

CRANFIELD UNIVERSITY

Youdao Wang

Optimisation of Deep Learning Techniques on Remaining Useful Life
Prediction of Complex Engineering Systems

Cranfield University
School of Aerospace, Transport and Manufacturing

PhD

Academic Year: 2019 - 2022

Primary Supervisor: Dr Yifan Zhao
Associate Supervisor: Dr Pavan Sri Addepalli

August 2022

CRANFIELD UNIVERSITY

Cranfield University
School of Aerospace, Transport and Manufacturing
PhD

Academic Year 2019 - 2022

Youdao Wang
Optimisation of Deep Learning Techniques on Remaining Useful Life
Prediction of Complex Engineering Systems
Primary Supervisor: Dr Yifan Zhao
Associate Supervisor: Dr Pavan Sri Addepalli
August 2022

This thesis is submitted in partial fulfilment of the requirements for the
degree of PhD

***(NB. This section can be removed if the award of the degree is based
solely on examination of the thesis)***

© Cranfield University 2022. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

ABSTRACT

Predictive maintenance based on performance degradation is a crucial way to reduce the operation and maintenance costs and potential failures in modern complex engineering systems. Reliable remaining useful life (RUL) prediction is the main criterion for predictive maintenance decisions. Data-driven techniques, especially artificial intelligence (AI) such as deep learning (DL) techniques, have attracted more and more attention in the manufacturing sector with the development of Big Data and Internet-of-Things. Different DL techniques recently have been used for RUL prediction and achieved great success. However, in many cases, the RUL prediction results vary greatly due to the measurement noise and selection of model parameters. This PhD research aims to develop a novel framework to optimise the performance of deep learning methods in the context of predicting the RUL of complex engineering systems. The project consists of four stages including literature review, optimisation investigations, construction of the new prognostic framework and validation. In the first stage, state-of-the-art DL-based approaches for RUL prediction are reviewed. Then, the optimisation investigations on different RNN models, feature engineering, model parameters and RUL target functions are carried out trying to improve the RUL prediction accuracy. After that, the prognostic framework is built based on the result of the optimisation investigation. Meanwhile, a novel three-stage feature selection method and a multi-scale RUL prediction approach are proposed in this stage. In addition, to accommodate the multiple operating conditions of complex engineering systems, this thesis presents an operation-based normalisation method to address the different degradation patterns from data. In the last stage, the C-MAPSS dataset is adopted to validate the proposed framework and the prediction performance is compared with the state-of-the-art RUL prediction approaches. A significant improvement can be observed in the RUL prediction performance using the proposed framework on most of the subsets of the C-MAPSS dataset. Therefore, a reliable and flexible RUL prediction strategy can be made based on this DL-based prognostics framework for complex engineering systems

Keywords: Remaining useful life prediction, Deep Learning, Recurrent Neural Network, Long Short-term Memory, Feature engineering, C-MAPSS

ACKNOWLEDGEMENTS

First and foremost, I'd like to show my deepest gratitude to my supervisor, Dr. Zhao Yifan, a respectable, responsible, and wise teacher. You have been taking care of me since the first day I arrived at the university. To me, you are a boss, a mentor, a friend, and most of the time a model I've most looked up to. You are everything I imagine a good scholar would be like and more. I have learned so much from you either as a professional or in my personal life. You provided me with valuable guidance in every stage of my PhD project including the writing of this thesis.

Next, I'd like to thank my associate Supervisor, Dr Pavan Sri Addepalli. I still remember how impressed I was by your perfect British accent the day we met. Language has never been one of my talents, I was very diffident about my English. You helped me a lot not only with my project but also with my English level.

Then, I want to show my sincerest gratitude to my parents for their unconditional love and support during my entire PhD duration. You believed in me even when I didn't. My most humble gratitude to my dear wife and daughter, you are always the strength in my heart that pushes me to become better.

At last, I'd also like to extend my thanks to all my colleges and dear friends, Weixiang Du, Lichao Yang, Jiaru Zhou, Jun cao, Kailun Deng, Aoxiang Dong, Suozhi Wang. Came to UK alone, you guys make my life much more colourful and make me much less homesick, especially during the Covid lockdown.

TABLE OF CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	IV
LIST OF FIGURES	VIII
LIST OF TABLES	X
LIST OF EQUATIONS	XI
LIST OF ABBREVIATIONS	XIV
1 CHAPTER ONE—INTRODUCTION	19
1.1 RESEARCH BACKGROUND.....	20
1.2 RESEARCH MOTIVATION.....	22
1.3 AIM AND OBJECTIVES	23
<i>Objective 1—Literature review.....</i>	<i>23</i>
<i>Objective 2—Parameter optimisation.....</i>	<i>23</i>
<i>Objective 3— Feature engineering.....</i>	<i>24</i>
<i>Objective 4— Prognostic model</i>	<i>24</i>
<i>Objective 5— Framework construction</i>	<i>24</i>
1.4 METHODOLOGY	25
1.5 THESIS STRUCTURE	26
1.6 MAIN CONTRIBUTIONS	27
2 CHAPTER TWO—LITERATURE REVIEW	28
ABSTRACT	29
2.1 INTRODUCTION.....	30
2.1.1 RUL prediction approaches.....	30
2.1.2 Research gaps	34
2.2 DEEP LEARNING APPROACHES	35

2.2.1 Auto-encoder	36
2.2.2 Deep Belief Network.....	37
2.2.3 Convolutional Neural Network.....	39
2.2.4 Recurrent Neural Network.....	41
2.3 CONCLUSION	43
ACKNOWLEDGEMENTS	44
REFERENCES.....	45
3 CHAPTER THREE—PARAMETER OPTIMISATION	53
ABSTRACT	54
3.1 INTRODUCTION	55
3.2 RECURRENT NEURAL NETWORK AND ITS VARIANTS.....	57
3.2.1 Recurrent Neural Network.....	57
3.2.2 Long Short-Term Memory (LSTM).....	59
3.2.3 Gated Recurrent Unit	61
3.2.4 Bi-directional LSTM.....	62
3.2.5 Research gaps.....	63
3.3 OPTIMISER	64
3.3.1 Adaptive Gradient (AdaGrad)	65
3.3.2 AdaDelta.....	66
3.3.3 RMSprop.....	67
3.3.4 Adam.....	67
3.4 ACTIVATION FUNCTION	68
3.4.1 The Sigmoid & hyperbolic tangent functions	69
3.4.2 Rectified linear unit function	69
3.4.3 Swish function.....	71
3.5 CASE STUDY	71
3.5.1 Benchmark dataset overview	71
3.5.2 Data pre-processing	72
3.5.3 Performance evaluation	72
3.6 CONCLUSIONS.....	80
REFERENCES.....	81

4 CHAPTER FOUR—FEATURE SELECTION	84
ABSTRACT	85
KEYWORDS	85
4.1 INTRODUCTION.....	86
4.2 METHODOLOGY	89
4.2.1 <i>Data Exploration</i>	90
4.2.2 <i>Three-stage Feature selection method</i>	91
4.2.3 <i>Construction of the predictive model</i>	93
4.2.3.1 <i>Long Short-Term Memory</i>	93
4.2.3.2 <i>Parameter selection</i>	94
4.2.4 <i>Benchmark dataset</i>	95
4.2.5 <i>Performance evaluation</i>	95
4.3 RESULTS	96
4.3.1 <i>Data exploration</i>	96
4.3.2 <i>Parameter optimisation</i>	97
4.3.3 <i>Performance of the proposed three-stage feature selection method</i>	101
4.3.4 <i>Performance of RUL prediction</i>	104
4.4 CONCLUSION.....	108
REFERENCES	110
APPENDIX: NORMALISATION APPROACHES	113
5 CHAPTER FIVE—PROGNOSIC MODEL	115
ABSTRACT	116
KEYWORDS	116
5.1 INTRODUCTION.....	117
5.2 DATASET DESCRIPTION	121
5.3 METHODOLOGY	124
5.3.1 <i>Feature selection</i>	125
5.3.2 <i>Data normalisation</i>	127
5.3.3 <i>RUL target function</i>	128
5.3.4 <i>Multi-scale RUL prediction</i>	129
5.4 RESULTS	131

5.4.1 <i>Feature selection</i>	131
5.4.2 <i>LSTM model for 3-stage classification</i>	134
5.4.3 <i>LSTM-based RUL prediction for the stage 2 and stage 3</i>	136
5.5 CONCLUSION	140
REFERENCES	142
6 CHAPTER SIX –OVERALL DISCUSSION	147
6.1 RESEARCH GAPS FILLED	148
6.2 CONTRIBUTION TO THE KNOWLEDGE	148
6.2.1 <i>The exploration of the RUL prediction approaches</i>	149
6.2.2 <i>The optimisation of the DL-based model for RUL prediction</i>	149
6.2.3 <i>The construction of a novel prognostic framework</i>	150
6.3 CONCLUSIONS TO THE RESEARCH FINDINGS	150
6.4 LIMITATIONS AND FUTURE WORK	152

LIST OF FIGURES

<i>Figure 1-1 Frameworks of physical-based, Conventional data-driven and DL-based RUL prediction approaches</i>	<i>20</i>
<i>Figure 1-2 Methodology of the project.....</i>	<i>25</i>
<i>Figure 3-1 Basic Feed-Forward Neural Network structure and Recurrent Neural Network structure.....</i>	<i>58</i>
<i>Figure 3-2 Basic structures of RNN, GRU and LSTM.....</i>	<i>60</i>
<i>Figure 3-3 Cell state and three gates in LSTM.....</i>	<i>60</i>
<i>Figure 3-4 A Bi-directional LSTM Structure.....</i>	<i>63</i>
<i>Figure 3-5 Different activation functions</i>	<i>68</i>
<i>Figure 3-6 Performance of different optimisers on the testing dataset.....</i>	<i>74</i>
<i>Figure 3-7 Performance of different activation functions on the test dataset</i>	<i>75</i>
<i>Figure 3-8 Performance of different sequence lengths on the test dataset.....</i>	<i>76</i>
<i>Figure 3-9 Performance of different Neuron numbers on the test dataset.....</i>	<i>78</i>
<i>Figure 3-10 Performance of different algorithms on Dataset FD001, where Optimiser: Adam; Activation function: ReLU; Neuron Number:128; Sequence Length: 50</i>	<i>79</i>
<i>Figure 4-1 Flow chart of the proposed RUL prediction framework</i>	<i>90</i>
<i>Figure 4-2 Cell state (Left) and three gates (Right) in LSTM</i>	<i>94</i>
<i>Figure 4-3 Result of k-medoids clustering model on FD002 (a: operation condition distribution; b: Setting values in the 6 operation conditions)</i>	<i>97</i>
<i>Figure 4-4 RUL prediction performance of FD002 using different RNN-based algorithms</i>	<i>98</i>
<i>Figure 4-5 RUL prediction performance using different optimisers for FD002.....</i>	<i>99</i>
<i>Figure 4-6 RUL prediction performance using different activation functions for FD002</i>	<i>100</i>
<i>Figure 4-7 The input signals after applying the operation-based normalisation.....</i>	<i>101</i>
<i>Figure 4-8 Comparison of 3 selected input variables without normalisation, with the global normalisation, and with the operation-based normalisation approach</i>	<i>102</i>
<i>Figure 4-9 Spearman's rank coefficients between each input and the output. (a): based on the global normalisation, (b): based on the operation-based normalisation</i>	<i>103</i>

<i>Figure 4-10 Pair-wise Pearson correlation coefficient for the input channels using (a) the global normalisation and (b) the operation-based normalisation</i>	<i>104</i>
<i>Figure 4-11 The predicted RUL vs the true RUL of each cycle and each engine for the (a) training and (b) testing dataset</i>	<i>105</i>
<i>Figure 4-12 Prediction results for all engines with lifecycles no less than 100.</i>	<i>106</i>
<i>Figure 4-13 Uncertainty quantification of the RUL prediction using the proposed method by combining all 1114 engines</i>	<i>107</i>
<i>Figure 5-1 Simplified diagram of the simulated engine.</i>	<i>121</i>
<i>Figure 5-2 Schematic representation of the CMAPSS model (Chao et al., 2021)</i>	<i>122</i>
<i>Figure 5-3 The flowchart of the proposed multi-scale RUL prediction. (a) the process of the 3-stage classification (low-scale RUL prediction), (b) the overall framework of multi-scale RUL prediction, and (c), the process of the high-scale RUL prediction.....</i>	<i>125</i>
<i>Figure 5-4 RUL Target Functions. (a) 1-stage RUL target function, (b) 2-stage RUL target function, and (c) the proposed 3-stage RUL target function</i>	<i>128</i>
<i>Figure 5-5 Cell state and three gates in LSTM</i>	<i>130</i>
<i>Figure 5-6 The sensor observations till the failure of Engine 2 in the dataset FD001</i>	<i>132</i>
<i>Figure 5-7 The correlation coefficients between the selected sensors of Engine 2 in the dataset FD001, where the discarded sensor is highlighted.....</i>	<i>133</i>
<i>Figure 5-8 The selections of p_1 and p_2 for four datasets</i>	<i>135</i>
<i>Figure 5-9 RUL Prediction results of the first 18 engines in the training set.....</i>	<i>137</i>
<i>Figure 5-10 Uncertainty quantification of the RUL prediction using the proposed method by combining all engines for each sub-dataset.</i>	<i>138</i>

LIST OF TABLES

<i>Table 2-1 Advantages and drawbacks of prognostic approaches</i>	<i>31</i>
<i>Table 3-1 C-MAPSS dataset.....</i>	<i>72</i>
<i>Table 3-2 Assessment of different optimisers using network structure RNN-1LAYER</i>	<i>73</i>
<i>Table 3-3 Optimal optimisers for different model structures.....</i>	<i>73</i>
<i>Table 3-4 Optimal activation functions for different model structures.....</i>	<i>74</i>
<i>Table 3-5 Optimal sequence length for different model structures</i>	<i>76</i>
<i>Table 3-6 Optimal Neuron number for different model structures.....</i>	<i>77</i>
<i>Table 3-7 The average MSE was recorded with regard to different parameters.</i>	<i>79</i>
<i>Table 4-1 C-MAPSS dataset.....</i>	<i>95</i>
<i>Table 4-2 Mean and standard deviation of all the variables of dataset FD002 (training set)</i>	<i>96</i>
<i>Table 4-3 RUL prediction performance (RMSE) without and with normalisation methods</i>	<i>98</i>
<i>Table 4-4 RMSE values of the RUL prediction of the final cycle for all engines with the global normalisation and the proposed operation-based normalisation.....</i>	<i>107</i>
<i>Table 4-5 Improvement of the proposed method for state-of-the-art methods.....</i>	<i>108</i>
<i>Table 5-1 C-MAPSS dataset.....</i>	<i>123</i>
<i>Table 5-2 Descriptive statistics of all the variables of dataset FD001</i>	<i>123</i>
<i>Table 5-3 Correlation values between each input and the output RUL for Engine 2 in the dataset FD001, where the discarded sensors are highlighted.....</i>	<i>132</i>
<i>Table 5-4 Feature selection process for the datasets FD002 – FD004</i>	<i>133</i>
<i>Table 5-5 Results of the binary classification of datasets FD001-FD004.....</i>	<i>136</i>
<i>Table 5-6 RUL prediction performance with different window lengths (wl) for both the train set and the test set (RMSE) (First 80% of engines in the train set are used for training and the other 20% are used for validation)</i>	<i>139</i>
<i>Table 5-7 Results of the RUL prediction performance between the proposed method and the state-of-the-art, where the best prediction of each dataset is highlighted....</i>	<i>139</i>

LIST OF EQUATIONS

(3-1) 58

(3-2) 58

(3-3) 60

(3-4) 60

(3-5) 60

(3-6) 60

(3-7) 60

(3-8) 61

(3-9) 62

(3-10) 62

(3-11) 62

(3-12) 62

(3-13) 62

(3-14) 62

(3-15) 62

(3-16) 65

(3-17) 65

(3-18) 65

(3-19) 66

(3-20) 66

(3-21) 66

(3-22) 66

(3-23) 66

(3-24) 66

(3-25) 66

(3-26) 66

(3-27) 67

(3-28) 67

(3-29) 67

(3-30) 67

(3-31)	67
(3-32)	67
(3-33)	67
(3-34)	69
(3-35)	69
(3-36)	70
(3-37)	70
(3-38)	70
(3-39)	70
(3-40)	71
(3-41)	72
(4-1)	91
(4-2)	91
(4-3)	92
(4-4)	92
(4-5)	92
(4-6)	94
(4-7)	94
(4-8)	94
(4-9)	94
(4-10)	94
(4-11)	94
(4-12)	<i>Error! Bookmark not defined.</i>
(5-1)	124
(5-2)	126
(5-3)	126
(5-4)	127
(5-5)	127
(5-6)	129
(5-7)	129
(5-8)	129
(5-9)	129

<i>(5-10)</i>	<i>130</i>
<i>(5-11)</i>	<i>130</i>
<i>(5-12)</i>	<i>130</i>
<i>(5-13)</i>	<i>130</i>
<i>(5-14)</i>	<i>130</i>
<i>(5-15)</i>	<i>130</i>
<i>(5-16)</i>	<i>130</i>
<i>(5-17)</i>	<i>131</i>
<i>(5-18)</i>	<i>131</i>

LIST OF ABBREVIATIONS

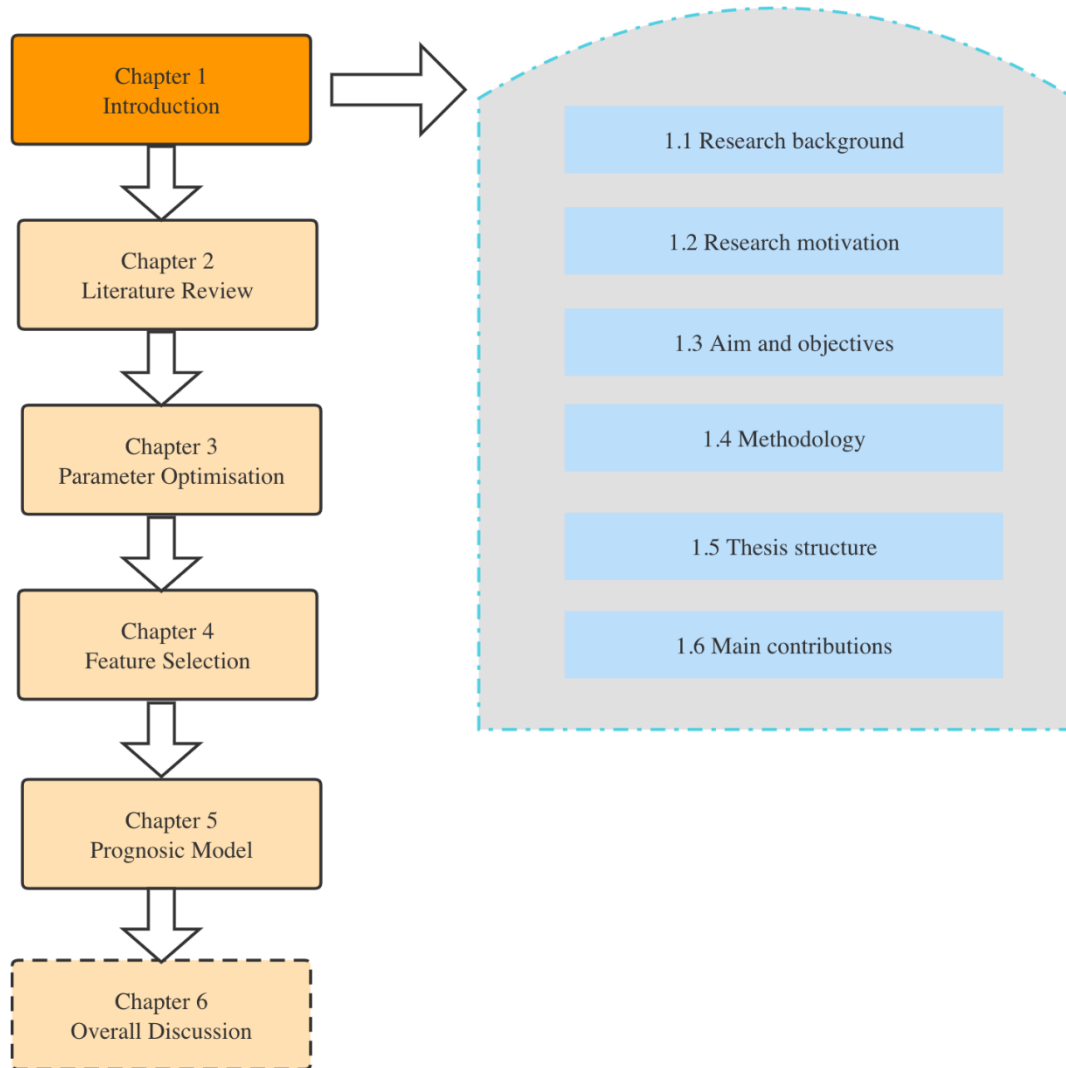
ACO-DDBN	Discriminative Deep Belief Network with Ant Colony Optimisation
AE	Auto-Encoders
AEC	Auto-Encoder-Correlation
AEQRNN	Autoencoder Quasi-Recurrent Neural Networks
AI	Artificial Intelligence
ANN	Artificial Neural Network
BHLSTM	Bidirectional Handshaking Long Short-Term Memory
BiGRU	Bidirectional Gated Recurrent Unit
BGD	Batch Gradient Descent
BiGRU	Bidirectional Gated Recurrent Unit
BMS	Boltzmann Machines
BOHB	Bayesian Optimization and HyperBand
CaConvNet	Causal Augmented Convolution Network
CAE	Convolutional Autoencoder
CBLSTM	Convolutional Bi-directional Long Short-Term Memory
CBM	Condition-Based Maintenance
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CNN	Convolutional Neural Networks
CVAE	Conditional Variational Autoencoder
DAG	Directed Acyclic Graph
DBN	Deep Belief Network
DDFF	Double Dynamic Forgetting Factors
DELTA	Degradation-aware Long short-term memory Autoencoder

DL	Deep Learning
DNN	Deep Neural Network
EA	Evolutionary Algorithm
ELU	Exponential Linear Unit
FFT	Fast Fourier Transform
FNN	Feedforward Neural Network
GRU	Gated Recurrent Unit
HI	Health Indicator
HPC	High-Pressure Compressor
HPT	High-Pressure Turbine
ICA	Independent Components Analysis
IoT	Internet of Things
KPCA	Kernel Principle Component Analysis
LDA	Linear Discriminant Analysis
LPC	Low-Pressure Compressor
LPT	Low-Pressure Turbine
LR	Logistic Regression
LSTM	Long Short-Term Memory
MLP	Multiple Layer Perceptron
MMoE	Multi-gate Mixture-of-Experts
MODBNE	Multi-Objective Deep Belief Networks
MSE	Mean Square Error
NN	Neural Network
OS-ELM	Online Sequential Extreme Learning Machine

PCA	Principal Component Analysis
PdM	Predictive Maintenance
PF	Particle Filter
PHM	Prognostics and Health Management
PoF	Physics of Failure
PreLU	Parametric Rectified Linear Unit
RBM	Restricted Boltzmann Machine
RCNN	Recurrent Convolutional Neural Network
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
RUL	Remaining Useful Life
RVM	Relevance Vector Machines
SAE	Sparse Auto-Encoder
SDA	Stacked Denoising Autoencoder
SGD	Stochastic Gradient Descent
SOM	Self-organizing Map
SSH	Single Scale-High
SSL	Single Scale-Low
SVM	Support Vector Machines
TR	Transfer Regression
USS	Updated Selection Strategy
WAFTR	Weibull Accelerated Failure Time Regression

WPT	Wavelet Packet Transform
-----	--------------------------

1 CHAPTER ONE—INTRODUCTION



1.1 Research background

The remaining useful life (RUL) of a component or a system is regarded as the length from the current time to the end of the usable life (Si *et al.*, 2011). The concept of the RUL has been widely used in operational research, reliability and statistics literature with important applications in many fields, such as material science, biostatistics, econometrics, military and manufacturing. In today's manufacturing industry, RUL prediction is indispensable because it establishes the basis for preventative and predictive maintenance planning and resource management (Zhang *et al.*, 2018a). The RUL of a system is a random variable that largely depends on the current condition of the system and the operational environment. Generally, RUL prediction approaches in the literature can be categorised into physics-based, data-driven, and hybrid approaches.

Physics-based approaches, also called model-based approaches, assess the system's health by building mathematical models based on the failure mechanisms or the first principle of damage (Cubillo *et al.*, 2016a). Data-driven approaches are most widely used in RUL prediction, where RUL is computed through statistical and probabilistic methods by utilizing historical information and routinely monitored data of the system (Kabir *et al.*, 2012). The hybrid RUL prediction is a collection methodology and technique of any former approaches.

Because of the complicated and noisy working conditions, it is challenging to build physical models of complex and dynamic systems (Pršić *et al.*, 2017). Unable to be updated with the online measured data is another limitation of the application of physics-based models. However, fully embracing the internet of things (IoT) and the big data revolution, data-driven RUL prediction approaches have become more and more attractive with the significant development of sensors, sensor networks and computing systems (Zhao *et al.*, 2019a). Machine learning techniques have been the most widely used tools to extract useful information from big data in recent years. Moreover, as the most popular branch of machine learning, significant growth can be found in the applications of Deep Learning (DL) in modern manufacturing fields. Due to the capability of addressing large-scale data and learning multi-level representation, DL techniques are regarded as an effective solution for RUL prediction.

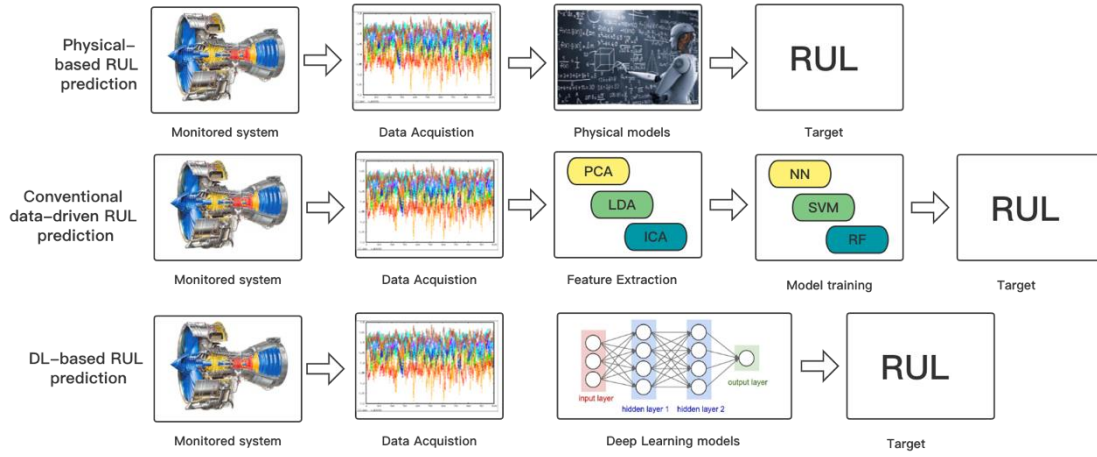


Figure 1-1 Frameworks of physical-based, Conventional data-driven and DL-based RUL prediction approaches

Figure 1-1 demonstrates the principles behind three kinds of RUL prediction methods: physical-based RUL prediction, conventional data-driven RUL prediction and DL-based RUL prediction. The physical-based RUL prediction approaches mainly use the physical models to simulate the monitored system, and predict the RUL based on the experiment results. The conventional data-driven RUL prediction methods contain two key parts, feature extraction and model training. Commonly used feature extraction method includes principal component analysis (PCA) (Khalid *et al.*, 2014), linear discriminant analysis (LDA) (Pohar *et al.*, 2004) and independent components analysis (ICA) (Lee *et al.*, 2004). The selected features are then fed to some shallow machine learning algorithms such as neural network (NN), support vector machines (SVM) and random forest (RF). It is complicated to perform the feature extraction process of complex systems and often requires a significant amount of expertise from the practitioner. Addressing this problem, DL-based RUL prediction methods are designed to extract hierarchical representations from input data through deep neural networks with multiple layers of nonlinear transformations (Zhao *et al.*, 2019a). All model parameters are trained jointly, so no extensive human labor and knowledge for the feature extraction method are demanded. The research on DL-based architectures is fast-growing, with new models being presented weekly. In the field of RUL prediction, the research is mainly focused on four primary deep learning techniques and their variants currently, including auto-encoders (AE), Restricted Boltzmann machine (RBM), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

1.2 Research motivation

The published literature on DL approaches to RUL prediction mainly focused on four representative deep architectures, including AE, RBM, CNN and RNN. AE can learn a new representation of the data through reconstructing the input data, which contains two phases: encoder and decoder (Ma *et al.*, 2018). A DBN is a stack of Restricted Boltzmann Machines (RBMs), which includes Boltzmann Machines (BMs) with a single layer of feature-detecting units and higher-order BMs (Hinton and Memisevic, 2010). AE and DBN are often used in data pre-processing, but are not end-to-end prediction methods.

CNN is a multi-layer feed-forward ANN, first put forward by LeCun (1998), focusing on two-dimensional inputs such as images, stacking convolutional layers and pooling layers to achieve the features learning. A RNN is a deep architecture that contains feedback connections from hidden or output layers to the preceding layers (Zheng *et al.*, 2017).

CNN and RNN are generally utilised as end-to-end predictive models in RUL prediction. They have been proven to outperform traditional prognosis algorithms, while CNN-based approaches are used more in fault diagnosis and surface integration inspection. RNN, on the other hand, gained much more attention and achievements in the research of RUL prediction because it can accommodate time sequence data and better capture time-dependent features.

Many RNN-based methods have been proposed in recent years, and the performance of the RUL prediction has been dramatically improved. However, very little research investigated how the model's components and parameters affect the RUL prediction performance, such as the optimiser, activation function, sequence length, etc. What is more, for a very long time, the study of DL focused on using fewer training samples to learn more information which no longer fits the real-world situation now in the industry. It is easier to obtain a large amount of data nowadays, and with the data augmentation technique, the size of training samples can be further increased. With enough amount of data, DL techniques can produce an incredibly high predictive accuracy. Therefore, the adaptation ability of algorithms in the real-world industrial environment is a more important target now, but there is still not enough attention in this area. In the field of RUL prediction, the adaptation ability can be evaluated in two aspects: (1) can the model deal with the data collected from the system that works under multiple working conditions,

and (2) can the trained model from samples with no noise achieve high accuracy when testing on noisy samples?

To address the abovementioned research gaps, the motivations of the thesis are listed below.

- To carry out a specific investigation on how different components (e.g., types of optimisers and activation functions) or parameters (e.g., sequence length, neuron quantities) of these approaches affect the overall performance (e.g., robustness, precision, accuracy) of the RUL prediction.
- To utilise feature engineering methods to address the different degradation patterns from data. With the degradation patterns addressed, the predictive model is able to extract more representative features to accommodate different operating conditions.
- To build a novel method to predict the RUL of complex engineering systems with multiple scales adopting the DL-based approaches.

1.3 Aim and objectives

This PhD research aims to develop a novel framework to optimise the performance of deep learning methods in the context of predicting the remaining useful life of complex engineering systems.

The aim of the thesis will be achieved through the following objectives.

Objective 1—Literature review

The literature review aims to obtain essential knowledge on RUL prediction and identify the research gaps. This thesis is in paper format. Therefore, the literature review is delivered in each chapter. Besides, a conclusive work on literature review is presented in Chapter 2, which gives a brief introduction to RUL prediction and reviews the state-of-the-art DL approaches with their associated research gaps in terms of four main representative deep architectures, including AE, DBN, CNN and RNN.

Objective 2—Parameter optimisation

Investigate the influence of different components and parameters of RNN and its variants on the RUL prediction performance, and then propose their optimisation solution to

achieve the best performance against the identified matrices. This objective is fulfilled in Chapter 3.

Objective 3— Feature engineering

Investigate different feature engineering techniques that are used to denoise the data and then evaluate their impact on RUL prediction performance. This objective is fulfilled in Chapter 4, and meanwhile, a novel feature selection method that accommodates the multiple operating conditions of complex engineering systems is developed.

Objective 4— Prognostic model

Address the different degradation patterns from data and propose a robust predictive model that accommodates the multiple operating conditions of complex engineering systems. This objective is fulfilled in Chapter 5.

Objective 5— Framework construction

Develop an end-to-end framework that optimises the performance of RUL prediction using deep learning approaches. This objective is fulfilled through integrating the methods developed in Objectives 2-4. The application of this framework is presented in Chapter 5.

1.4 Methodology

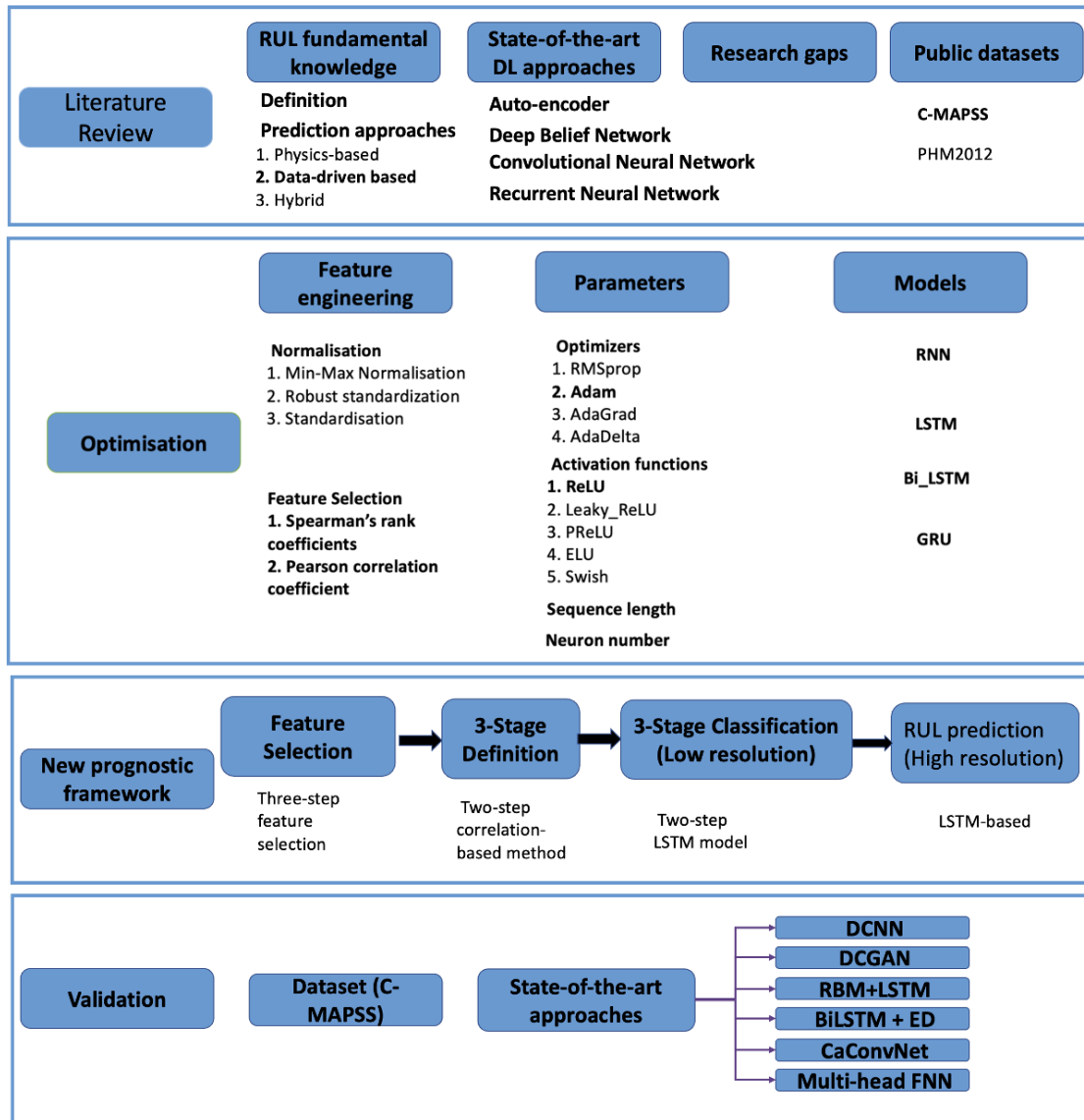


Figure 1-2 Methodology of the project

As shown in Figure 1-2, the project methodology is mainly divided into four phases: literature review, model optimisation, framework construction and validation. In the first phase, a great effort has been made to study the fundamental knowledge of RUL and the state-of-the-art DL application in this field. Based on this study, some research gaps have been identified and discussed. Besides, two commonly used public datasets have been obtained. In the second phase, many tests were conducted to evaluate the performance of selected DL-based techniques on RUL prediction. During this process, optimisation work has been completed focusing on various feature engineering methods and the parameters

of the models. In addition, two different RUL target functions have been compared, and a novel one has been proposed. The framework to optimise the performance of deep learning methods in the context of predicting the RUL of complex engineering systems is constructed in the third phase. The optimisation results in the second phase are considered and utilised. Moreover, a multi-scale RUL prediction approach has been proposed. The proposed framework is validated using the selected dataset as a case study with state-of-the-art DL-based techniques.

1.5 Thesis structure

This thesis consists of six chapters, and the construction of each chapter can be seen on the cover page of each chapter. A summary of each chapter is listed below.

Chapter 1 briefly introduces the research background, presents the research motivations, and describes the aim and objectives of the thesis.

Chapter 2 deliver the background knowledge on RUL prediction and presents a detailed literature review on the state-of-the-art DL-based RUL prediction approaches.

Chapter 3 reviews RNN and its variants for RUL prediction and proceeds with an investigation of how different components or parameters of the model affect their performance on RUL prediction.

Chapter 4 proposes a three-stage feature selection approach for DL-based RUL prediction models to address the limitation of the current methods for data under multiple operating conditions. An operation-based normalisation method is presented, and Spearman's rank coefficient and Pearson's correlation coefficients are utilised to select representative features.

Chapter 5 proposes a novel multiple scales RUL prediction method which adopts the Long Short-Term Memory (LSTM) neural network. An LSTM-based classification model is first used to distinguish the degradation process and present the low-scale RUL prediction. Then another LSTM-based regression prediction model is used for the high-scale RUL prediction.

Chapter 6 summarises the key research findings of the thesis, concludes the contributions to the knowledge, presents the limitation of the work and suggests potential future work.

