2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)

Estimation of remaining useful life using Convolutional Neural Network (CNN)

Shabaz Malik
CSE Department, School of Engineering and technology
Sharda University
Greater Noida, Uttar Pradesh 201310, India
2017012950.shabaz@ug.sharda.ac.in

Abdul Ahad
CSE Department, School of Engineering and technology
Sharda University
Greater Noida, Uttar Pradesh 201310, India
2017014308.abdul@ug.sharda.ac.in

Vipin Sharma
CSE Department, School of Engineering and technology
Sharda University
Greater Noida, Uttar Pradesh 201310, India
2017005791.vipin@ug.sharda.ac.in

Ankit
CSE Department, School of Engineering and technology
Sharda University
Greater Noida, Uttar Pradesh 201310, India
2017013991.ankit@ug.sharda.ac.in

***Abstract*—We systematically summarize the latest literature on Remaining Useful Life (RUL) turbofan engine prediction with deep learning (DL) algorithms in this survey Research. While conventional methods of Turbofan engine maintenance require manual checking of engine time to time. Although traditional machine learning (ML) approaches, including artificial neural network, main component analysis, support vector machines, etc have been successfully applied in the estimation of RUL. In this survey we provide a brief review of estimating RUL using machine learning, in addition we obtain a more accurate and efficient method for RUL estimation. And we will also conclude the benefits of using ML over the conventional techniques used for ages**

Keywords*—*Multivariate time series analysis, Deep Learning, Convolutional Neural Networks, Supervised Learning, Regression Methods, Prognostics, Remaining Useful Life

Introduction

In condition-based maintenance for various application areas such as manufacturing, aerospace, automotive, heavy industry, power generation, and transportation, forecasting technologies are very crucial. Here, we are referring to the Turbofan Engine Because they are key aircraft components, it is crucial to improve the safety, reliability and economy of engines. The formatter will need to create these components, incorporating the applicable criteria that follow.

I. Problem Overview

Ensuring a flight Protection and repair costs during the operation of aircraft engines, a method of forecasting and health management that In order to address the issues, the emphasis is on fault identification, health assessment, and life prediction. Forecasting the remainder Useful Life (RUL) is the most important knowledge for the operation and maintenance and decision making of aircraft engines. It relies primarily on the option of features for performance degradation. The selection of such characteristics is extremely important, but there are There are some shortcomings in the current RUL prediction algorithm, notably, the inability to obtain data trends. Extracting useful degradation features from multisensory data with complex correlations is a key feature, especially with aircraft engines. A technological issue that has delayed the application of the evaluation of deterioration.

II. Progonistic

Deep learning has to overcome these problems, Multiple layers of nonlinear knowledge processing for unsupervised self-learning have been proposed in recent years to exploit Characteristics. This Research introduces a profound learning technique to predict an aircraft engine's RUL based on a stacked sparse Logistic regression and auto-encoder. To automatically extract output degradation, the stacked sparse autoencoder is used to Features on the aircraft engine from multiple sensors and to fuse multiple features by self-learning multilayer. Logistic the Logistic Regression is used to estimate the useful life that remains. Nevertheless, the deep learning hyperparameters, which are important in most cases, the effect on feature extraction and prediction efficiency is calculated on the basis of expert experience. This Research introduces the grid search method to optimize the hyperparameters of the proposed RUL estimation of the aircraft engine. About model. To illustrate the feasibility of the proposed methodology, an application of this method of predicting the RUL of an aircraft engine with a benchmark dataset is used.

