Deep Learning for Automobile Predictive Maintenance under Industry 4.0

Chong Chen

School of Engineering
Cardiff University

A thesis submitted to Cardiff University
for the degree of Doctor of Philosophy

October 2020

Acknowledgements

Firstly, I would like to express my sincere gratitude to my main supervisors Dr. Ying Liu for his knowledge, continuous support, patience, criticism, and encouragement throughout my PhD research. Under Dr. Liu's supervision, I have learnt how to think critically and carry out my research with rigour. I could not have asked for a better supervisor to train and mentor me.

My sincere thanks also go to my co-supervisor Dr. Xianfang Sun for his academic guidance and support throughout this process. Many thanks to Dr. Jian Qin, Dr. Shixuan Wang, the colleagues in our groups, and Miss Zhi Xin Wong, for their help and encouragement.

Finally, I am thankful to my parents for their mental and financial support for my PhD research.

Abstract

Industry 4.0 refers to the fourth industrial revolution, which has boosted the development of the world. An important target of Industry 4.0 is to maximize the asset uptime so to improve productivity and reduce the production and maintenance cost. The emerging techniques such as artificial intelligence (AI), industrial Internet of things (IIoT) and cyber-physical system (CPS) have accelerated the development of data-orientated application such as predictive maintenance (PdM). Maintenance is a big concern for an automobile fleet management company. An accurate maintenance prediction can be helpful to avoid critical failure and avoid further loss. Deep learning is a type of prevailing machine learning algorithm which has been widely used in big data analytics. However, how to establish a maintenance prediction model based on historical maintenance data using deep learning has not been investigated. Moreover, it is worthwhile to study how to build a prediction model when the labelled data is insufficient. Furthermore, surrounding factors which may impact automobile lifecycle have not been concerned in the state-of-the-art. Hence, this thesis will focus on how to pave the way for automobile PdM under Industry 4.0.

This research is structured according to four themes. Firstly, different from the conventional PdM research that only focuses on modelling based on sensor data or historical maintenance data, a framework for automobile PdM based on multi-source data is proposed. The proposed framework aims at automobile TBF modelling, prediction, and decision support based on the multi-source data. There are five layers designed in this framework, which are data collection, cloud data transmission and storage, data mapping, pre-processing and integration, deep learning for automobile TBF modelling, and decision support for PdM. This framework covers the entire knowledge discovery process from data collection to decision support.

Secondly, one of the purposes of this thesis is to establish a Time-Between-Failure (TBF) prediction model through a data-driven approach. An accurate automobile TBF prediction can bring tangible benefits to a fleet management company. Different from the existing studies that adopted sensor data for failure time prediction, a new approach called Cox proportional hazard deep learning (CoxPHDL) is proposed based on the historical maintenance data for TBF modelling and prediction. CoxPHDL is able to tackle the data sparsity and data censoring issues that are common in the analysis of historical maintenance data. Firstly, an autoencoder is adopted to convert the nominal data into a robust representation. Secondly, a Cox PHM is researched to estimate the TBF of the censored data. A long-short-term memory (LSTM) network is then established to train the TBF prediction model based on the pre-processed maintenance data. Experimental results have demonstrated the merits of the proposed approach.

Thirdly, a large amount of labelled data is one of the critical factors to the satisfactory algorithm performance of deep learning. However, labelled data is expensive to collect in the real world. In order to build a TBF prediction model using deep learning when the labelled data is limited, a new semi-supervised learning algorithm called deep learning embedded semi-supervised learning (DLeSSL) is proposed. Based on DLeSSL, unlabelled data can be estimated using a semi-supervised learning approach that has a deep learning technique embedded so to expand the labelled dataset. Results derived using the proposed method reveal that deep learning (DLeSSL based) outperforms the benchmarking algorithms when the labelled data is limited. In addition, different from existing studies, the effect on algorithm performance due to the size of labelled data and unlabelled data is reported to offer insights for the deployment of DLeSSL.

Finally, automobile lifecycle can be impacted by surrounding factors such as weather, traffic, and terrain. The data contains these factors can be collected and processed via geographical information system (GIS). To introduce these GIS data into automobile TBF modelling, an integrated approach is proposed. This is the first time that the surrounding factors are considered in the study of automobile TBF modelling. Meanwhile, in order to build a TBF prediction model based on multi-source data, a new deep learning architecture called merged-LSTM (M-LSTM) network is designed.

Experimental results derived using the proposed approach and M-LSTM network reveal the impacts of the GIS factors.

This thesis aims to research automobile PdM using deep learning, which provides a feasibility study for achieving Industry 4.0. As such, it offers great potential as a route to achieving a more profitable, efficient, and sustainable fleet management.