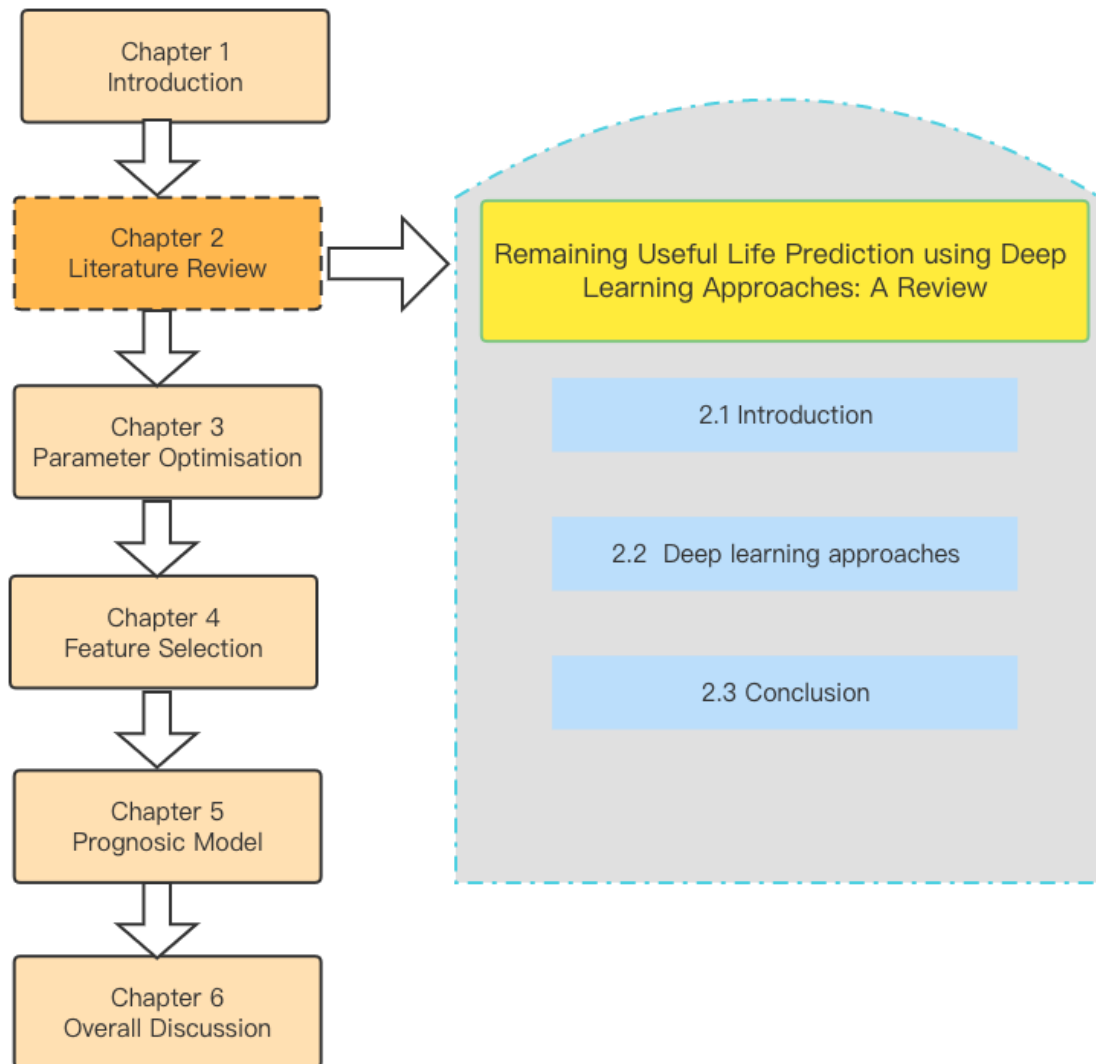
1.6 Main contributions

This thesis is arranged in paper forma, and its main content is derived from the following published papers:

- Paper 1: ‘Remaining Useful Life Prediction using Deep Learning Approaches: A Review’, *8th International Conference on Through-life Engineering Services*, Cleveland, Ohio, 2019, doi:10.1016/j.promfg.2020.06.015
- Paper 2: ‘Recurrent Neural Networks and its variants in Remaining Useful Life prediction’, *4th IFAC Workshop on Advanced Maintenance Engineering, Service and Technology*, September 10-11, 2020. Cambridge, UK. doi:10.1016/j.ifacol.2020.11.022
- Paper 3: ‘Practical Options for Adopting Recurrent Neural Network and Its Variants on Remaining Useful Life Prediction’, *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, p. 69, Dec. 2021. doi:10.1186/s10033-021-00588-x
- Paper 4: ‘Three-stage feature selection approach for deep learning-based remaining useful life prediction methods’, *Quality and Reliability Engineering International*, under review.
- Paper 5: ‘Multi-scale remaining useful life prediction using Long Short-Term Memory’, *Sustainability*, vol. 14, no. 23, p. 15667, Nov. 2022, [doi: 10.3390/su142315667](https://doi.org/10.3390/su142315667)

Paper 1 delivers a review work on DL-based applications on RUL prediction (chapter 2). Paper 2 shrinks the review work of applications on RUL prediction from DL-based to RNN and its variants (chapter 3). Paper 3 conducts a detailed investigation of how different components or parameters of the RNN-based model affect their performance on RUL prediction (chapter3). Paper 4 investigates some feature engineering methods, and presents a novel feature selection approach for DL-based RUL prediction approaches (chapter 4). Paper 5 proposes a multi-scale RUL prediction approach which can significantly improve the prediction performance at the later stage, and offers more degree of flexibility in the way maintenance strategy is organised (chapter 5).

2 CHAPTER TWO—LITERATURE REVIEW



Remaining Useful Life Prediction using Deep Learning Approaches: A Review

Abstract

Data-driven techniques, especially artificial intelligence (AI), such as deep learning (DL) techniques, have attracted more and more attention in the manufacturing sector because of the growth of the industrial Internet of Things (IoT) and Big Data. Tremendous research of DL techniques has been applied in machine health monitoring, but still very limited works focus on applying DL to the Remaining Useful Life (RUL) prediction. Precise RUL prediction can significantly improve industrial component' reliability and operational safety, avoid fatal breakdowns and reduce maintenance costs. This paper gives a brief introduction to RUL prediction and reviews the state-of-the-art DL approaches in terms of four main representative deep architectures, including Auto-encoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). It has been observed that DL techniques attract growing interest in RUL prediction, suggesting a promising future for their applications in manufacturing

Keywords: Remaining useful life prediction, Deep learning, Auto-encoder, Deep Belief Network, Convolutional Neural Network, Recurrent Neural Network

2.1 Introduction

Prognostics is an engineering discipline that predicts the future state or response of a given system based on synthesis observations, calibrated mathematical models, and simulation (Leser, 2017a). It generally refers to the study of predicting the specific time the system will no longer be able to have its intended functional performance. Prognostics attempt to predict the RUL of an engineering system of a component. Salunkhe (2014) regards RUL as the time left before observing a failure. Remaining useful life (RUL) is also called remaining service life or remnant life, referring to the time left before observing a failure, given the current machine age, condition, and past operation profile. Okoh *et al.* (2014) define RUL as the time remaining for a component to perform its functional capabilities before failure. In recent years, prognostics have attracted vast attention from academic researchers and industrial operators. For instance, according to Lei *et al.* (2018), there is a rapid rise in publications in the area of machinery prognostics.

2.1.1 RUL prediction approaches

RUL prediction approaches can be classified in different ways, either based on the type of available data and knowledge about the system or according to the type of methodology or algorithms used. In the early years, the approaches were simply classified into model-based (physics-based), data-driven-based and hybrid-based. More detailed classifications of the RUL prediction approaches have been proposed with the increasing study in this area. The data-driven approaches are further classified into artificial intelligence (AI) approaches and statistical approaches by Dawn *et al.* (2015). Okoh *et al.* (2014) divided the approaches into model-based, analytical-based, knowledge-based and hybrid-based simulation algorithms and tools. The model-based RUL prediction refers to statistics and computational intelligence approaches. The analytical-based RUL prediction method refers to the physical failure technique. The knowledge-based RUL prediction is a combination of computational intelligence and experience. The hybrid RUL prediction is a collection methodology and technique of any former approaches.

Physics-based approaches, also called model-based approaches, assess the system's health by building mathematical models based on the failure mechanisms or the first principle of damage (Cubillo *et al.*, 2016a). Since system degradation modelling depends

on laws of nature, these approaches are generally quite efficient and descriptive. The advantage of this type of approach is that it is relatively accurate and precise concerning the model fidelity, ease of validation, certification, and verification. While the drawbacks of these approaches are that building a high-fidelity model for RUL prediction is expensive and time-consuming, computationally intensive and with a risk of not achieving the desired outcome (Elattar *et al.*, 2016). Moreover, even if high-fidelity models are obtained, the application could be limited only to a specific component or system, or to similar ones. For these drawbacks, the research on physics-based approaches is much less attractive than data-driven approaches.

Data-driven approaches are most widely used in RUL prediction, where RUL is computed through statistical and probabilistic methods by utilizing historical information and routinely monitored data of the system (Kabir *et al.*, 2012). The precondition for setting up the data-driven models for RUL prediction is the availability of multivariate historical data about the system behavior, which must encompass all phases of the system operation and degradation scenarios under certain operating conditions. There are three ways to obtain these run-to-failure data: fielded applications, experimental test beds and computer simulations (Elattar *et al.*, 2016). Although the accuracy and precision of the physics-based approaches make them preferred over data-driven ones, obtaining a physics of failure (PoF) degradation model is often challenging. In contrast, data-driven approaches are generally user-friendly because they mainly rely on techniques from AI tools that could be applied directly or with minor modifications. The faster implementation and deployment of data-driven prognostics result in a much quicker spread speed than the physics-based ones in the RUL prediction field. Moreover, the non-requirement of knowledge about the physics of the system is another main contribution to the population of these types of approaches among the prognostics system developers. The drawbacks of these types of approaches are transparent so are their advantages. As stated by Leser, a struggle with extrapolation and data generation are two of the most common difficulties highlighted in the myriad of data-driven approaches (Leser, 2017a).

Table 2-1 Advantages and drawbacks of prognostic approaches

Prognostic approaches	Advantages
-----------------------	------------

Physics-based approaches	<ul style="list-style-type: none"> • Efficient and descriptive • High accuracy and precision • Easy to validate, certificate and verify • The dynamics of the states can be estimated at each time interval • Extrapolation outside of the training data enables more robust predictions
Data-driven approaches	<ul style="list-style-type: none"> • Quick implementation and deployment • Low cost of algorithms development and little knowledge requirement about system physics • Algorithms can be tuned to be used for other systems • No requirement for the assumptions and empirical estimations of physics parameters • Able to transform high-dimensional noisy data into lower-dimensional information for prognostic decisions
Hybrid approaches	<ul style="list-style-type: none"> • Eliminate the drawbacks of physics-based approaches and data-driven approaches and gain their benefits • Do not require highly accurate models or an enormous amount of data • Flexible and valuable for uncertainty management through retaining the intuitiveness of the model • Higher accuracy
	Drawbacks
Physics-based approaches	<ul style="list-style-type: none"> • High-fidelity models are costly, time-consuming and computationally intensive • Limited reusability

	<ul style="list-style-type: none"> • Difficult in understanding the physics of damage for complex mechanical systems • Too stochastic and complicated to model the defect <p>Require the examination of simplifying assumptions</p>
Data-driven approaches	<ul style="list-style-type: none"> • Requirement for a robust algorithm • Requirement of uncertainty management • Accuracy is affected by the overgeneralisation and overfitting during the training of the algorithm • Inadequate data for newly developed systems • Counter-intuitive results due to the absence of physical knowledge of the system • Require a large amount of data • Have a short prediction horizon • Low accuracy with high dimensional data
Hybrid approaches	<ul style="list-style-type: none"> • Carry the drawbacks of both methods but to a minor extent • Computational resource required • Require both models and data • Inaccuracy caused by an incorrect model of noisy data

While most of the researchers concentrated on the data-driven and physics-based approaches, a relatively small amount of research focused on the hybrid approaches. Till 2017, only 8% of the literature in this field could be found working on hybrid prognostic approaches (Lei *et al.*, 2018). However, a hybrid approach can often be the better option since it attempts to integrate the advantages of both physics-based and data-driven-based approaches. Meanwhile, based on a certain amount of data and relatively accurate models, a hybrid approach can usually achieve higher accuracy for RUL prediction. Nevertheless, the drawback of the hybrid approaches is that they also carry the shortcomings of both approaches and the increased complexity in achieving the solution. compares these three types of approaches in terms of advantages and limitations.

2.1.2 Research gaps

There have been numerous review papers concerning RUL prediction. The earliest review work can be dated back to 2005 by Schwabacher (2005). His survey focused on the work in data-driven approaches for prognosis and related work in model-based and data-driven fault detection and diagnosis. Heng *et al.* (2009) presented a brief review of the RUL prediction approaches with a specific focus on the prognostics of rotating machinery. Sikorska *et al.* (2009) produced classification tables and process flow diagrams to help the industry and researchers to select the appropriate predictive models for predicting the RUL of engineering assets. Lee *et al.* (2010) delivered an introduction to the systematic Prognostics and Health Management (PHM) design methodology for converting data to prognostics information, which includes identifying critical components and selecting the most appropriate algorithms for specific applications. Si *et al.* (2011) reviewed some modelling developments for estimating the RUL centred on statistical data-driven approaches. Baraldi *et al.* (2013) discussed the choice of the predictive method in different information settings in the RUL prediction of different prognostic approaches taking the accuracy and the ability to provide measures of confidence into consideration. Zadeh and Lundberg (2013) compared some typical RUL predicting approaches based on the level of accuracy and availability of data and discussed their advantages and disadvantages. Dawn *et al.* (2013) provided a practical review of predictive methods for beginners, introducing various prognostics algorithms and the responding pros and cons using some simple examples. Tsui *et al.* (2015) took a holistic view to summarize mainstream methods in all major aspects of the PHM framework and their relationships, focusing on data-driven approaches. Thamo *et al.* (2015) reviewed the status of algorithms and methods and provided a structured and comprehensive classification of the existing state-of-the-art PHM approaches, data-driven approaches, and algorithms. Huang *et al.* (2017) summarized the problems and datasets of the PHM Data Challenge competitions and analysed the responding solutions and algorithms. Lei *et al.* (2018) reviewed machinery prognostics, including all four technical processes, i.e., data acquisition, health indicator construction, health stage division and RUL prediction. Jia *et al.* (2018) provided a review of the cutting-edge PHM methodologies and analytics based on the data competitions from 2008 to 2017, where the authors summarized the methods and analytics for the PHM practices in terms of

failure detection, diagnosis, assessment, and prediction. Gregory *et al.* (2019) reviewed the challenges, needs, methods, and best practices for PHM within manufacturing systems, highlighting diagnostics, prognostics, dependability analysis, data management and business areas. Zonta *et al.* (2020) presented a more specific discussion of predictive maintenance (PdM) and emphasized using PdM in the context of Industry 4.0. Zeki *et al.* (2020) reviewed the state-of-the-art ML techniques widely applied to PdM for smart manufacturing in industry 4.0. They first classify the research according to the ML algorithms, ML category, machinery and equipment used, the device used in data acquisition, classification of data, size and type, highlight the key contributions of the researchers, and then offer guidelines for further research. Xu *et al.* (2021) reviewed the use of ML in reliability and safety applications. They examined plenty of publications and discussed the use of DL to highlight its growing popularity and distinctive advantages.

As mentioned above, many reviews have been completed focusing on data-driven prognostic methodologies. In recent years, AI techniques, particularly Deep learning (DL) techniques, are becoming more and more attractive because of the growth in the industrial Internet of Things (IoT) and Big Data. However, as one of the most cutting-edge data-driven prognostic approaches, DL-based approaches have not been systematically reviewed yet. Thus, the attention of this review will be laid on the deep learning approaches focusing on the most state-of-the-art research work.

2.2 Deep learning approaches

Deep learning is one of the sub-branches of machine learning featured by multiple nonlinear processing layers and originated from the artificial neural network (ANN). With the rapid development of computational infrastructure, DL has become one of the main research topics in the field of prognostics, given its capability to capture the hierarchical relationship embedded in deep structures (Ma *et al.*, 2017). The characteristic of DL is its deep network architecture, where multiple layers are stacked to fully capture the representative information from raw input data (Geoffrey Hinton and Ruslan Salakhutdinov, 2006). DL models have gained significant attention and remarkable

achievement in many fields, such as image recognition (Krizhevsky *et al.*, 2015), speech recognition and natural language processing (Hinton *et al.*, 2012). However, it has not been fully exploited in RUL prediction (Navathe *et al.*, 2016).

The published literature on DL mainly focused on four representative deep architectures, including Auto-encoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) (Zhao *et al.*, 2015). This section aims to review the start-of-the-art based on these four representative deep architectures.

2.2.1 Auto-encoder

AE can learn a new representation of the data through reconstructing the input data, which contains two phases: encoder and decoder. Thus, it is often used for pre-training the network. The stacked sparse auto-encoder (SAE) is one of the most used deep neural networks (DNN) approaches to dealing with the data because it contains multilayer AE like sparse auto-encoder and denoising AE (Ma *et al.*, 2018).

In machine health monitoring, AE models are majorly used for fault diagnostics. For RUL prediction, AE is usually used to extract degradation features. Very limited direct applications of AE on RUL prediction can be found in the literature. For instance, Ramin *et al.* (2017) developed a novel predictive method for machine bearings called the auto-encoder-correlation-based (AEC) prognostic algorithm. They collected some test-to-failure experiments sensory data and extracted unsupervised features by training a sparse AE. Then, a moving average filter was used to pass through the output. The health condition of the system can be illustrated after the AEC algorithm normalizes the output of the filter. Jian *et al.* (2018) predicted the RUL of an aircraft engine using a stacked SAE combined with logistic regression (LR). The stacked SAE was used to extract performance degradation features and fuse multiple features through multilayer self-learning. Although the authors claimed that the RUL was predicted based on a stacked SAE, it was achieved by an LR model. Lei *et al.* (2018) developed a DL-based prediction framework to predict the RUL of bearings combining deep AE and DNN. The deep AE was used to select bearing degradation features and reduce the number of prediction network parameters. Yu *et al.* (2019) trained a bidirectional recurrent neural network-based autoencoder in an unsupervised way to convert high-dimensional historical run-to-failure instances to low-dimensional embeddings. And then, the health degradation

patterns of the instances are reflected by the one-dimensional health index (HI), which is constructed by these low-dimensional embeddings. Jiao *et al.* (2020) adopted a conditional variational autoencoder (CVAE) based particle filter (PF) framework to predict the RUL of batteries. The CVAE is embedded into the PF framework to substitute the traditional prior distribution and thus alleviate particle degradation. Wu *et al.* (2021) proposed a novel RUL prediction approach called degradation-aware long short-term memory (LSTM) autoencoder (DELTA). The author claimed that the proposed method could dynamically model the degradation factor and explore latent variables to improve RUL prediction accuracy. Wang *et al.* (2021) combined the convolutional autoencoder (CAE) and LSTM to predict the RUL for electric valves of nuclear power systems. In this work, CAE was used to extract deeper features, and LSTM was used to predict the RUL. Duan *et al.* (2021) introduced a bidirectional gated recurrent unit (BiGRU) decoding approach to carry out the RUL prediction. To improve the decoding capabilities, the authors proposed a skip connection at each time step which can reduce the decoding burden on a single embedding vector. Cheng *et al.* (2022) presented a novel autoencoder quasi-recurrent neural networks (AEQRNN)based prognostic approach for RUL prediction of the engineering systems. In this work, the autoencoder is also used for automatically extracting hidden features from monitoring signals without manual feature design.

2.2.2 Deep Belief Network

A DBN is a stack of Restricted Boltzmann Machines (RBMs), which includes Boltzmann Machines (BMs) with a single layer of feature-detecting units and higher-order RBMs (Hinton and Memisevic, 2010). The greedy layer-by-layer learning algorithm of RBMs can pre-train the model in an unsupervised way with no strict demand on the amount of training data.

Liao *et al.* (2016) proposed an enhanced RBM with a novel regularization term to automatically generate features for RUL prediction. Their work was to fill the research gap of regularizing the RBM model to output a feature space that can highly represent a degradation pattern. Zhang *et al.* (2017) presented a multi-objective deep belief networks (MODBNE) ensemble model for RUL estimation, which employs a robust multi-objective evolutionary algorithm (EA) based on the decomposition integrated with the

traditional DBN training technique. The proposed approach can evolve multiple DBNs simultaneously, subject to two conflicting objectives: accuracy and diversity. Ma *et al.* (2017) applied a Discriminative deep belief network with ant colony optimisation (ACO-DDBN) to predict the health status of the machine. DDBN utilises a deep architecture to combine the advantages of DBN and the discriminative ability of the back-propagation strategy. The ACO can discover the best parameter combinations when selecting the parameters for DDBN. The structure of the DDBN model is determined automatically without prior knowledge through optimisation. Deutsch and He (2017) presented a DBN-feedforward neural network (DBN-FNN) algorithm to predict the RUL of rotating components using a vibration sensor. They developed the DBN-FNN approach combines the advantages of the self-taught feature learning capability of the DBN and the predicting ability of the FNN, which can either take processed vibration features or extract features from the vibration data for predicting the RUL. Although the RUL prediction performance of DBF-FNN was not as good as that of the particle filter, the proposed method accomplished the objective of automatic feature extraction and RUL prediction without the intervention of humans in the age of big data. Deutsch *et al.* (2017) then developed a new integrated method that combined a DBN with a particle filter for RUL prediction of hybrid ceramic bearings using vibration signals. Actual vibration data of hybrid ceramic bearing run-to-failure tests were collected and used to validate the proposed prognostic approach. The proposed integrated system presents a better RUL prediction performance than the particle filter-based approach and the RBN method. Peng *et al.* (2019) combined a particle filter with a DBN-based unsupervised health indicator construction method for RUL prediction. The DBN was trained to extract the hidden characteristics corresponding to the fault state of a system, and the RUL is predicted by particle filter. Hu *et al.* (2020) also combined DBN with another algorithm to present the RUL prediction, in which DBN was also used to extract features. Pan *et al.* (2020) addressed that background noise in signals restricts the applicability of existing PHM approaches in feature extraction. Thus, they developed a novel performance degradation assessment method based on DBN and self-organizing map (SOM) which can denoise and merge multi-sensor vibration signals. Haris *et al.* (2021) claimed that the time to acquire data imposes restrictions on developing a data-driven RUL prediction model for supercapacitors. Therefore, they presented a novel RUL prediction approach which

combines DBN and Bayesian Optimization and HyperBand (BOHB). The authors indicated that the proposed method could provide a successful prediction of the degradation curve using the data of the initial 15 thousand cycles when the supercapacitor has yet to show much degradation.

2.2.3 Convolutional Neural Network

CNN is a multi-layer feed-forward ANN, first put forward by LeCun (1998), focusing on two-dimensional inputs such as images, stacking convolutional layers and pooling layers to achieve the features learning. It was then widely well-recognized for its capability to reveal abstract visual features, such as color contrast and gradient, using the corresponding filters. Although the CNN-based approaches have achieved excellent results in machinery fault diagnosis and surface integration inspection (Wang et al., 2018), there are few research reports on their application in RUL prediction.

Navathe *et al.* (2016) developed the first CNN model specifically applied to solve an RUL estimation problem. It was also the first attempt to leverage the deep learning approach in RUL prediction. Their work mainly uses CNN to conduct alternative processing units such as convolution, pooling, sigmoid/hyperbolic tangent squashing, rectifier, and normalisation. The authors claimed that the proposed deep CNN-based regression approach for RUL estimation is efficient and accurate compared to several state-of-the-art algorithms on two publicly available data sets. Inspired by this work, Li *et al.* (2018) developed a new DL architecture for RUL estimation in prognostics. A time window approach is utilised for sample preparation to improve the feature extraction by CNN. The proposed approach adopted the raw sensor measurements directly as the model inputs. They claimed that no prior expertise in prognostics and signal processing was required using this approach, which facilitates its application in the industrial area. The predictive performance of the proposed approach was validated based on the experiments carried out on the popular C-MAPSS dataset. The experiment results of this proposed approach were compared to the CNN approach provided by Navathe *et al.* (2016) and the LSTM approach provided by Zheng *et al.* (2017a). The comparison outcomes demonstrated that the proposed approach achieves high accuracy in the RUL prediction. Li *et al.* (2019a) developed a novel intelligent RUL prediction approach for bearings, where the time-frequency domain information was explored, and the CNN was used for multi-scale feature extraction. To demonstrate the superiority of the proposed method, the rolling

bearing dataset prepared from the PRONOSTIA platform was used for experiments. Furthermore, different implementations of the DNN-based approaches were compared, such as DNN, Single Scale-Low (SSL) and Single Scale-High (SSH). Good predictive results of the proposed approach suggest its potential in industrial applications. Wang *et al.* (2019) developed a two-dimensional deep CNN approach for rapidly evaluating reliability and predicting RUL. To fulfil the input requirement of the 2-D CNN, a signal conversion method is proposed to convert the one-dimensional signals into 2-D images. Li *et al.* (2019) presented a directed acyclic graph (DAG) network which combines LSTM and CNN to predict the RUL. Instead of just using CNN as the feature extractor, it was placed with LSTM in parallel into the DAG network. Yang *et al.* (2019) proposed a double-CNN-based RUL prediction model. The first CNN model is used to identify the incipient fault point, and the second CNN is used to predict the RUL. Cheng *et al.* (2020) adopted a deep CNN-based RUL prediction approach in which CNN is used to identify the hidden pattern between the extracted degradation energy indicator and the raw vibrations of training bearings. To capture common features between different relative problems, Liu *et al.* (2020) developed a joint-loss CNN architecture. The authors claimed that the proposed approach could not only learn multiple tasks in parallel but also could improve the performance of both tasks by sharing features. Wang *et al.* (2020) addressed the limitations of the traditional CNN-based RUL prediction approaches, and then proposed a recurrent convolutional neural network (RCNN) for RUL prediction of machinery. In RCNN, the recurrent convolutional layers are constructed to model the temporal dependencies of different degradation states. Li *et al.* (2022) presented a two-stage transfer regression (TR)-based RUL prediction for bearing. A CNN classifier was first used to detect the incipient fault point, and then, a transfer regression CNN with multi-loss is constructed for RUL prediction. Majali *et al.* (2022) introduced a robust and precise method of monitoring bearing conditions which combines the Poincare Maps, Fast Fourier Transform (FFT) and CNN. CNN process the bearing Continuous Wavelet Transform images and provide the Remaining Useful Life (RUL) prediction of the bearing. It is also used to provide the fraction of the RUL.

2.2.4 Recurrent Neural Network

An RNN is a deep architecture that contains feedback connections from hidden or output layers to the preceding layers. Thus, it can process dynamic information (Malhi *et al.*, 2011). Sequential data is a standard input data format, such as pressure-time and temperature-time series data. While RNN is one of the most common sequential modelling techniques, it has limitations on long-term RUL predictions because of the lack of trend identification by the weights of the trained network resulting from the weight updated on the presentation of each input pattern. For this reason, LSTM, which conquers the long-term time dependency issues by controlling information flow using an input gate, forget gate and output gate, has been developed (Zheng *et al.*, 2017a).

The RNN and its variation LSTM networks gained great attraction in many applications which have a sequential nature. Researchers working on RUL prediction started to explore the value of RNN in this area these years, especially LSTM. Heimes (2016) proposed an RNN approach to solving the IEEE 2008 Prognostics and Health Management conference challenge problem, which is related to the RUL prediction of a complex system. The RNN training algorithm consists of a Truncated Back Propagation through Time gradient calculation, an Extended Kalman Filter training method and evolutionary algorithms. Yuan *et al.* (2016) proposed an LSTM approach for diagnosis and RUL prediction of a complex system like the aero engine. While LSTM was only used for fault diagnosis, the RUL estimation was still based on an SVM model. Similarly, Guo *et al.* (2017) developed an RNN-based health indicator (RNN-HI) to enhance the accuracy of RUL prediction of bearings. However, the RUL of bearings is calculated using an exponential model with pre-set failure threshold of RNN-HI instead of the trained RNN directly. Zhao *et al.* (2017) presented an integrated approach of CNN and bi-directional LSTM for machining tool wear prediction named Convolutional Bi-directional Long Short-Term Memory (CBLSTM) networks. CNN was first used to extract local robust features from the sequential input. Then, LSTM was utilised for encoding temporal information. The proposed CBLSTM's capability of predicting the RUL of actual tool wear based on raw sensory data was verified with a real-life tool wear test.

Many other researchers have paid attention to the applications of LSTM. Zheng *et al.* (2017a) developed an LSTM-based approach for RUL estimation using sensor data. The

research investigated the hidden patterns from the sensor and operational data with multiple operating conditions, fault and degradation models by combining multiple layers of LSTM cells with standard feed-forward layers. The superiority of the LSTM model in RUL prediction was validated on three widely used data sets, C-MAPSS Data Set, PHM08 Challenge Data Set and Milling Data Set. Wu *et al.* (2018) implemented vanilla LSTM networks to achieve good RUL prediction accuracy in the cases of complicated operations, working conditions, model degradations and intense noises. A Relevance Vector Machines (RVM) was used to detect the starting time of degradation, and vanilla LSTM was used to calculate the RUL. The drawback of this approach is that the RUL requires labelling at every time step for each sample. Some experiential knowledge is required since an appropriate threshold needs to be defined before implementing the Support Vector Machine (SVM). Inspired by the Vanilla LSTM networks, Wu *et al.* (2018) developed another LSTM network focusing on fault prognosis with a degradation sequence of equipment, in which the RUL can be predicted without any pre-defined threshold. To achieve that goal, a one-hot vector was used as the input indicator from which the shutdown time was calculated by the model. Zhang *et al.* (2018b) presented a bi-directional LSTM network to discover the underlying patterns embedded in time series, track the system degradation and, consequently, predict the RUL. Nevertheless, the bi-directional LSTM network was only implemented to track the variation of the health index, and the RUL was predicted by the recursive one-step ahead method. Ahmed *et al.* (2019a) built a new LSTM architecture for RUL prediction when short sequences of monitored observations were given with random initial wear. The proposed LSTM was able to predict the RUL with random starts, which makes it more suitable for real-world cases as the initial condition of physical systems is usually unknown, especially in terms of its manufacturing deficiencies. A new, asymmetric objective function that penalizes late predictions rather than earlier ones was presented as well to ensure safer predictions. Chen *et al.* (2019) adopted a gated recurrent unit-based (GRU) RUL prediction approach, which used kernel principle component analysis (KPCA) to extract the informative features. Li *et al.* (2020) proposed a predictive framework shared by multiple batteries for RUL prediction based on the design of a variant LSTM neural network called AST-LSTM NN. One innovation of the proposed AST-LSTM NN is that it uses a fixed connection to couple the input and forget gates so that the previous information and new

data can be determined simultaneously. Park *et al.* (2020) presented a novel LSTM-based method to predict the RUL even in the presence of the capacity regeneration phenomenon. Instead of matching the input layer with the output layer as a one-to-one structure, they adopted a many-to-one structure which is flexible for various input types. Thereby, the number of parameters could be significantly reduced for better generalisation. Liu *et al.* (2021) combined LSTM with elastic net to produce the RUL prediction of the rolling bearing. The elastic net is used to solve the over-fitting problem of the LSTM. Chui *et al.* (2021) proposed to combine RNN and LSTM for RUL prediction because the former is good at short-term prediction, and the latter performs better in long-term prediction. They designed the weights to combine RNN and LSTM using a non-dominated sorting genetic algorithm II. Xiang *et al.* (2021) constructed a multi-cellular LSTM to overcome the problem that most neural networks cannot process the input data in different update modes based on its importance degree. The authors claimed that the proposed algorithm could well mine different degradation trends. Liu *et al.* (2021) present a multi-stage RUL prediction algorithm which combines clustering and LSTM. The experimental results using the CMAPSS dataset indicate that the proposed approach effectively reduces RUL's prediction error. Zhang *et al.* (2022) introduced a dual task network structure based on BiGRU and multi-gate mixture-of-experts (MMoE), which can perform the health status assessment and RUL prediction simultaneously. BiGRU is used to extract the shared information from sensor signals, and MMoE structure is applied to differentiate between HS assessment and RUL prediction tasks adaptively. Liu *et al.* (2022) developed a stacked BiLSTM neural network to predict the RUL of supercapacitors. They suggested that the stacked network could ensure enough capacity space.

2.3 Conclusion

The emerging infrastructure presented by the Internet of Things and data science has been a revolutionary factor in the manufacturing industry. RUL prediction has been increasingly important in machine health monitoring in modern manufacturing systems. Compared with traditional physics-based models, data-driven models gain more attention due to the significant development of sensors, sensor networks and computing systems. Machine learning techniques, especially DL techniques, are regarded as powerful

solutions due to their ability to provide more agility to process data associated with highly nonlinear and complex feature abstraction through a cascade of multiple layers.

DL provides decision-makers with new visibility into their operations, real-time performance measures and costs (Wang *et al.*, 2018). This paper presented an overview of the latest DL-based works on the related topic covering four main DL variants: AE, DBN, CNN and RNN. It has been observed that DL-based techniques were mainly used for fault diagnostics, and very limited studies applied DL-based techniques in RUL prediction until recent years. The growth trend of the literature shows an increasing interest and suggests a promising future of DL in RUL prediction. Besides, DL-related RUL prediction approaches are purely data-driven approaches. Thus, to increase user confidence, an extensive database of run-to-fail trajectories should be obtained and compared to the observed data based on the similarity before the application. Additional research could focus on combining DL approaches with other data-driven approaches or physics-based approaches because these hybrid approaches have great potential and a chance to provide a more effective and precise RUL prediction.

Acknowledgements

This work was partly supported by the UK EPSRC Platform Grant: Through-life performance: From science to instrumentation (Grant number EP/P027121/1).

References

- Ahmadzadeh, F., Lundberg, J., 2013. Remaining useful life estimation: review. *International Journal of Systems Assurance Engineering and Management* 5, 461–474. <https://doi.org/10.1007/s13198-013-0195-0>
- Ahmed, Elsheikh, Soumaya, Yacout, Mohamed-Salah, Ouali, Elsheikh, A., Yacout, S., Ouali, M.S., 2018. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* 323, 148–156. <https://doi.org/10.1016/j.neucom.2018.09.076>
- An, D., Choi, J.H., Kim, N.H., 2013. Options for Prognostics Methods: A review of data-driven and physics-based prognostics 1–14. <https://doi.org/10.2514/6.2013-1940>
- An, D., Kim, N.H., Choi, J.H., 2015. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliab Eng Syst Saf* 133, 223–236. <https://doi.org/10.1016/j.ress.2014.09.014>
- Baraldi, P., Cadini, F., Mangili, F., Zio, E., 2013. Model-based and data-driven prognostics under different available information. *Probabilistic Engineering Mechanics* 32, 66–79. <https://doi.org/10.1016/j.probengmech.2013.01.003>
- Bin Huang, Yuan Di, C.J. and J.L., 2017. Review of Data-Driven Prognostics and Health Management Techniques: Lessons Learned From Phm Data Challenge Competitions 1–17.
- Chen, J., Jing, H., Chang, Y., Liu, Q., 2019. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliab Eng Syst Saf* 185, 372–382. <https://doi.org/10.1016/j.ress.2019.01.006>
- Cheng, C., Ma, G., Zhang, Y., Sun, M., Teng, F., Ding, H., Yuan, Y., 2020. A Deep Learning-Based Remaining Useful Life Prediction Approach for Bearings. *IEEE/ASME Transactions on Mechatronics* 25, 1243–1254. <https://doi.org/10.1109/TMECH.2020.2971503>
- Cheng, Y., Hu, K., Wu, J., Zhu, H., Shao, X., 2022. Autoencoder Quasi-Recurrent Neural Networks for Remaining Useful Life Prediction of Engineering Systems. *IEEE/ASME Transactions on Mechatronics* 27, 1081–1092. <https://doi.org/10.1109/TMECH.2021.3079729>

- Chui, K.T., Gupta, B.B., Vasant, P., 2021. A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine. *Electronics (Switzerland)* 10, 1–15. <https://doi.org/10.3390/electronics10030285>
- Çinar, Z.M., Nuhu, A.A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B., 2020. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability (Switzerland)* 12. <https://doi.org/10.3390/su12198211>
- Cubillo, A., Perinpanayagam, S., Esperon-Miguez, M., 2016. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering* 8, 1–21. <https://doi.org/10.1177/1687814016664660>
- Deutsch, J., He, D., 2017. Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components. *Journal of Zoo and Wildlife Medicine* 40, 321–327. <https://doi.org/10.1109/TSMC.2017.2697842>
- Deutsch, J., He, M., He, D., 2017. Remaining Useful Life Prediction of Hybrid Ceramic Bearings Using an Integrated Deep Learning and Particle Filter Approach. *Applied Sciences* 7, 649. <https://doi.org/10.3390/app7070649>
- Duan, Y., Li, H., He, M., Zhao, D., 2021. A BiGRU Autoencoder Remaining Useful Life Prediction Scheme with Attention Mechanism and Skip Connection. *IEEE Sens J* 21, 10905–10914. <https://doi.org/10.1109/JSEN.2021.3060395>
- Elattar, H.M., Elminir, H.K., Riad, A.M., 2016. *Prognostics: a literature review, Complex & Intelligent Systems*. Springer Berlin Heidelberg. <https://doi.org/10.1007/s40747-016-0019-3>
- Geoffrey Hinton, Ruslan Salakhutdinov, C., 2006. Reducing the dimensionality of data with neural networks. *Science (1979)* 28, 502–504.
- Guo, L., Li, N., Jia, F., Lei, Y., Lin, J., 2017. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* 240, 98–109. <https://doi.org/10.1016/j.neucom.2017.02.045>
- Haris, M., Hasan, M.N., Qin, S., 2021. Early and robust remaining useful life prediction of supercapacitors using BOHB optimized Deep Belief Network. *Appl Energy* 286. <https://doi.org/10.1016/j.apenergy.2021.116541>

- Hasani, R.M., Wang, G., Grosu, R., 2017. An Automated Auto-encoder Correlation-based Health-Monitoring and Prognostic Method for Machine Bearings.
- Heimes, F., Heimes, F.O., Systems, B.A.E., 2016. Recurrent neural networks for remaining useful life estimation Recurrent Neural Networks for Remaining Useful Life Estimation. <https://doi.org/10.1109/PHM.2008.4711422>
- Heng, A., Zhang, S., Tan, A.C.C., Mathew, J., 2009. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mech Syst Signal Process* 23, 724–739. <https://doi.org/10.1016/j.ymssp.2008.06.009>
- Hinton, G., Deng, L., Et., A., 2012. *Ieee Signal Processing Magazine* [82] November 2012 82–97. <https://doi.org/10.1109/MSP.2012.2205597>
- Hinton, G.E., Memisevic, R., 2010. Learning to Represent Spatial Transformations with Factored Higher-Order Boltzmann Machines. *Neural Comput* 22, 1473–1492.
- Hu, C.H., Pei, H., Si, X.S., Du, D.B., Pang, Z.N., Wang, X., 2020. A prognostic model based on DBN and diffusion process for degrading bearing. *IEEE Transactions on Industrial Electronics* 67, 8767–8777. <https://doi.org/10.1109/TIE.2019.2947839>
- Jia, X., Huang, B., Feng, J., Cai, H., Lee, J., 2018. A Review of PHM Data Competitions from 2008 to 2017. *PHM Society Conference* 10. <https://doi.org/10.1234/PHMCONF.2018.V10I1.462>
- Jiao, R., Peng, K., Dong, J., 2020. Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Conditional Variational Autoencoders-Particle Filter. *IEEE Trans Instrum Meas* 69, 8831–8843. <https://doi.org/10.1109/TIM.2020.2996004>
- Kabir, A., Bailey, C., Lu, H., Stoyanov, S., 2012. A review of data-driven prognostics in power electronics. *Proceedings of the International Spring Seminar on Electronics Technology* 189–192. <https://doi.org/10.1109/ISSE.2012.6273136>
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2015. Alexnet_Tugce_Kyunghee.
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., Siegel, D., 2010. Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications. *Mech Syst Signal Process* 42, 314–334. <https://doi.org/10.1016/j.ymssp.2013.06.004>