Since they are the key components of the aircraft, the failure of the aircraft Engines are also a significant cause of major accidents and injuries, Therefore, the safety and reliability of the engines are important. Important for aircraft quality. It is difficult, however, to Ensure their safety and reliability due to their complexity Structures, and engine failures have eventually occurred. Effects of aging, climate,

ISBN: XX-X-XX-XX-X/19/$31.00 ©2021 IEEE

and variable loading Working time is rising. It is necessary, for this purpose, to Detecting the underlying deterioration, estimating the speed of the engine Failed successfully, initiated timely repairs, and in the end, avoid a catastrophic failure. In the field of aircraft repair, conventional maintenance is either strictly reactive (fixing or repairing an aircraft). Failed engine component) or blindly constructive (assuming a certain level of performance degradation without input) The aircraft engine itself and the repair of the aircraft Engine on a regular schedule, if maintenance is actually completed Required or not). Both situations are rather inefficient. Inefficient, and neither is carried out in real time in view of Scheduling repair activities on the basis of a fault diagnosis, Output degradation assessment and prediction assessment the continued useful life of the aircraft equipment and the need Prognosis and Health Management (PHM) are increasingly being replaced in order to avoid pre-existing faults. The plan. Prognosis, as the foundation of PHM, includes controlling the process of degradation of results or faults in PH. Aircraft engine and predictions for components / systems the engine will be broken or the output will be broken It will hit an intolerable amount. RUL prediction methods are composed of three primary classes: data-driven. methods, model-based methods for physics, methods that combine physics and data-driven methods and Methods focused on the model Data-driven techniques are used by Data from past condition reporting, the existing health status of the device, and the degradation data for similar systems. System-specific approaches based on physics models use Awareness of processes, control of failures and conditions Monitoring information to predict a device or component's RUL. Prognostics-based problems are two big obstacles. On physics: There is not enough knowledge of physics to Constructing a model of physical deterioration and the values the parameters of the physical model are hard to determine. Exactly that. It is therefore important to understand the failure. Proper system mechanisms and experienced staff are needed for physics-based models. Additionally, during system operation, the peripheral environment the success of several data-driven prognostics the methods rely heavily on the collection of the performance degradation data to which they are applied. Engines have several sensor parameters, however. The sensitiveness of the information from various sensors varies in terms of display. Degradation of engine performance; data from some sensors It is fragile and the knowledge from other sensors is not delicate. Reasonable sensor parameters must therefore be selected. Whose data is more sensitive to the performance of the engine Trend of degradation as the training details for the RUL prediction About model. By observing the characteristic variations of the details, the quadratic fitting curve of all sensor parameters is used to suit the degradation information from various sensors and ranks. Deep learning, a new methodology that has been proposed It has been possible to obtain multilevel characteristics in recent years. From data, which implies that the system can convey data at Different abstraction levels. Deep learning is a machine learning system that is end-to - end. It is able to process automatically, Identify a discriminative and pattern feature in an initial signal the layer of input data by layer, and then the output directly Outcome of classification regression. The entire functionality phase Training in the learning and classifier regression model is based on Optimization of an objective total purpose. There are a number of methods of deep learning: deep neural Networks (DNNs), deep neural coevolutionary networks (CNNs), networks of deep beliefs (DBNs) and so on, A new deep learning architecture for prognostic RUL estimation is proposed in this Research. For sample collection, the time window method is used for better feature extraction by CNN. Measurements of raw sensors with normalization are directly used There is no need for prior experience in prognostics and signal processing as model inputs to the proposed network, which enables the industrial implementation of the proposed method. be calculated. Making use of time the proposed method is supposed to be a window, data normalization and deep CNN structure, in contrast with conventional machine learning approaches, achieve greater prognostic accuracy. Comprehensive review of the methodology proposed and contrasts with current approaches in this report, are presented. In recent years, the advancement of modern aeronautical technology has led to Towards a Complex In very tough environments, the aircraft system requires a high degree of reliability, performance and safety. The engine is the aircraft's main component and there is always an urgent need to Develop new methods to better assess the deterioration of engine performance and estimate the remaining lifetime of use. The RUL for aero-engines is calculated in this Research as a case of the analysis and the best-known NASA C-MAPSS dataset available to the public are used to validate the efficiency of the approach suggested. Comparisons of the state-of-the-art studies on the superiority of the network proposed is shown by the same dataset.

III. EXISTING SYSTEM

[1]LeCun first suggested coevolutionary neural networks (CNNs) for image processing, and It has two functions, i.e. weights exchanged spatially and spatial pooling. CNNs have their own Important performance in many fields of science and industry, including computer vision, [25], processing natural language, comprehension of speech [33] and so on. Coevolutionary ones with raw input data, layers combine multiple filters and generate features, and the following Afterwards, pooling layers remove the most important local characteristics. The data from the input is CNNs usually have 2-dimensional (2D) data to learn abstract spatial characteristics by alternating Stacking and convolutional kernels and process of pooling.