Research Achievements

Journal papers:

1. **Chen, C**., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S. and Syntetos, A.A., 2020. Predictive maintenance using cox proportional hazard deep learning. Advanced Engineering Informatics, 44, p.101054. 10.1016/j.aei.2020.101054 **(Impact factor: 3.879)**

2. **Chen, C**., Liu, Y., Kumar, M., Qin, J. and Ren, Y., 2019. Energy consumption modelling using deep learning embedded semi-supervised learning. Computers & Industrial Engineering, 135, pp.757-765. 10.1016/j.cie.2019.06.052 **(Impact factor: 4.135)**

3. Liang, Y., Liu, Y., **Chen, C**. and Jiang, Z., 2018. Extracting topic-sensitive content from textual documents—A hybrid topic model approach. Engineering Applications of Artificial Intelligence, 70, pp.81-91. 10.1016/j.engappai.2017.12.010 **(Impact factor: 4.201)**

4. **Chen, C.**, Liu, Y., Sun, X., Cairano-Gilfedder, D., Titmus, S. An Integrated Deep Learning Based Approach for Automobile Maintenance Prediction with GIS Data. Reliability Engineering & System Safety. **(Impact factor: 5.040) (Submitted)**

Conference papers:

1. **Chen, C.**, Liu, Y., Sun, X., Cairano-Gilfedder, D., Titmus, S. Automobile maintenance modelling using gcForest. In 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE) 10.1109/CASE48305.2020.9216745

2. **Chen, C.**, Liu, Y., Sun, X., Cairano-Gilfedder, D., Titmus, S. Automobile maintenance prediction using deep learning with GIS data. In 52nd CIRP CMS, Procedia CIRP 81 (2019). 10.1016/j.procir.2019.03.077

3. **Chen, C**., Liu, Y., Sun, X., Wang, S., Cairano-Gilfedder, D., Titmus, S. and Syntetos, A.A., 2018, August. Reliability analysis using deep learning. In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers Digital Collection. 10.1115/DETC2018-86172

4. **Chen, C**., Liu, Y., Kumar, M. and Qin, J., 2018. Energy consumption modelling using deep learning technique—a case study of EAF. In 51st CIRP CMS, Procedia CIRP, 72, pp.1063-1068. 10.1016/j.procir.2018.03.095

Table of Contents

List of Tables

List of Figures

List of Symbols

a_i: Actual values of the data label

$\bar{a}$: The average actual value

b_i: The output of the LSTM memory cell

b_j: The bias in fully connected layer

b_ι: The factors of input gate of the LSTM memory cell

b_Φ: The factors of forget gate of the LSTM memory cell

b_ω: The factors of the output gate of the LSTM memory cell

β_p: The covariate of Cox PHM

c_i: The input of the LSTM memory cell

$f()$: The activation function in a fully connected layer

$h_0(t)$: The maximum likelihood estimator of Cox PHM

p_i: Predicted values of algorithm

$\bar{p}$: The average of the predicted value

ΔR: The difference in reliability of Cox PHDL

S_j: The input of the activation function in a fully connected layer

ΔT: The difference of TBF in Cox PHDL

x_n: The input for a neuron in a fully connected layer

X_p: The input for Cox PHM

I: The total number of labelled data

J: The total number of unlabelled data

l_i: Instance in the labelled dataset $l_i \in \{l_1, \dots, l_I\}$

y_i: The labels of the instances in the labelled dataset, $y_i \in \{y_1, \dots, y_I\}$

u_j: Instance in the unlabelled dataset $u_j \in \{u_1, \dots, u_J\}$

$\hat{l}_i$: The most similar l_i for u_j, $\hat{l}_i \in \{l_1, \dots, l_I\}$

y_j: The new label of u_j, $y_j \in \{y_1, \dots, y_J\}$

Δy_j: Bias between new label y_j and the actual label of u_j, $\Delta y_j \in \{\Delta y_1, \dots, \Delta y_J\}$

Δu_j: Difference of $\hat{l}_i$ and u_j, $\Delta u_j \in \{\Delta u_1, \dots, \Delta u_J\}$

Δl_i: Difference of each two l_i, $\Delta l_i \in \{\Delta l_1, \dots, \Delta\ l_N\}$, $N = \frac{I(I-1)}{2}$

Δy_i: Difference of each two y_i, $\Delta y_i \in \{\Delta y_1, \dots, \Delta\ y_N\}$, $N = \frac{I(I-1)}{2}$-