- Lei, Y., Li, Naipeng, Guo, L., Li, Ningbo, Yan, T., Lin, J., 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech Syst Signal Process* 104, 799–834. <https://doi.org/10.1016/j.ymssp.2017.11.016>
- Leser, P.E., 2017. Probabilistic Prognostics and Health Management for Fatigue-critical Components using High-fidelity Models (Sissertation). North Carolina.
- Li, J., Li, X., He, D., 2019. A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction. *IEEE Access* 7, 75464–75475. <https://doi.org/10.1109/ACCESS.2019.2919566>
- Li, P., Zhang, Z., Xiong, Q., Ding, B., Hou, J., Luo, D., Rong, Y., Li, S., 2020. State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network. *J Power Sources* 459. <https://doi.org/10.1016/j.jpowsour.2020.228069>
- Li, X., Ding, Q., Sun, J.Q., 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab Eng Syst Saf* 172, 1–11. <https://doi.org/10.1016/j.ress.2017.11.021>
- Li, X., Zhang, K., Li, W., Feng, Y., Liu, R., 2022. A Two-Stage Transfer Regression Convolutional Neural Network for Bearing Remaining Useful Life Prediction. *Machines* 10. <https://doi.org/10.3390/machines10050369>
- Li, X., Zhang, W., Ding, Q., 2019. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab Eng Syst Saf* 182, 208–218. <https://doi.org/10.1016/j.ress.2018.11.011>
- Liao, L., Jin, W., Pavel, R., 2016. Enhanced Restricted Boltzmann Machine with Prognosability Regularization for Prognostics and Health Assessment. *IEEE Transactions on Industrial Electronics* 63, 7076–7083. <https://doi.org/10.1109/TIE.2016.2586442>
- Liu, C., Zhang, Y., Sun, J., Cui, Z., Wang, K., 2022. Stacked bidirectional LSTM RNN to evaluate the remaining useful life of supercapacitor. *Int J Energy Res* 46, 3034–3043. <https://doi.org/10.1002/er.7360>
- Liu, J., Lei, F., Pan, C., Hu, D., Zuo, H., 2021. Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion. *Reliab Eng Syst Saf* 214. <https://doi.org/10.1016/j.ress.2021.107807>

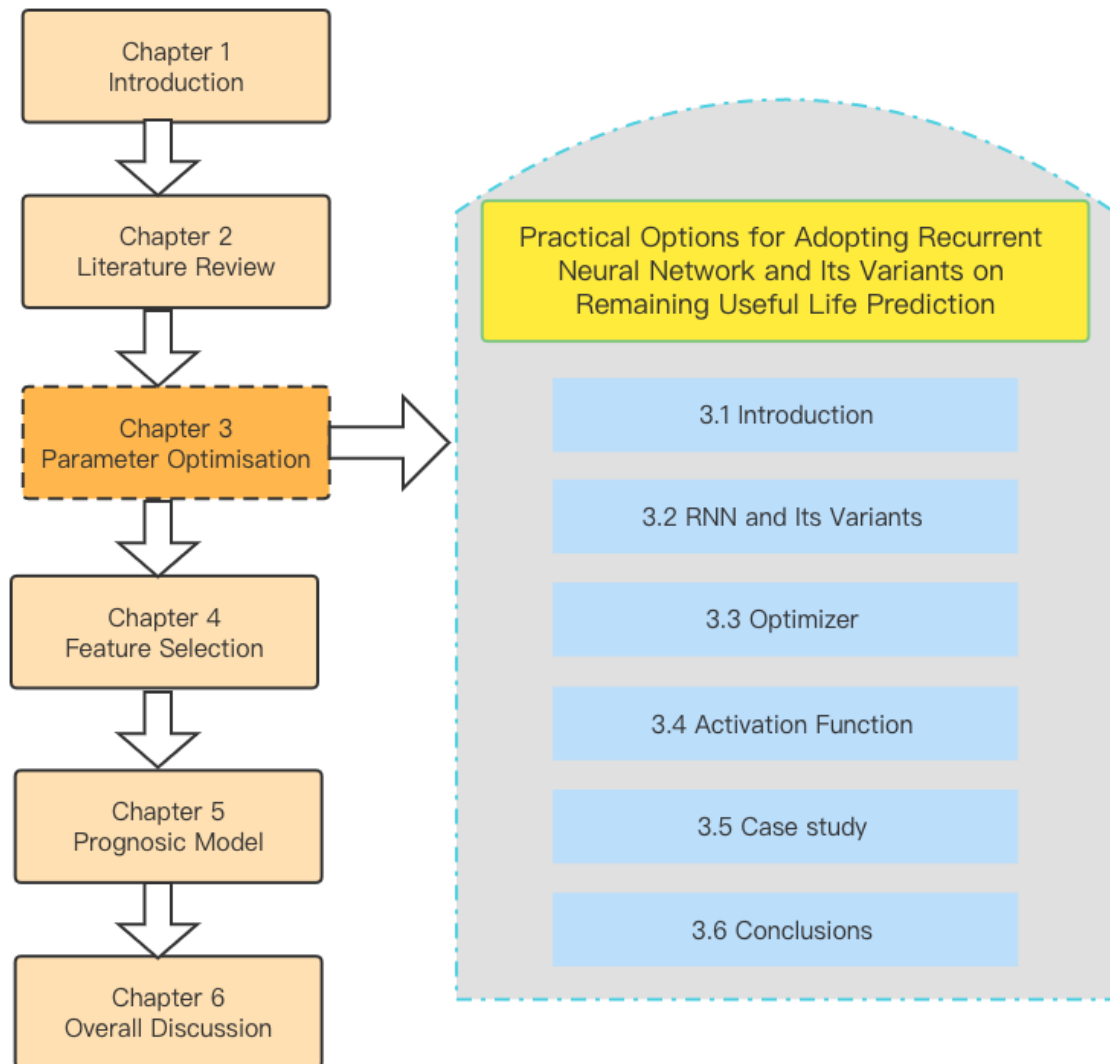
- Liu, R., Yang, B., Hauptmann, A.G., 2020. Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network. *IEEE Trans Industr Inform* 16, 87–96. <https://doi.org/10.1109/TII.2019.2915536>
- Ma, J., Su, H., Zhao, W., Liu, B., 2018. Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning. *Complexity* 2018, 1–13. <https://doi.org/10.1155/2018/3813029>
- Ma, M., Sun, C., Chen, X., 2017. Discriminative Deep Belief Networks with Ant Colony Optimization for Health Status Assessment of Machine. *IEEE Trans Instrum Meas* 66, 3115–3125. <https://doi.org/10.1109/TIM.2017.2735661>
- Majali, A., Mulay, A., Iyengar, V., Nayak, A., Singru, P., 2022. Fault identification and remaining useful life prediction of bearings using Poincare maps, fast Fourier transform and convolutional neural networks. *Mathematical Models in Engineering* 8, 1–14. <https://doi.org/10.21595/mme.2022.22364>
- Malhi, A., Yan, R., Gao, R.X., 2011. Prognosis of defect propagation based on recurrent neural networks. *IEEE Trans Instrum Meas* 60, 703–711. <https://doi.org/10.1109/TIM.2010.2078296>
- Navathe, S.B., Wu, W., Shekhar, S., Du, X., Sean Wang, X., Xiong, H., 2016. Deep convolutional neural network based regression approach for estimation of remaining useful life E1–E1. https://doi.org/10.1007/978-3-319-32049-6_29
- Okoh, C., Roy, R., Mehnen, J., Redding, L., 2014. Overview of Remaining Useful Life prediction techniques in Through-life Engineering Services. *Procedia CIRP* 16, 158–163. <https://doi.org/10.1016/j.procir.2014.02.006>
- Pan, Y., Hong, R., Chen, J., Wu, W., 2020. A hybrid DBN-SOM-PF-based prognostic approach of remaining useful life for wind turbine gearbox. *Renew Energy* 152, 138–154. <https://doi.org/10.1016/j.renene.2020.01.042>
- Park, K., Choi, Y., Choi, W.J., Ryu, H.Y., Kim, H., 2020. LSTM-Based Battery Remaining Useful Life Prediction with Multi-Channel Charging Profiles. *IEEE Access* 8, 20786–20798. <https://doi.org/10.1109/ACCESS.2020.2968939>
- Ren, L., Sun, Y., Cui, J., Zhang, L., 2018. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. *J Manuf Syst* 48, 71–77. <https://doi.org/10.1016/j.jmsy.2018.04.008>

- Salunkhe, T., Jamadar, N.I., Kivade, S.B., 2014. Prediction of Remaining Useful Life of Mechanical Components-A Review 3, 125–135.
- Schwabacher, M., 2005. A Survey of Data-Driven Prognostics 1–5.
<https://doi.org/10.2514/6.2005-7002>
- Si, X.S., Wang, W., Hu, C.H., Zhou, D.H., 2011. Remaining useful life estimation - A review on the statistical data driven approaches. *Eur J Oper Res* 213, 1–14.
<https://doi.org/10.1016/j.ejor.2010.11.018>
- Sikorska, J.Z., Hodkiewicz, M., Ma, L., 2009. Prognostic modelling options for remaining useful life estimation by industry. *Mech Syst Signal Process* 25, 1803–1836. <https://doi.org/10.1016/j.ymssp.2010.11.018>
- Sutharssan, T., Stoyanov, S., Bailey, C., Yin, C., 2015. Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms. *The Journal of Engineering* 2015, 215–222.
<https://doi.org/10.1049/joe.2014.0303>
- Tsui, K.L., Chen, N., Zhou, Q., Hai, Y., Wang, W., 2015. Prognostics and Health Management: A Review on Data Driven Approaches. *Math Probl Eng* 2015, 1–17. <https://doi.org/10.1155/2015/793161>
- Vogl, G.W., Weiss, B.A., Helu, M., 2019. A review of diagnostic and prognostic capabilities and best practices for manufacturing. *J Intell Manuf* 30, 79–95.
<https://doi.org/10.1007/s10845-016-1228-8>
- Wang, B., Lei, Y., Yan, T., Li, N., Guo, L., 2020. Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. *Neurocomputing* 379, 117–129. <https://doi.org/10.1016/j.neucom.2019.10.064>
- Wang, H., Peng, M. jun, Miao, Z., Liu, Y. kuo, Ayodeji, A., Hao, C., 2021. Remaining useful life prediction techniques for electric valves based on convolution auto encoder and long short term memory. *ISA Trans* 108, 333–342.
<https://doi.org/10.1016/j.isatra.2020.08.031>
- Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., 2018. Deep learning for smart manufacturing: Methods and applications. *J Manuf Syst* 48, 144–156.
<https://doi.org/10.1016/j.jmsy.2018.01.003>
- Wang, Q., Zhao, B., Ma, H., Chang, J., Mao, G., 2019. A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional

- convolutional neural network with signal conversion. *Journal of Mechanical Science and Technology* 33, 2561–2571. <https://doi.org/10.1007/s12206-019-0504-x>
- Wu, J.Y., Wu, M., Chen, Z., Li, X.L., Yan, R., 2021. Degradation-Aware Remaining Useful Life Prediction with LSTM Autoencoder. *IEEE Trans Instrum Meas* 70. <https://doi.org/10.1109/TIM.2021.3055788>
- Wu, Q., Ding, K., Huang, B., 2018. Approach for fault prognosis using recurrent neural network. *J Intell Manuf* 1–13. <https://doi.org/10.1007/s10845-018-1428-5>
- Xiang, S., Qin, Y., Luo, J., Pu, H., Tang, B., 2021. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. *Reliab Eng Syst Saf* 216. <https://doi.org/10.1016/j.ress.2021.107927>
- Xu, Z., Saleh, J.H., 2021. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliab Eng Syst Saf*. <https://doi.org/10.1016/j.ress.2021.107530>
- Y, L., L, B., Y, B., P, H., 1998. LeNet-Gradient Based Learning Applied to Document Recognition. *Proceedings of the IEEE* 86, 2278–2324.
- Yang, B., Liu, R., Zio, E., 2019. Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Transactions on Industrial Electronics* 66, 9521–9530. <https://doi.org/10.1109/TIE.2019.2924605>
- Yu, W., Kim, I.Y., Mechefske, C., 2019. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mech Syst Signal Process* 129, 764–780. <https://doi.org/10.1016/j.ymssp.2019.05.005>
- Yuan, M., Wu, Y., Dong, S., Lin, L., Liu, Y., 2018. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* 275, 167–179. <https://doi.org/10.1016/j.neucom.2017.05.063>
- Yuan, M., Wu, Y., Lin, L., 2016. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. *AUS 2016 - 2016 IEEE/CSAA International Conference on Aircraft Utility Systems* 135–140. <https://doi.org/10.1109/AUS.2016.7748035>
- Zhang, C., Lim, P., Qin, A.K., Tan, K.C., 2017. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Trans*

- Neural Netw Learn Syst 28, 2306–2318.
<https://doi.org/10.1109/TNNLS.2016.2582798>
- Zhang, J., Wang, P., Yan, R., Gao, R.X., 2018. Long short-term memory for machine remaining life prediction. *J Manuf Syst* 48, 78–86.
<https://doi.org/10.1016/j.jmsy.2018.05.011>
- Zhang, Y., Xin, Y., Liu, Z. wei, Chi, M., Ma, G., 2022. Health status assessment and remaining useful life prediction of aero-engine based on BiGRU and MMoE. *Reliab Eng Syst Saf* 220. <https://doi.org/10.1016/j.res.2021.108263>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2015. Deep Learning and Its Applications to Machine Health Monitoring : A Survey 14, 1–14.
- Zhao, R., Yan, R., Wang, J., Mao, K., 2017. Learning to monitor machine health with convolutional Bi-directional LSTM networks. *Sensors (Switzerland)* 17, 1–18.
<https://doi.org/10.3390/s17020273>
- Zheng, S., Ristovski, K., Farahat, A., Gupta, C., 2017a. Long Short-Term Memory Network for Remaining Useful Life estimation, in: 2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017. Institute of Electrical and Electronics Engineers Inc., pp. 88–95.
<https://doi.org/10.1109/ICPHM.2017.7998311>
- Zheng, S., Ristovski, K., Farahat, A., Gupta, C., 2017b. Long Short-Term Memory Network for Remaining Useful Life estimation. 2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017 88–95.
<https://doi.org/10.1109/ICPHM.2017.7998311>
- Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., Li, G.P., 2020. Predictive maintenance in the Industry 4.0: A systematic literature review. *Comput Ind Eng* 150. <https://doi.org/10.1016/j.cie.2020.106889>

3 CHAPTER THREE—PARAMETER OPTIMISATION



Practical Options for Adopting Recurrent Neural Network and Its Variants on Remaining Useful Life Prediction

Abstract

A system's remaining useful life (RUL) is generally predicted by utilising the data collected from the sensors that continuously monitor different indicators. Different deep learning (DL) techniques have recently been used for RUL prediction and achieved great success. Because the data is often time-sequential, recurrent neural networks (RNNs) have attracted significant interest due to their efficiency in dealing with such data. This paper reviews RNN and its variants for RUL prediction, specifically focusing on understanding how different components (e.g., types of optimisers and activation functions) or parameters (e.g., sequence length, neuron quantities) affect their performance. After that, a case study using the well-studied NASA's C-MAPSS dataset is presented to quantitatively evaluate the influence of various state-of-the-art RNN structures on the RUL prediction performance. The result suggests that the variant methods usually perform better than the original RNN. Bi-directional Long Short-Term Memory generally has the best performance in terms of robustness, precision, and accuracy. Certain model structures may fail to produce valid RUL prediction results due to the gradient vanishing or gradient exploding problem if the parameters are not chosen appropriately. It is concluded that parameter tuning is a crucial step to achieving optimal prediction performance.

Keywords: Remaining useful life prediction, Deep learning, Recurrent neural network, Long Short-term Memory, Bi-directional long short-term memory, Gated recurrent unit

3.1 Introduction

Remaining useful life (RUL) prediction is an engineering discipline that predicts a given system's future state or response based on synthesis observations, calibrated mathematical models, and simulations (Leser, 2017b). It generally refers to the study of predicting the specific time at which the system or the component will no longer be able to have its intended functional performance. Salunkhe *et al.* (2014) regard RUL as the time left before observing a failure. Okoh *et al.* (2014) define RUL as the time remaining for a component to perform its functional capabilities before failure. Predicting the RUL of a component or a system in the industrial world is crucial, as it helps prevent failures or accidents. For example, the failure of aircraft engines would often lead to major accidents and casualties (Ma *et al.*, 2018). Thus, it is essential to predict the RUL of the engine, implement maintenance accordingly and eventually prevent catastrophic failure. The degradation process of an operating device is a process of gradual deterioration and can be detected to a certain extent through the measurement of covariate variables (Deutsch and He, 2017). In recent years, RUL prediction has attracted vast attention from academic researchers and industrial operators.

RUL prediction approaches are generally catalogued into model-based (physics-based) methods, data-driven methods, and hybrid models, which are a combination of the first two methods (Deutsch and He, 2017). As the complex and noisy working condition impedes the construction of the physical systems, it results in difficulties in developing the modelling of complex dynamic systems (Zhang *et al.*, 2017). In addition, the difficulty to be updated the online measured data limits the effectiveness and flexibility of the physics-based models. In contrast, data-driven approaches are gaining popularity due to their quick implementation, widespread deployment of low-cost sensors, and internet connection. RUL is computed through statistical and probabilistic methods by utilising historical information and routinely monitored data of the system (Kabir *et al.*, 2012). The precondition for setting up the data-driven models for RUL prediction is the availability of multivariate historical data about the system behaviour, which must encompass all phases of the system operation and degradation scenarios under certain operating conditions. In recent years, Artificial intelligence (AI) techniques, particularly deep learning (DL) techniques, are becoming more and more attractive because of the rapid growth in the industrial Internet of Things (IoT), Big Data and increasing computing

power (Zhao *et al.*, 2019b). Researchers have exploited applications of AI techniques for RUL prediction as well.

Deep learning is one of the sub-branches of machine learning, which originated from Artificial Neural Networks (ANNs) and featured multiple nonlinear processing layers. It intends to model hierarchical representations and predicts patterns behind data through building stacked multiple layers of information processing modules in hierarchical architectures. With the rapid development of computational infrastructure and the availability of a large volume of data, DL has become one of the main research topics in the field of prognostics, given its capability to capture the hierarchical relationship embedded in deep structures (Kraus and Feuerriegel, 2019). The published literature on DL approaches for RUL prediction mainly focus on four representative deep architectures, including Auto-encoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) (2016). AE and DBN are often used for the pre-training of networks. For instance, Jia *et al.* (2016) developed a stacked denoising autoencoder (SDA) fed with the frequency spectra of time-series data to diagnose rotating machinery. Wang *et al.* (2017) proposed an SDA to identify the health state of certain systems with signals containing ambient noise and working condition fluctuations. Shao *et al.* (2017) developed a deep AE-based method to diagnose rotating machinery faults. Liao *et al.* (2016) proposed combining an enhanced Restricted Boltzmann Machine (RBM) with a novel regularisation term to extract the features which are suitable for RUL prediction automatically. Gan *et al.* (2016) presented a hierarchical diagnosis network that combines a wavelet packet transform (WPT) and DBN for consecutive identification of bearing fault location and severity. Thus, research suggests that CNN and RNN are generally used as predictive models and have proved to outperform traditional prognosis algorithms in RUL prediction. CNN-based approaches are used more in fault diagnosis and surface integration inspection (Wang *et al.*, 2018). RNN, on the other hand, gained much more attention and achievements in the research of RUL prediction because of its ability to accommodate time sequence data (Zheng *et al.*, 2017b). Therefore, this paper systematically reviews the applications of RNN and its variants for RUL prediction in recent years. Many novel RNN-based methods have been proposed, and the RUL prediction performance has been dramatically improved. However, most of these works focused on achieving a better prediction performance using

a particular approach. Few researchers paid attention to other factors that also affect the prediction result, such as the optimiser, activation function, neuron number and sequence length. For example, the optimisers are used to shape the model into its most accurate form through fusing with the weights. To the best of our knowledge, there is no research discussing how different optimisers affect the performance of RUL prediction using DL-based approaches and what is the underlying principle to optimise the selection. To fill these research gaps, this paper not only presents an evaluation of the basic RNN and its variants on RUL prediction based on a case study in a publicly available dataset, but a specific investigation has also been carried out on how different components (e.g., types of optimisers and activation functions) or parameters (e.g., sequence length, neuron quantities) of these approaches affect the overall performance (e.g., robustness, precision, accuracy) of the RUL prediction.

The remainder of this paper is organized as follows. Section 3.2 briefly introduces the fundamental conception of RNN and its variants. Section 3.3 presents the different optimisers that are normally used in DL. Section 3.4 explains how activation functions affect the training of the network and demonstrate the advantages and drawbacks of different activation functions. Section 3.5 presents a case study that aims to evaluate the factors that influence the performance of RUL prediction based on a publicly available dataset.

3.2 Recurrent Neural Network and Its Variants

3.2.1 Recurrent Neural Network

In a traditional neural network, inputs are independent, while in RNN, the front neurons pass the information to the following neurons. As illustrated in Figure 3-1, unlike a traditional feed-forward neural network, an RNN can be regarded as numerous copies of the same neural network cell, in which each cell passes the message to the next through the hidden state. In other words, the output from a recurrent neuron is connected to the next one to characterise the current system state as a function of current sensing data and the preceding system state.

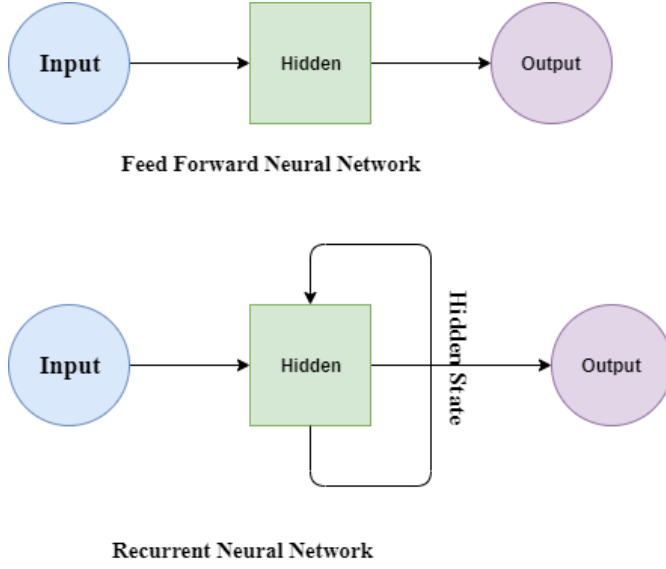


Figure 3-1 Basic Feed-Forward Neural Network structure and Recurrent Neural Network structure

In an unrolled RNN, the sensing data $(\dots x_{t-1}, x_t, x_{t+1} \dots)$ are fed simultaneously into the corresponding neurons, which generate the corresponding neuron time series $(\dots h_{t-1}, h_t, h_{t+1} \dots)$. The output of a single recurrent neuron can be expressed as:

$$h_t = \sigma(Ux_t + W_h h_{t-1} + b) \quad (3-1)$$

$$y_t = \text{softmax}(W_y h_t + c) \quad (3-2)$$

where U , W_h and W_y represent the weight vectors respectively. The symbol b and c denote the bias term, and σ is the activation function, with the hyperbolic tangent or Relu commonly used in RNN. y_t is the output of the recurrent neuron based on the output of the hidden state h_t , which can be referred to a memory space containing the information of the current input and the former hidden state h_{t-1} . It is worth mentioning that all the weight vectors are shared at every step, which means that the same task is repeated at every step with different inputs, and the memory is renewed accordingly. The basic structures of RNN, LSTM and Gated Recurrent Unit (GRU), are illustrated in Figure 3-2 Basic structures of RNN, GRU and LSTM. The upper diagram shows how information transform from the former neurons to the latter ones. The symbol A refers to the hidden state, and the lower three figures indicates the detailed composite struct diagram of the cells.

3.2.2 Long Short-Term Memory (LSTM)

The main issue of the standard RNN is gradient exploding and gradient vanishing. These issues might happen when the network is too deep. In other words, when the number of the time step is too large, the information carried in the front neuron will be lost because no structure in a standard recurrent layer individually controls the flow of the memory itself. The LSTM network, a modified structure of the recurrent cell that incorporates the standard recurrent layer along with additional “memory” control gates, has been proposed to solve this problem.

The original LSTM was developed by Hochreiter and Schmidhuber (1997) when researchers discovered a vanishing and exploding gradient issue in traditional RNNs. LSTM uses storage elements to transfer information from the past output instead of having the RNN cell output to be a nonlinear function of the weighted sum of the current input and previous output. In other words, instead of using a hidden state h only, LSTM adopts a cell state C to keep the long-term information, as shown in Figure 3-3. The central concept of LSTM is utilising three gates, such that each is a nonlinear function σ activated by a weighted sum of the current input observation x_t and previous hidden state h_{t-1} , to control the cell state C (forget gate f_t , input gate i_t and output gate o_t). The forget gate f_t is used to manage the information from the previous cell state C_{t-1} to the current cell state C_t ; the input gate i_t decides how many inputs should be kept in the current cell state C_t ; and the output gate o_t determines the output h_t from the current cell state C_t .

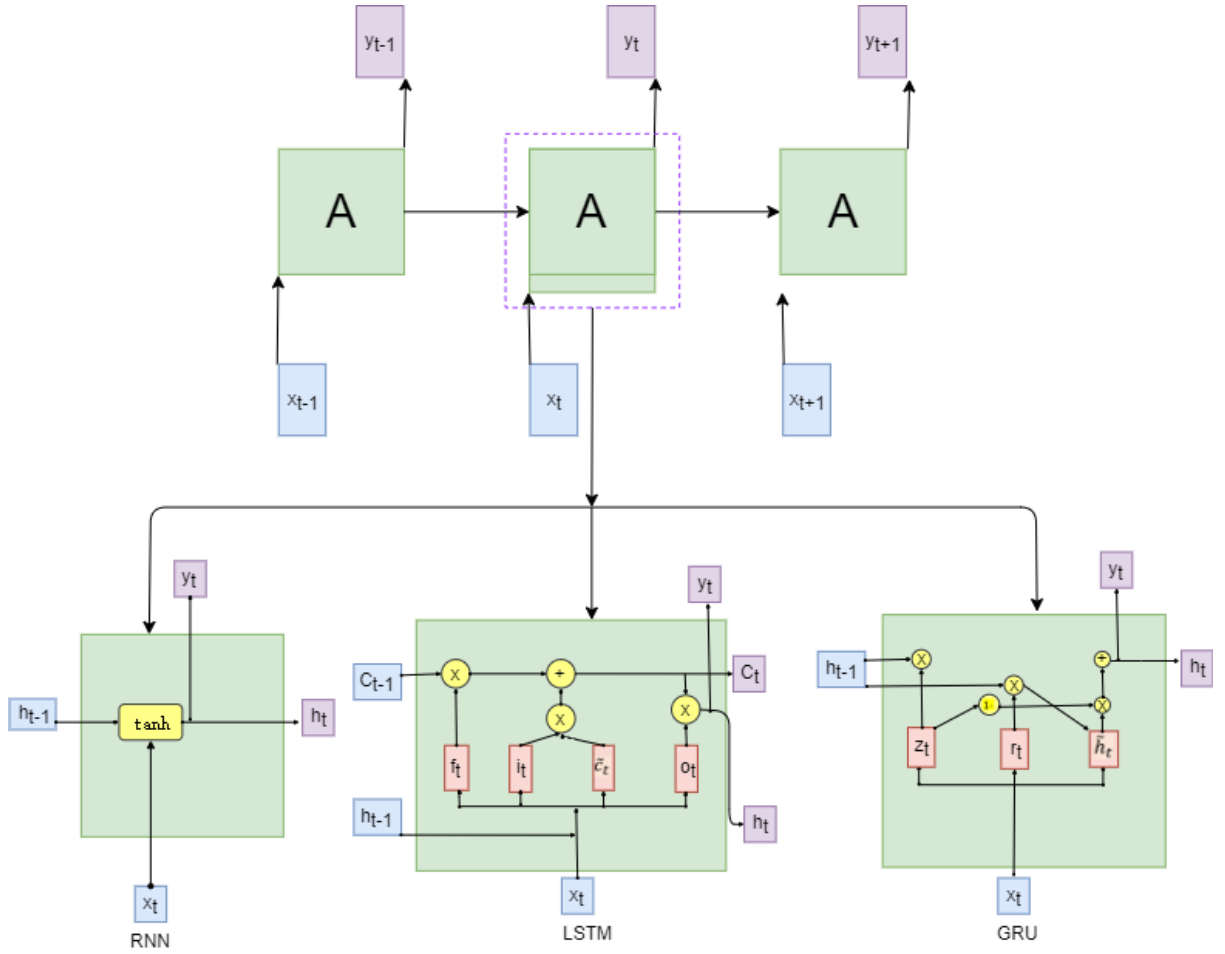


Figure 3-2 Basic structures of RNN, GRU and LSTM

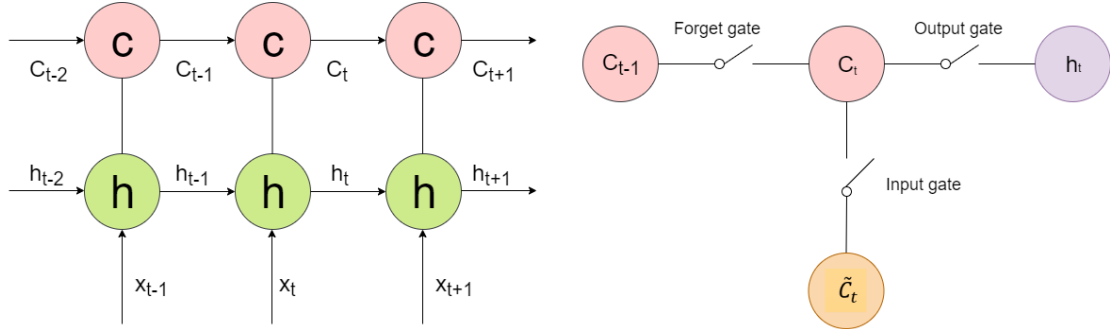


Figure 3-3 Cell state and three gates in LSTM

The output of LSTM at step t is calculated using the following equations:

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \quad (3-3)$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \quad (3-4)$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \quad (3-5)$$

$$\tilde{c}_t = \tanh(U_{\tilde{c}} x_t + W_{\tilde{c}} h_{t-1} + b_{\tilde{c}}) \quad (3-6)$$

$$C_t = C_{t-1} \odot f_t + \tilde{c}_t \odot i_t \quad (3-7)$$

$$h_t = \tanh(c_t) \odot o_t \quad (3-8)$$

where U , W and b are the trainable weights and biases, and the Hadamard multiplication is denoted by $\odot$. The three gates have the same shape with different parameters U and W , which need to be learned from the training process. $\tilde{c}_t$ is the candidate stage, which cannot be used directly. It must pass through the input gate and then be used to calculate the internal storage C_t . While C_t is not only affected by the hidden state but also by C_{t-1} that is controlled by the forget gate. Based on C_t , a layer of $\tanh$ function is applied to the output information h_t , which is constrained by the output gate. The existence of the gates enables LSTM to fulfil the long-term dependencies in the sequence, and by learning the gate parameters, the network can find the appropriate internal storage. Therefore, LSTMs are naturally suited for RUL prediction tasks using sensor data with an inherent sequential nature due to their capability of remembering information over long periods. Yuan *et al.* (2016) proposed an LSTM approach for different types of faults, where the C-MPASS dataset was used as the study case. Compared to the traditional RNN, Gated Recurrent Unit LSTM (GRU-LSTM) and AdaBoost-LSTM showed improved performance in all cases. They developed a vanilla LSTM approach two years later, which further improving the prediction performance (Yuan *et al.*, 2018). A multi-layer LSTM approach was provided by Zheng *et al.* (2017b). The research investigated the hidden patterns from sensors and operational data with multiple operating conditions, fault, and degradation models through combining multiple layers of LSTM cells with standard feed-forward layers. The superiority of this approach in RUL prediction was validated by three widely used data sets, C-MAPSS Data Set, PHM08 Challenge Data Set and Milling Data Set.

3.2.3 Gated Recurrent Unit

The shortcoming of LSTMs is that it is usually time-consuming due to the forget gate, input gate and output gate added to the structure of the memory blocks. Addressing this problem, an improved structure, GRU, was proposed (Cho *et al.*, 2014).

GRU is the latest generation of RNN which looks very similar to LSTM. Instead of using the cell state, GRU uses the hidden state to transfer information. Moreover, it only has two gates (a reset gate and an update gate) instead of three. Like the forget and input gate

of LSTM, the function of the update gate is to decide what information to keep and what to throw away. The function of the reset gate is to determine what to keep from past information.

The output of GRU at step t is calculated using the following equations:

$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z) \quad (3-9)$$

$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r) \quad (3-10)$$

$$\tilde{h}_t = \tanh(U_{\tilde{h}} x_t + W_{\tilde{h}} h_{t-1} \odot r_t + b_{\tilde{h}}) \quad (3-11)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \cdot \tilde{h}_t \quad (3-12)$$

where z_t and r_t represent the update gate and reset gate, respectively; $\tilde{h}_t$ is the candidate hidden state, and h_t is the hidden state. Since there are fewer tensor operations in GRU, it runs relatively faster when training the structure than LSTM. However, the accuracy is behind LSTM due to fewer gates. Thus, GRU could be a good option when the computational resource is limited or short training is required. For instance, Chen *et al.* (2019a) adopted a GRU network to predict the RUL for a complex system featured with multiple components and states, and many parameters.

3.2.4 Bi-directional LSTM

Another variant of RNN called Bi-directional LSTM can be seen frequently in literature in recent years. The Bi-directional LSTM is proposed with the information flowing back to the former LSTM cells. The forward flow of information can discover the system variation, which flows back to smooth the predictions, as illustrated in Figure 3-4. The outputs of the forward path and the backward path are then concatenated. The governing equations of Bi-directional LSTM can be presented as follows:

$$h_i^1 = f(U^1 x_i + W^1 h_{i-1}) \quad (3-13)$$

$$h_i^2 = f(U^2 x_i + W^2 h_{i-1}) \quad (3-14)$$

$$y_i = \text{softmax}(V \odot [h_i^1; h_i^2]) \quad (3-15)$$

where the superscript 1 refers to the forward path and superscript 2 refers to the backward path, y_i is the output of the Bi-directional LSTM obtained by fusing the results from both directional paths.

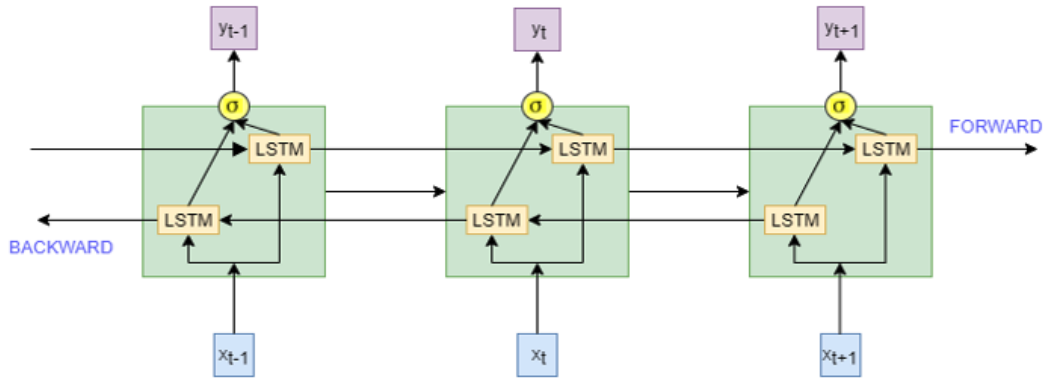


Figure 3-4 A Bi-directional LSTM Structure

As for the application, Zhao *et al.* (2017) presented an integrated approach of CNN and bi-directional LSTM for machining tool wear prediction named Convolutional Bi-directional Long Short-Term Memory (CBLSTM) networks. CNN was first used to extract local robust features from the sequential input. Then, Bi-directional LSTM was utilised to encode temporal information. The proposed CBLSTM's capability of predicting the RUL of actual tool wear based on raw sensory data was verified with a real-life tool wear test. Zhang *et al.* (2018b) presented a Bi-directional LSTM network to discover the underlying patterns embedded in time series to track system degradation. The Bi-directional LSTM network was implemented to track the variation of the health index, and the RUL was predicted by the recursive one-step ahead method. Elsheikh *et al.* (2019b) built a Bidirectional Handshaking LSTM (BHLSTM) network for RUL prediction, where short sequences of monitored observations were given with random initial wear. This method was able to predict the RUL with a random start, which makes it more suitable for real-world application as the initial condition of physical systems is usually unknown, especially in terms of its manufacturing deficiencies.

3.2.5 Research gaps

Many novel RNN-based methods have been proposed in recent years, and the performance of the RUL prediction has been significantly improved. However, most of these works focused on achieving a better prediction performance using a particular approach. Very little research paid attention to other factors that also affect the prediction result, such as the optimiser, activation function and sequence length. Taking the optimisers as an example, they are used to shape the model into its most accurate form through futzing with the weights. Gradient Descent is the most popular algorithm used to

optimize the neural network. Recently, many different optimisers based on Gradient Descent have been developed, especially for DL. To the best of our knowledge, there is no research discussing how different optimisers affect the performance of RUL prediction using DL-based approaches and the principle to optimise the selection. Additionally, as one of the most state-of-the-art algorithms suitable for time-series data related to RUL prediction, RNN and its main variants like LSTM, Bidirectional LSTM, and GRU have not been systematically compared in terms of accuracy, reliability, and repeatability. To fill these research gaps, this paper presents an evaluation of the basic RNN and its variants with different optimisers, activation functions, neuron quantities and sequence lengths based on a case study in a publicly available dataset.

3.3 Optimiser

Gradient descent, by far, is the most used way to optimise neural networks (Ruder, 2016). It is an iterative optimisation algorithm used to find the values of parameters or coefficients of a function that minimizes a cost function. Although various algorithms have been developed to optimize gradient descent, they are usually used as black-box optimisers because it is hard to figure out practical explanations of their strengths and weaknesses.

Different in how much data is used to compute the gradient of the objective function, the gradient descent variants are classified into two categories: batch gradient descent (BGD) and stochastic gradient descent (SGD). BGD, also known as the vanilla gradient descent, is guaranteed to converge to a global minimum for convex error surfaces and a local minimum for non-convex surfaces. However, BGD can be very time-consuming because it needs to calculate the gradients for the whole dataset to perform just one update, and thus it is intractable for datasets that do not fit in memory. In addition, BGD cannot be used to update models online.

In contrast, SGD performs one update at a time. Thus it will not have any redundant computations for large datasets as BGD does. As a result, SGD is usually much faster than BGD. Meanwhile, it can be used to learn the model online. The drawback of SGD is that frequent updates with a high variance would lead to a heavy fluctuation in the objective function. While if the learning rate is slowly decreased over time, SGD shows the same

convergence behaviour as BGD, it almost certainly converges to a local or the global minimum for non-convex optimisation.

Although SGD can often lead to good convergence, a few challenges need to be addressed. For instance, it is difficult to determine a proper learning rate and an annealing schedule, or it is hard to update features to a different extent and avoid suboptimal minima. Ruder (2016) outlines some algorithms widely used by the deep learning community that can deal with these challenges, including Momentum, Nesterov accelerated gradient, AdaGrad, AdaDelta, RMSprop, Adam, AdamMax and Nadam. Ruder also stated that AdaGrad, AdaDelta, RMSprop and Adam could all significantly improve the robustness of SGD and do not need much manual tuning of the learning rate. These four optimisers are therefore selected and discussed in more detail in this paper.