The input data is prepared in 2D format in this analysis, where one dimension is the number of the feature and the other is the time sequence of each feature. However, the relationship between the spatially adjacent features in the data sample is not remarkable considering the collected machinery features are from different sensors in this prognostic problem. Therefore, the convolution filters in the proposed network are 1-dimensional (1D) in practice, while the input and the corresponding function maps have 2 dimensions. The 1D CNN is introduced briefly below.

It is assumed that the 1-dimensional sequential data input is x = [x1, x2, ..., xN] where N denotes the sequence length. The convolution operation in the convolution layer can be described as a multiplying operation between a filter kernel w, w, RFL and a xi:i+FL-1 representation of the concatenation vector, which can be expressed as,

Where xi:i+FL−1 represents a sequential FL-long window starting from the i-th stage, and each data sample is concatenated into a longer embedding. The final convolution operation shall be described as

Where * T denotes the transpose of a matrix,* b and φ are the bias term and the function of non-linear activation, respectively. The output zi can be considered as the learned function of the filter kernel w in the corresponding subsequence xi:i+FL−1. By dragging the filter window from the first point to the last point in the sample data, the j-th filter feature map can be obtained, which is denoted as;

Where j is the filter kernel j-th. In CNNs, multiple filter kernels with different FL filter lengths can be added to the convolution layer. The effects on the network efficiency of the filter number and duration

Usually, a pooling layer is added to the function maps created by the coevolutionary layer. On the one hand, the pooling will extract the most relevant local information in the coevolutionary layer. A map of each feature. On the other hand, the dimensionality of the function, that is, the model number Parameters can be substantially lowered by this process. For very high-dimensional problems such as image processing, pooling is therefore well adapted. However, although this operation can enhance computational performance, visible useful information is to some degree filtered. Hence, despite the widespread use of pooling in the neural network of convolution, in this prognostic problem, pooling is not suggested where the raw feature dimension is relatively low.

The CNN deep learning architecture for RUL estimation from multi-variate sensor signals is provided in this section. For time series. Normalized sensor signals are the inputs in Except for the extracted features corresponding to the history of the operating condition. THE Target values are the system's RUL at the relevant time cycle. The target that has been considered as described in previous sections, the RUL function is a piece-wise linear function. Convolutionary neural networks have a great potential to recognize the various salient networks. Sensor Signals patterns. Specifically, the local layer processing units obtain the local layer processing units Salience of signals. Signals. Processing units of the higher layers obtain the salient patterns Elevated-level representation of signals. Note that there may be a number of each layer as described below, convolution or pooling operators (specified by various parameters) are jointly regarded as multiple On CNN, salient patterns learned from various aspects are jointly considered. When these operators with the same parameters are applied to local signals (or their mapping) in various time segments, a type of translation invariance is used. Consequently, instead of their positions or scales, what matters is only the salient patterns of signals. In the RUL estimate, however, we are confronted with several Time series signal channels, where the traditional CNN cannot be directly used. The challenges in our problem include (i) it is necessary to apply processing units in CNN Throughout the temporal dimension and (ii) the sharing or unification of CNN units between multiple with sensors. We will define the convolution and pooling filters along the way, in what follows. Temporal dimension, and then present the complete CNN architecture used in RUL Estimation.