$\hat{y}_j$: The compensated label of u_j $\hat{y}_j \in \{\hat{y}_1, \dots \hat{y}_J\}$

w_{jn}: The unique weight of a neuron in the fully connected layer

List of Abbreviation

AI: Artificial intelligence

CNN: Convolutional neural network

Cox PHDL: Cox proportional hazard deep learning

Cox PHM: Cox Proportional Hazard Model

CPS: Cyber-physical system

DCNN: Deep convolutional neural network

DLeSSL: Deep learning embedded semi-supervised learning

FCNN: Fully connected neural network

GIS: Geographical information system

GPS: Global positioning system

GRU: Gate recurrent unit

IIoT: Industrial Internet of things

LSTM: Long-short-term memory

M-LSTM: Merged-LSTM

MCC: Model correlation coefficient

MLP: Multilayer perceptron

PdM: Predictive maintenance

PtP: Peak to peak

RBF: Radial basis function

RNN: Recurrent neural network

RMSE: Root mean square error

RUL: Remaining useful life

StD: Standard deviation

SVM: Support vector machine

TBF: Time-between-failures

Chapter 1 Introduction

1.1 Background

Industry 4.0 stands for the fourth industrial revolution, which is bringing sweeping change to industries worldwide. The digitisation and intelligentisation are two key parts in Industry 4.0, which have generated meaningful impact in different parts of the industry (Behrendt et al., 2017, Vaidya et al., 2018). Under this context, new technologies such as AI, IIoT and CPS have accelerated the development of the industry (Rauch et al., 2020). AI is a cognitive science with the focuses in the areas of, image processing, natural language processing and robotics and so on. It has become one of the core technologies in Industry 4.0, which have gained increasing attention from both industry and academia (Lee et al., 2018). IIoT refers to interconnected sensors, instruments, and other devices connected with computers' industrial applications, which has greatly promoted the information process in real-time data collection (Boyes et al., 2018). With the deployment of IIoT, a large amount of industrial data is generated and collected, which greatly promotes the information process in monitoring, tracking, and interaction. The data collected from IIoT requires big data analytics techniques to build an integrated environment which can offer a transparent view of the production process and can be helpful in process control and management (Kang et al., 2016a). CPS is the integrations of computation, networking, and physical processes, which has been used to implement the efficient management of industrial big data collected from IIoT and implement analytics using AI techniques

(Zhang et al., 2018d). These techniques can be beneficial to different aspects of the industry, such as asset lifecycle management.

Asset is important to the operation in the industry. Asset failure leads to great losses in revenues and productivity. Taking automotive manufacturing as an example, every minute of unplanned downtime can cost them as much as $15,000 - $20,000 and a single downtime event can cost approximately $2 Million (Cloudera, 2017). Since the business impact of the unexpected failure is significant, it is not surprising that the asset lifecycle management has become a key concern in modern industry. Big data has prevailed under the context of Industry 4.0. The analytic for industrial big data is a key concern for the industry which can bring valuable insights for decision-making. Industrial big data is labelled with '5V' characteristics, which are volume, velocity, variety, veracity, value (Erl et al., 2016). These characteristics impose challenges to process industrial big data via traditional data processing techniques. The big data collected from the industry contains various information which is about the process and event from the assets. These data can be relevant to asset's lifecycle. With the advancement of data analytics, it is possible to get insights to achieve better maintenance management (Sezer et al., 2018, Chen et al., 2020).

Besides digitisation and intelligentisation, sustainability is another important topic in Industry 4.0 (Stock and Seliger, 2016). The role of asset maintenance is crucial in sustainable manufacturing (Rødseth and Schjølberg, 2016). If an asset breaks down seriously, some of the critical parts may need to be replaced or major repaired. The replacement and major repair will lead to high cost, extra energy consumption and low resource-efficiency, which have a negative impact on the environment. Moreover, critical failure can shorten the overall lifespan of the asset (Wang et al., 2014), which accelerate its pace to be scrapped or recycled. Achieving better asset lifecycle management can not only lower the economic loss and increase productivity, but it can promote the sustainability of the industry.

Maintenance is critical as it is highly relevant to asset lifespan. The useful life of a system can be extended with the implementation of PdM. Figure 1.1 compares the health status change under the strategy of PdM and passive maintenance. Passive maintenance is also known as run-to-failure. The maintenance is implemented when the health status of an asset decreases to 0. Then the downtime for the asset will be extremely long. In strike contrast, with the help of PdM, the occurrence of failure and then implement maintenance can be predicted in the early stage of failure, which can dramatically shorten the asset downtime (Patil et al., 2017). With an accurate prediction of equipment failure time, maintenance can be scheduled beforehand so to decrease the probability of accidents, economic losses, and human casualty. Nowadays, PdM has been widely applied in different industries such as automobile (Prytz et al., 2015), aircraft (Aremu et al., 2019), manufacturing (Baruah and Chinnam, 2005). The implementation of PdM can help a company to increase asset availability by 5% to 15%, reduce maintenance costs by 18% to 25%, and reduce machine downtime by 30% to 50% (Behrendt et al., 2017). The automobile is a type of expensive asset. Automobile fleet management is the management of a business's cars and vans, which is important to the logistics.