3.3.1 Adaptive Gradient (AdaGrad)

AdaGrad is a gradient-based optimiser that adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent updates. Thus, it is very suitable for sparse data. It uses a different learning rate for every parameter θ_i at every time step t , so the gradient of the objective function $g_{t,i}$ regarding the parameter θ_i at time step t is written as:

$$g_{t,i} = \nabla_{\theta_i} J(\theta_{t,i}) \quad (3-16)$$

The SGD updates for every parameter θ_i at each time step t following

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} \quad (3-17)$$

AdaGrad modifies the general learning rate η at each time step t for every parameter θ_i based on the past gradients:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} \cdot g_{t,i} \quad (3-18)$$

where $G_t \in R^{d \times d}$ is a diagonal matrix where each diagonal element i is the sum of the squares of the gradients regarding the parameter θ_i at time step t , ϵ is a smoothing term used to avoid division by zero.

One of the main advantages of AdaGrad is that it is not required to manually tune the learning rate. The default value is set as 0.01. The main drawback of this optimiser is that its accumulation of the squared gradients in the denominator would shrink the learning

rate, and become infinitesimally small, which means that at a certain point, the algorithm can no longer acquire additional knowledge.

3.3.2 AdaDelta

To reduce the monotonically decreasing learning rate, an extension optimiser of AdaGrad has been promoted named AdaDelta. It uses a fixed-size window of accumulated past gradients instead of accumulating all past squared gradients. The sum of the gradient is recursively defined as a decaying average of all past squared gradients. Thus, the running average of the squared gradients of the objective function at time step t depends on the previous average and the current gradient:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2 \quad (3-19)$$

where γ is the fraction of the update vector of the past time step to the current update vector, which is normally set to 0.9 (Ruder, 2016).

The SGD update for parameter $\Delta\theta_t$ at each time step t therefore becomes:

$$\Delta\theta_t = -\eta \cdot g_{t,i} \quad (3-20)$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (3-21)$$

And according to the update rule, through simply replacing the diagonal matrix G_t with the decaying average of past squared gradients $E[g^2]_t$, the parameter update vector of AdaDelta can be derived as:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_{t,i} \quad (3-22)$$

As $\sqrt{E[g^2]_t + \epsilon}$ is the root mean squared error criterion of the gradient, it can then be written as:

$$\Delta\theta_t = -\frac{\eta}{RMS[g]_t} \cdot g_{t,i} \quad (3-23)$$

Since the update should have the same hypothetical units as the parameter, the exponentially decaying average of the squared parameter should be used:

$$E[\Delta\theta^2]_t = \gamma E[\Delta\theta^2]_{t-1} + (1 - \gamma)\Delta\theta_t^2 \quad (3-24)$$

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} \cdot g_t \quad (3-25)$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (3-26)$$

Based on the update rule of AdaDelta, there is no need to set a default learning rate.

3.3.3 RMSprop

RMSprop is an adaptive learning rate method designed for neural networks, which has grown in popularity in recent years. Like AdaDelta, the central idea of RMSprop is to keep the moving average of the squared gradients for each weight and then divide the gradient by square rooting the mean square. However, a good set default values of decay parameter γ and learning rate is set to 0.9 and 0.001:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad (3-27)$$

$$\theta_{t+1} = \theta_t - \frac{0.001}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_{t,i} \quad (3-28)$$

3.3.4 Adam

Another method that computes adaptive learning rates for each parameter is Adaptive Moment Estimation (Adam) (Kingma and Ba, 2015). Adam not only stores an exponentially decaying average of past squared gradients v_t , but also keeps an exponentially decaying average of past gradients m_t , as shown in Eq. (3-29) and (3-30).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t \quad (3-29)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2 \quad (3-30)$$

where m_t refers to the estimate of the first moment (the mean) of the gradients and v_t refers to the second moment (the uncentered variance) of the gradients. As the initial value of m_t and v_t are vectors of zeros, it is observed that when the decay rates are small during the initial time, they are biased towards zero. The biases are counteracted by computing bias-corrected first and second-moment estimates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3-31)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (3-32)$$

Therefore, the update rule of Adam can be derived as:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t \quad (3-33)$$

The proposed values for β_1 , β_2 and ϵ are 0.9, 0.999 and $\epsilon 10^{-8}$, respectively.

3.4 Activation Function

Activation function is a function working on a neuron in an ANN and mapping the input of the neuron to the output. More specifically, each neuron node in the neural network adapts the output of the neuron in the upper layer as the input and passes it to the next layer (hidden layer or output layer). Thus, the activation function refers to the functional relationship between the output of the upper node and the input of the lower node in the multilayer neural network. Without an activation function, the input of each layer will be linear to the output of the upper layer. No matter how many layers the neural network has, the output is just a linear combination of the input, similar to the original perceptron. The activation function introduces a nonlinear factor to the neuron to enable the neural network arbitrarily to any nonlinear function. The nonlinear activation functions allow the network to learn complex form data and complex function mappings that represent nonlinearity between input and output.

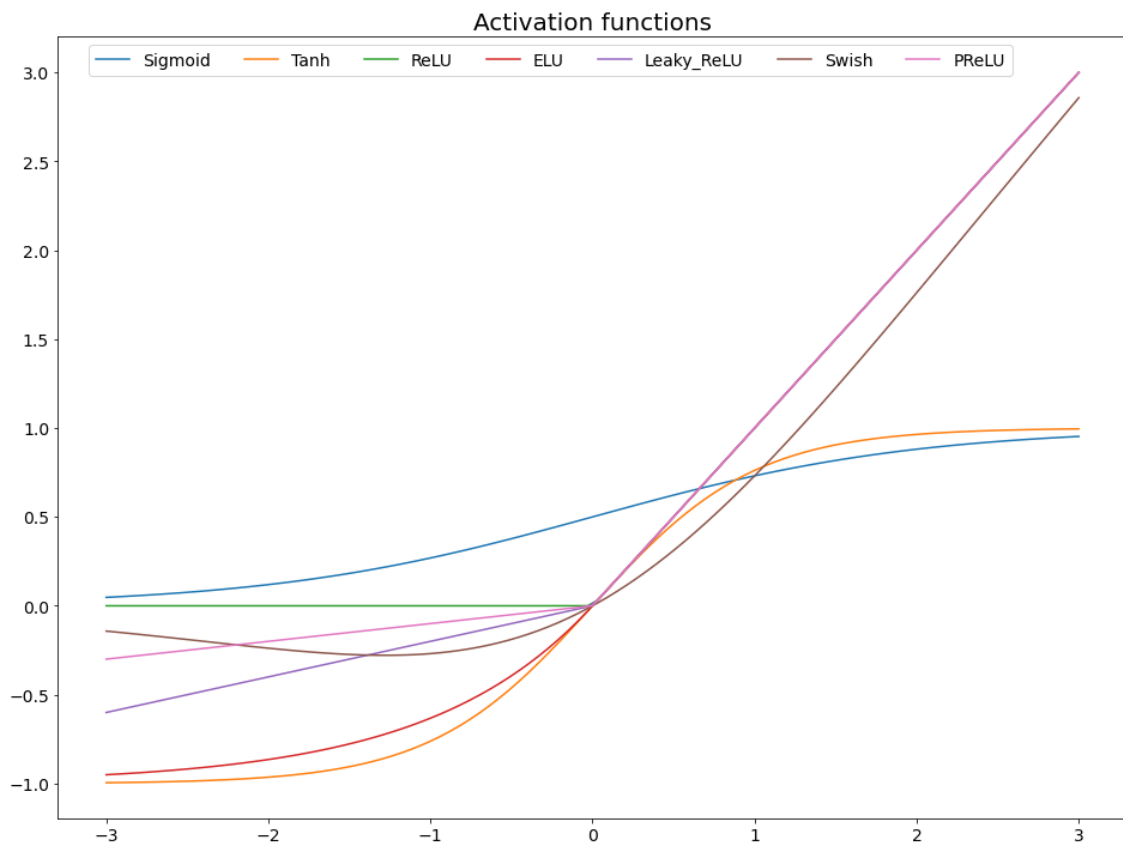


Figure 3-5 The curves of different activation functions

Three activation functions are normally used in the DL area: *tanh* & *sigmoid*, Rectified linear unit (ReLU) and *swish*. In this section, the primary mathematical expression of

these three types of activation functions is reviewed with their advantages and drawbacks. The expression of these activation functions and their variants are demonstrated in Figure 3-5.

3.4.1 The Sigmoid & hyperbolic tangent functions

The Sigmoid function, expressed in Eq. (3-34), also known as the Logistic function, is normally used to the output of the hidden layer neurons.

$$f(x) = \frac{1}{1 + e^{-x}} \quad x \in (-\infty, +\infty) \quad (3-34)$$

The advantage of the Sigmoid function is that the output of the activation function is limited between 0 and 1, which results in a stable optimisation and thus good to be used as the output layer. The drawback is that the function could be very insensitive to small changes in input when a variable takes a very large positive or negative value. During the back propagation, the weight will hardly be updated when the gradient gets close to zero. Therefore, the gradient will disappear, and the network can complete its training. In addition, the output of the sigmoid function is not zero mean, which leads to the input of neurons in the back layer being non-zero mean, and then affects the gradient. Besides, the computational complexity is very high due to the exponential form in the sigmoid function,.

Tanh function, expressed in Eq. (3-35), is also called the hyperbolic tangent function:

$$f(x) = \tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad x \in (-\infty, +\infty) \quad (3-35)$$

Tanh function is the translation and contraction of sigmoid function: $\tanh(x) = 2 \cdot \sigma(2x) - 1$. *Tanh* function often outperforms sigmoid in practice because its output is zero mean. Therefore, the output value can be easily mapped as strongly negative, neutral, or strongly positive. Nevertheless, it still suffers from gradient saturation and computational complexity.

3.4.2 Rectified linear unit function

ReLU is the most used DL neural network activation function. It is the default activation function for most of the feed-forward neural networks. The ReLU function is written as:

$$f(x) = \max(0, x) \quad (3-36)$$

The advantage of the ReLU function is that the computation of the function and its derivative is easier and faster comparing to the sigmoid and tanh functions. Thereby, the training process of the neural network model with ReLU function converges faster no matter which gradient-based optimisation strategy is employed. When the weight is larger than zero, there are no problems like gradient saturation and gradient disappearance. Since there is no need to carry out the exponential operation, the computational complexity is relatively low. A threshold is needed to achieve the activation value. The limitation of the ReLU function is that the output is not zero mean either. Besides, the Dead ReLU Problem will occur when the weight is in the negative field. During the training, when x is less than zero, the gradient of the current neuron and the neurons after it is always zero. In other words, it will no longer respond to any data, and the corresponding parameters will never be updated. To solve this problem, Leaky ReLU, Parametric Rectified Linear Unit (PReLU) and Exponential Linear Unit (ELU) were introduced.

Leaky ReLU function:

$$f(x) = \max(0.01x, x) \quad (3-37)$$

PReLU function:

$$f(x) = \max(\alpha x, x) \quad (3-38)$$

ELU:

$$f(x) = \begin{cases} \alpha(e^x - 1), & x \leq 0 \\ x, & x > 0 \end{cases} \quad (3-39)$$

The Leaky ReLU uses a small value of 0.01 to initialize the neuron so that the ReLU function can be activated in the negative region. The difference between Leaky ReLU and the PReLU is that the α of PReLU function is learned through backpropagation. ELU has all the advantages of ReLU and no Dead ReLU Problem. It can make the average activation mean value of neurons close to zero simultaneously, which suggests that it is more robust to noise. However, the calculation complexity is relatively higher because of the exponential form.

3.4.3 Swish function

Swish is a self-gated activation function proposed by Prajit *et al.* (2018), who attempted to use an automated search technique to find novel activation functions to replace the ReLU function without changing the network architecture. By a combination of exhaustive and reinforcement learning-based search, they found a few novel promising activation functions and named the best one of them as Swish.

The Swish function can be written as:

$$f(x) = x \cdot \text{sigmoid}(\beta x) \quad (3-40)$$

where β is a constant or trainable parameter.

3.5 Case study

3.5.1 Benchmark dataset overview

The case study focuses on investigating the influence of various practical options of optimisers, activation functions and other parameters like sequence length and neuron number when adopting RNN and its variants on RUL prediction. We selected NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, aiming at modelling the damage propagation of aircraft gas turbine engines (Saxena *et al.*, 2008a). This engine simulator produced four datasets which consisted of three operating condition indicators. Each subset has different numbers of engines with varied operating cycles. In the dataset, engine profiles were simulated with different initial degradation conditions. The maintenance was not considered during the simulation. The dataset includes one training set and one testing set for each engine. The training set consists of the historical run-to-failure measurement records of the engines from 21 onboard sensors. The objective is to predict the RUL of each engine based on the given sensor measurements. The information on the four subsets is listed in Table 3-1. FD001 refers to the engine failure arising from the high-pressure compressor under a single operating condition. FD002 refers to the engine failure from the high-pressure compressor under six operation conditions. FD003 refers to the engine failure from the high-pressure compressor and fan under a single operating condition. FD004 refers to the engine failure from both high-pressure compressor and fan under six operation conditions. In this study case, FD001

was used because the data volume is relatively small. Therefore it is more time-efficient to do the test.

Table 3-1 C-MAPSS dataset

Dataset	FD001	FD002	FD003	FD004
Number of engines for training	100	260	100	249
Number of engines for testing	100	259	100	248
Operating conditions	single	multiple	single	multiple
Fault conditions	compressor	compressor	compressor & fan	compressor & fan

3.5.2 Data pre-processing

The raw sensor data were normalised to [0, 1]. No dimension reduction and feature extraction have taken place in this case study, and the entire sensor data stack was used as input for training. In addition, since there is no target output in raw datasets, the RUL was labelled at every cycle for each sample before training the models.

3.5.3 Performance evaluation

In this case study, the mean square error (MSE) was used to evaluate the performance of the trained neural networks. The mathematical expression is:

$$MSE = \frac{1}{n} \sum_{i=1}^n d_i^2 \quad (3-41)$$

where n is the total number of actual RUL targets in the related test set and d_i refers to the difference between the true RUL and the predicted RUL.

The RNN algorithm and its variants were tested with the dataset FD001, and three different layer structures for each method were used. Each algorithm and structure were tested five times to achieve the statistical result illuminated by box charts. The results were presented and discussed according to four main factors: optimisers, activation functions, neuron numbers and sequence lengths against three assessment criteria: robustness, precision, and accuracy. The base parameters are: Optimiser=ReLU, Activation function=AdaDelta, Neuron Number=128, Sequence Length=50. The ranks for precision and accuracy for these four factors will be presented. As for robustness, if

the network can produce a reliable RUL prediction result, it will be marked as 1; otherwise, it will be marked as 0.

3.5.3.1 Optimisers

As indicated in Figure 3-6, gradient exploring or gradient vanishing took place when adopting AdaGrad in RNN_2LAYERS, RNN_3LAYERS, LSTM_2LAYERS, LSTM_3LAYERS and GRU_3LAYERS and AdaDelta in RNN_LAYERS, Bi_LSTM_3LAYERS. This observation suggests that RMSprop and Adam are less sensible to the parameters than AdaGrad and AdaDelta, which means they are more workable. All four optimisers can produce generally accurate results, while the precision of RMSprop is relatively poor. More specifically, AdaGrad and AdaDelta are more likely to lose their workability when the network gets more complicated. However, using a simple network structure can often produce more accurate results.

The assessment of the four optimisers has been made for all network structures, such as the example set in Table 3-2, which only one layer RNN is used in the model. The optimal optimisers have been summarized in Table 3-3.

Table 3-2 Assessment of different optimisers using network structure RNN-1LAYER

Optimisers	Robustness	Precision (rank)	Accuracy (rank)	Assessment
RMSprop	1	4	4	
Adam	1	3	3	
AdaGrad	1	1	1	Recommended
AdaDelta	1	2	2	

Table 3-3 Optimal optimisers for different model structures

Model structures	Optimal optimisers
RNN_1LAYER	AdaGrad
RNN_2LAYERS	Adam
RNN_3LAYERS	AdaGrad
LSTM_1LAYER	AdaGrad
LSTM_2LAYERS	Adam
LSTM_3LAYERS	Adam, AdaDelta

Bi_LSTM_1LAYER	Adam, AdaGrad, AdaDelta
Bi_LSTM_2LAYERS	AdaGrad
Bi_LSTM_3LAYERS	AdaGrad
GRU_1LAYER	Adam, AdaGrad
GRU_2LAYERS	AdaGrad
GRU_3LAYERS	AdaGrad

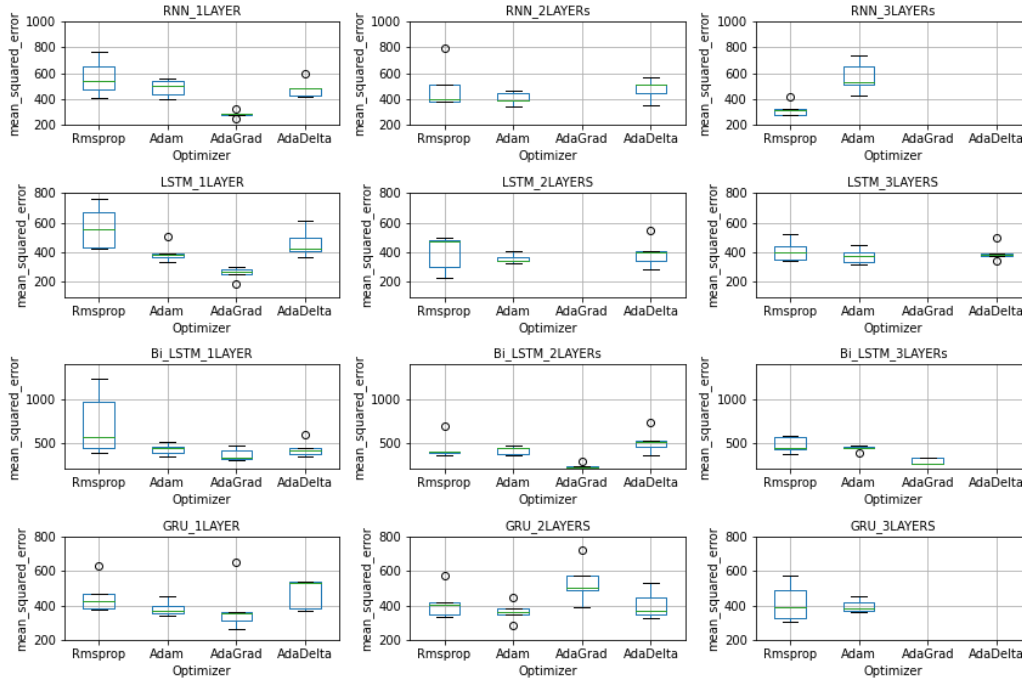


Figure 3-6 Performance of different optimisers on the testing dataset

3.5.3.2 Activation functions

Both *Sigmoid* and *Tanh* functions were tested, but these two activation functions were significantly affected by gradient vanishing and gradient exploring problems. Therefore, these two functions are not shown in the figure. As demonstrated in Figure 3-7, the performance of ReLU, Leaky_ReLU, PReLU and ELU is quite similar, and they are generally better than Swish in precision and accuracy perspective. However, gradient exploring, or gradient vanishing occurred when adopting ReLU, PReLU and ELU in RNN_1LAYER and GRU_3layer, which suggests that Swish and Leaky_ReLU are more robust than these three activation functions. The Optimal activation functions in this case for different model structures are listed in Table 3-4.

Table 3-4 Optimal activation functions for different model structures

Model structures	Optimal activation functions
RNN_1LAYER	Leaky_ReLU
RNN_2LAYERS	Leaky_ReLU, ELU
RNN_3LAYERS	ELU
LSTM_1LAYER	ReLU
LSTM_2LAYERS	PreLU
LSTM_3LAYERS	ELU
Bi_LSTM_1LAYER	PreLU
Bi_LSTM_2LAYERS	ReLU, Leaky_ReLU
Bi_LSTM_3LAYERS	PreLU
GRU_1LAYER	ReLU
GRU_2LAYERS	ReLU
GRU_3LAYERS	Swish

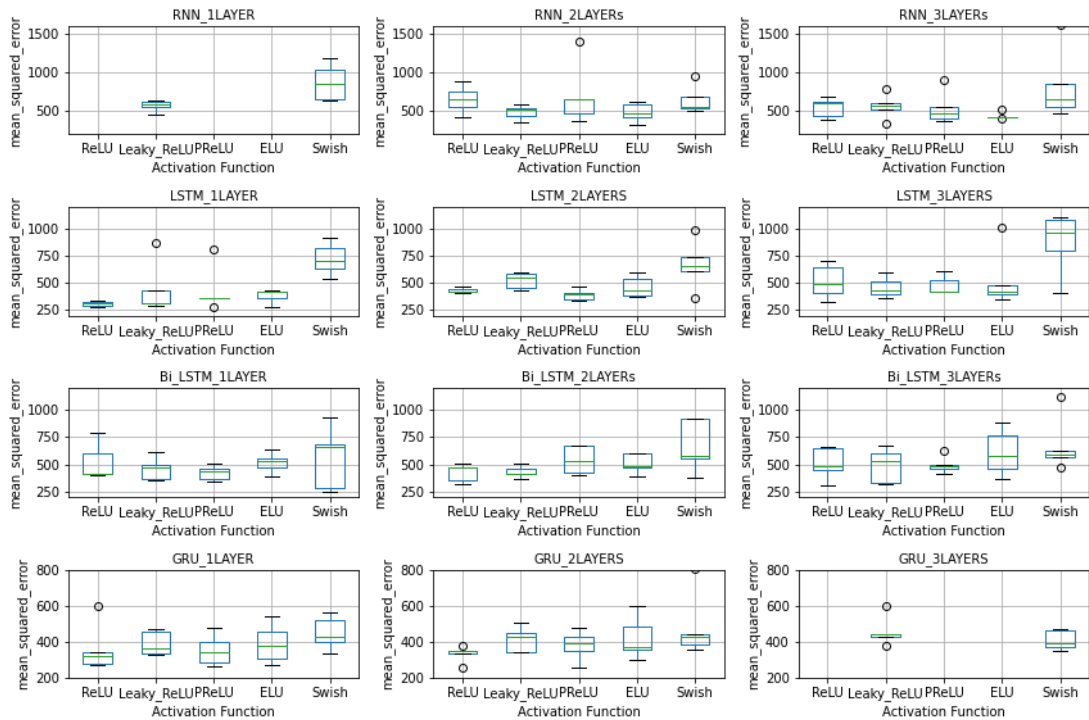


Figure 3-7 Performance of different activation functions on the test dataset

3.5.3.3 Sequence length

As indicated in Figure 3-8, generally, the longer the sequence length, the better the algorithms achieve the RUL prediction performance. In this case, gradient vanishing

happened when adopting the GRU_3LAYERS network structure, which suggests that the choice of the sequence length may also affect the workability of GRU. The optimal sequence length for different model structures is listed in Table 3-5, considering the workability, precision, and accuracy.

Table 3-5 Optimal sequence length for different model structures

Model structures	Optimal sequence length
RNN_1LAYER	50
RNN_2LAYERS	50,75,100
RNN_3LAYERS	75
LSTM_1LAYER	100
LSTM_2LAYERS	100
LSTM_3LAYERS	100
Bi_LSTM_1LAYER	100
Bi_LSTM_2LAYERS	75
Bi_LSTM_3LAYERS	100
GRU_1LAYER	75,100
GRU_2LAYERS	100
GRU_3LAYERS	100

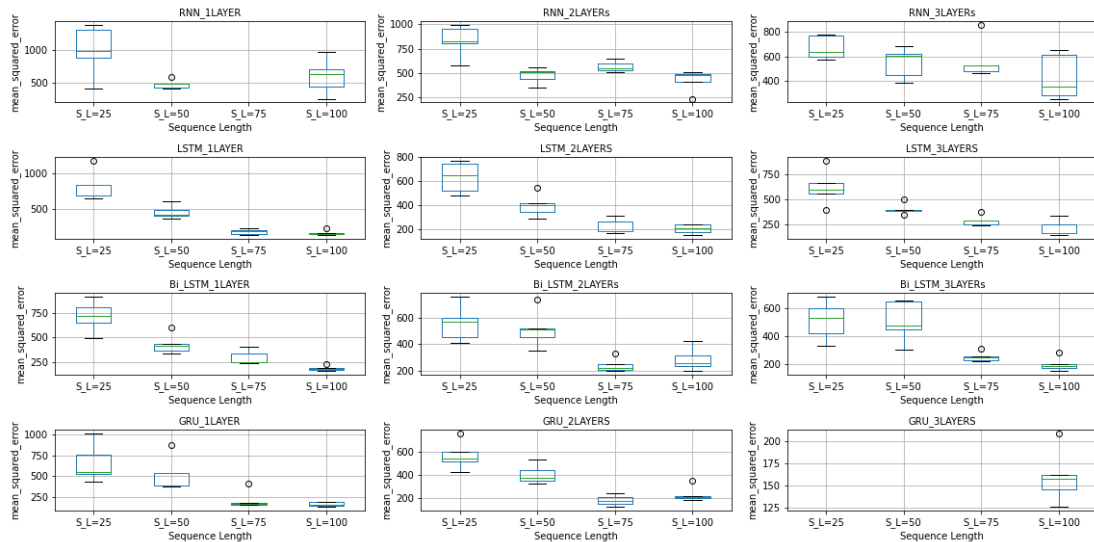


Figure 3-8 Performance of different sequence lengths on the test dataset

3.5.3.4 Neuron number

Figure 3-9 displays the influences of different Neuron numbers on the performance of RUL prediction. Gradient exploring, or gradient vanishing occurs when using network structure RNN_1LAYER and all GRU structures, which may suggest that neuron number is a sensitive parameter for RNN and GRU in terms of workability. The performance of different neuron numbers varies significantly using different model structures. Taking the LSTM network structure as an example, the influence of different neuron numbers is smaller when using LSTM_1LAYER than in the other two. In addition, for LSTM_3LAYERS, the observation shows that the more neuron number used, the less accurate the result turns out to be, while this tendency cannot be found in the other two network structures. As only three different neuron numbers were tested, the optimal neuron for each network structure cannot be achieved. Nevertheless, the optimal neuron number for different model structures, in this case, is listed in Table 3-6.

Table 3-6 Optimal Neuron number for different model structures

Model structures	Optimal Neuron number
RNN_1LAYER	64,32
RNN_2LAYERS	32
RNN_3LAYERS	32
LSTM_1LAYER	128
LSTM_2LAYERS	32
LSTM_3LAYERS	32
Bi_LSTM_1LAYER	32
Bi_LSTM_2LAYERS	64
Bi_LSTM_3LAYERS	64
GRU_1LAYER	128
GRU_2LAYERS	128
GRU_3LAYERS	64

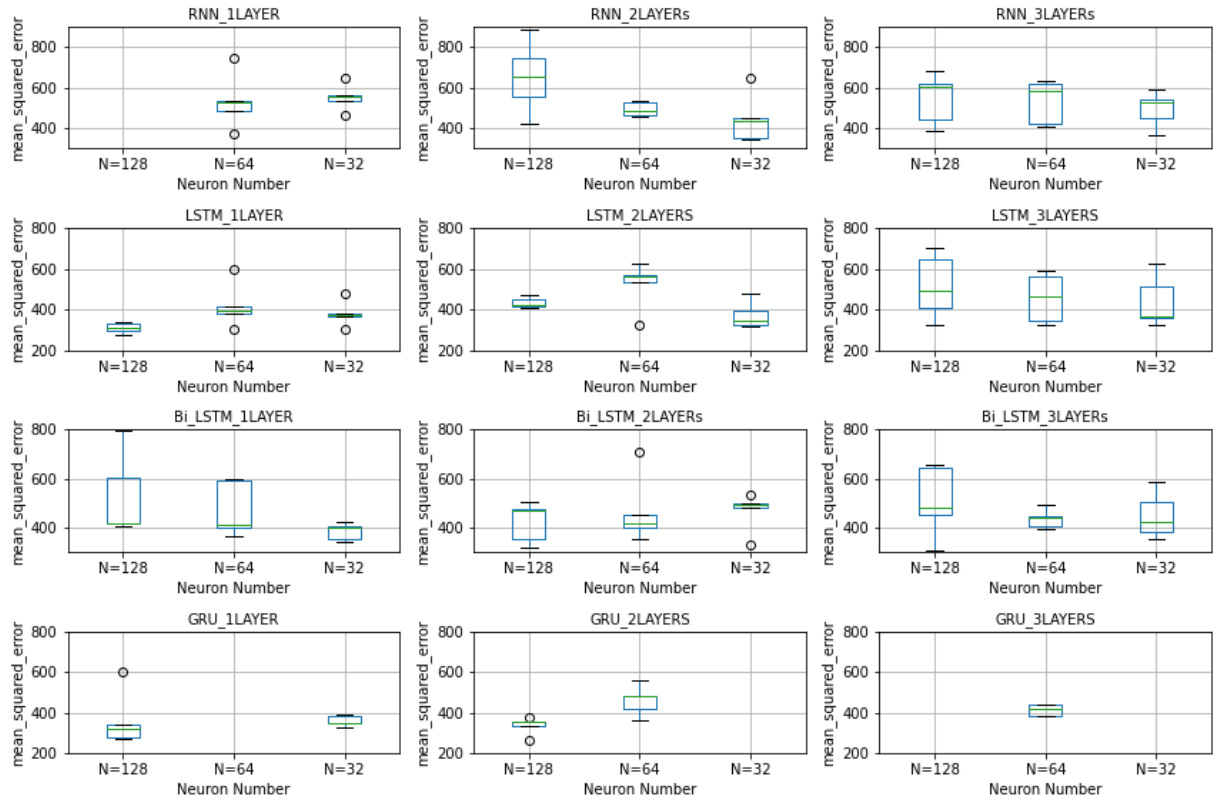


Figure 3-9 Performance of different Neuron numbers on the test dataset

3.5.3.5 Overall performance of each Algorithm

Figure 3-10 demonstrates the performance of different model structures using a certain group of parameters. As in this case, gradient exploring or gradient vanishing occurred when using RNN_1LAYER and GRU_3LAYRES. Generally, the performance of LSTM, Bi_LSTM and GRU network structures seems relatively close and significantly better than RNN. The accuracy of LSTM is close to Bi_LSTM and GRU, but the precision is relatively poor. GRU turns out to be very accurate and precise, but it suffers from robustness problems. Thus, a Bi_LSTM structure might be a better option in this case.

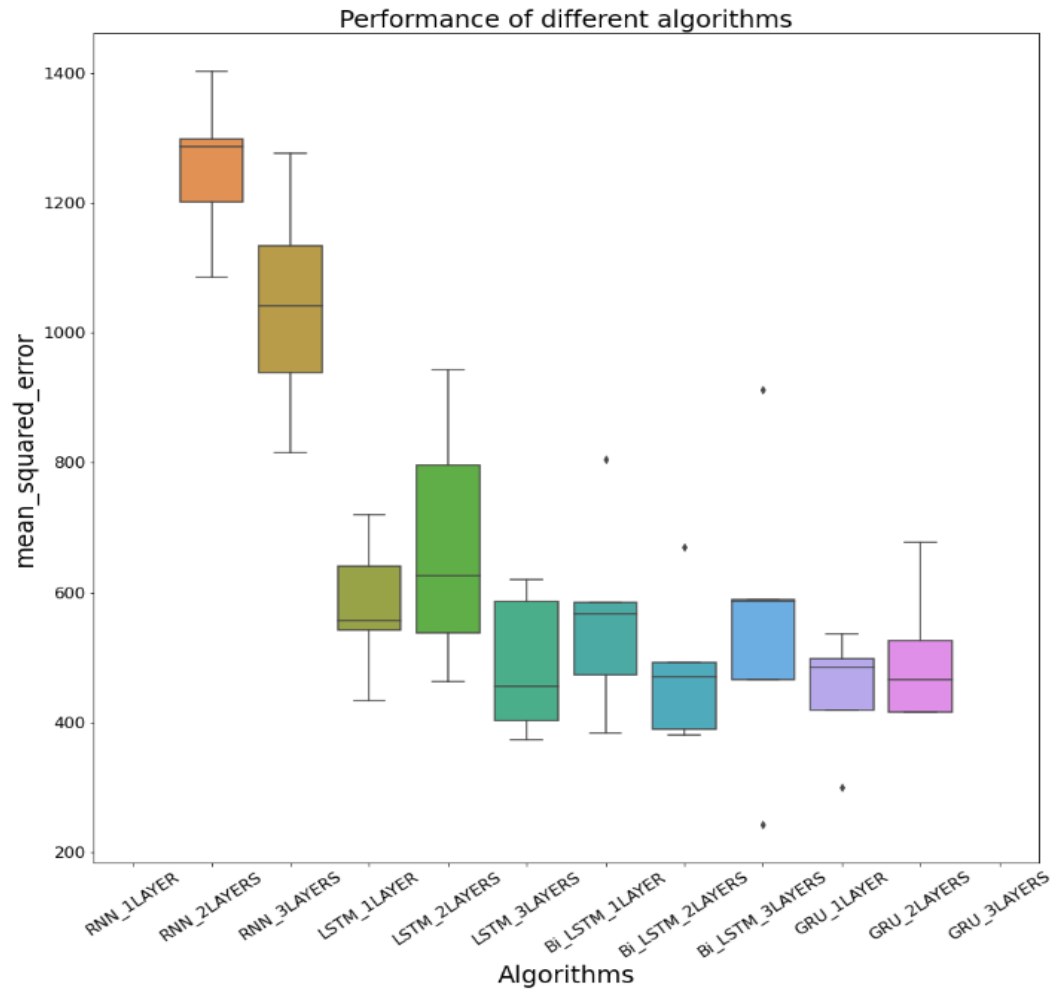


Figure 3-10 Performance of different model structures on Dataset FD001, where Optimiser: Adam; Activation function: ReLU; Neuron Number:128; Sequence Length: 50

Using the variable-controlling approach, different parameters of the model are tested using different model structures. A more detailed comparison is displayed in Table 3-7. The optimal parameters for this dataset are pointed out using a purple hatch according to every network structure. Although the optimal parameters can still not be selected for the dataset based on this table, it provides some useful options.

Table 3-7 The average MSE was recorded with regard to different parameters.

Model structures	Activation function					Optimiser				Neuron Number			Sequence Length			
	ReLU	L_ReLU	PReLU	ELU	Swish	Rms prop	Adam	AdaGrad	AdaDelta	128	64	32	25	50	75	100
RNN_1L		569				569	487	286	478		533	553	994	478		596
RNN_2L	651	491	710	488	646	494	406		476	651	494	444	930	476	565	425
RNN_3L	547	563	539	440	831	325	572		547	547	531	493	670	547	570	431
LSTM_1L	310	443	439	382	725	566	397	260	464	310	419	379	836	464	196	181

LSTM_2L	432	526	391	464	670	398	358		396	432	522	369	632	396	221	199
LSTM_3L	515	460	480	531	872	411	375		397	515	458	436	617	397	284	209
Bi_LSTM_1L	526	458	423	516	562	718	422	362	428	526	473	385	714	428	294	185
Bi_LSTM_2L	423	431	538	507	672	441	413	228	513	423	465	467	557	513	243	286
Bi_LSTM_3L	508	492	494	611	673	473	434	285	508	508	435	450	512	508	253	198
GRU_1L	363	393	356	392	450	457	383	389	540	363		360	658	540	220	166
GRU_2L	336	415	383	423	486	417	365	535	405	336	459		570	405	179	231
GRU_3L		458			410	417	398				414					160

3.6 Conclusions

In the age of big data, deep learning offers modern manufacturing great potential with its advanced computational analytical capability. With the ability to discover the variation pattern underlying time series, RNN gains increasing attention in RUL prediction. In this paper, an evaluation of RNN and its variants has been conducted using NASA's C-MAPSS dataset, which uses the sensing data onboard to predict the engines' RUL. Different parameters, such as optimisers, activation functions, neuron quantities and sequence length, which might affect the prediction performance, are discussed using a sensitivity analysis. Some of the network structures are very parameter sensitive from the result of the evaluation. Take a one-layer RNN as an example, gradient exploring problem or gradient vanishing problem may occur when using certain sets of parameters, such as an ELU activation function or a sequence length of 75. The influence of these parameters on the performance of RUL prediction is different according to different network structures. Instead of giving the optimal parameters and the network structures for the dataset, the result of this case study offers some practical choices of parameters for different network structures. Although the conclusions from this case study could not be directly applied to other cases, it at least suggests the influence of these factors on the RUL prediction. Moreover, it provides some options for researchers when considering adopting DL to carry out similar prediction tasks in their own cases.

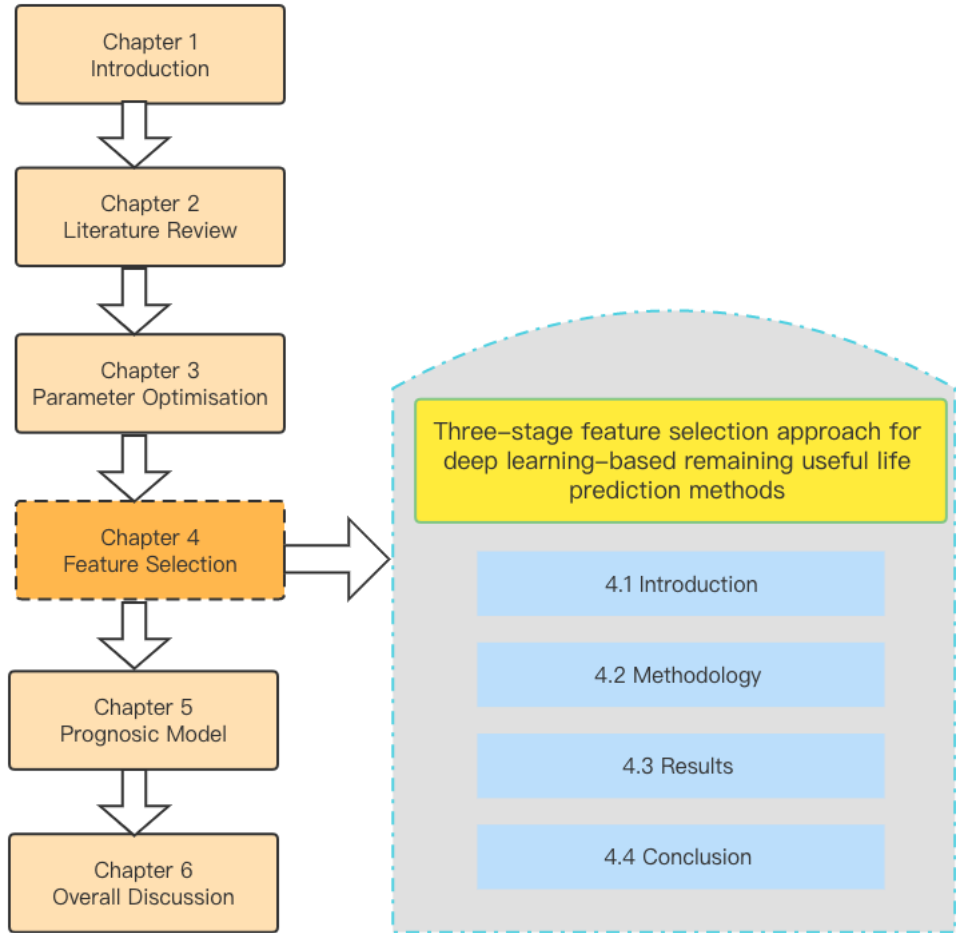
References

- Chen, J., Jing, H., Chang, Y., Liu, Q., 2019. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliab Eng Syst Saf* 185, 372–382. <https://doi.org/10.1016/j.ress.2019.01.006>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* 1724–1734. <https://doi.org/10.3115/v1/d14-1179>
- Deutsch, J., He, D., 2017. Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components. *Journal of Zoo and Wildlife Medicine* 40, 321–327. <https://doi.org/10.1109/TSMC.2017.2697842>
- Elsheikh, A., Yacout, S., Ouali, M.S., 2019. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* 323, 148–156. <https://doi.org/10.1016/j.neucom.2018.09.076>
- Gan, M., Wang, C., Zhu, C., 2016. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mech Syst Signal Process* 72–73, 92–104. <https://doi.org/10.1016/j.ymssp.2015.11.014>
- Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Comput* 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Jia, F., Lei, Y., Lin, J., Zhou, X., Lu, N., 2016. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech Syst Signal Process* 72–73, 303–315. <https://doi.org/10.1016/j.ymssp.2015.10.025>
- Kabir, A., Bailey, C., Lu, H., Stoyanov, S., 2012. A review of data-driven prognostics in power electronics. *Proceedings of the International Spring Seminar on Electronics Technology* 189–192. <https://doi.org/10.1109/ISSE.2012.6273136>
- Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings 1–15.