IV. Proposed System

We're starting with the CNN notes that are used. In order to follow a sliding window technique, Segment the signal from the time series into a set of short signal bits. In concrete words, A two-dimensional matrix containing r data samples is an example used by CNN. Each sample with D attributes (Sub-data sets D in case of a single operating condition Attributes are taken as raw sensor signals in the case of multiple operating conditions and D attributes of sub-data sets included d raw sensor signals along with extracted features. As explained in the operating state, referring to the operating condition history Subsection in the section on problem settings). Here, r is selected to be the sampling rate as (15 used in the tests as there are only 15 times for one of the test engine trajectories Cycle data samples), and the sliding window phase size is selected to be 1. One may be permitted to reduce the number of cases for less computational computation, choose larger phase sizes Free. Rate. The real RUL of the matrix instance is calculated for training data by the true RUL of the matrix instance The Last Record RUL. Conventional CNN is updated in this proposed architecture, as shown in Figure 3, Multi-variate time series regression is implemented as follows: we jointly conduct feature learning on each segmented multivariate time series. At the end of learning the function, we standard multi-layer perceptron (MLP) for RUL estimation is concatenated. Especially in We use 2 pairs of convolution layers and pooling layers for this job, and one regular layer.Multi layered perceptron, totally connected. This includes inputs from the D-channel and the duration of Each input amounts to 15. This segmented time series of multivariates (D x 15) is fed into a 2-Stages of layers of convolution and pooling. Then, all end layer features are concatenated, As the MLP input for RUL estimation, it maps into a vector. The preparation stage requires the training process. Estimation of CNN parameters by the normal back propagation algorithm using stochastic propagation algorithm Method of gradient descent to optimize objective function, cumulative square, The CNN mode error

Convolution Layer: In the convolution layers, feature maps of the previous layer are Converged (to be learned in the training process) with several convolutional kernels. Performance applied by a bias (to be learned) and a feature map of the convolution operators the next layer is determined by the activation function. The map of output features of Computed convolution layer as shown below:

$$\mathbf{x}_j^l = sigm\left(\mathbf{z}_j^l\right), \quad \mathbf{z}_j^l = \sum_i \mathbf{x}_i^{l-1} * \mathbf{k}_{ij}^l + b_j^l$$

Where ∗ denotes the convolution operator, x l−1 i and x l j are the convolution filter input and output, *sigm*() denotes the sigmoid function, and z l j is the input of non-linear sigmoid function

Pooling Layer: The input features are sub-sampled by sufficient samples in the pooling layers in order to increase the invariance of features on the inputs for distortions, the resolution of the feature maps is reduced. Without overlapping, we utilize average pooling for the entire stage of our work. The average pooling of the input feature maps is partitioned A set of non-overlapping regions and results. The output for each sub-region is Value average. The output layer pooling feature map is computed as shown below.

$$\mathbf{x}_j^{l+1} = down\left(\mathbf{x}_j^l\right)$$

Where x l j is the input and x l+1 j is the output of pooling layer, and down(.) represents the sub-sampling function for average pooling.

V. Training Process

Forward Propagation: The purpose of the forward propagation is to decide the On segmented multi-variate time series data, the CNN model expected performance. In particular, maps of each layer's output function are computed. Each step contains a convolution layer, followed by a pooling layer, as stated in the previous sections. Using above Eqs, we compute the performance of convolution and pooling layers. . Finally, with feature extract, a single completely connected layer is linked

Backward Propagation: Once one iteration of forward propagation is done, we will have the error value, with the squared error loss function. The predicted error propagates back on each layer parameters from last layer to first layer, derivatives chain commonly applied for this procedure .For the backward propagation of errors in the second stage pooling layer, the xl−1 j's derivative is calculated by the up-sampling function up(.), it is an inverse operation of the sub-sampling function down

Deep neural networks are capable of extracting information from the representation of Raw input signals through so many non-linear and complex transformations approximate transformations Non-linear functions, and this analysis uses them as the core architecture. The proposed deep learning approach consists of two sub-structures in general., i.e. multiple neural convolutions, Regression networks and a fully-connected layer. In above Figure Proposed method of calculating RULs demonstrates the architecture.

