Systems for Advanced Applications* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, Apr. 2016, pp. 214–228.

[54] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.

[55] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Rel. Eng. Syst. Saf.*, vol. 182, pp. 208–218, Feb. 2019.

[56] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5525–5534, Jul. 2019.

[57] P. R. D. O. Da Costa, A. Akçay, Y. Zhang, and U. Kaymak, "Remaining useful lifetime prediction via deep domain adaptation," *Rel. Eng. Syst. Saf.*, vol. 195, Mar. 2020, Art. no. 106682.

[58] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[59] A. Malhi, R. Yan, and R. X. Gao, "Prognosis of defect propagation based on recurrent neural networks," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 703–711, Mar. 2011.

[60] N. Gugulothu, V. TV, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff, "Predicting remaining useful life using time series embeddings based on recurrent neural networks," Sep. 2017, *arXiv:1709.01073*. [Online]. Available: https://arxiv.org/abs/1709.01073

[61] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning longterm dependencies," in *A Field Guide to Dynamical Recurrent Networks*. New York, NY, USA: IEEE Press, 2001.

[62] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1310–1318.

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

[64] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," Sep. 2016, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144

[65] C.-S. Hsu and J.-R. Jiang, "Remaining useful life estimation using long short-term memory deep learning," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Apr. 2018, pp. 58–61.

[66] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Chongqing)*, Oct. 2018, pp. 1037–1042.

[67] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, p. 273, Jan. 2017.

[68] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[69] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "Hybrid deep neural network model for remaining useful life estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3872–3876.

[70] J. Li, X. Li, and D. He, "A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction," *IEEE Access*, vol. 7, pp. 75464–75475, 2019.

[71] P. Malhotra, V. TV, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder," Aug. 2016, *arXiv:1608.06154*. [Online]. Available: https://arxiv.org/abs/1608.06154

[72] A. Graves, "Generating sequences with recurrent neural networks," Aug. 2013, *arXiv:1308.0850*. [Online]. Available: https://arxiv.org/abs/1308.0850

[73] M. M. Suarez-Alvarez, D.-T. Pham, M. Y. Prostov, and Y. I. Prostov, "Statistical approach to normalization of feature vectors and clustering of mixed datasets," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 468, no. 2145, pp. 2630–2651, Sep. 2012.

[74] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[75] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, May 2011.

[76] M. E. Tipping, "The relevance vector machine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, p. 7.

[77] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2017, pp. 88–95.

[78] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, p. 5.

• • •