- Kraus, M., Feuerriegel, S., 2019. Forecasting remaining useful life: Interpretable deep learning approach via variational Bayesian inferences. *Decis Support Syst* 125. <https://doi.org/10.1016/j.dss.2019.113100>
- Leser, P.E., 2017. Probabilistic Prognostics and Health Management for Fatigue-critical Components using High-fidelity Models (Sissertation). North Carolina.
- Liao, L., Jin, W., Pavel, R., 2016. Enhanced Restricted Boltzmann Machine with Prognosability Regularization for Prognostics and Health Assessment. *IEEE Transactions on Industrial Electronics* 63, 7076–7083. <https://doi.org/10.1109/TIE.2016.2586442>
- Ma, J., Su, H., Zhao, W., Liu, B., 2018. Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning. *Complexity* 2018, 1–13. <https://doi.org/10.1155/2018/3813029>
- Okoh, C., Roy, R., Mehnen, J., Redding, L., 2014. Overview of Remaining Useful Life prediction techniques in Through-life Engineering Services. *Procedia CIRP* 16, 158–163. <https://doi.org/10.1016/j.procir.2014.02.006>
- Ruder, S., 2016. An overview of gradient descent optimization algorithms 1–14.
- Salunkhe, T., Jamadar, N.I., Kivade, S.B., 2014. Prediction of Remaining Useful Life of Mechanical Components-A Review 3, 125–135.
- Saxena, A., Goebel, K., Simon, D., Eklund, N., 2008. Damage propagation modeling for aircraft engine run-to-failure simulation. 2008 International Conference on Prognostics and Health Management, PHM 2008. <https://doi.org/10.1109/PHM.2008.4711414>
- Shao, H., Jiang, H., Zhao, H., Wang, F., 2017. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. *Mech Syst Signal Process* 95, 187–204. <https://doi.org/10.1016/j.ymssp.2017.03.034>
- Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., 2018. Deep learning for smart manufacturing: Methods and applications. *J Manuf Syst* 48, 144–156. <https://doi.org/10.1016/j.jmsy.2018.01.003>
- Wang, P., Ananya, Yan, R., Gao, R.X., 2017. Virtualization and deep recognition for system fault classification. *J Manuf Syst* 44, 310–316. <https://doi.org/10.1016/j.jmsy.2017.04.012>

- Yuan, M., Wu, Y., Dong, S., Lin, L., Liu, Y., 2018. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* 275, 167–179. <https://doi.org/10.1016/j.neucom.2017.05.063>
- Yuan, M., Wu, Y., Lin, L., 2016. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. *AUS 2016 - 2016 IEEE/CSAA International Conference on Aircraft Utility Systems* 135–140. <https://doi.org/10.1109/AUS.2016.7748035>
- Zhang, J., Wang, P., Yan, R., Gao, R.X., 2018. Long short-term memory for machine remaining life prediction. *J Manuf Syst* 48, 78–86. <https://doi.org/10.1016/j.jmsy.2018.05.011>
- Zhang, W., Jia, M.-P.P., Zhu, L., Yan, X.-A.A., 2017. Comprehensive Overview on Computational Intelligence Techniques for Machinery Condition Monitoring and Fault Diagnosis. *Chinese Journal of Mechanical Engineering (English Edition)* 30, 782–795. <https://doi.org/10.1007/s10033-017-0150-0>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2019. Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process* 115, 213–237. <https://doi.org/10.1016/j.ymssp.2018.05.050>
- Zhao, R., Yan, R., Wang, J., Mao, K., 2017. Learning to monitor machine health with convolutional Bi-directional LSTM networks. *Sensors (Switzerland)* 17, 1–18. <https://doi.org/10.3390/s17020273>
- Zheng, S., Ristovski, K., Farahat, A., Gupta, C., 2017a. Long Short-Term Memory Network for Remaining Useful Life estimation. *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017* 88–95. <https://doi.org/10.1109/ICPHM.2017.7998311>
- Zheng, S., Ristovski, K., Farahat, A., Gupta, C., 2017b. Long Short-Term Memory Network for Remaining Useful Life estimation. *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017* 88–95. <https://doi.org/10.1109/ICPHM.2017.7998311>
- Zoph, B., Le, Q. V., 2018. Searching for activation functions. *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings* 1–13.

4 CHAPTER FOUR—FEATURE SELECTION



Three-stage feature selection approach for deep learning-based remaining useful life prediction methods

Abstract

The remaining useful life (RUL) prediction plays an increasingly important role in predictive maintenance. With the development of Big Data and the Internet of Things, Deep Learning (DL) techniques have been widely adopted for RUL prediction. Addressing the limitation of the current approaches for data under multiple operating conditions, this paper proposes a three-stage feature selection approach for DL-based RUL prediction models. The K-medoids cluster is initially used to sort raw data based on different operating conditions. In the first stage of feature selection, an operation-based normalisation approach is applied to reconstruct the data. Afterwards, Spearman's rank and pair-wise Pearson correlation coefficients are used to eliminate irrelevant and redundant features in the second and third stages, respectively. A case study using NASA's C-MAPSS dataset is presented to quantitatively evaluate the influence of the proposed feature selection method using the Recurrent Neural Network (RNN) and its' variants, enhanced by an optimised activation function and optimiser. The results confirm that the proposed method can improve the robustness of DL models and achieve about 7.3% improvement in the RUL prediction for popular and state-of-the-art DL models.

Keywords

Remaining useful life prediction, Deep Learning, Long Short-Term Memory, Feature engineering

4.1 Introduction

The economic cost caused by the failure of highly complex and automated machines has become increasingly expensive. Thus, a more effective maintenance strategy is crucial to improve machines' overall reliability and productivity. With the rapid development of Internet-of-Things (IoT) and cyber manufacturing techniques in recent years, the manufacturing industry has adopted condition-based maintenance (CBM). The rapid development of modern sensor technology and condition monitoring systems enhances the possibilities of utilising real-time and historical data to comprehensively manage maintenance (Vogl *et al.*, 2019). The remaining useful life (RUL) prediction of components or systems is the core and major challenging role in the CBM of machines (Li *et al.*, 2016).

RUL prediction generally refers to the study of predicting the specific time length from the current time to the end of the useful life of an asset or system (Salunkhe *et al.*, 2014). Approaches to RUL prediction can be catalogued into model-based methods, data-driven methods, and hybrid methods. Model-based approaches, also called physics-based approaches, evaluate the health condition of a system by building mathematical models based on the failure mechanisms or the first principle of damage (Cubillo *et al.*, 2016b). In data-driven approaches, RUL is computed through statistical and probabilistic methods by utilising historic information and routinely monitored data of the system (Javed *et al.*, 2017). Complex and noisy working conditions often impede the construction of a physical model. Meanwhile, it is often difficult to determine the parameters of the physical model. In contrast, the requirements for data-driven methods to model the degradation and predict RUL are much easier to be satisfied, especially with the fast development of big data and IoT. Therefore, data-driven approaches are more widely used in RUL prediction at present. Hybrid approaches, combining model-based approaches and data-driven approaches, aim to leverage the advantages of both categories. Dawn *et al.* (2015) further classified the data-driven approaches into artificial intelligence (AI) approaches and statistical approaches. AI approaches intend to learn the degradation patterns from available observations directly, which can deal with prognostic issues of complex mechanical systems (Lei *et al.*, 2018). Statistical approaches predict the RUL by fitting

the empirical model as close as possible to the collected data and extrapolating the fitted curve to failure criteria.

Deep learning (DL), a subset of AI approaches that has been attracting significant investigations in the last few years in RUL prediction, can be used to extract multilevel features from data (Ma *et al.*, 2018). As an end-to-end machine learning method, it can automatically process the original signal, identify discriminative and trend features in the input data layer by layer, and then directly output the classification or regression result. Because of its strength in self-learning features, DL has already achieved great success in applications in the manufacturing industry (Zhao *et al.*, 2019b). There are mainly four representatives of DL-based architectures for RUL prediction, including Auto-encoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) (Zhao *et al.*, 2015). AE and DBN are often used for pre-training of networks while CNN and RNN are generally used as predictive models. For example, Lu *et al.* (2017) proposed a stacked denoising autoencoder (SDA) which is fed with the signals containing ambient noise and working condition fluctuations to conduct the fault diagnosis of rotary machinery components. Gan *et.al.* (2016) developed a hierarchical diagnosis network to identify the bearing fault location and severity by combining a wavelet packet transform and DBN. It has been proven that both CNN and RNN outperform the traditional prognosis algorithms in RUL prediction, while CNN-based approaches are used more in fault diagnosis and surface integration inspection (Wang *et al.*, 2018). RNN, on the other hand, gained much more attention and achievements in the field of RUL prediction because it is good at dealing with time sequence data (Zheng *et al.*, 2017b). RNNs can be trained by backpropagation through time for supervised tasks with sequential input data and target outputs. The limitation of a vanilla RNN is that it may not capture long-term dependencies during the backpropagation for model training, resulting from the vanishing gradient or exploding gradient problem. Therefore, Long-short term memory (LSTM), a modified structure of the recurrent cell that incorporates the standard recurrent layer along with additional “memory” control gates, was developed. Besides, Gated Recurrent Unit (GRU), Bi-directional Long-short term memory (Bi-LSTM) and many other variants of RNN are also designed and adopted in the manufacturing field in the last few years. Zheng *et al.*

(2017b) presented a multi-layer LSTM approach to investigate the hidden patterns from sensors and operational data with multiple operating conditions, fault and degradation models. Chen *et al.* (2019a) utilised a GRU network to establish the nonlinear deterioration model for RUL prediction. Elsheikh *et al.* (2018) built a Bidirectional Handshaking LSTM (BHLSTM) network for RUL prediction, where short sequences of monitored signals were provided with random initial wear.

The main challenge of degradation monitoring is to extract useful representative features from the collected raw data (Li *et al.*, 2019b). Time-domain and frequency-domain-based features are commonly adopted in signal processing, and these features usually consider the degradation model stationary until the occurrence of a fault. The limitation is that some of these features may be only effective for a specific operating stage (Tran *et al.*, 2012). Moreover, in the modern manufacturing field, machines often work under more complex environments such as multiple operating conditions, making it even harder to extract the representative features. Different operating conditions may lead to different degradation models for the same machine, often reflected by the dynamic of the data collected from the sensors, which challenges the applicability of the existing stationary features. The most popular solution to address this challenge nowadays is adopting DL-based methods to extract features from data automatically. Although DL methods are claimed to be able to find out the hidden patterns of the dynamic data, it remains a black box and lacks the transparency required for further system improvement. Zhu *et al.* (2020) developed a novel transfer learning method based on multiple layer perceptron (MLP) to solve the distribution discrepancy problem caused by multiple working conditions. The domain adaptation modules are used to learn the time-invariant features, which have been proven effective and advantageous in RUL prediction. Another solution is to utilise feature engineering methods to address data degradation patterns. The predictive model can extract more representative features to accommodate different operating conditions with the degradation patterns addressed. Kundu *et al.* (2019) propose a Weibull Accelerated Failure Time Regression (WAFTR) model to extract representative frequency domain features of the rolling element bearings that operate at different operating conditions. When the operating condition data is not considered in the RUL prediction model, a high underestimation and overestimation of RUL is observed. In

contrast, the prediction accuracy improves significantly when the operating condition data is involved in the model.

In this paper, a three-stage feature selection approach, enhanced by an optimised DL-based model is presented to better predict the RUL of a system under multiple operating conditions. An operation-based normalisation method is proposed to better expose the hidden degradation patterns from the data and enhance the RUL prediction performance. A public dataset is then employed to demonstrate the superior performance of the introduced feature engineering methods and DL optimisation.

4.2 Methodology

As shown in Figure 4-1, the proposed RUL prediction framework consists of three phases: data exploration, feature selection, and RUL prognostics. In the first phase (the green block), a k-medoids clustering method is applied to address the multiple operating conditions. An operation-based normalisation method is used to reconstruct the raw dataset first in the second phase (the yellow block). Then significant and lean sensors are selected using two correlation analysis methods for preparing training and testing datasets. In the third phase (the blue-green block), the transformed training data are then used to train a two-layer LSTM predictive model, where different activation functions and optimisers are optimised to form the best-performed model (the purple block). After that, the transformed testing data are fed into the trained predictive model to produce the RUL estimation for the model deployment. Details of each phase are presented below.

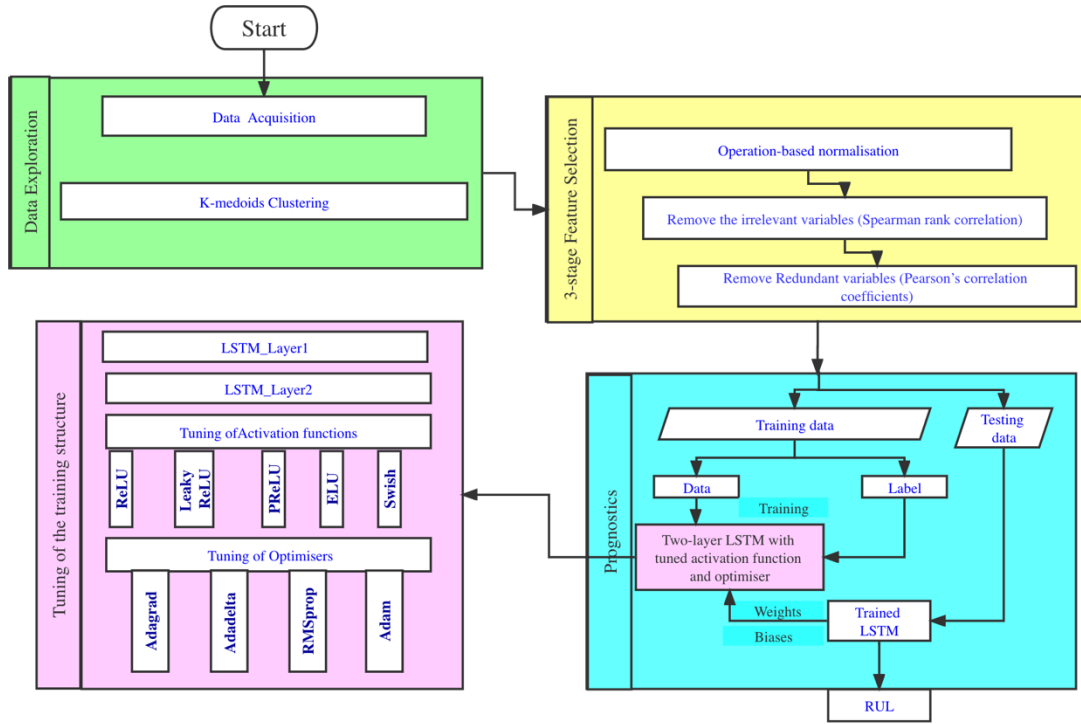


Figure 4-1 Flow chart of the proposed RUL prediction framework

4.2.1 Data Exploration

The data collected for the RUL prediction of a system is generally time series data. In many cases, data sets are simply made up of columns of numbers, sometimes even with recording mistakes. An exploration and pre-processing of the data sets are often necessary to ensure reliability before directly feeding the data into the model.

When the system works under multiple operating conditions, the degradation mechanism is often different for different operating conditions. While in many cases, the information on the operating conditions is not recorded in the dataset directly, such as the CMAPSS dataset that is used in this paper. A classical partitional clustering algorithm is adopted to classify the operating condition from the original data sets naming k-medoids, which divides the dataset of n objects into k clusters. In the k-medoids algorithm, the most representative objects in the clusters are picked as medoids to represent the clusters. A medoid is defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal (Madhulatha, 2011). Each remaining object is clustered with the

medoid to which it is the most similar. The following equation calculates the total absolute error for the clustering configuration:

$$E(M, X) = \sum_{i=1}^n \min_{j=1}^k |M_j - X_i| \quad (4-1)$$

where M is the set of medoids, X is the data set, n is the number of objects, k is the number of clusters, M_j is the j^{th} medoid, X_i is the i^{th} object in data set X . The function $\min$ is used to find the medoid that is the closest to the given object (Chitrakar and Chuanhe, 2012).

4.2.2 Three-stage Feature selection method

Feature selection or feature reduction is the process of reducing the number of input variables to reduce the computational cost of modelling and improve the model's prediction performance. It primarily focuses on removing non-informative or redundant variables from the data that do not contribute to the predictive model performance or even cause overfitting problems. In this paper, we present a three-stage feature selection method.

In stage 1, we proposed an operation-based normalisation approach. This method is used to cope with datasets with multiple operating conditions. By utilising the unsupervised k-medoids clustering based on the selected input variables, which contains the operating condition information, all raw input variables X_i ($X_1, X_2, \dots, X_n$) in terms of different operating conditions OC ($OC_1, OC_2, \dots, OC_N$) can be written as

$$X_i = \bigcup_{j=1}^N OC_j(X_k) \quad (4-2)$$

where n is the number of input variables and N the number of operation conditions, i is the index of the raw input, and k is the index of the input in the new series OC_j . Then the commonly used normalisation method (Standardisation, Min-Max Normalisation or Robust standardisation) is applied to each operation condition of each input variable. And then each input variable $\hat{X}_i$ is reconstructed by connecting with all normalised segments with the same time index shown in Eq. (4-3), written as

$$\hat{X}_l = \bigcup_{j=1}^N \text{Norm}|OC_j(X_k)| \quad (4-3)$$

where $\text{Norm}|\cdot|$ denotes the normalisation operation which can be any of the basic normalisation methods that is normally used in Machine learning area (see Appendix). Based on the data exploration of the input variable $\hat{X}_l$ (including data description and visualisation), a preliminary decision of feature selection can be made.

In stage 2, we propose to use pair-wise Spearman's rank coefficient between each input and each output to remove the irrelevant input variables. The limitation of Pearson correlation is that it cannot be adopted to tell the nonlinear relationship between a dependent variable and an independent variable. Spearman rank correlation does not have any assumptions about the data distribution; Therefore, it is the appropriate correlation analysis method when the variables are measured on a scale that is at least ordinal. The formula used to calculate the Spearman rank correlation is written as

$$p = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (4-4)$$

where p is the Spearman rank correlation, d_i is the difference between the ranks of corresponding variables (x and y) and n is the number of observations. The rank is calculated as the average value when there are ties in input or output. The scores range between -1 and 1, where -1 refers to perfectly negatively correlated, and 1 refers to perfectly positively correlated.

In stage 3, since the RUL prediction of a system is a regression predictive modelling problem with numerical input variables and output variables, we propose to inspect the pair-wise Pearson's correlation coefficients among all input variables to remove the redundant input variables. Pearson's correlation coefficient measures the strength of the linear relationship between two data samples, either time-series data or non-time-series data. It is calculated as the covariance of two variables (x and y) divided by the product of the standard deviation of each variable, written as:

$$r_{xy} = \frac{S_{xy}}{S_x S_y} \quad (4-5)$$

where r_{xy} is the correlation coefficient, S_{xy} is the covariance, and S_x, S_y are the standard deviations. r_{xy} ranges from -1 to +1. A correlation coefficient of 1 indicates that for every

positive increase in one variable, there is a positive increase of fixed proportion in another variable. A correlation coefficient of -1 means that for every positive increase in one variable, there is a negative decrease of fixed proportion in another variable. A correlation coefficient of 0 means that these two variables are unrelated. The absolute value of the correlation coefficient refers to the relationship strength between the two variables.

4.2.3 Construction of the predictive model

After the three-stage feature selection, the run-to-failure training data is used to train the DL-based network. Then, the prediction of the RUL of the test engines is achieved by feeding the pre-processed testing data into the trained predictive model. Since RNN is good at dealing with time series data, various RNN-type network structures are tested to validate the practical value of the proposed feature section approach.

4.2.3.1 Long Short-Term Memory

Long Short-Term Memory network is a modified structure of the recurrent cell that incorporates the standard recurrent layer along with additional “memory” control gates. The original LSTM was developed by Hochreiter and Schmidhuber (1997) when researchers discovered a vanishing and exploding gradient issue in traditional RNNs. LSTM uses storage elements to transfer information from the past output instead of having the output of the RNN cell be a nonlinear function of the weighted sum of the current input and previous output. In other words, instead of using a hidden state h only, LSTM adopts a cell state C to keep the long-term information as shown in Figure 4-2. The central concept of LSTM is utilising three gates, such that each is a nonlinear function σ activated by a weighted sum of the current input observation x_t and previous hidden state h_{t-1} , to control the cell state C (forget gate f_t , input gate i_t and output gate o_t). The forget gate f_t is used to manage the information from the previous cell state C_{t-1} to the current cell state C_t ; the input gate i_t decides how many inputs should be kept in the current cell state C_t ; and the output gate o_t determines the output h_t from the current cell state C_t .

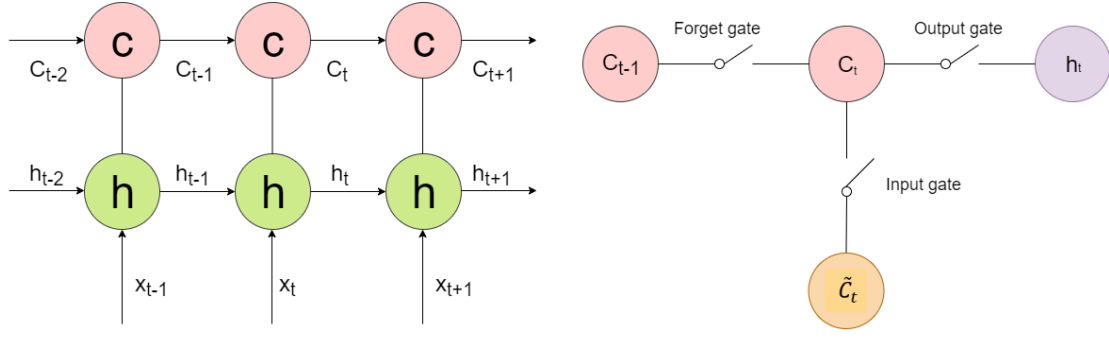


Figure 4-2 Cell state (Left) and three gates (Right) in LSTM

The output of LSTM at step t is calculated using the following equations:

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \quad (4-6)$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \quad (4-7)$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \quad (4-8)$$

$$\tilde{c}_t = \tanh(U_{\tilde{c}} x_t + W_{\tilde{c}} h_{t-1} + b_{\tilde{c}}) \quad (4-9)$$

$$C_t = C_{t-1} \cdot f_t + \tilde{c}_t \cdot i_t \quad (4-10)$$

$$h_t = \tanh(c_t) \cdot o_t \quad (4-11)$$

where U , W and b are the trainable weights and biases. These three gates have the same shape with different parameters U and W , which need to be learned from the training process. The candidate state $\tilde{c}_t$ cannot be used directly. It must pass through the input gate and then be used to calculate the internal storage C_t . While C_t is not only affected by the hidden state but also by C_{t-1} that is controlled by the forget gate. Based on C_t , a layer of $\tanh$ function is applied to the output information h_t , which is constrained by the output gate. The gates enables LSTM to fulfil the long-term dependencies in the sequence, and by learning the gate parameters, the network can find the appropriate internal storage. Therefore, LSTMs are naturally suited for the RUL prediction task using sensor data with the inherent sequential nature due to their capability of remembering information over long periods.

4.2.3.2 Parameter selection

The optimiser and activation functions are two integral parts of constructing the predictive model. The choice of optimiser and activation function will largely affect the model's prediction performance. A preceding work introduced some popular optimisers (RMSprop, Adam, AdaGrad and AdaDelta) and activation functions (ReLU, ELU,

Leaky_ReLU, PReLU, and Swish) and systematically analysed their performance in RUL prediction (Wang *et al.*, 2021).

4.2.4 Benchmark dataset

NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset is a high-fidelity computer model simulating run-to-failure trajectories of a small fleet of turbofan engines with the main elements (Saxena *et al.*, 2008b). A more in-depth introduction of the dataset can be found in the C-MAPSS User's Guide (Liu *et al.*, 2012). The dataset consists of 4 subsets, each containing different numbers of engines with varied operational cycles. The dataset contains one training set and one testing set for each engine both of which contain a multivariate time series of 26 features. In this dataset, engine profiles were simulated with different initial degradation conditions. The maintenance was not taken into consideration during the simulation. The information of the four subsets is listed in Table 4-1.. In this case study, FD002 was primarily used to validate the proposed approach.

Table 4-1 C-MAPSS dataset

Dataset	FD001	FD002	FD003	FD004
Number of engines for training	100	260	100	249
Number of engines for testing	100	259	100	248
Number of operating conditions	1	6	1	6
Number of Fault conditions	1	1	2	2

4.2.5 Performance evaluation

In this case study, the Root Mean square error (RMSE) was used to evaluate the performance of the trained neural networks. The mathematical expression is:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (4-12)$$

where n is the total number of actual RUL targets in the related test set and e_i refers to the difference between the true RUL and the predicted RUL.

4.3 Results

4.3.1 Data exploration

The employed data set FD002 (training set) is composed of information relating to a total of 260 different turbines with a total number of observations that varies from 128 to 378. The mean and standard deviation of the response and the predictor variables are summarised in Table 4-2. Since this dataset has six operating conditions ($k=6$), the three columns of setting data are fed to a k-medoids clustering model with 6 clusters. To make sure the labels of the training set and testing set are the same for the same operation condition, the setting data in both the training dataset and the testing dataset is used. The distribution of the six operating conditions in FD002 is demonstrated in Figure 4-3(a) with the setting values shown in Figure 4-3(b).

Table 4-2 Mean and standard deviation of all the variables of dataset FD002 (training set)

Index	Symbol	Input variables	Mean	Standard deviation
1	ID	Engine ID		
2	T	Time cycle	109.15	69.18
3	set_1	Operation condition 1	24	14.74
4	set_2	Operation condition 2	0.57	0.31
5	set_3	Operation condition 3	94.05	14.24
6	sensor_1	Total temperature at fan inlet (°R)	472.91	26.39
7	sensor_2	Total temperature at LPC outlet (°R)	579.67	37.29
8	sensor_3	Total temperature at HPC outlet (°R)	1419.97	105.94
9	sensor_4	Total temperature at LPT outlet (°R)	1205.44	119.12
10	sensor_5	Pressure at fan inlet (psia)	8.03	3.61
11	sensor_6	Total pressure in bypass-duct (psia)	11.6	5.43
12	sensor_7	Total pressure at HPC outlet (psia)	282.61	146
13	sensor_8	Physical fan speed (rpm)	2228.88	145.21
14	sensor_9	Physical core speed (rpm)	8525.2	335.81

15	sensor_10	Engine pressure ratio (P50/P2)	1.09	0.13
16	sensor_11	Static pressure at HPC outlet (psia)	42.99	3.23
17	sensor_12	Ratio of fuel flow to Ps30 (pps/psi)	266.07	137.66
18	sensor_13	Corrected fan speed (rpm)	2334.56	128.07
19	sensor_14	Corrected core speed (rpm)	8066.6	84.84
20	sensor_15	Bypass ratio	9.33	0.75
21	sensor_16	Burner fuel–air ratio	0.02	0.005
22	sensor_17	Bleed enthalpy	348.31	27.75
23	sensor_18	Demanded fan speed (rpm)	2228.8	145.33
24	sensor_19	Demanded corrected fan speed (rpm)	97.76	5.36
25	sensor_20	HPT coolant bleed (lbm/s)	20.79	9.87
26	sensor_21	LPT coolant bleed (lbm/s)	12.47	5.92

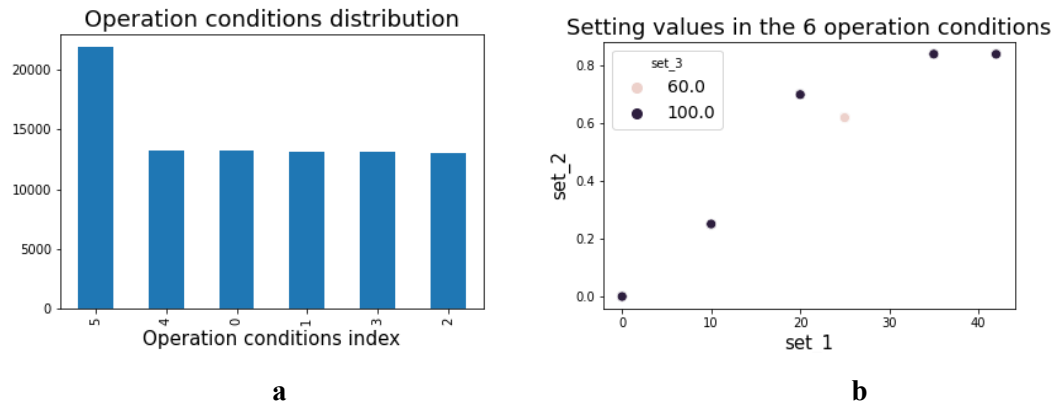


Figure 4-3 Result of k-medoids clustering model on FD002 (a: operation condition distribution; b: Setting values in the 6 operation conditions)

4.3.2 Parameter optimisation

This section reports the results of the RUL prediction which inform the selection of the normalisation method, RNN models, and model parameters, where the operation-based data scaling method and feature selection are not considered. It should be noted, to better evaluate the uncertainty of prediction, each model was repeated 20 times and the statistical results are presented below.

Normalisation methods: Before feeding the raw data into the RNN-based model for training or testing, the data need to be normalised. If not, gradient explosion or gradient

vanishing problems will occur which leads to invalid results. We applied three classic normalisation methods for FD0002 including Min-Max Normalisation, Robust standardization, and Standardisation. For the RNN model, we used two- layer LSTM with the optimiser of *Adam*, the activation function of *ReLU*, the sequence length of 50, and the neuron number of 128. The selection of the parameters is based on the previse work which can be found in (Wang *et al.*, 2021). The RUL prediciton results using these three normalization methods can be found in Table 4-3 (mean $\pm$ standard deviation over 20 tests), which suggests the robustness of the model prediction is largely improved after the normalisation. Generally, Robust standardisation scaling outperforms both Min-Max normalisation scaling and Standardisation scaling. Therefore, Robust standardization is selected for further analysis.

Table 4-3 RUL prediction performance (RMSE) without and with normalisation methods

Dataset	Normalisation methods			
	Raw data	Min-Max Normalisation	Robust standardisation	Standardisation
FD002	Fail	37.92 $\pm$ 12.61	32.17 $\pm$ 10.51	37.06 $\pm$ 11.67

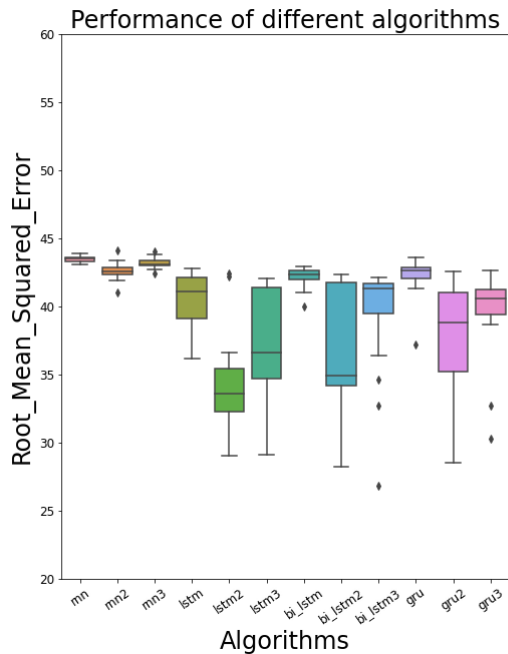


Figure 4-4 RUL prediction performance of FD002 using different RNN-based algorithms

RNN model selection: Figure 4-4 displays the RUL prediction performance for FD002 using 12 algorithms constructed by four different model types and three layers. The

optimiser of *Adam*, the activation function of *ReLU*, the sequence length of 50, and the neuron number of 128 were used. The data was transformed using Robust standardisation. Inspection of Figure 4-4 indicates that, for this dataset, LSTM, Bi_LSTM and GRU are all outperformed the original RNN models in terms of precision. Considering the balance of prediction accuracy, precision, and model complexity, a two-layer LSTM is selected for further optimisation using feature engineering and operation-based data scaling.

RNN model parameters selection: To select the optimiser, the activation function of *ReLU*, the neuron number of 128, and the sequence length of 50 were used to test four different optimisers, *RMSprop*, *Adam*, *AdaGrad* and *AdaDelta* are tested using different neural network structures. It has been observed that gradient explosion occurs while applying *AdaGrad* and *AdaDelta* for FD002, but this problem does not take place when applied to FD002 (Wang *et al.*, 2021). Therefore, Figure 4-5 only presents the prediction performance of *RMSprop* and *Adam*. It can be observed that *RMSprop* outperforms *Adam* for most models in terms of accuracy. Therefore, we select *RMSprop* for further analysis. To select the activation function, the same parameters and *Adam* were used to test the prediction performance of five activation functions including *ReLU*, *ELU*, *Leaky_ReLU*, *PReLU*, and *Swish*. Inspection of the results, shown in Figure 4-6, suggests that the performance is similar. *ReLU* is selected for further analysis due to its popularity.

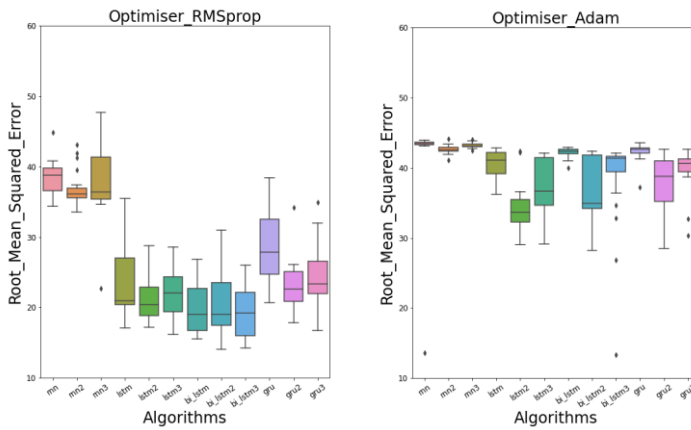


Figure 4-5 RUL prediction performance using different optimisers for FD002

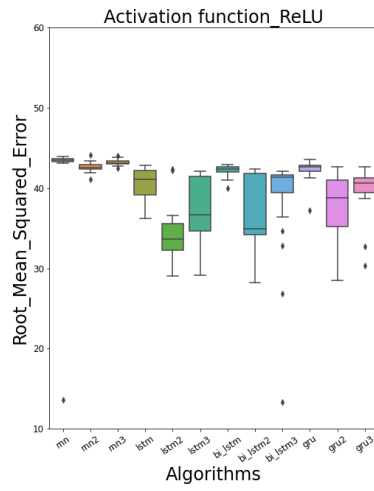
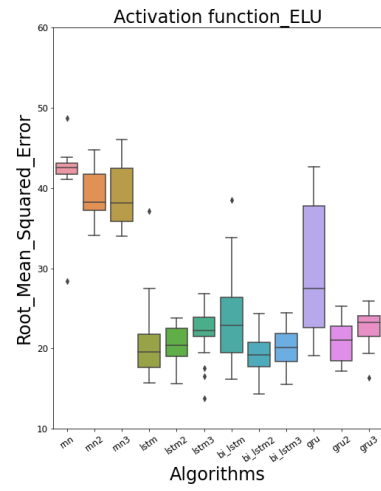
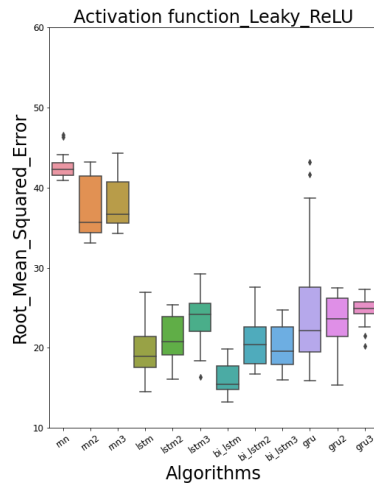
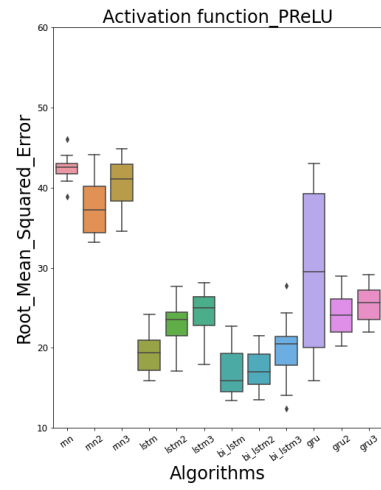
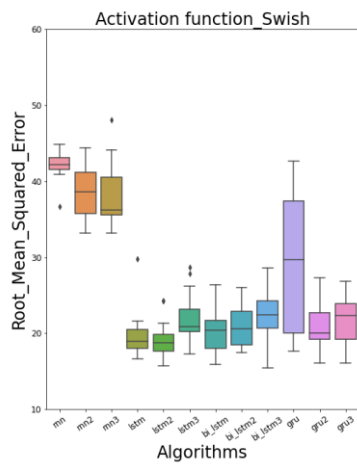


Figure 4-6 RUL prediction performance using different activation functions for FD002

4.3.3 Performance of the proposed three-stage feature selection method

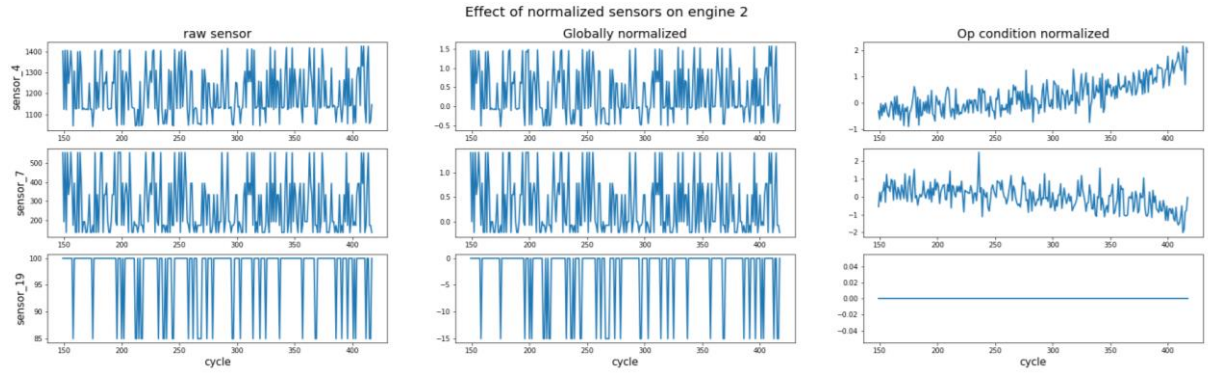


Figure 4-7 The input signals after applying the operation-based normalisation

Figure 4-7 gives a detailed comparison of 3 selected sensors between the data without normalisation, with global normalization (data are normalized regardless of the operating condition), and operation-based normalisation. It can be observed that the signal after the global normalisation remains the same pattern as the raw signal, and the only change is the scale. However, after the operation-based normalisation, the data reveals the trends hidden in the raw signals. For example, there is an apparent increasing trend in sensor_4 and a decreasing trend in sensor_7, particularly after the data point of 200, which is unlikely observed in the raw data or the data after the global normalisation. Such a trend could play an important role in improving the performance of RUL prediction. Furthermore, there is a variation in the raw data and the data after the global normalisation for sensor_19, but this variation disappears after the operation-based normalisation. The reason is that sensor_19 is only influenced by the operating condition. Such a sensor should be removed before proceeding with the RUL prediction.