Secondly, the 2-dimensional (2D) format of the input data sample is prepared to enable the implementation of convolution operations. The input dimension is Ntw x Nf t, where Ntw the time series dimension is denoted and the number of characteristics selected is Nf t. The Crude Features are normally obtained from several measurements of sensors. First, 4 convolution layers are stacked in the network for feature extraction. The specifics of the data. Four of the layers They have the same design used by FN filters and the filter size is FL 1. the padding operation is introduced. So Far, the performance obtained is FN feature maps whose size is Ntw x Nf t which is the same with an initial sample of the input. To merge the convolution layer, we use another convolution layer with 1 filter. To be a new one, previous function maps are used. The scale of the filter is 3 x 1. In this respect, the high-level Representation is obtained for each raw function. The 2-dimensional feature map is then flattened and connected to a completely connected function. Layer, layer. Notice that the dropout method is used on the last map of functions, i.e. the flattened sheet, to remove overfitting. Finally, at the end of the network proposed for RUL estimation, one neuron is linked. Tanh is used as the activation function of all layers, and Xavier's standard initializer is used for weight initializations

To further boost the performance of the prognosis,
A fine-tuning approach is implemented using the algorithm of back-propagation (BP), where the back-propagation algorithm is used To minimize the training error, the parameters of the model proposed are revised. Adam for optimization, the algorithm is used. It should be noted that the process of

convolution is performed in 1 dimension only, i.e. the time series, whereas the 2D convolution neural network is used for function extraction. For each function, dimension. Initially, the multiple stacked convolution layers are then targeted at for each raw function, learn the high-level representations and the completely linked ones, respectively. For the final regression, the layer uses all the studied depictions of the characteristics. Compared with most of the current deep CNN prognostic methods that aim to learn the spatial at the outset, relationships of various characteristics and further details are extracted from the with several layers of learned abstract representation, the proposed technique is more fitting for the extraction of features from various measurements of sensors.

A .Methodology

This segment presents the applicable calculations utilized in this research. the entire method for RUL expectation for an airplane motor comprises of two primaries steps: information pre-processing and RUL expectation.

Data Pre-processing. Determination of sensors that are sensitive to performance degradation and normalization of sensor information with various measurements is the essential errands important to acquire a high RUL expectation exactness. Two stages are expected to pre-process the information

Selection of Sensor. In an aircraft engine, numerous sensors have very different responses to the phase of performance degradation. Because of noise or insensitivity to deterioration patterns, some sensors exhibit ambiguous tendencies. Choosing insensitive parameter data could decrease the accuracy of RUL prediction. Sensors that are more susceptible to the performance degradation process are chosen as inputs to the RUL prediction model to increase the performance of the prediction model. For sensitivity assessment, a procedure called slope analysis is suggested. The three key phases are the following:

Phase 1: For each parameter of each engine, curve fitting is done on the degradation data. Then, to analyse the sensitivity of the degradation results, the parameters of the best-fit curves are used, called slopes.

Phase 2: In step 1, the average values of all the engine parameters belonging to the same sensor are computed.

Phase 3: With higher slopes, the degradation data is Picked for the engine RUL prediction

Data Normalization. The linear function that best preserves the aircraft engine 's original performance degradation pattern is selected to map the data for each chosen sensor to [0, 1].

X1=0.8(x-a)/(b-a) +0.1

X1=Normalised data

A=Minimum value data

B=Maximum Value data

RUL Normalization. The proposed prediction technique yields an outcome in the reach from 0 to 1. In the preparation stage of the prediction model, the RUL of each cycle of airplane engine should be normalised to [0, 1] using a linear function. The test outputs of the prediction model need to be inversely mapped from [0, 1] to the real RUL

VI. DATASET

C-MAPSS Dataset: The proposed method is assessed in this Research on a prognostic benchmarking problem, I.e. Degradation problem of the NASA turbofan engine [30, 41]. Includes this famous dataset Simulated data that a model-based simulation program generates, i.e. NASA-developed Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). The CMAPSS dataset consists of 4 sub-datasets consisting of multi-variate temporal data collected from 21 sensors. There is one training set and one test set in each sub-dataset. The training datasets provide records of run-to-failure sensors of multiple aero-engines gathered under various operating conditions and fault modes. Each engine unit starts with distinct degrees of initial wear and variance in manufacturing that is unknown and considered to be healthy. As time passes, before the machine failures are reached, the engine units start to degrade, i.e. the last data entry corresponds to the time period considered unhealthy by the engine unit. On the other hand, some time before system failure, the sensor records in the testing datasets terminate and the purpose of this task is to estimate the remaining useful life of each engine in the test dataset.