Figure 4-8 demonstrates the input signals of engine 2 in FD002 after operation-based normalisation, where it can be observed that signals of set_3, sensor_1, sensor_5, sensor_18, and sensor_19 are constant. Therefore, these five features are dropped first. Then, Spearman's rank coefficient between each input and the RUL was calculated to determine the relevant variables, as shown in Figure 9(a). Set 1, set 2, Sensor_8 and Sensor_13 are removed since Spearman's rank coefficients with RUL are much smaller than the other sensors. After that, the pair-wise Pearson correlation coefficients among the input channels are calculated, and the operation-based normalisation results are shown in Figure 10(b). Sensor_9 and sensor_14 have a high correlation coefficient of 0.96. Therefore, Sensor_9 were removed according to this observation.

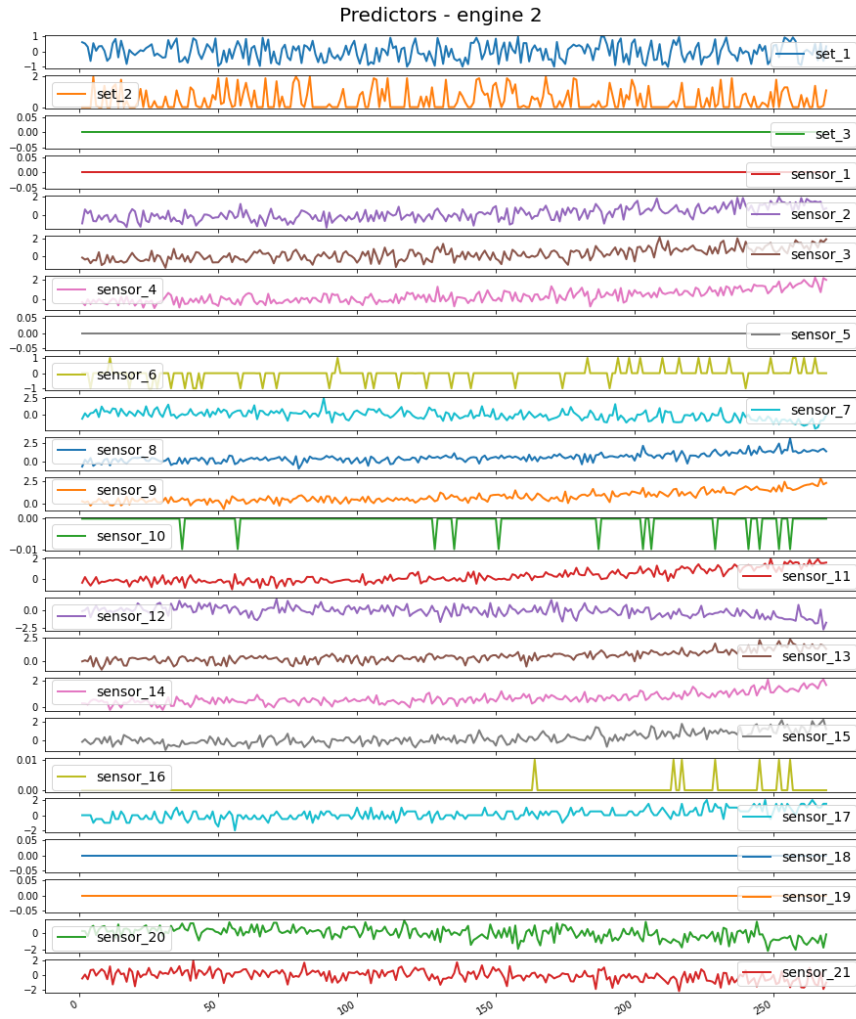
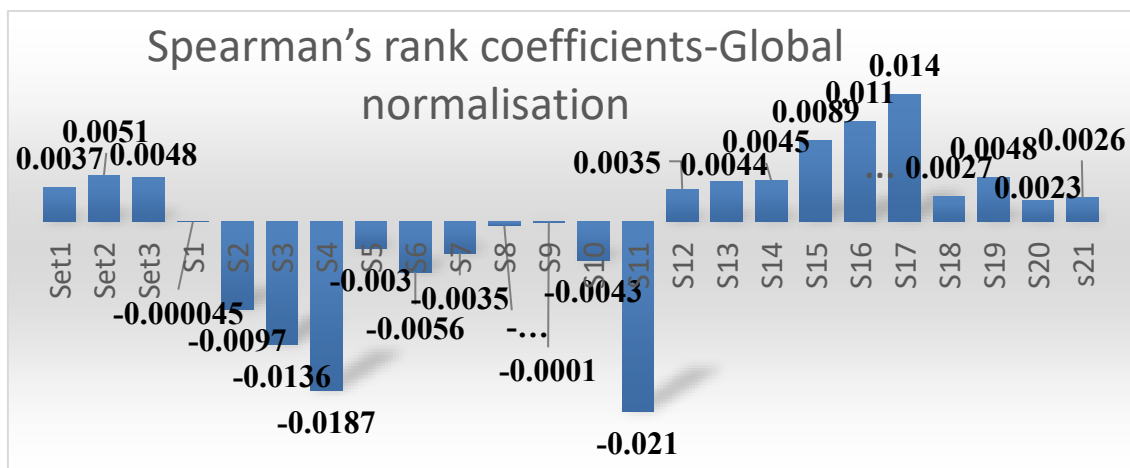
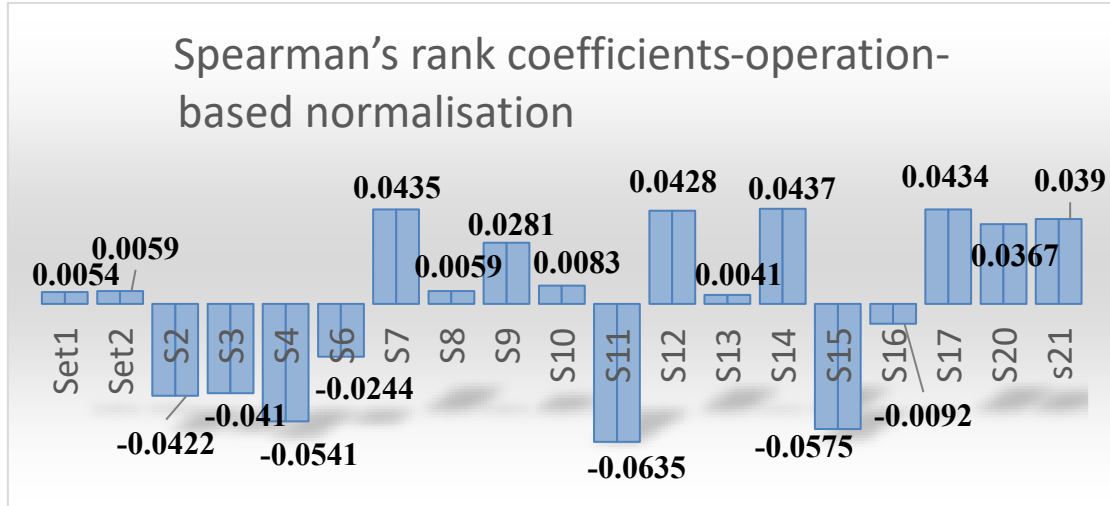


Figure 4-8 Comparison of 3 selected input variables without normalisation, with the global normalisation, and with the operation-based normalisation approach



(a)

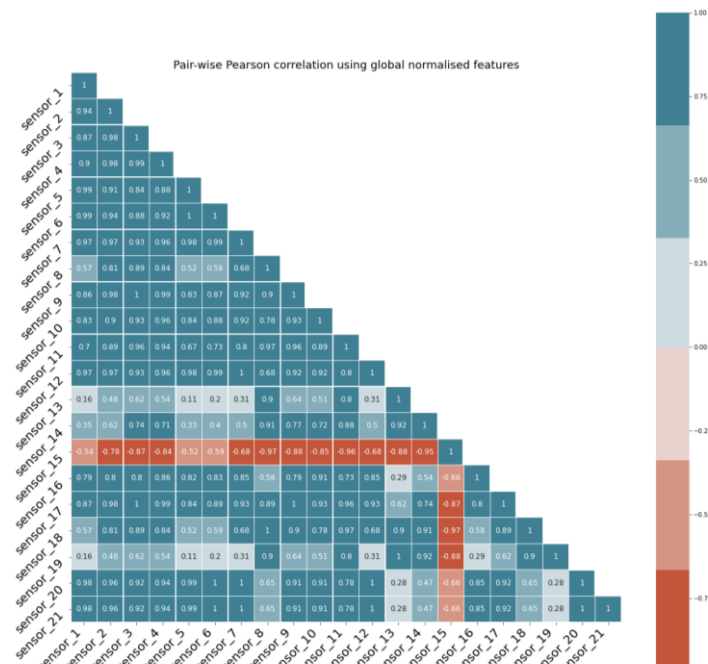


(b)

Figure 4-9 Spearman's rank coefficients between each input and the output. (a): based on the global normalisation, (b): based on the operation-based normalisation

The results of Spearman's rank coefficients between the sensors and the output RUL using two different normalisation methods are displayed in Figure 9. It is shown that the operation-based method (Figure 9(b)) produces higher correlations than the global normalisation method (Figure 9(a)). The comparison suggests that the proposed operation-based method produces the input variables, which have a better chance of modelling the output (RUL). Furthermore, revealing irrelevant features in Figure 9(b) is more straightforward.

(a)



(b)

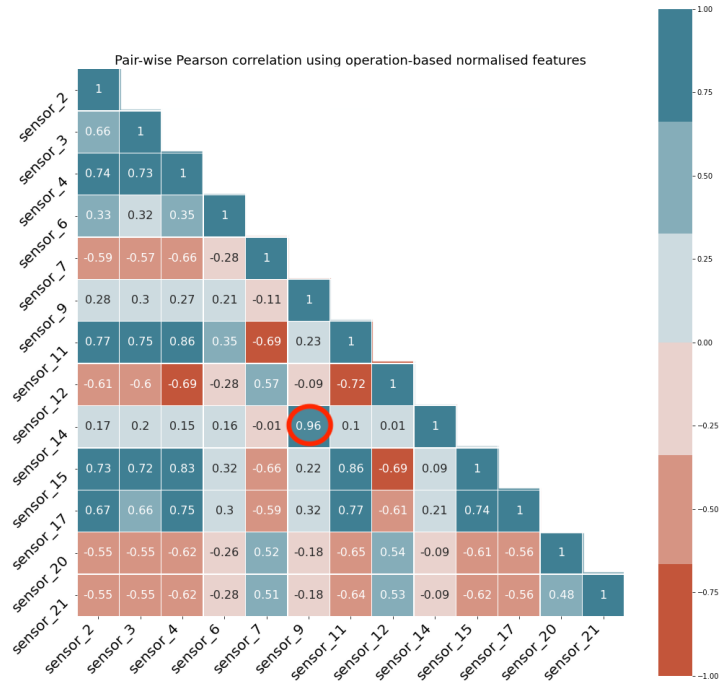


Figure 4-10 Pair-wise Pearson correlation coefficient for the input channels using (a) the global normalisation and (b) the operation-based normalisation

It can be observed that different normalisation methods have a significant impact on the correlation coefficients among sensors and lead to a different strategy of feature selection. It has been observed that the pair-wise Pearson correlation coefficient of many features are almost the same after the global normalisation, as shown in Figure 4-10(a). However, when the operation-based normalisation approach is used, the Pearson correlation coefficients among the input channels are smaller in most cases as shown in Figure 4-10(b). This is because the measurements recorded are significantly affected by the operating conditions, which makes it much more difficult to find out the real relationship between two variables. However, when the operating normalisation is applied, the actual degradation trend of the sensors is revealed.

4.3.4 Performance of RUL prediction

The selected features are then fed to a two-layer LSTM model with the fixed parameters, selected from the above results. It should be noted there are 24 engines whose number of cycles in the test dataset is less than the sequence length of 50 used in the model. Therefore, these engines were removed from the testing dataset due to insufficient

sampling data. The degradation of the engines is generally classified into two stages: the normal performance stage shows relative flat RUL values at the initial cycles, and the performance degradation stage shows an exponential drop trend of the RUL values at the later cycles. In this study, the RUL is assumed to be constant in the normal stage until it crosses a specific cut-off value. In the paper, the cut-off value is set to 120. The results shown below are only for the data points whose RUL values are less than 120.

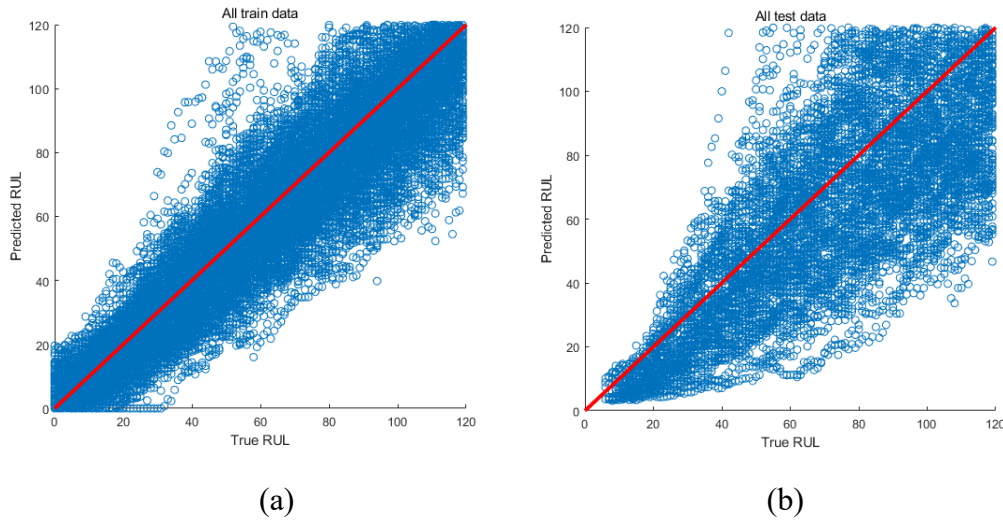


Figure 4-11 The predicted RUL vs the true RUL of each cycle and each engine for the (a) training and (b) testing dataset

The results of the predicted RUL using the proposed method on the training and testing dataset of FD002 are shown in , where the left plot illustrates the training result for every cycle and each engine. The right plot illustrates the testing results. It is interesting to observe that the prediction error for different cycles is similar to the training dataset while the error is reduced following the decrease of RUL for the testing dataset.

To detail the prediction performance of each engine, Figure 4-12 demonstrates the RUL prediction result for engines with lifecycles no less than 100. It can be observed that in general, the RUL prediction becomes increasingly accurate following the increase in cycle. This is the reason why most related studies used the RUL value of the last cycle to evaluate the model performance. It has also been observed that the prediction performance compromises if the available testing data number is smaller. This is probably because the system has not wholly entered the performance degradation stage.

To better show the prediction performance, Figure 4-13 plots the uncertainty quantification of the RUL prediction of 114 engines with cycles length larger than 100. It can be observed that the solid red curve, the prediction average overall aligns with the

ground truth, illustrated by the blue dot line. For most of the cycles, the predicted value is larger than the ground truth, particularly for the initial cycles (< 60 cycles). The prediction has a superior performance for the later cycles (> 60 cycles). The grey area, indicating the 95% confidence interval, suggests that the uncertainty of the prediction decreases following the increment of the lifecycle.

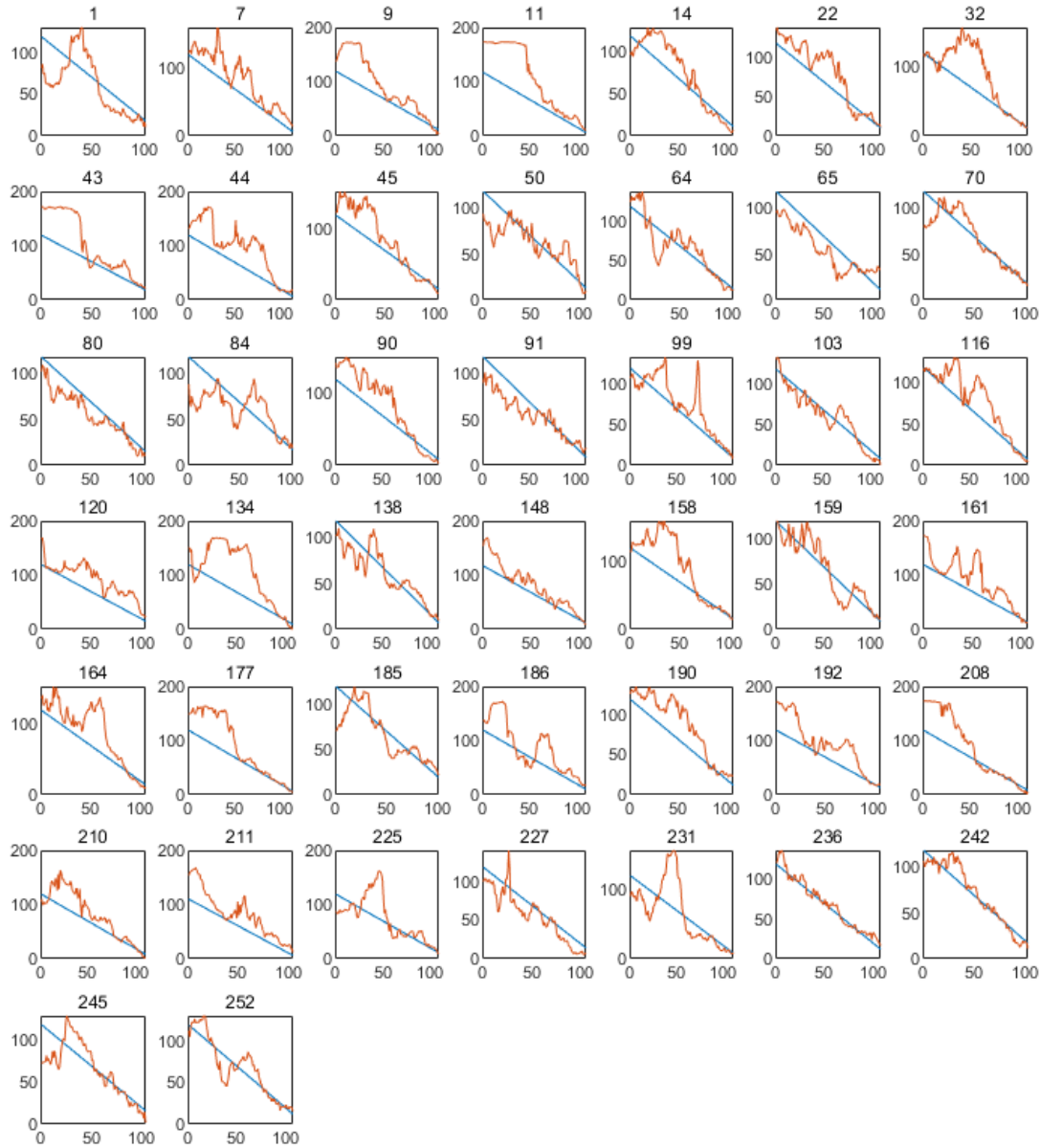


Figure 4-12 Prediction results for all engines with lifecycles no less than 100.

It should be noted that the proposed framework can accommodate all other deep learning methods. To demonstrate this, Table 4-4 shows the RMSE values of the RUL prediction of the final cycle for all engines with the global normalisation and the proposed operation-

based normalisation, where 12 popular models are tested. An improvement of 6.84% has been achieved for all models, demonstrating the proposed solution's contribution.

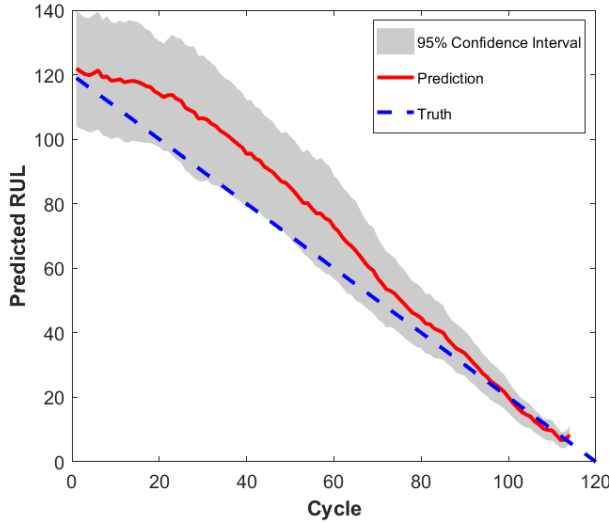


Figure 4-13 Uncertainty quantification of the RUL prediction using the proposed method by combining all 1114 engines

Table 4-4 RMSE values of the RUL prediction of the final cycle for all engines with the global normalisation and the proposed operation-based normalisation

Model	Global Normalisation (GN)	Operation-based normalisation (ON)	Improvement $\left(\frac{GN-ON}{GN}\right)$
RNN-1LAYER	35.88	31.40	12.4%
RNN-2LAYERS	32.12	30.80	4.0%
RNN-3LAYERS	33.47	30.65	8.4%
LSTM-1LAYER	30.78	30.48	1.0%
LSTM-2LAYERS	33.01	30.32	8.1%
LSTM-3LAYERS	34.26	31.23	8.8%
Bi_LSTM-1LAYER	31.66	30.66	3%
Bi_LSTM-2LAYERS	32.59	31.01	4.8%
Bi_LSTM-3LAYERS	35.62	32.86	7.5%
GRU1-LAYER	32.41	31.48	2.9%
GRU2-LAYERS	34.74	30.76	11.4%
GRU3-LAYERS	35.46	32.00	9.8%

To further extend the application of the proposed method, Table 4-5 shows the RMSE values of the RUL prediction using state-of-the-art methods without and with the operation-based normalisation. The superior performance of this approach is demonstrated again with an average improvement of 7.3%.

Table 4-5 Improvement of the proposed method for state-of-the-art methods

Model	Global Normalisation (GN)	Operation-based normalisation (ON)	Improvement $(\frac{GN-ON}{GN})$
Baseline LR	34.10	32.30	5.3%
Baseline Ridge	34.10	32.10	5.9%
XGBoost	29.70	28.50	4%
Attention-based LSTM (Ayodeji <i>et al.</i> , 2021)	24.50	22.40	8.6%
Attention-based GRU (Ayodeji <i>et al.</i> , 2021)	28.50	24.90	12.6%

4.4 Conclusion

In this study, a novel three-stage feature engineering approach enhanced by an optimised DL-based model was presented to improve the RUL prediction performance of a system under multiple operating conditions. The activation function and optimiser optimisation of the DL-based model was also introduced to improve the prediction results. A k-medoids clustering model was used to categorise the raw data into 6 clusters based on the operation conditions. Then, the operation-based normalisation approach is applied to find the true degradation trend hidden in the raw signal. The preliminary feature selection decision is made by plotting the operation-based normalised signals. After that, two correlation analysis methods are used to eliminate the irrelevant and redundant input variables to further reduce the computational cost and avoid the over-fitting problem. The result demonstrates that the operation-based normalisation improves the correlation between the input variable and the output (RUL), which leads to higher prediction accuracy. Furthermore, it greatly enhances the robustness of the model by reducing the gradient explosion or gradient vanishing problems. The results demonstrate that the proposed framework can consistently improve the performance of popular RNN variants and state-of-the-art for RUL prediction. This DL-based prognostics framework provides a reliable RUL prediction strategy for a complex system that works under multiple operating conditions. However, it should be noted that the optimisation is based on the testing of every feature engineering method and prognostic parameter, which could be very time consuming and can not necessarily present the most accurate RUL prediction

performance. Therefore, future work will focus on designing new implementation algorithms to speed up the training progress and achieve better prediction results.

References

- Ahmed, Elsheikh, Soumaya, Yacout, Mohamed-Salah, Ouali, Elsheikh, A., Yacout, S., Ouali, M.S., 2018. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* 323, 148–156.
<https://doi.org/10.1016/j.neucom.2018.09.076>
- An, D., Kim, N.H., Choi, J.H., 2015. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliab Eng Syst Saf* 133, 223–236. <https://doi.org/10.1016/j.ress.2014.09.014>
- Ayodeji, A., Wang, W., Su, J., Yuan, J., Liu, X., 2021. An empirical evaluation of attention-based multi-head models for improved turbofan engine remaining useful life prediction.
- Chen, J., Jing, H., Chang, Y., Liu, Q., 2019. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliab Eng Syst Saf* 185, 372–382. <https://doi.org/10.1016/j.ress.2019.01.006>
- Chitrakar, R., Chuanhe, H., 2012. Anomaly detection using Support Vector Machine classification with k-Medoids clustering, in: Asian Himalayas International Conference on Internet. <https://doi.org/10.1109/AHICI.2012.6408446>
- Cubillo, A., Perinpanayagam, S., Esperon-Miguez, M., 2016. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*.
<https://doi.org/10.1177/1687814016664660>
- Gan, M., Wang, C., Zhu, C., 2016. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mech Syst Signal Process* 72–73, 92–104.
<https://doi.org/10.1016/j.ymssp.2015.11.014>
- Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Comput* 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Javed, K., Gouriveau, R., Zerhouni, N., 2017. State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mech Syst Signal Process* 94, 214–236. <https://doi.org/10.1016/j.ymssp.2017.01.050>

- Kundu, P., Darpe, A.K., Kulkarni, M.S., 2019. Weibull accelerated failure time regression model for remaining useful life prediction of bearing working under multiple operating conditions. *Mech Syst Signal Process* 134, 106302. <https://doi.org/10.1016/j.ymssp.2019.106302>
- Lei, Y., Li, Naipeng, Guo, L., Li, Ningbo, Yan, T., Lin, J., 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech Syst Signal Process* 104, 799–834. <https://doi.org/10.1016/j.ymssp.2017.11.016>
- Li, Q., Gao, Z., Tang, D., Li, B., 2016. Remaining useful life estimation for deteriorating systems with time-varying operational conditions and condition-specific failure zones. *Chinese Journal of Aeronautics* 29, 662–674. <https://doi.org/10.1016/j.cja.2016.04.007>
- Li, X., Zhang, W., Ding, Q., 2019. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab Eng Syst Saf* 182, 208–218. <https://doi.org/10.1016/j.ress.2018.11.011>
- Liu, Y., Frederick, D.K., Decastro, J.A., Litt, J.S., Chan, W.W., 2012. User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). Nasa/Tm 2012–21743, 1–40.
- Lu, C., Wang, Z.Y., Qin, W.L., Ma, J., 2017. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing* 130, 377–388. <https://doi.org/10.1016/j.sigpro.2016.07.028>
- Ma, J., Su, H., Zhao, W., Liu, B., 2018. Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning. *Complexity* 2018, 1–13. <https://doi.org/10.1155/2018/3813029>
- Madhulatha, T.S., 2011. Comparison between K-Means and K-Medoids Clustering Algorithms, CCIS.
- Salunkhe, T., Jamadar, N.I., Kivade, S.B., 2014. Prediction of Remaining Useful Life of Mechanical Components-A Review 3, 125–135.
- Saxena, A., Goebel, K., Simon, D., Eklund, N., 2008. Damage propagation modeling for aircraft engine run-to-failure simulation. 2008 International Conference on Prognostics and Health Management, PHM 2008. <https://doi.org/10.1109/PHM.2008.4711414>

- Tran, V.T., Thom Pham, H., Yang, B.S., Tien Nguyen, T., 2012. Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine. *Mech Syst Signal Process* 32, 320–330. <https://doi.org/10.1016/j.ymssp.2012.02.015>
- Vogl, G.W., Weiss, B.A., Helu, M., 2019. A review of diagnostic and prognostic capabilities and best practices for manufacturing. *J Intell Manuf* 30, 79–95. <https://doi.org/10.1007/s10845-016-1228-8>
- Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D., 2018. Deep learning for smart manufacturing: Methods and applications. *J Manuf Syst* 48, 144–156. <https://doi.org/10.1016/j.jmsy.2018.01.003>
- Wang, Y., Zhao, Y., Addepalli, S., 2021. Practical Options for Adopting Recurrent Neural Network and Its Variants on Remaining Useful Life Prediction. *Chinese Journal of Mechanical Engineering (English Edition)* 34. <https://doi.org/10.1186/s10033-021-00588-x>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2019. Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process* 115, 213–237. <https://doi.org/10.1016/j.ymssp.2018.05.050>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2015. Deep Learning and Its Applications to Machine Health Monitoring : A Survey 14, 1–14.
- Zheng, S., Ristovski, K., Farahat, A., Gupta, C., 2017. Long Short-Term Memory Network for Remaining Useful Life estimation. 2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017 88–95. <https://doi.org/10.1109/ICPHM.2017.7998311>
- Zhu, J., Chen, N., Shen, C., 2020. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. *Mech Syst Signal Process* 139, 106602. <https://doi.org/10.1016/j.ymssp.2019.106602>

Appendix: Normalisation approaches

1. Standardisation

Standardisation is a transform for data with a Gaussian distribution. It standardizes features by subtracting the mean and dividing the result by the standard deviation of the data sample. It can be regarded as subtracting the mean value or centring the data. The standard score z of the variable x is calculated as:

$$\hat{x} = (x - u)/s \quad (4-13)$$

where u and s are the mean and standard deviation of the variable, respectively. Standardising a dataset means rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1. Centring and scaling are applied to every feature by calculating the relevant statistics on the samples in the training dataset. The stored mean u and standard deviation s are then used on the test dataset.

2. Min-Max Normalisation

Min-Max Normalisation is a transform that rescales the data from the original range to a new fixed range of 0 and 1. It is based on the assumption that the data's minimum and maximum observable values are provided or can be accurately estimated. The transformation equation is given by:

$$\hat{x} = (x - \min)/(max - \min) \quad (4-14)$$

where the minimum and maximum values pertain to the value x being normalised. Normalisation is often used as an alternative to standardisation. However, normalisation may not be the best method to estimate these expected values if the time series trends up or down.

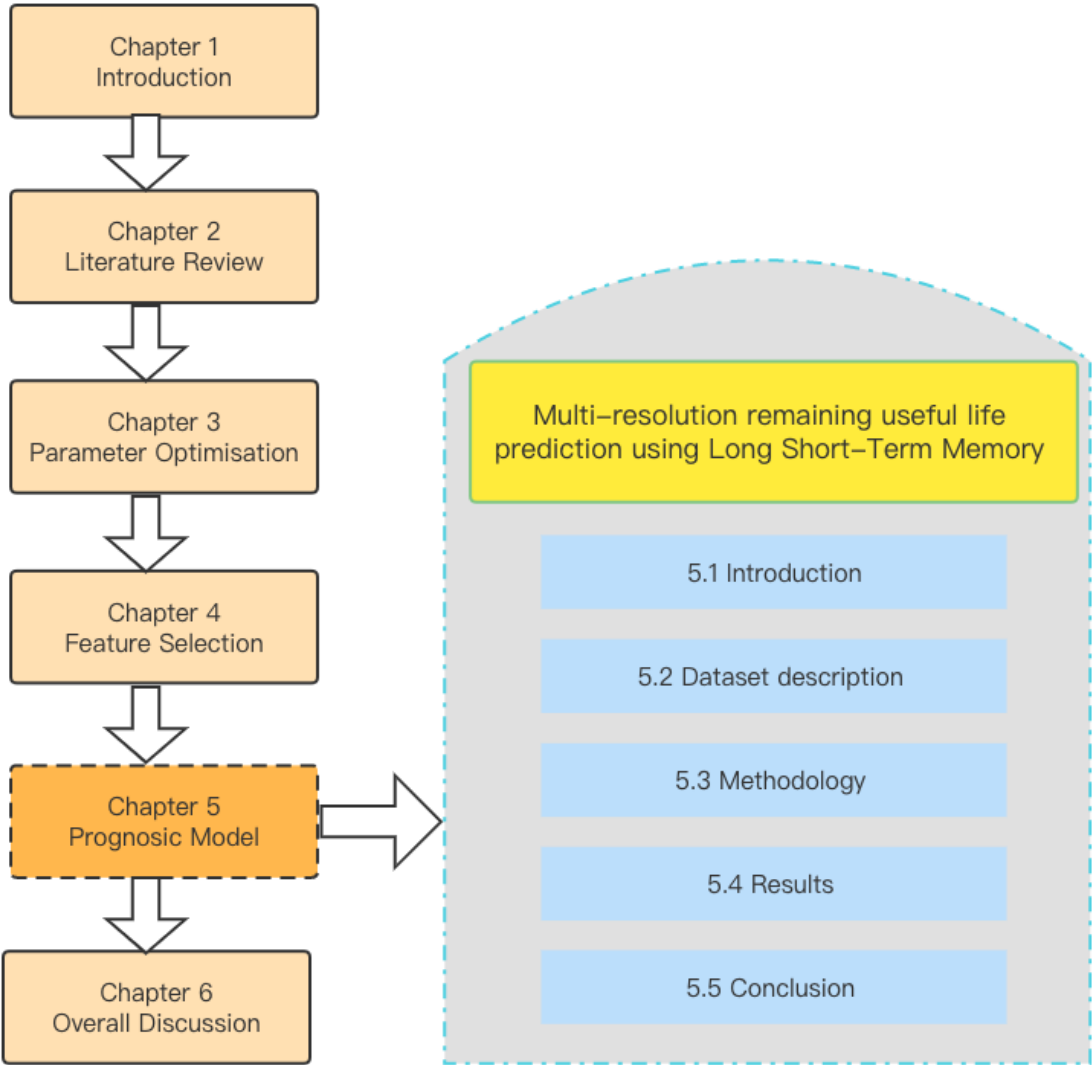
3. Robust standardisation

Robust standardisation is used to scale input variables in the presence of outliers. The probability distribution will be skewed if some outlier values exist in the input variable. Therefore, data standardisation will be very difficult because these outliers skew the calculated mean and standard deviation. Robust standardisation is presented by removing these outliers from the mean and standard deviation calculation. This is fulfilled by calculating the median and the 25 and 75 percentiles as shown below:

$$\hat{x} = (x - median)/(p75 - p25) \quad (4-15)$$

The scaled data has a zero mean and median, and a standard deviation of 1. Although outliers do not skew the scaled data, these outliers still have the same relative relationships to other values.

5 CHAPTER FIVE—PROGNOSIC MODEL



Multi-scale remaining useful life prediction using Long Short-Term Memory

Abstract

Predictive maintenance based on performance degradation is a crucial way to reduce the operation and maintenance costs and potential failures in modern complex engineering systems. Reliable remaining useful life (RUL) prediction is the main criterion for predictive maintenance decisions. Conventional model-based methods and data-driven approaches often fail to achieve an accurate prediction result using a single model for a complex system featuring multiple components and multiple states, as the degradation pattern is usually nonlinear and time-varying. This paper proposes a novel multi-scale RUL prediction approach adopting the Long Short-Term Memory neural network. In the feature engineering phase, Pearson's correlation coefficient is applied to extract the representative features; An operation-based data normalisation approach is presented to deal with the cases where multiple degradation patterns are concealed in the sensor data. Then, a three-stage RUL target function is proposed, which segments the degradation process of the system into the non-degradation stage, the transition stage, and the linear degradation stage. The classification of these stages is regarded as the low-scale RUL prediction, and it is achieved through processing sensor signals using a novel LSTM-based binary classification algorithm combined with a correlation method. After that, a specific LSTM-based predictive model is built for the last two stages to produce a high-scale RUL prediction. The proposed approach is validated by comparing it with many state-of-the-art techniques using the widely used C-MAPSS dataset. A significant improvement is achieved in RUL prediction performance in most subsets. For instance, a 40% reduction is achieved in Root Mean Square Error over the best existing method in subset FD001. Another contribution of the multi-scale RUL prediction approach is that it offers more degree of flexibility in the way maintenance strategy is organised.

Keywords

Classification, C-MAPSS, Feature engineering, RNN, RUL target function

5.1 Introduction

Predictive maintenance (PdM) becomes the most promising maintenance strategy in Industry 4.0 (Ayodeji *et al.*, 2021a) which can significantly reduce the maintenance cost and ensure the critical components' reliability and safety (Li *et al.*, 2019b). Remaining useful life (RUL) prediction of engineering components and systems is one of the most important tasks in PdM. RUL prediction generally refers to the study of predicting the specific time length from the current time to the end of the useful life of an asset or system (Salunkhe *et al.*, 2014). It is a critical step to minimise catastrophic failures and it has become an important measurement to achieve the ultimate goal of zero-downtime performance in industrial systems (Listou Ellefsen *et al.*, 2019).

Approaches for RUL prediction can be catalogued into model-based methods, data-driven methods, and hybrid methods. Model-based approaches, also called physics-based approaches, evaluate the health condition of a system by building mathematical models based on the failure mechanisms or the first principle of damage (Cubillo *et al.*, 2016c). The construction of a physical model is often impeded by the complex and noisy working conditions, which is usually difficult in the model. Meanwhile, it is often difficult to determine the parameters of the physical model. In data-driven approaches, RUL is computed through statistical and probabilistic methods by utilising historic information and routinely monitored data of the system (Javed *et al.*, 2017). With the presence of multivariate time sequence signals derived from parallel measurement of hundreds of process variables with diverse sensors, the application of many machine learning models for RUL prediction has been greatly promoted (Ayodeji *et al.*, 2021b). Hybrid approaches, combining model-based approaches and data-driven approaches, aim to leverage the advantages of both categories. Deep learning (DL), a subset of data-driven approaches that has been attracting significant investigations in the last few years in the RUL prediction, can be used to extract multilevel features from data (Ma *et al.*, 2018). As an end-to-end machine learning method, DL algorithms can automatically process the original signal, identify discriminative and trend features in the input data layer by layer, and improve generalisation performance (Bai *et al.*, 2021). Because of its strength in self-learning features, DL has already achieved great success in the manufacturing industry (Zhao *et al.*, 2019b).