Two data indexes picked in this work, in particular the NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) informational collection and the PHM 2008 Data Challenge informational index [19]. The C-MAPSS informational index is additionally separated into 4 sub-informational indexes as given in Table 1. Both datasets contain simulated data produced using model-based stimulation program C-MAPSS created by NASA [20].

Both data sets are orchestrated in a n-by-26 matrix where n relates to the quantity of data points in every part. Each column is a depiction of information taken during a single working time cycle and in 26 columns, where 1st column speaks to the motor number, 2nd column speaks to the operational cycle number, 3 - 5 columns represent the three working settings and 6 - 26 columns speak to the 21 sensor qualities. More data about the 21 sensors can be found in [22]. Motor execution can be affected by three working settings in the data altogether. Each trajectory inside the train and test trajectories is thought to be life-cycle of a motor. While every motor is simulated with various beginning conditions, these conditions are viewed as typical conditions (no flaws). For every motor trajectory inside the training sets, the last data sets relate to the moments motor is pronounced failure status. Then again, test sets contain data sometime before the failure and the point here is to anticipate RUL in the test set for every motor. For every one of the C-MAPSS data set, the real RUL estimation of the test trajectories was made accessible to people in general, while the real RUL estimation of the test trajectories in PHM 2008 Data Challenge data set isn't accessible. To reasonably think about the assessment model execution on the test data, we need a few target executions measures. In this work, we essentially utilize 2 measures: scoring capacity, and Root Mean Square Error (RMSE), which are presented in subtleties as follows:

Data sets	FD001	FD002	FD003	FD004	Phm 2008
Train Trajectories					
Operating Conditions					
Faults Conditions					

A. Scoring function: The scoring function utilized in this Research is indistinguishable from that utilized in PHM 2008 Data Challenge. This scoring function is represented in Eq. (1), where N is the quantity of motors in test set, S is the figured score, and h = (Estimated RUL− T mourn RUL).

$$S = \begin{cases} \sum_{i=1}^{N} \left(e^{-\frac{h_i}{13}} - 1\right) & for \;\; h_i < 0 \\ \sum_{i=1}^{N} \left(e^{\frac{h_i}{10}} - 1\right) & for \;\; h_i \geq 0 \end{cases}$$

This scoring feature further penalizes late predictions (too late for maintenance) Early predictions (no major harm, although maintenance resources may be wasted). This is in line with the aerospace industry 's risk-adverse mindset. Nonetheless, there is There are some limitations to this feature. A single outlier (with a much late prediction) would dominate the overall performance score, the most significant drawback being (pls. refer to the rapid increase in the right side of Figure 1), thereby masking the real overall accuracy of the algorithm. Another downside is the lack of concern for the algorithm's projection period. The prognostic horizon is evaluated Period before failure that the algorithm can correctly estimate the value of the RUL at a certain degree of trust. At the end of the day, this score function favours algorithms Artificially eliminates the score by underestimating RUL. In spite of all these shortcomings, the score feature is still used in this Research to provide comparable results with other scores. Methods of literature

B. RMSE: The Root Mean Square Error (RMSE) of, in addition to the scoring function, as a performance metric, approximate RUL's are also appointed. RMSE as it is picked Both early and late predictions are given equal weight. Using RMSE in association with the scoring feature would avoid preferring an algorithm that artificially decreases the Score by underestimating it, but with a higher RMSE performance. The RMSE is characterized as Indicated below

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

VII. CONCLUSION

Clearly, accurate estimation of RUL has great benefits and advantages in many real world applications across different industrial verticals. As the first attempt to adapt deep learning to estimate RUL for prognostic problem, this paper investigated a novel deep architecture CNN based regressor to estimate the RUL of complex system from multivariate time series data. This proposed deep architecture mainly employs the convolution and pooling layers to capture the salient patterns of the sensor signals at different time scales. All identified salient patterns are systematically unified and finally mapped into the RUL in the estimation model. To evaluate the proposed algorithm, we examined its empirical performance on two public data sets and our experimental results shows that it significantly outperforms the existing state-of-the-art shallow regression models that have been utilized extensively for RUL estimation in literature. As in our future study, we would like to further explore novel deep learning techniques to tackle a variety of emerging real-world problems in prognostics field.