An inherent challenge for predicting the RUL of a system is to determine the desired output value for each input data point. Without an accurate physics-based model, it is impossible to accurately identify the system health status at each time step in real-world applications (Navathe *et al.*, 2016). Generally, there are two solutions to this problem. One is to simply assign the desired output as the actual time left before functional failure, and another is to derive the desired output values based on a suitable degradation model. We would like to discuss this challenge based on NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset as one of the most popular multivariate benchmark datasets used in evaluating predictive DL algorithms. For C-MAPSS, a piece-wise linear degradation model has been proposed (Heimes, 2008). This model assumes that the degradation of the system typically only begins after a certain degree of usage. The threshold value is determined based on the observations and its numerical value is different for each dataset. Since then, this model was adopted by most of the related research.

It should be noted that the threshold value achieved in this paper is 130 based on the first subset (FD001), however, the application of this threshold value has been extended to other subsets (FD00x) in plenty of later research without sufficient justification. Yuan *et al.* (2018) proposed a novel dynamic differential technology that extracts inter-frame information, promoting cognition about the model degradation process. A support vector machine (SVM) is used as an anomaly detector to identify the point where the system starts degrading. An RUL prediction approach is based on degradation pattern learning using a back-propagation neural network (Zhao *et al.*, 2017). A piece-wise linear distribution function promotes a proportion of the tail samples with a threshold of 120. Ahmed *et al.* (2018) designed a new LSTM architecture that uses the sensor readings to estimate the health of the system, which is then mapped to the target RUL. It is claimed that the proposed method requires no assumptions about the degradation function patterns, nor the point at which the degradation starts. A Denoising Online Sequential Extreme Learning Machine with Double Dynamic Forgetting Factors (DDFF) and Updated Selection Strategy (USS) is proposed by Berghout *et al.* (2020). The Online Sequential Extreme Learning Machine (OS-ELM) is used to fit the non-accumulative linear degradation function of the engine to address dynamic programming by tracking the new coming data and forgetting gradually the old ones based on DDFF. The piece-wise linear

function is used as the RUL target function and the threshold value used is 130. Chen *et al.* (2019b) presented a Gated Recurrent Unit (GRU) based neural network that has also been used to extract and learn the nonlinear degradation patterns in the dataset, where the linear function is used as the RUL target function. An unsupervised pre-trained model for turbofan RUL prediction is proposed by Listou Ellefsen *et al.* (2019). This approach utilised an unsupervised and semi-supervised method to train a model to extract the features and then use the Genetic Algorithm (GA) to determine the threshold value. Shi and Chehade (2019) proposed a dual LSTM with changing detection capability. A health index construction function is utilised to help to detect change points in sensor measurements, and therefore determine the threshold value. A causal augmented convolution network (CaConvNet) has been presented by Ayodeji *et al.* (2021b). The network is further optimized with a dynamic hyperparameter search algorithm to reduce uncertainties and the cost of manual hyperparameter selection. A piece-wise linear function and the threshold value of 130 are adopted. Shi and Chehade (2021) proposed a dual-LSTM framework that combines change point detection and RUL prediction. They also adopted the piecewise RUL target function but assumed the threshold value to be the mid-point of the degradation process for every engine. Li *et al.* (2022) proposed a GRU-based high-level feature fusion block to replace the traditional fully connected layer and introduced a novel activation function Mish to perform the aero-engine RUL prediction. They also used the piece-wise linear function, with a fixed threshold of 120 for all four subsets. Zhang *et al.* (2022) used the piece-wise linear function too, but with a novel approach to identify the threshold. The thresholds for all en-gines are different and determined using a health status assessment which is evaluated by a bidirectional gated recurrent unit (BiGRU) and multi-gate mixture-of-experts (MMoE).

It should be noted that most of the aforementioned research adopted a nonlinear RUL target function except (Chen *et al.*, 2019b) in which the degradation process of the system is assumed to be linear with usage. The widely used Piece-wise linear RUL target function with a fixing threshold value of 130, assumes the degradation process of a system transitions directly from a plateau to a linear decline, and the yielding point is located at the RUL of 130. In a real-life application, once the yielding point is reached, the degradation speed of the system will accelerate till the occurrence of failure. The maintenance strategy should be based on the degradation rate, and the majority of the

solutions are to predict the RUL directly using the whole past data. However, the prediction accuracy varies from engine to engine due to the inconsistent volumes of data collected from different engines and different initial conditions. Therefore, the prediction performance can be unreliable for some engines.

This paper proposes a multi-scale RUL prediction solution using LSTM with a novel RUL target function. Multi-scale prediction solution includes a low-scale RUL prediction and a high-scale RUL prediction, which refers to the prediction of each input data point's stage or RUL value, respectively. The new RUL target function includes a transition stage between the non-degradation and linear degradation stages, which better simulates the engine system's degradation process. Besides the higher RUL prediction accuracy, the proposed approach also aims to provide more flexibility in the maintenance strategy with the multi-scale RUL prediction solution. For instance, based on the low-scale RUL prediction, maintenance can be arranged when the engine system steps into the linear degradation stage. Or, based on the high-scale RUL prediction, maintenance can be arranged when the prediction value is lower than a certain threshold.

Four main contributions are presented in this paper in order to predict the RUL:

- (1) Extracting the valuable Features by using Pearson's correlation coefficient. Pearson's correlation coefficient is used to find the correlation between the sensors and output RUL and the correlation between the sensors themselves.
- (2) Adopting the operation-based normalisation approach. For the engine system working under multiple operation conditions, operation-based normalization is used to reveal the actual degradation patterns concealed in the sensor data.
- (3) Proposing a new RUL target function for the training process. To define an approximation for the actual RUL, we assume the degradation process of the engine system will go through a constant stage, a transition stage and a linear degradation stage. The turning points of these three stages differ from engine to engine because of the different initial, operating and fault conditions.
- (4) Proposing a multi-scale RUL prediction solution using LSTM. An LSTM-based classification model is used to sort each input data point into the three stages defined in the new RUL target function as the low-scale RUL prediction. Then, the data in the transition and linear degradation stage is fed into another LSTM-based regression model to achieve the high-scale RUL prediction.

The paper is structured as follows: Section 5.2 describes the C-MAPSS datasets. Section 5.3 introduces the proposed multi-scale RUL prediction approach. Section 5.4 reports the performance of the proposed method and the comparison with the state-of-the-art methods for RUL prediction, and the conclusion is given in Section 5.5.

5.2 Dataset description

NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) is a high-fidelity computer model aiming at modelling the damage propagation of aircraft gas turbine engines (Saxena *et al.*, 2008b). Figure 5-1 shows the diagram of the simulated run-to-failure trajectories of a small fleet of turbofan engines with the main elements. The simulated engine is a tow spool turbofan engine with a high level of thrust up to 400340 (Newton) which operates under operating conditions ranging from sea level to 12192 (meters) of altitude and ambient temperature varying from -51°C to 39°C (Berghout *et al.*, 2020). The air is introduced into the low-pressure compressor (LPC) through the fan at first in this type of engine. Then, it travels through the high-pressure compressor (HPC) and meanwhile is heated in the combustor. In the combustor, the compressed air is mixed with fuel and ignited. The fuel combustion provides enough thrust to drive both low-pressure turbines (LPT) and high-pressure turbines (HPT) (Ordóñez *et al.*, 2019). A more in-depth explanation can be found in the C-MAPSS User's Guide (Liu *et al.*, 2012).

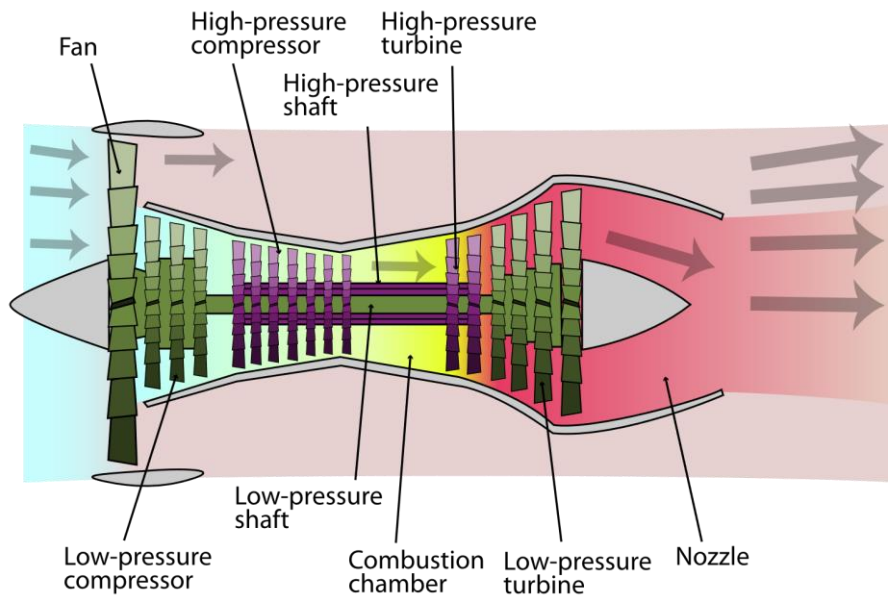


Figure 5-1 Simplified diagram of the simulated engine.

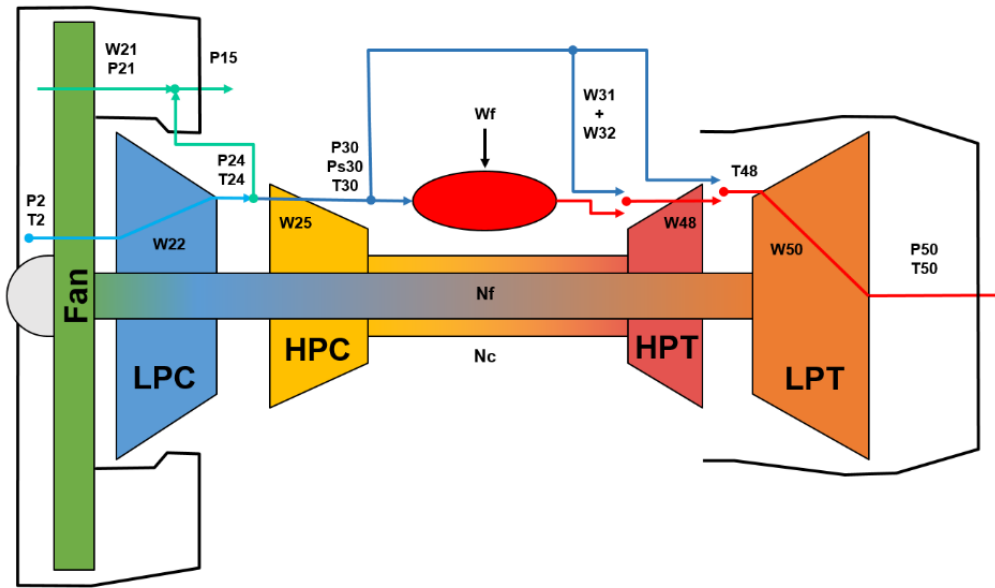


Figure 5-2 Schematic representation of the CMAPSS model (Chao *et al.*, 2021)

Retrieved data from C-MAPSS software is provided for the public as a benchmark for research on RUL prediction for aircraft engines (Saxena *et al.*, 2008). The dataset consists of four subsets and each subset has different numbers of engines with varied operating cycles. In this dataset, engine profiles were simulated with different initial degradation conditions. The maintenance was not considered during the simulation. The dataset includes one training set and one testing set for each engine both of which contain a multivariate time series of 26 features (engine number, time cycles, three operating condition measurements and 21 sensor measurements). The objective is to predict the RUL of each engine based on the given sensor measurements. The information on the four subsets is listed in Table 5-1. Specifically, FD001 refers to the engine failure arising from the high-pressure compressor under a single operating condition. FD002 refers to the engine failure from the high-pressure compressor under six operation conditions. FD003 refers to the engine failure from both high-pressure compressor and fan under a single operating condition. FD004 refers to the engine failure from both high-pressure compressor and fan under six operation conditions. In this study, FD001 was primarily used to demonstrate the proposed solution because the data volume is relatively small to achieve time efficiency. The other three datasets were also tested to validate this method.

Table 5-1 C-MAPSS dataset

Dataset	FD001	FD002	FD003	FD004
Number of engines for training	100	260	100	249
Number of engines for testing	100	259	100	248
Operating conditions	single	multiple	single	multiple
Fault conditions	compressor	compressor	compressor & fan	compressor & fan

The employed dataset is composed of information relating to a total of 100 different turbines with a total number of observations that varies from 128 to 362. The descriptive statistics of the response and the predictor variables are summarized in Table 5-2. The location of each sensor is illustrated in Figure 5-2.

Table 5-2 Descriptive statistics of all the variables of dataset FD001

Index	Symbol	Input variables	Mean	Standard deviation
1	ID	Engine ID		
2	T	Time cycle	108.81	68.88
3	OP1	Operation condition 1	-8.9×10^{-6}	0.003
4	OP2	Operation condition 2	2.4×10^{-6}	0.003
5	OP3	Operation condition 3	100.00	10^{-6}
6	T2	Total temperature at fan inlet (°R)	518.67	0
7	T24	Total temperature at LPC outlet (°R)	642.68	0.5
8	T30	Total temperature at HPC outlet (°R)	1590.52	6.1
9	T50	Total temperature at LPT outlet (°R)	1408.93	9
10	P2	Pressure at fan inlet (psia)	14.62	3.39×10^{-6}
11	P15	Total pressure in bypass-duct (psia)	21.61	0.001
12	P30	Total pressure at HPC outlet (psia)	553.37	0.89
13	Nf	Physical fan speed (rpm)	2388.10	0.07
14	Nc	Physical core speed (rpm)	9065.24	22.08

15	Epr	Engine pressure ratio (P50/P2)	1.30	4.66*10 ⁻¹³
16	Ps30	Static pressure at HPC outlet (psia)	47.54	0.27
17	Phi	Ratio of fuel flow to Ps30 (pps/psi)	521.41	0.74
18	NRf	Corrected fan speed (rpm)	2388.10	0.72
19	NRc	Corrected core speed (rpm)	8143.75	19.08
20	BPR	Bypass ratio	8.44	0.04
21	farB	Burner fuel–air ratio	0.03	1.56*10 ⁻¹⁴
22	htBleed	Bleed enthalpy	393.21	1.55
23	Nf_dmd	Demanded fan speed (rpm)	2388.00	0
24	PCNfR_dmd	Demanded corrected fan speed (rpm)	100.00	0
25	W31	HPT coolant bleed (lbm/s)	38.82	0.18
26	W32	LPT coolant bleed (lbm/s)	23.28	0.11

Since Root Mean Square Error (RMSE) is the most widely used indicator in residual life prediction. It was used to evaluate the performance of the trained neural networks in this case study. The mathematical expression is:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n d_i^2} \quad (5-1)$$

where n is the total number of true RUL targets in the related test set and d_i refers to the difference between the true RUL and the predicted RUL at each time cycle.

5.3 Methodology

The proposed scheme of multi-scale RUL prediction is shown in Figure 5-3(b). Figure 5-3(a) demonstrates the flow chart of 3-stage classification model; Figure 5-3(c) illustrates the flow chart of the LSTM-based regression model. At first, the feature selection method is used to refine the number of input variables to reduce the computational cost of modelling and improve the model's prediction performance by reducing interference from noise. Then, in order to reduce data redundancy and improve

data integrity, an operation-based normalization approach is applied to the selected variables. After that, based on the assumption that the degradation process of the system includes three stages, a correlation-based method is proposed to estimate the two thresholds that can divide the original training set into three parts. An LSTM based 3-stage classification model is then established to automatically determine the degradation stage for each cycle time purely based on the selected inputs, which acts as a low-scale RUL prediction. Finally, another LSTM based model for the last two stages is performed to predict the high-scale RUL. Each step will be explained in more detail.

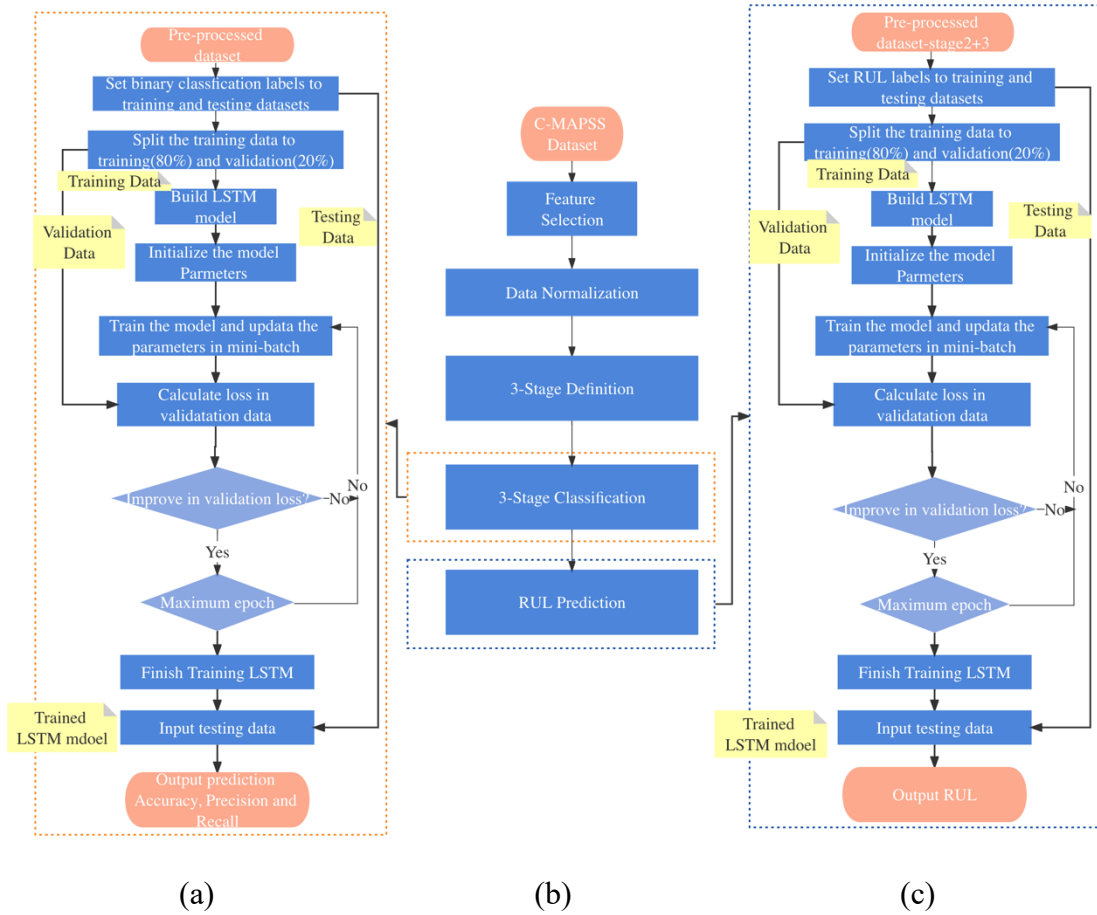


Figure 5-3 The flowchart of the proposed multi-scale RUL prediction framework. (a) the process of the 3-stage classification (low-scale RUL prediction), (b) the overall framework of multi-scale RUL prediction, and (c), the process of the high-scale RUL prediction

5.3.1 Feature selection

An accurate prognosis relies on effective feature extraction from the whole series to capture valuable information and discard irrelevant noise (Xingyu Li *et al.*, 2022). Feature selection is primarily focused on removing non-informative or redundant variables from

the data that do not contribute to the prediction accuracy of the model or even cause overfitting problems. Since the RUL prediction of a system is a regression predictive modelling problem with numerical input variables and output variables, Pearson's correlation coefficient is the most common method to use in this area (Hauke and Kossowski, 2011). Pearson's correlation coefficient is used to measure the strength of the linear relationship between two data samples.

In this case, we assume the input variables as $x_i(t)$, where $i = 1, 2, \dots, 26$; $t = 1, 2, \dots, n$; n is the number of observations, and we assume the output RUL is $y(t)$. We selected the most useful features through three steps. Take dataset FD001 as an example, the first step is to find out the Descriptive statistics of all the variables. Based on Table 5-2, the sensor measurements $T2$, Nf_dmd and $PCNfR_dmd$ are constant and the standard deviation of $OP3$, $P2$, Epr and $farB$ are almost equal to zero. Therefore, these seven nugatory variables are discarded to reduce the computation cost. The second step is to calculate the Pearson correlation between the selected input variable $x_p(t)$ and $x_q(t)$ following formula Eq. (5-2). The last step is to figure out the Pearson correlation between the selected input variable $x_p(t)$ and output $y(t)$ following formula Eq. (5-3) calculate the Pearson correlation between the selected input variable $x_p(t)$ and $x_q(t)$,:

$$r_{x_p(t), x_q(t)} = \frac{n \sum x_p(t)x_q(t) - \sum x_p(t) \sum x_q(t)}{\sqrt{n \sum x_p(t)^2 - (\sum x_p(t))^2} \sqrt{n \sum x_q(t)^2 - (\sum x_q(t))^2}} \quad (5-2)$$

$$r_{x_p(t), y(t)} = \frac{n \sum x_p(t)y(t) - \sum x_p(t) \sum y(t)}{\sqrt{n \sum x_p(t)^2 - (\sum x_p(t))^2} \sqrt{n \sum y(t)^2 - (\sum y(t))^2}} \quad (5-3)$$

$r_{x_p(t), x_q(t)}$, $r_{x_p(t), y(t)}$ are the correlation coefficient between $x_p(t)$ and $x_q(t)$, $x_p(t)$ and $y(t)$, ranges from -1 to +1. A correlation coefficient of 1 indicates that for every positive increase in one variable, there is a positive increase of fixed proportion in another variable. A correlation coefficient of -1 means that for every positive increase in one variable, there is a negative decrease of fixed proportion in another variable. A correlation coefficient of 0 means that these two variables are unrelated. The absolute value of the correlation coefficient refers to the relationship strength between the two variables. The selection of the thresholds will affect the volume of the dataset that feeds into the model and then affect the prediction performance. Therefore, we only discarded the highly correlated

variables. In this case, one of the two variables will be discarded if $r_{x_p(t), x_q(t)}$ is higher than 0.95. We have tested different thresholds, and it has been found that the value of 0.95 achieves the best result for this dataset. Similarly, we only discarded the extremely irrelevant variables, in other words, when $r_{x_p(t), y(t)}$ is extremely close to zero. Based on many tests, the threshold is set to 0.01.

5.3.2 Data normalisation

Normalisation techniques are generally necessary to effectively train deep neural networks (Lubana *et al.*, 2021). The beneficial properties of this technique include the ability to propagate informative activation patterns in deeper layers, reduced dependence on initialisation, smoothing of loss landscape and so on. In this case, we applied the Min-Max scaler to rescale the data from the original range to a new fixed range between 0 and 1. This scaler assumes that the minimum and maximum observable values of the data are provided or can be accurately estimated. For the selected input $x_p(t)$, the transformation equation is given by:

$$\widehat{x_p(t)} = \frac{x_p(t) - \min(x_p(t))}{\max(x_p(t)) - \min(x_p(t))} \quad (5-4)$$

To address the cases where multiple degradation patterns are concealed in the sensor data such as datasets FD002 and FD004, we propose an operation-based data scaling. The concept is to scale the data that works under the same operation condition and then reconstruct them back to their original order. This way, the influence of different operating conditions can be minimised. We first utilise the unsupervised k-Means clustering based on the selected input variables to classify all raw input variables $x_p(t)$ in terms of different operation conditions ($OC_1, OC_2, \dots, OC_N$). Then each input variable $\widehat{x_p(t)}$ is reconstructed by connecting with all normalised segments with the same time index shown in Eq. (5-5), written as

$$\widehat{x_p(t)} = \bigcup_{j=1}^N \text{Norm}[x_p(t(OC_j))] \quad (5-5)$$

where N is the number of operation conditions and $\text{Norm}[\cdot]$ denotes the MinMax normalisation operation.

5.3.3 RUL target function

Accurately determining the system health status at each time step is an inherent challenge for data-driven prognostic problems. One sensible solution is to simply assign the desired output as the time left before the occurrence of the failure which indicates that the health of the system degrades linearly with usage. The advantage of this solution is that it follows the definition of the RUL strictly which is defined as the time to failure. When the degradation model is unknown, the linear model is the most natural option, as shown in Figure 5-4(a). Another approach is to apply a suitable degradation model, specifically the piece-wise linear degradation model for this dataset. The RUL is assumed to be constant until it crosses a certain threshold value which can be seen in Figure 5-4(b). This approach is more reasonable because the degradation of the system usually starts after a certain degree of usage. Also, it is more likely to avoid the model from overestimating the RUL. Because the data used to train the network is run-to-failure samples, and the samples in the testing set are still within the service life. The prediction error will be substantial for the engines still in the initial use stage.

Similar to the piece-wise linear degradation model, this study assumes that the degradation of the system only starts after a certain degree of usage. However, this paper also assumes that it will not turn into the linear degradation mode directly after the threshold value. Instead, the degradation speed should be slow at first and then increase over time. Therefore, we propose a three-stage RUL target function as illuminated in Figure 5-4(c), which includes the constant stage, the transition stage where the system degrades nonlinearly with usage, and the linear degradation stage.

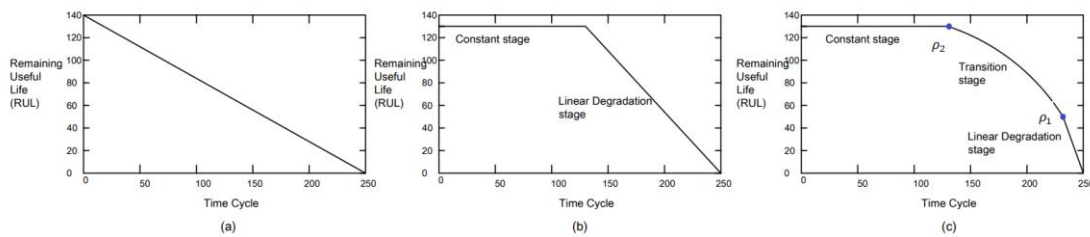


Figure 5-4 RUL Target Functions. (a) 1-stage RUL target function, (b) 2-stage RUL target function, and (c) the proposed 3-stage RUL target function

One challenge for the proposed three-stage degradation process is to find out the two thresholds that can distinguish the whole dataset into three parts. A two-step correlation-

based method is introduced to accomplish this work. First, we assume that the RUL degradation is linear temporally, so the RUL can be labelled for each time cycle. Therefore, the RUL of the first time cycle for each engine is the maximum time cycle of that engine. The first threshold ρ_1 can be calculated by the following equations:

$$\rho_1 = \underset{\tau}{argmax} \frac{1}{m} \sum_{p=1}^m |r_{\hat{x}_p, \hat{y}_\tau}|, \tau = 1, 2, \dots, n \quad (5-6)$$

$$\hat{y}_\tau(t) = \begin{cases} 0, & y(t) > \tau \\ 3, & y(t) \leq \tau \end{cases} \quad (5-7)$$

where m is the number of the selected input variables. The range of τ is selected referring to the average numbers of observations of 100 engines. The second threshold ρ_2 is then computed by

$$\rho_2 = \underset{\tau}{argmax} \frac{1}{m} \sum_{p=1}^m |r_{\hat{x}_p, \hat{y}_\tau}|, \tau = \rho_1, \rho_1 + 1, \dots, n \quad (5-8)$$

$$\hat{y}_\tau(t) = \begin{cases} 1 & y(t) > \tau \\ 2 & \rho_1 < y(t) \leq \tau \\ 3 & y(t) \leq \rho_1 \end{cases} \quad (5-9)$$

5.3.4 Multi-scale RUL prediction

5.3.4.1 Two-step LSTM model for 3-stage classification

Long Short-Term Memory (LSTM) network is a modified structure of the recurrent cell that incorporates the standard recurrent layer along with additional “memory” control gates. LSTM uses storage elements to transfer information from the past output instead of having the output of the RNN cell be a nonlinear function of the weighted sum of the current input and previous output. In another word, instead of using a hidden state h only, LSTM adopts a cell state C to keep the long-term information as shown in Figure 5-5. The central concept of LSTM is utilising three gates, such that each is a nonlinear function σ activated by a weighted sum of the current input observation x_t and previous hidden state h_{t-1} , to control the cell state C (forget gate f_t , input gate i_t and output gate o_t). The forget gate f_t is used to manage the information from the previous cell state C_{t-1} to the current cell state C_t ; the input gate i_t decides how many inputs should be kept in the current cell state C_t ; and the output gate o_t determines the output h_t from the current cell state C_t .

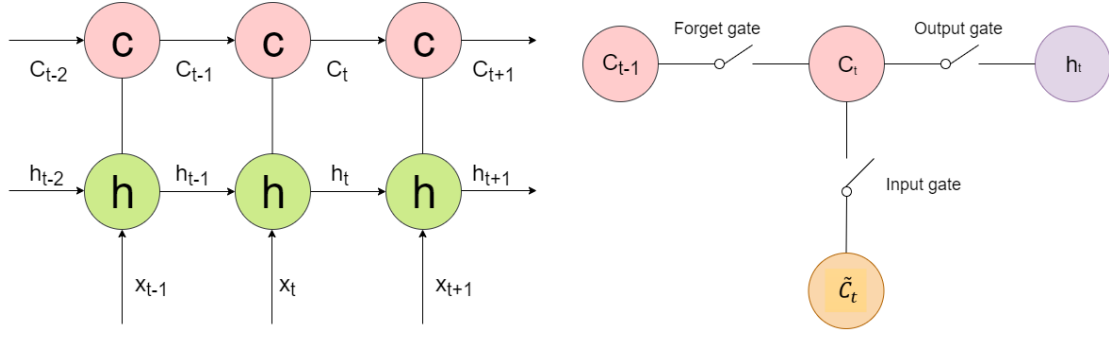


Figure 5-5 Cell state and three gates in LSTM

The output of LSTM at step t is calculated using the following equation:

$$i(t) = \sigma(U_i \widetilde{x_p(t)} + W_i h(t-1) + b_i) \quad (5-10)$$

$$f(t) = \sigma(U_f \widetilde{x_p(t)} + W_f h(t-1) + b_f) \quad (5-11)$$

$$o(t) = \sigma(U_o \widetilde{x_p(t)} + W_o h(t-1) + b_o) \quad (5-12)$$

$$\widetilde{C(T)} = \tanh(U_{\tilde{c}} \widetilde{x_p(t)} + W_{\tilde{c}} h(t-1) + b_{\tilde{c}}) \quad (5-13)$$

$$C(t) = C(t-1) \cdot f(t) + \widetilde{C(T)} \cdot i(t) \quad (5-14)$$

$$h(t) = \tanh(C(t)) \cdot o(t) \quad (5-15)$$

where U , W and b are the trainable weights and biases, respectively. These three gates have the same shape with different parameters U and W , which need to be learned from the training process. The candidate state $\widetilde{C(T)}$ cannot be used directly. It must pass through the input gate and then be used to calculate the internal storage $C(t)$. While $C(t)$ is not only affected by the hidden state but also by $C(t-1)$ which is controlled by the forget gate. Based on $C(t)$, a layer of $\tanh$ function is applied to the output information h_t , which is constrained by the output gate. The existence of the gates enables LSTM to fulfil the long-term dependencies in the sequence, and by learning the gate parameters, the network can find the appropriate internal storage.

The proposed two-step binary classification model is demonstrated in Figure 5-3(a). To implement this LSTM model, the two threshold values ρ_1 , ρ_2 are required to label the output for the training set. In the first step, ρ_1 is used to label the data, written as

$$y_1(t) = \begin{cases} 3, & y(t) \leq \rho_1 \\ 0, & y(t) > \rho_1 \end{cases} \quad (5-16)$$

Then $y_1(t)$ and $\widetilde{x_p(t)}$ (or $\widetilde{x_p(t)}$) are used to train the first model. In the second step, ρ_1 and ρ_2 are used to label the data, written as

$$y_2(t) = \begin{cases} 2, & \rho_1 < y(t) \leq \rho_2 \\ 1, & y(t) > \rho_2 \end{cases} \quad (5-17)$$

The data in stage $y(t) \leq \rho_1$ is neglected in this step. Then $y_2(t)$ and $\widehat{x_p(t)}$ (or $\widehat{x_p(t)}$) are used to train the second model. Based on these two models, the testing data can be classified into three stages purely based on the input variables $\widehat{x_p(t)}$. It should be noted that the original training set is split into two parts, the first 80% engines for training and the last 20% engines for validation. The parameters used in the structure include activation function: sigmoid; neuron number: 128; optimiser: Adam; loss function: the binary cross-entropy.

5.3.4.2 LSTM-based high-scale RUL prediction

Once the engine enters stage 2, the degradation rate is accelerated. A higher scale RUL prediction is then demanded to closely monitor the system performance. The data in stage 2 and stage 3 is extracted before feeding selected variables into the LSTM model. The cycle time threshold t_1 is determined when

$$y(t_1) = \rho_1 \quad (5-18)$$

The dataset $y(t)$ and $\widehat{x_p(t)}$, where $t = t_1, t_1 + 1, \dots, n$, are then used to train the model. Since the volume of the data has been reduced, the validation split is set to be 5% of the original dataset. The parameters used in the structure include the activation function: ReLU; optimiser: Adam; and the loss function: Mean Square Error.

5.4 Results

5.4.1 Feature selection

Figure 5-6 plots the raw sensor measurements of Engine 2 in the dataset FD001, where the variation of each sensor can be observed. There are 7 sensors (OP3, T2, P2, P15, farB, Nf_dmd, and PCNfR_dmd) that have no variation and were discarded in the first step. Then, the correlation method was applied to find out the correlation between each selected input variable and the output RUL using Eq. (5-3). Based on the result displayed in Table 5-3, the measurements of OP1 and OP2 are dropped because their correlation values are small than a pre-set threshold of 0.01.

In the third step, the pairwise correlation coefficients among the selected sensors were calculated using Eq. (5-2) and the results are illustrated in Figure 5-7. A pre-set threshold of 0.95 was used in this case study. It can be observed that Nc and NRc are highly correlated ($r=0.98$), which suggests there is no additional benefit to including both measures for the RUL prediction. Therefore, in this study, we dropped the sensor Nc for the following analysis. Through these three steps, the 14 sensors of T24, T30, T50, P15, P30, Nf, Ps30, Phi, NRf, NRc, BPR, htBleed, W31 and W32 are finally selected as the features for the multi-stage RUL prediction in the dataset FD001.

The feature selection process and the outcome for the datasets FD002 – FD004 can be seen in Table 5-4.

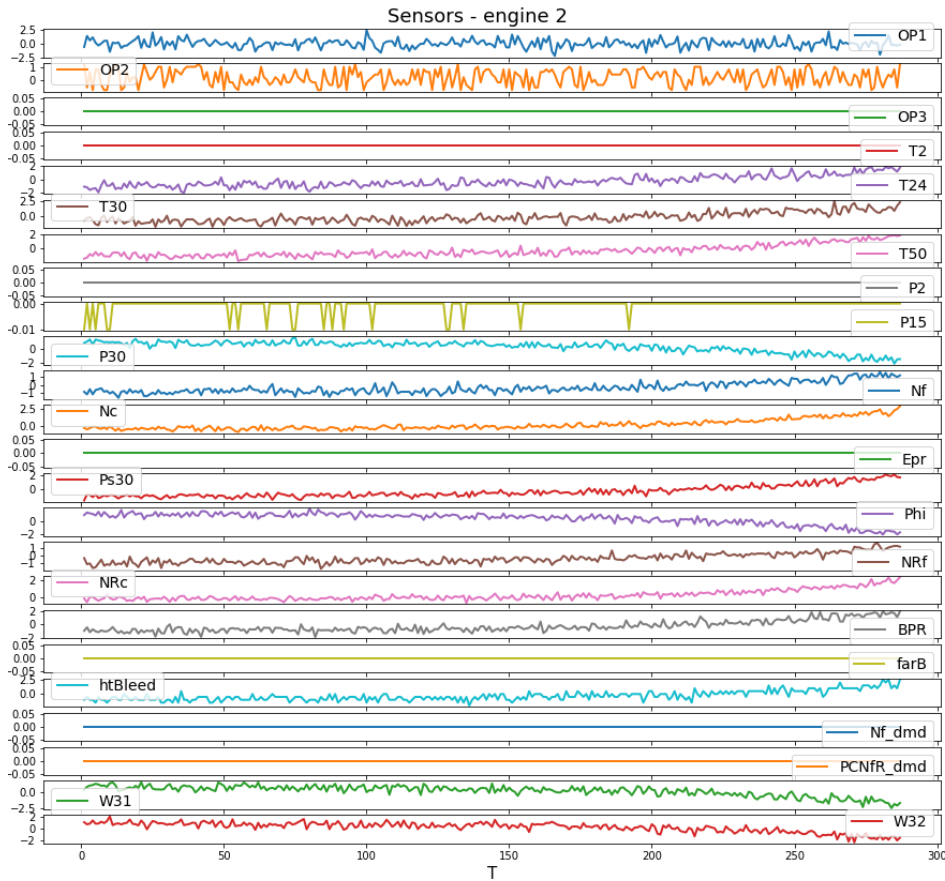


Figure 5-6 The sensor observations till the failure of Engine 2 in the dataset FD001

Table 5-3 Correlation values between each input and the output RUL for Engine 2 in the dataset FD001, where the discarded sensors are highlighted

Sensor	Correlation value	Sensor	Correlation value
Ps30	0.727	T30	0.616

T50	0.711	NRf	0.573
Phi	0.701	Nf	0.571
P30	0.685	Nc	0.448
BPR	0.673	NRc	0.362
W32	0.666	P15	0.121
W31	0.663	OP2	0.010
htBleed	0.640	OP1	0.008
T24	0.632		

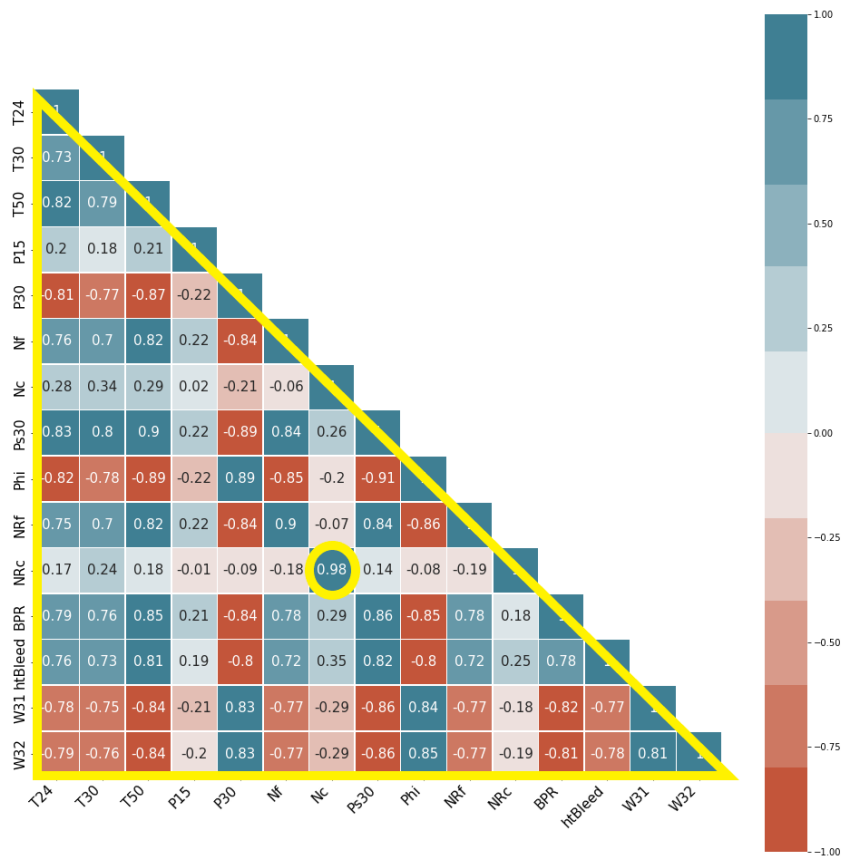


Figure 5-7 The correlation coefficients between the selected sensors of Engine 2 in the dataset FD001, where the discarded sensor is highlighted

Table 5-4 Feature selection process for the datasets FD002 – FD004

FEATURES	FD001	FD002	FD003	FD004
OP1	X(IR)	X(IR)	X(IR)	X(IR)
OP2	X(IR)	X(IR)	X(IR)	X(IR)

OP3	X(CO)	X(CO)	X(CO)	X(CO)
T2	X(CO)	X(CO)	X(CO)	X(CO)
T24	√	√	√	√
T30	√	√	√	√
T50	√	√	√	√
P2	X(CO)	X(CO)	X(CO)	X(CO)
P15	X(CO)	√	√	X(IR)
P30	√	√	X(RE)	X(RE)
Nf	√	X(RE)	X(RE)	X(RE)
Nc	X(RE)	X(RE)	X(RE)	X(RE)
Epr	√	√	√	√
Ps30	√	√	√	√
Phi	√	√	√	√
NRf	√	√	√	√
NRc	√	√	√	√
BPR	√	√	√	√
farB	X(CO)	√	X(CO)	√
htBleed	√	√	√	√
Nf_dmd	X(CO)	X(CO)	X(CO)	X(CO)
PCNfR_dmd	X(CO)	X(CO)	X(CO)	X(CO)
W31	√	√	√	√
W32	√	√	√	√
*X=Feature dropped √=Feature selected IR=Irrelevant CO=Constant RE=Redundant				

5.4.2 LSTM model for 3-stage classification

To achieve the 3-stage classification, the two thresholds ρ_1 and ρ_2 need to be determined after the raw sensor signals are scaled by the MinMax normalisation. Figure 5-8 plots the mean correlation value against different τ for the four datasets using Eq. (5-6) (the left column of Figure 5-8) and Eq. (5-8) (the right column of Figure 5-8). The τ values of the

peaks are selected as ρ_1 and ρ_2 . It is observed that ρ_1 can be easily set with a clear peak, which suggests that the linear degradation stage (the third stage) is relatively easy to be distinguished. The value of ρ_1 is between 50 to 65 for these datasets. The selection of ρ_2 is tricky as the mean correlation value doesn't drop dramatically with large τ values. This indicates that the boundary between the transition stage (the second stage) and the constant stage (the first stage) is usually vague. The selection of ρ_2 in datasets FD001 and FD002 are relatively smaller than that in FD003 and FD004, which suggests that the engines with more fault conditions may enter the stage 2 sooner.

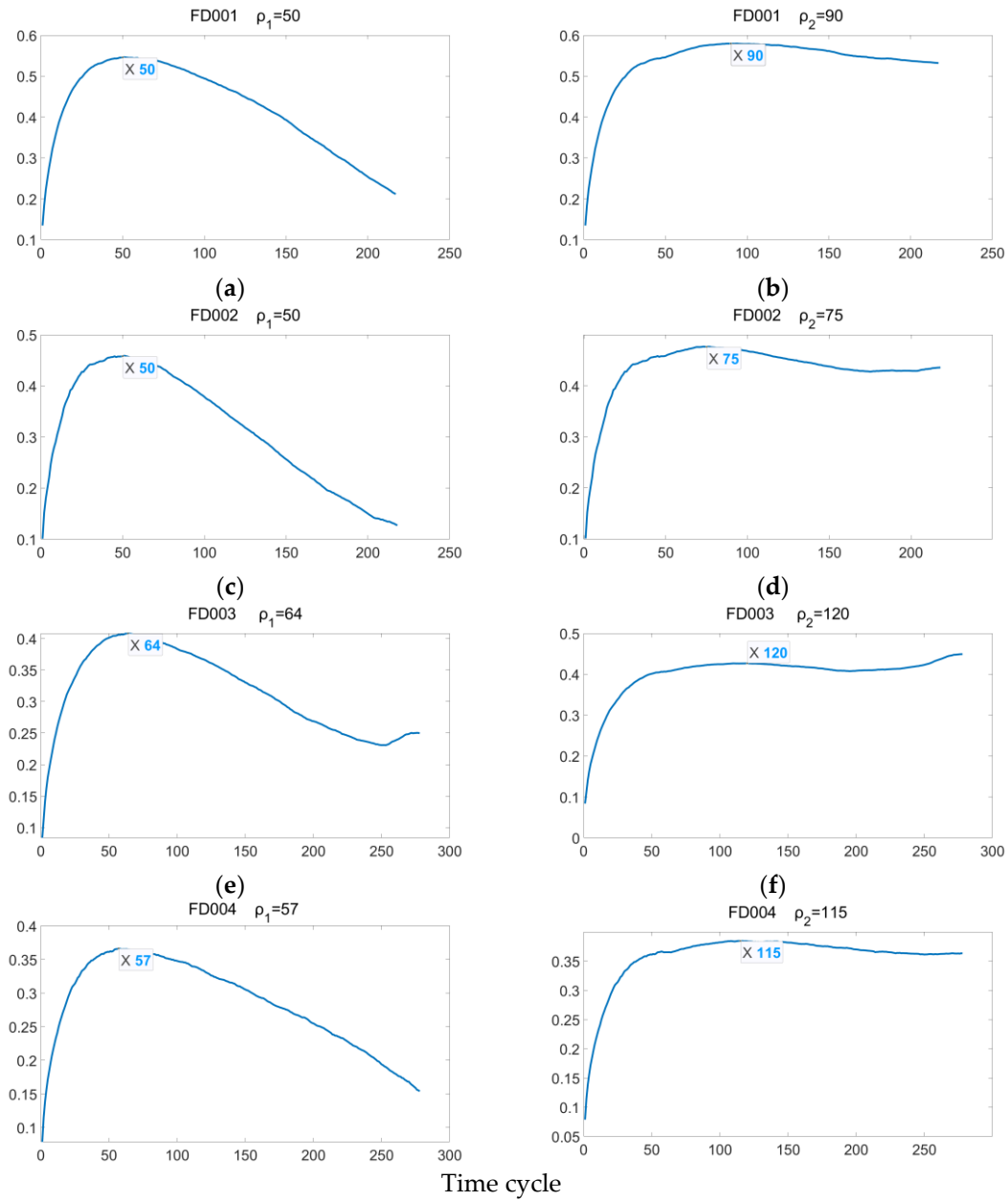


Figure 5-8 The selections of ρ_1 and ρ_2 for four datasets

After the determination of ρ_1 and ρ_2 , a two-step LSTM model for the 3-stage classification is established. In the first step, we labelled the normalised data based on Eq. (5-7) and fed it into the LSTM-based binary classification model. In the second step, we labelled the data based on Eq. (5-9) and then discarded the data $\tilde{y}_t(t) = 3$. The rest of the data were then fed to another binary classification model. The classification result of both models is displayed in Table 5-5, where the model has been run 3 times and the average values are recorded. It can be observed that stages 2 and 3 can be distinguished with high accuracy (0.96, 0.92, 0.95 and 0.88 for FD001, FD002, FD003 and FD004 respectively). The accuracy in distinguishing stages 1 and 2 is relatively low (0.86, 0.87, 0.81 and 0.74, respectively). The reason results in the prediction difference in accuracy between ρ_1 and ρ_2 is that the degradation process is relatively slow in the initial phase. As seen in Figure 6, the increase or decrease trend in sensor observations is inconspicuous at the beginning. In addition, that life span is relatively long compared to the engines' total lifetime, making it harder for the model to learn the features. Another reason is the selection of the threshold of ρ_2 using the correlation-based method is not as precise as ρ_1 . FD004, which works under more fault and multiple operational conditions, has the lowest low-scale RUL prediction. Since the operation-based normalisation approach minimized the impact caused by multiple operating conditions, the difference in prediction accuracy mainly results from the fault conditions.

Table 5-5 Results of the binary classification of datasets FD001-FD004

	Threshold (ρ_1/ρ_2)	Accuracy (ρ_1/ρ_2)	
		Train set	Test set
FD001	50/90	0.94/0.85	0.96/0.86
FD002	50/75	0.93/0.88	0.92/0.87
FD003	64/120	0.92/0.84	0.95/0.81
FD004	57/115	0.93/0.85	0.88/0.74

5.4.3 LSTM-based RUL prediction for the stage 2 and stage 3

In this part, the data in the constant stage ($y(t_1) \geq \rho_2$) is neglected and only the data in the transition stage ($\rho_1 < y(t_1) < \rho_2$) and the linear degradation stage ($y(t_1) \leq \rho_1$) is

adopted. These data are then fed to a two-layer LSTM model with labelled RUL. Figure 5-9 plots the prediction result of the first 18 engines in the training dataset. Generally, the RUL prediction accuracy is relatively poor at the beginning and is in line with the ground truth when closing to the failure point. The most likely reason for this result is that the turning points from the constant and transition stages in different engines in the same dataset are also different. Although these engines suffer from the same operating and fault conditions, the degradation process would not be precisely the same due to other factors like the working environment.

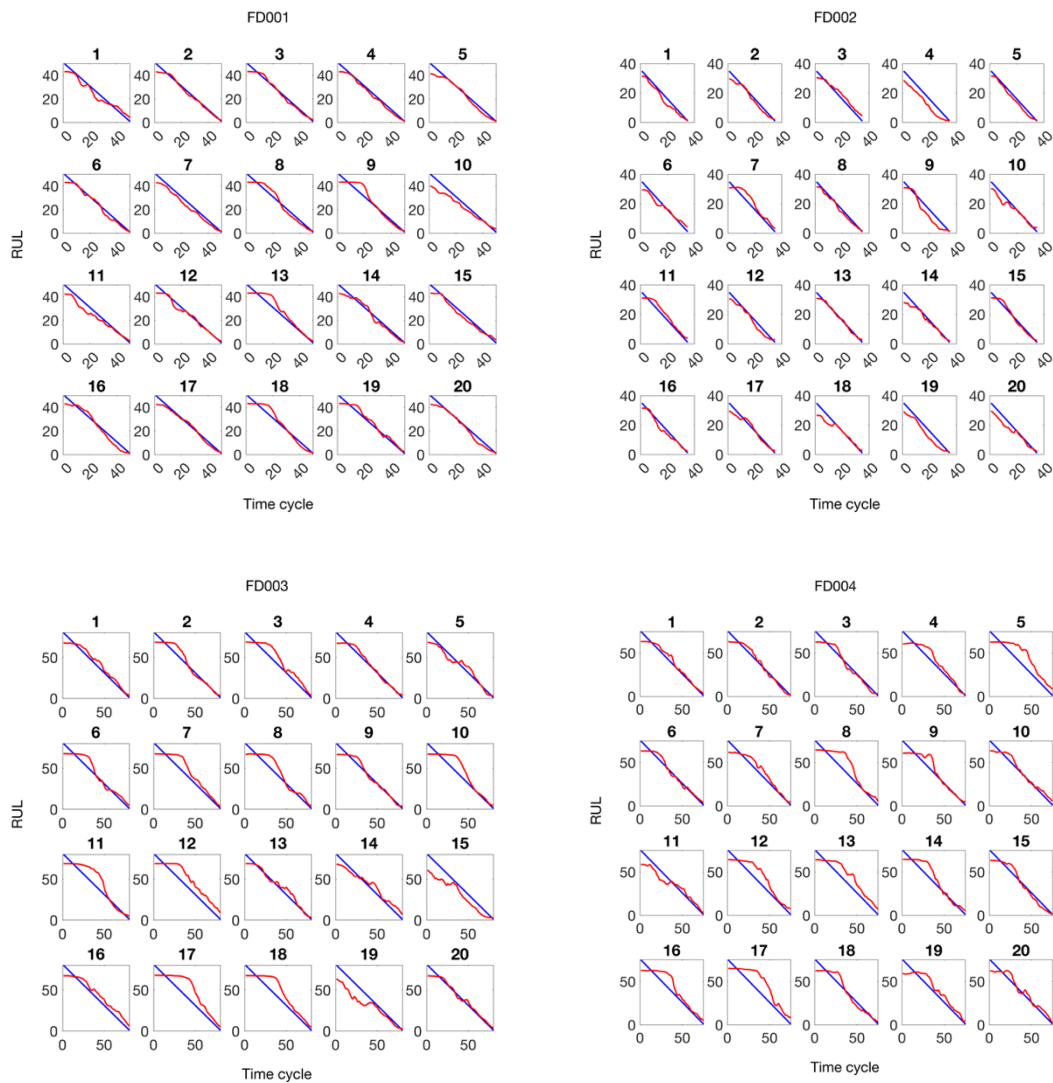


Figure 5-9 RUL Prediction results of the first 18 engines in the training set

Figure 5-10 demonstrates the uncertainty quantification of the RUL prediction of all engines in the training set for each subset. It can be observed that the average RUL

prediction, indicated by the solid curves, overall aligns with the ground truth. The pink area (indicating the 95% confidence interval) shows the decreasing trend of the uncertainty of RUL prediction following the increment of the lifecycle. The RUL prediction of the first two subsets outperforms that of FD003 and FD004 as they have only one fault condition.

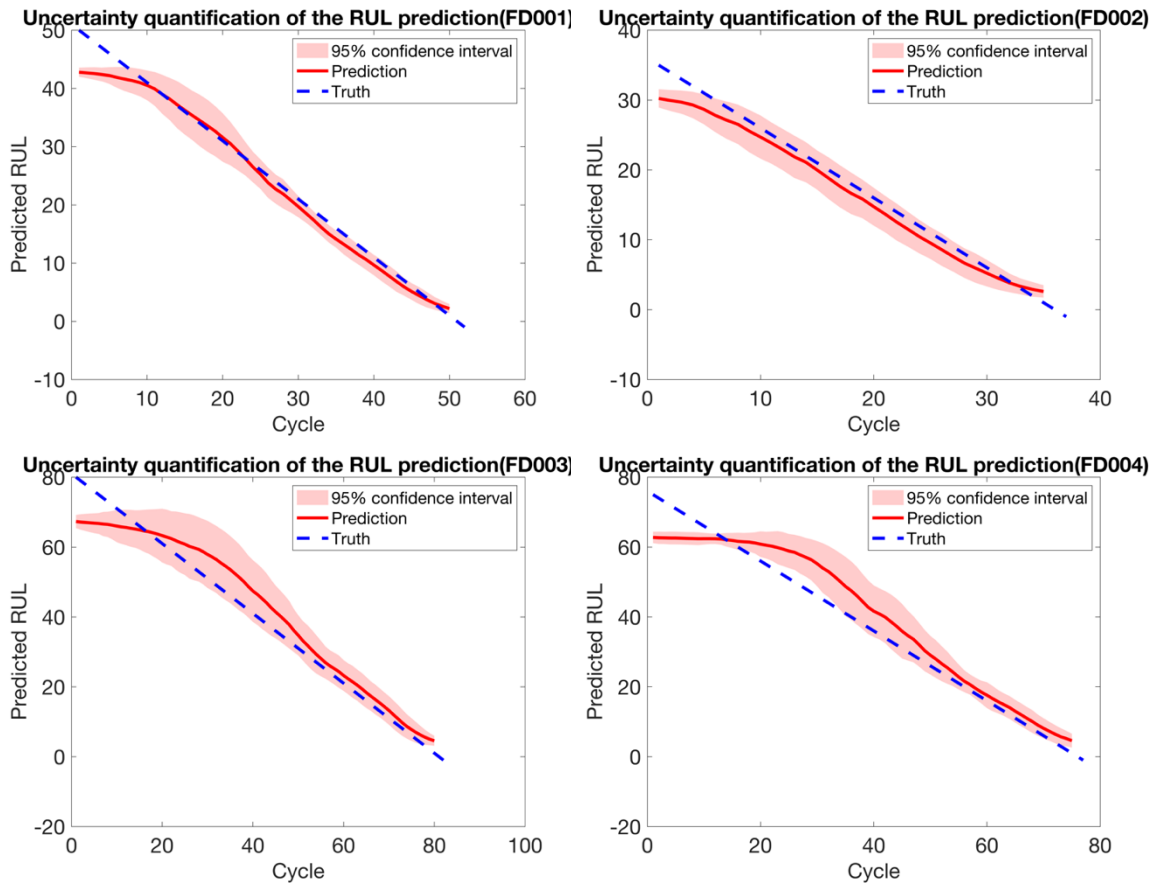


Figure 5-10 Uncertainty quantification of the RUL prediction using the proposed method by combining all engines for each sub-dataset.

Table 5-6 demonstrates the high-scale RUL prediction performance of the LSTM-based model on both the training dataset and testing dataset for FD001-FD004 with different window lengths. The result indicates that when the volume of the dataset is guaranteed, a longer window size usually leads to a better RUL prediction performance. Therefore, one drawback of the proposed model for the RUL prediction is that it segments the dataset into different stages which reduces the data volume of each stage and thereby restricts the

options of the window size. For instance, in the dataset FD002, the data that can be used for training is very limited since the threshold to distinguish stage 1 and stage 2 is relatively small ($\rho_2 = 75$). Therefore, to ensure the quality of the training, the window size needs to be quite small in this case. In addition, the RUL prediction performance on dataset FD001 is generally better than the others. The reason is that the engines in this dataset are working only under one operation condition and there is only one fault condition which means much less noise is involved when recording the measurements.

Table 5-7 compares the proposed method's result with the window length of 40 with other state-of-the-art approaches on all turbofan engine benchmark datasets, where we ran the model three times and the average values were used. A significant improvement can be seen in the RUL prediction performance on the datasets FD001, FD002 and FD004. The improvement over the best existing method for these three datasets is 40%, 22% and 4% respectively. For the dataset FD003, the proposed three-stage model is also better than most of the other methods and remarkably close to the best existing method. The improvement of the proposed method on datasets FD001 and FD002 is much more apparent than that on datasets FD003 and FD004. The reason is that the classification accuracy of these two datasets demonstrated in Section 5.4.2 is relatively higher.

Table 5-6 RUL prediction performance with different window lengths (wl) for both the train set and the test set (RMSE) (First 80% of engines in the train set are used for training and the other 20% are used for validation)

Dataset	Training data set				Testing data set			
	wl=10	wl=20	wl=30	wl=40	wl=10	wl=20	wl=30	wl=40
FD001	9.15	6.88	5.03	4.21	14.16	9.41	5.95	5.21
FD002	8.71	7.4	6.85	4.22	16.54	16.13	14.95	11.76
FD003	12.55	10.61	9.66	8.37	17.07	17.29	14.35	11.61
FD004	15.88	13.41	10.5	8.75	25.49	22.49	20.18	18.89

Table 5-7 Results of the RUL prediction performance between the proposed method and the state-of-the-art, where the best prediction of each dataset is highlighted

Model	FD001	FD002	FD003	FD004
DCNN(Li <i>et al.</i> , 2018)	12.61	22.36	12.64	23.31
DCGAN (Hou <i>et al.</i> , 2020)	10.71	19.49	11.48	19.71
RBM+LSTM (Listou Ellefsen <i>et al.</i> , 2019)	12.56	22.73	12.1	22.66

BiLSTM+ED (Ahmed , Elsheikh <i>et al.</i> , 2018)	14.47	N/A	17.48	N/A
CaConvNet (Ayodeji <i>et al.</i> , 2021b)	11.83	15.12	9.24	22.31
Multi-head FNN (Ayodeji <i>et al.</i> , 2021a)	8.68	N/A	9.69	N/A
The proposed method	5.21	11.76	11.61	18.89

5.5 Conclusion

This paper proposed a multi-scale RUL prediction solution using the LSTM neural network that assumes the operating life of aero-engines can be divided into three stages: the constant stage, the transition stage and the linear degradation stage. Experiments were carried out on the popular C-MAPSS dataset to demonstrate the effectiveness of the proposed method. The low-scale RUL prediction by the LSTM-based classification model achieves a relatively good accuracy for all subsets. The high-scale RUL prediction by the LSTM-based regression model obtains a generally higher accurate prediction performance than some state-of-the-art approaches. The enhancement of RUL prediction benefits from three elements:

- Multiple solutions were used to extract the features for further RUL modelling. Firstly, the standard deviations of the measurements are observed to get rid of the constant variables which have no contribution to the dynamics of RUL. Then, Pearson's correlation coefficient is used to drop the irrelevant variables and redundant variables. The Min-Max normalisation is used to adjust the scale of the selected features. For the dataset FD002 and FD004 that worked under multiple operating conditions, the operation-based data scaling method is used to reveal the hidden degradation process. These feature engineering techniques help the proposed model to achieve a better performance in RUL prognostics in terms of reducing computational cost and over-fitting problems.
- Instead of the widely used piece-wise linear degradation model with a fixing threshold value of 130, a novel RUL target function has been introduced. We introduced a transition stage between the non-degradation stage and the linear degradation stage. By cataloguing the engine operating cycle into these three stages purely based on the selected sensors, we achieve a low-scale RUL prediction using

a two-step LSTM-based binary classification model. A correlation-based method was introduced to find out the two thresholds that can divide the dataset into these three stages. As the degradation process varies from engine to engine, the thresholds of all four datasets are calculated separately which is more reasonable than adopting a fixed threshold as in some research.

- By feeding the data in stage 2 and stage 3 into an LSTM-based model with labelled RUL, high-scale RUL prediction is obtained. The experimental result demonstrated the superiority of the proposed solution against the state-of-the-art approaches for most datasets.

It should be noted that the model performance can be further improved by optimising the parameters in the two used LSTMs. One limitation of this solution is that data volume to train the LSTM RUL prediction model for stages 2 and 3 is reduced. Generally, a longer window size leads to a better RUL prediction result when the volume of the dataset is large enough. The proposed method restricts the options of window length. This is the reason why the data in stages 2 and 3 are combined for the high-scale prediction. If the data length is sufficient, each stage can be modelled separately, and the RUL performance should be better. Future work could extend the proposed structure to more advanced neural networks instead of the basic LSTM. The innovation of the present work mainly focuses on the separation of the 3 degradation stages and the manipulation of the features of the dataset.

References

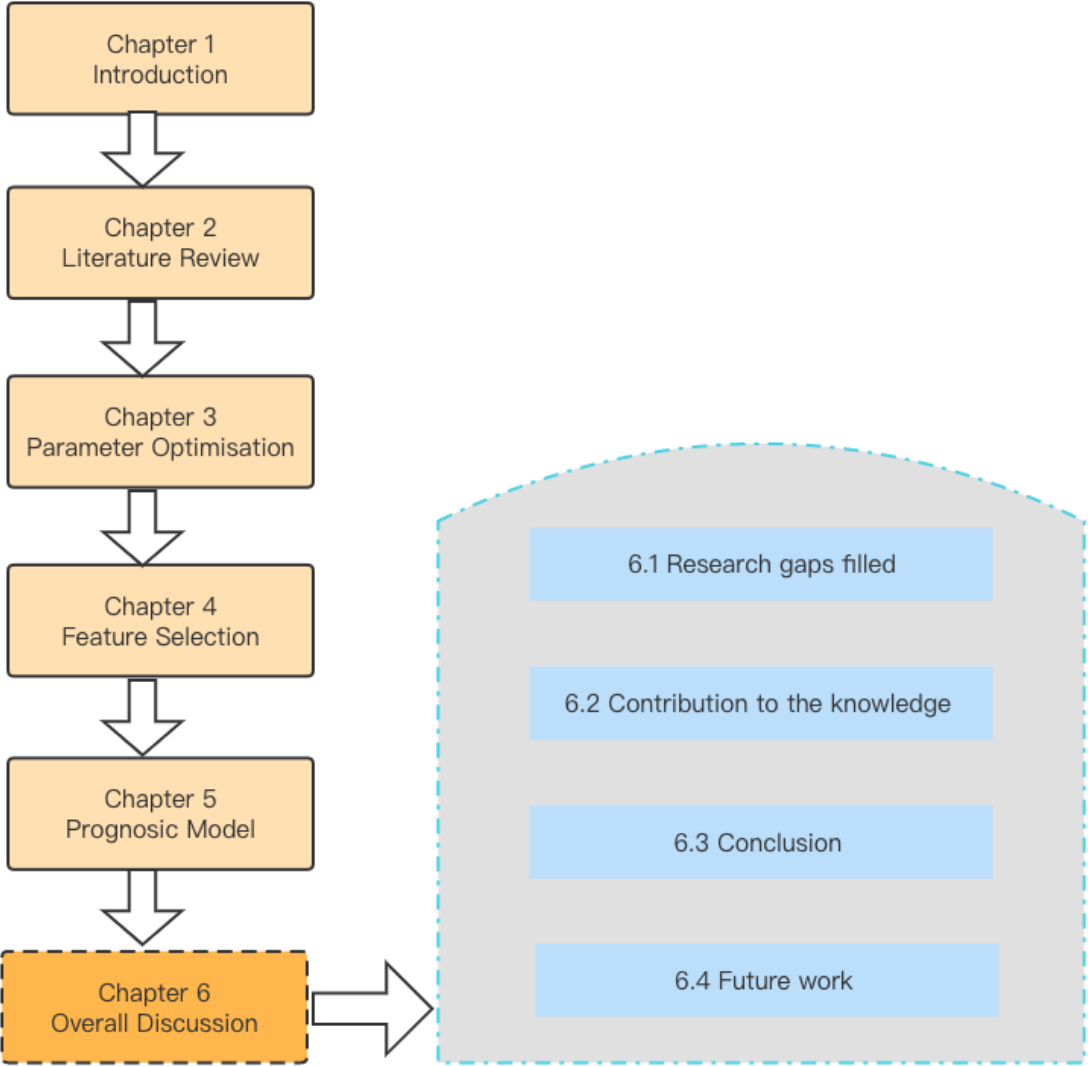
- Ahmed, Elsheikh, Soumaya, Yacout, Mohamed-Salah, Ouali, Elsheikh, A., Yacout, S., Ouali, M.S., 2018. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* 323, 148–156.
<https://doi.org/10.1016/j.neucom.2018.09.076>
- Ayodeji, A., Wang, W., Su, J., Yuan, J., Liu, X., 2021a. An empirical evaluation of attention-based multi-head models for improved turbofan engine remaining useful life prediction.
- Ayodeji, A., Wang, Z., Wang, W., Qin, W., Yang, C., Xu, S., Liu, X., 2021b. Causal augmented ConvNet: A temporal memory dilated convolution model for long-sequence time series prediction. *ISA Trans.*
<https://doi.org/10.1016/j.isatra.2021.05.026>
- Bai, R., Xu, Q., Meng, Z., Cao, L., Xing, K., Fan, F., 2021. Rolling bearing fault diagnosis based on multi-channel convolution neural network and multi-scale clipping fusion data augmentation. *Measurement (Lond)* 184.
<https://doi.org/10.1016/j.measurement.2021.109885>
- Berghout, T., Mouss, L.H., Kadri, O., Saïdi, L., Benbouzid, M., 2020. Aircraft engines Remaining Useful Life prediction with an adaptive denoising online sequential Extreme Learning Machine. *Eng Appl Artif Intell* 96, 103936.
<https://doi.org/10.1016/j.engappai.2020.103936>
- Chao, M.A., Kulkarni, C., Goebel, K., Fink, O., 2021. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data (Basel)* 6, 1–14. <https://doi.org/10.3390/data6010005>
- Chen, J., Jing, H., Chang, Y., Liu, Q., 2019. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliab Eng Syst Saf* 185, 372–382. <https://doi.org/10.1016/j.ress.2019.01.006>
- Cubillo, A., Perinpanayagam, S., Esperon-Miguez, M., 2016. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*.
<https://doi.org/10.1177/1687814016664660>

- Hauke, J., Kossowski, T., 2011. Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data. *Quaestiones Geographicae* 30, 87–93. <https://doi.org/10.2478/v10117-011-0021-1>
- Heimes, F.O., 2008. Recurrent neural networks for remaining useful life estimation. 2008 International Conference on Prognostics and Health Management, PHM 2008. <https://doi.org/10.1109/PHM.2008.4711422>
- Hou, G., Xu, S., Zhou, N., Yang, L., Fu, Q., 2020. Remaining Useful Life Estimation Using Deep Convolutional Generative Adversarial Networks Based on an Autoencoder Scheme. *Comput Intell Neurosci* 2020. <https://doi.org/10.1155/2020/9601389>
- Javed, K., Gouriveau, R., Zerhouni, N., 2017. State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mech Syst Signal Process* 94, 214–236. <https://doi.org/10.1016/j.ymssp.2017.01.050>
- Li, X., Ding, Q., Sun, J.Q., 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab Eng Syst Saf* 172, 1–11. <https://doi.org/10.1016/j.ress.2017.11.021>
- Li, Xingqiu, Jiang, H., Liu, Y., Wang, T., Li, Z., 2022. An integrated deep multiscale feature fusion network for aeroengine remaining useful life prediction with multisensor data. *Knowl Based Syst* 235. <https://doi.org/10.1016/j.knosys.2021.107652>
- Li, Xingyu, Krivtsov, V., Arora, K., 2022. Attention-based deep survival model for time series data. *Reliab Eng Syst Saf* 217. <https://doi.org/10.1016/j.ress.2021.108033>
- Li, X., Zhang, W., Ding, Q., 2019. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab Eng Syst Saf* 182, 208–218. <https://doi.org/10.1016/j.ress.2018.11.011>
- Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S., Zhang, H., 2019. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliab Eng Syst Saf* 183, 240–251. <https://doi.org/10.1016/j.ress.2018.11.027>

- Liu, Y., Frederick, D.K., Decastro, J.A., Litt, J.S., Chan, W.W., 2012. User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). Nasa/Tm 2012-21743, 1-40.
- Lubana, E.S., Dick, R.P., Tanaka, H., 2021. Beyond BatchNorm: Towards a Unified Understanding of Normalization in Deep Learning 1-21.
- Ma, J., Su, H., Zhao, W., Liu, B., 2018. Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning. *Complexity* 2018, 1-13. <https://doi.org/10.1155/2018/3813029>
- Navathe, S.B., Wu, W., Shekhar, S., Du, X., Sean Wang, X., Xiong, H., 2016. Deep convolutional neural network based regression approach for estimation of remaining useful life E1-E1. https://doi.org/10.1007/978-3-319-32049-6_29
- Ordóñez, C., Sánchez Lasheras, F., Roca-Pardiñas, J., Juez, F.J. de C., 2019. A hybrid ARIMA-SVM model for the study of the remaining useful life of aircraft engines. *J Comput Appl Math* 346, 184-191. <https://doi.org/10.1016/j.cam.2018.07.008>
- Salunkhe, T., Jamadar, N.I., Kivade, S.B., 2014. Prediction of Remaining Useful Life of Mechanical Components-A Review 3, 125-135.
- Saxena, A., Goebel, K., Simon, D., Eklund, N., 2008. Damage propagation modeling for aircraft engine run-to-failure simulation. 2008 International Conference on Prognostics and Health Management, PHM 2008. <https://doi.org/10.1109/PHM.2008.4711414>
- Shi, Z., Chehade, A., 2021. A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliab Eng Syst Saf* 205, 107257. <https://doi.org/10.1016/j.ress.2020.107257>
- Yuan, M., Wu, Y., Dong, S., Lin, L., Liu, Y., 2018. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* 275, 167-179. <https://doi.org/10.1016/j.neucom.2017.05.063>
- Zhang, Y., Xin, Y., Liu, Z. wei, Chi, M., Ma, G., 2022. Health status assessment and remaining useful life prediction of aero-engine based on BiGRU and MMoE. *Reliab Eng Syst Saf* 220. <https://doi.org/10.1016/j.ress.2021.108263>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2019. Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process* 115, 213-237. <https://doi.org/10.1016/j.ymssp.2018.05.050>

Zhao, Z., bin Liang, Wang, X., Lu, W., 2017. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliab Eng Syst Saf* 164, 74–83. <https://doi.org/10.1016/j.ress.2017.02.007>

6 CHAPTER SIX –OVERALL DISCUSSION



6.1 Research gaps filled

To fill the research gaps identified in the literature review, a great effort has been made in this project. Details can be found below.

- *Research gap 1: Very little research investigated how the model's components and parameters affect the RUL prediction performance.*

A detailed investigation of the factors influencing RUL prediction performance is presented to fill this gap. The factors that are concerned in this project include the optimiser, activation function, neuron number and sequence length. Many tests have been taken to evaluate the influence of these factors, as can be seen in Chapter 3. Meanwhile, four RNN techniques with different layer numbers are evaluated as well. Moreover, the performance of various feature engineering methods is also tested and assessed, as demonstrated in Chapter 4.

- *Research gap 2: More research should focus on the adaptation ability of algorithms in the real-world industrial environment Contribution to the knowledge.*

In the field of RUL prediction, the adaptation ability can be evaluated in two aspects: (1) can the model deal with the data collected from the system that works under multiple working conditions, and (2) can the trained model from samples with no noise achieve high accuracy when testing on noisy samples? To cope with the first challenge, an operation-based normalisation method is presented in Chapter 4. For the second challenge, a three-stage feature selection method is developed and utilised to reduce noise in the data in Chapter 5. Then, to improve the prediction accuracy, a new RUL target function is introduced, and a novel multi-scale remaining useful life prediction approach is developed and tested on the public dataset in Chapter 5.

6.2 Contribution to the knowledge

The contribution of this research can be categorised into three parts: the exploration of the RUL prediction approaches, the optimisation of the DL-based model for RUL prediction, and the construction of a novel prognostic framework.

6.2.1 The exploration of the RUL prediction approaches

This research carried out a detailed literature review on the fundamental knowledge of RUL prediction approaches. The main branches of the review work are listed below.

- The definition of the RUL varies based on its application scenario. A summary of these definitions in literature has been presented. (See in 2.1)
- This research also summed up different category ways of the RUL prediction approaches. (See in 2.1.1)
- The RUL prediction approaches are classified into model-based (physics-based), data-driven based and hybrid-based in this research. The advantages and drawbacks of these approaches are listed. (See in Table 2-1)
- A literature review is laid on the deep learning approaches focusing on the most state-of-the-art research work. (See in CHAPTER TWO—LITERATURE REVIEW)
- Finding that RNNs are effective at dealing with time series data, this research reviews RNN and its variants for RUL prediction. (See in CHAPTER THREE—PARAMETER OPTIMISATION)

6.2.2 The optimisation of the DL-based model for RUL prediction

To achieve a more accurate RUL prediction, many tests have been done to evaluate the factors that have an influence on the prediction performance. These factors are categorised into three different fields, as listed below.

- Four DL-based models with different layers are tested and evaluated, including RNN, LSTM, GRU, and Bi-LSTM. (See in Figure 3-10)
- A specific study on how different components (types of optimisers and activation functions) or parameters (sequence length, neuron quantities) affect the models' performance on RUL prediction has been carried out. (See in 3.5)
- Three different normalisation techniques are evaluated, and an operation-based normalisation is developed to cope with the dataset with different operating conditions. (See in Table 4-3, Figure 4-7)

6.2.3 The construction of a novel prognostic framework

A novel prognostic framework has been developed in this research. The selection of the model parameters is based on the optimisation results. The contributions of the novel prognostic framework are listed below.

- An operation-based normalisation is applied to find the actual degradation trend hidden in raw data.
- A three-stage feature selection approach is utilised to obtain the most valuable features for RUL prediction. Two correlation analysis methods are used to eliminate the irrelevant and redundant input variables to further reduce the computational cost and avoid the over-fitting problem. (See in CHAPTER FOUR—FEATURE SELECTION)
- A three-stage RUL target function is proposed, which includes the constant stage, the transition stage, where the system degrades nonlinearly with usage, and the linear degradation stage. (See in Figure 5-4)
- A two-step correlation-based method is introduced to determine the two thresholds for the three-stage RUL target function. (See in 5.3.4.1)
- A multi-scale RUL prediction approach is developed, which includes a two-step LSTM model for 3-stage classification (low scale), and an LSTM-based high-scale RUL prediction. (See in Figure 5-3)

6.3 Conclusions to the research findings

The key findings of the research are listed below.

1. It has been observed that DL-based techniques have attracted increasing interest in RUL prediction in recent years. These techniques are capable of providing more agility to process data associated with highly nonlinear and complex feature abstraction through a cascade of multiple layers and therefore suggest a promising future in this area. (See in CHAPTER TWO—LITERATURE REVIEW)

2. With the ability to discover the variation pattern underlying time series, RNN gains increasing attention in RUL prediction. (See in CHAPTER TWO—LITERATURE REVIEW)
3. Some of the network structures are very parameter sensitive. Gradient exploring or gradient vanishing problems may occur when using certain sets of parameters. The influence of these parameters on the performance of RUL prediction is different according to different network structures. (See in CHAPTER THREE—PARAMETER OPTIMISATION)
4. The operation-based normalisation approach can find the actual degradation trend hidden in the raw signal. (See in CHAPTER FOUR—FEATURE SELECTION)
5. The operation-based normalisation improves the correlation between the input variables and the output and thereby be able to help the model to achieve higher prediction accuracy. (See in CHAPTER FOUR—FEATURE SELECTION)
6. The proposed three-stage feature selection approach can consistently improve the performance of popular RNN variants and state-of-the-art for RUL prediction. An improvement of 6.84% has been achieved for all RNN-based models, and an improvement of 7.3% has been achieved compared to other state-of-the-art RUL prediction approaches. (See in CHAPTER FOUR—FEATURE SELECTION)
7. As the degradation process varies from engine to engine, the thresholds of all four datasets should be calculated separately. Therefore, the proposed two-step correlation-based approach to finding the threshold for the proposed three-stage RUL target function is more reasonable than adopting a fixed threshold, as in some research. (See in CHAPTER FIVE—PROGNOSIC MODEL)
8. A significant improvement can be observed in the RUL prediction performance using the proposed framework on the datasets FD001, FD002 and FD004. The improvement over the best existing method for these three datasets is 40%, 22% and 4%, respectively. For the dataset FD003, it is also better than most of the other methods and remarkably close to the best existing method. (See in CHAPTER FIVE—PROGNOSIC MODEL)

6.4 Limitations and Future work

Although many accomplishments have been achieved in this research, there are still many limitations, which along with some future works addressed below.

- Because of the Big Data and IoT, most of the recent research on RUL prediction is focused on data-driven approaches. Additional research could be focused on combining DL approaches with physics-based approaches because these hybrid approaches have great potential and the chance to provide a more effective and precise RUL prediction.
- The optimisation is based on the testing of every feature engineering method and prognostic parameter, which could be very time-consuming and can not necessarily present the most accurate RUL prediction performance. Therefore, future work will focus on the design of new implementation algorithms to speed up the training progress and achieve better prediction results
- Currently, the model's parameters selection is based on the optimisation result achieved in Chapter 3. An automatic optimisation approach could be developed for each specific model.
- The 3-stage RUL prediction approach can achieve outstanding prediction performance based on the assumption that the data volume is large enough. The data volume will limit the selection of the sequence length, which can significantly affect the prediction results. Therefore, data enhancement techniques can be taken into consideration in the future.
- The proposed framework mainly focuses on optimising the feature engineering method and prognostic parameter and separating the three degradation stages. Not much innovation in designing novel network structures has been involved in this research. Therefore, more research work could focus on the innovation of designing of new network structures.

The proposed framework only adopts the basic LSTM network. The More advanced network could be involved in future