ACKNOWLEDGMENT

The authors would like to thank Dean Frederick for providing valuable insight into the operation of C-MAPSS dataset.

REFERENCES

[1] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8), 1798–1828(Aug 2013)

[2] Bengio, Y.: Learning deep architectures for AI. Foundations and trends R in Machine Learning 2(1), 1–127 (2009)

[3] Bouvrie, J.: Notes on convolutional neural networks (November 2006), http://cogprints.org/5869/1/cnn_tutorial.pdf

[4] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. pp. 160–167. ACM (2008)

[6] Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks 5(2), 240–254 (1994)

[7] Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing (2014)

[8] Fukushima, K.: Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics 36(4), 193–202 (1980)

[9] Heimes, F.O.: Recurrent neural networks for remaining useful life estimation. In: International Conference on Prognostics and Health Management, 2008. PHM 2008. pp. 1–6 (Oct 2008)

[10] G. Sateesh Babu, Peilin Zhao, Xiao-Li Li

[11] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE 29(6), 82–97 (2012)

[12] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105(2012)

[13] LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on.pp. 253–256 (May 2010)

[14] LeCun, Y., Bengio, Y.: The handbook of brain theory and neural networks. chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. MIT Press, Cambridge, MA, USA (1998)

[15] LeCun, Y., Bottou, L., Orr, Genevieve, B., Müller, K.R.: Efficient backprop. In: Orr, G.,Müller, K.R. (eds.) Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science, vol. 1524, pp. 9–50. Springer Berlin Heidelberg (1998)

[16] Lim, P., Goh, C.K., Tan, K.C., Dutta, P.: Estimation of remaining useful life based on switching kalman filter neural network ensemble. In: Annual Conference of the prognostics and Health Management Society 2014. pp. 1–8 (2014)

[17] Peel, L.: Data driven prognostics using a kalman filter ensemble of neural network models. In International Conference on Prognostics and Health Management, 2008. PHM 2008. pp.1–6 (Oct 2008)

[18] Ramasso, E., Saxena, A.: Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset. In: Annual Conference of the Prognostics and Health Management Society 2014. pp. 1–11 (2014)

[19] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Neurocomputing: Foundations of research. chap. Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, Cambridge, MA, USA (1988), http://dl.acm.org/citation.cfm?id=65669104451

[20] Saxena, A., Goebel, K.: PHM08 challenge data set, NASA AMES prognostics data repository. Tech. rep., Moffett Field, CA (2008)

[21] Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In: International Conference on Prognostics and Health Management, 2008. PHM 2008. pp. 1–9 (Oct 2008)

[22] Tipping, M.E.: The relevance vector machine. In: Solla, S.A., Leen, T.K., Müller, K.R. (eds.) Advances in Neural Information Processing Systems. vol. 12, pp. 652–658. MIT Press (2000)

[23] Wang, P., Youn, B.D., Hu, C.: A generic probabilistic framework for structural health prognostics and uncertainty management. Mechanical Systems and Signal Processing 28, 622–637 (2012)

[24] Wang, T., Yu, J., Siegel, D., Lee, J.: A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In: International Conference on Prognostics and Health Management, 2008. PHM 2008. pp. 1–6 (Oct 2008)

[25] Yang, J.B., Nguyen, M.N., San, P.P., Li, X.L., Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: Proceedings of the 24th International Conference on Artificial Intelligence. pp. 3995–4001. AAAI Press (2015)

[26] Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., Zhang, J.: Convolutional neural networks for human activity recognition using mobile sensors. In: Mobile Computing, Applications and Services (MobiCASE), 6th International Conference on. pp. 197–205.IEEE (2014)

[27] Zheng, Y., Liu, Q., Chen, E., Ge, Y., Zhao, J.L.: Time series classification using multichannels deep convolutional neural networks. In: Web-Age Information Management, pp.298–310. Springer (2014)