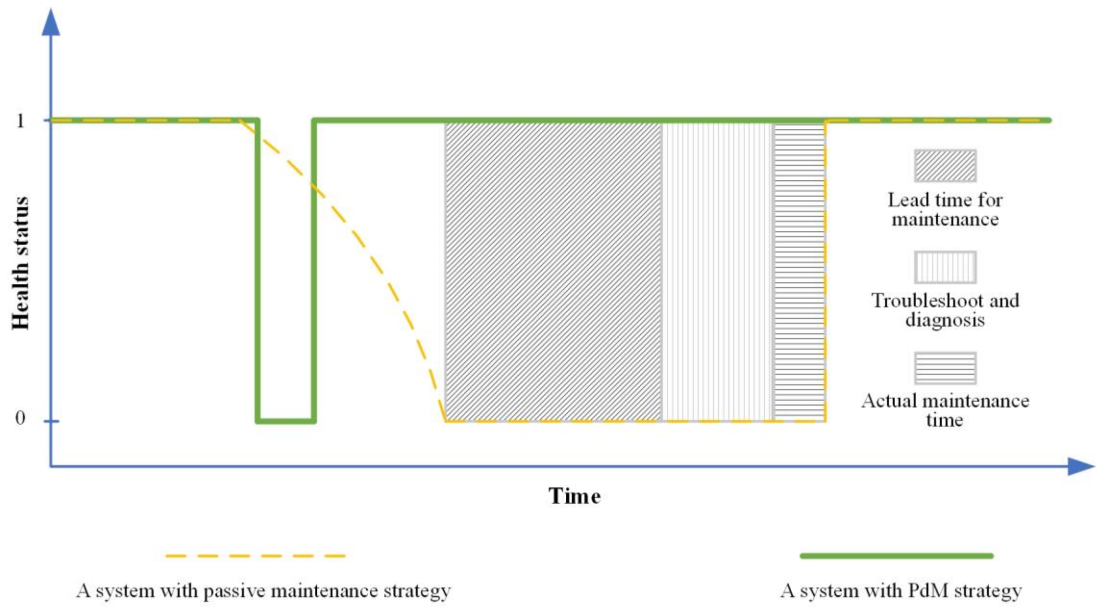

Figure 1.1 The comparison of passive maintenance and PdM

For an automobile fleet management company, the maintenance of the possessed automobile is a significant concern because it is directly relevant to the maintenance planning, job scheduling and spare parts management (Chen et al., 2018). The asset's pending failures can be detected and failure time can be predicted in advance using data analytic tools such as defined health factors, statistical inference methods, and engineering approaches (Susto et al., 2015). In PdM, it is essential to predict the next failure time accurately. If maintenance is implemented too early in advance, the benefits of more extended usage are lost. In contrast, if it is carried out too late, the asset may fail and result in a larger loss (Allah Bukhsh et al., 2019). If the upcoming failure of an automobile can be predicted, maintenance can be scheduled in advanced to avoid critical failure, which can bring tangible benefits to the fleet management company. Hence, achieving better prediction accuracy is one of the principal research targets in the era of Industry 4.0.

1.2 Motivations

The maintenance of the automobile is a major concern for fleet management companies. If the engine of an automobile fails when it is running, it might cause the accident and economy loss (Zhang and Liu, 2002). Fleet management companies need to take better maintenance management to ensure an automobile's health status. There are two main types of maintenance strategies widely deployed in fleet management, which are run-to-failure and preventive maintenance (Mobley, 2002). Run-to-failure is a passive management technique, where maintenance is not carried out until failure occurs. Preventive maintenance is deemed as a time-driven maintenance strategy. With the deployment of preventive maintenance, an automobile takes a scheduled check after a certain period (Mobley, 2002). Apparently, run-to-failure management cannot lower maintenance cost. As for preventive maintenance, a critical issue part is that the scheduled check period is challenging to determine. If it is scheduled too frequently, the maintenance cost will increase, and a part of automobile usage will be sacrificed. However, if it is scheduled less frequently, an accident will happen (Cresci et al., 2017). Different from preventive maintenance and run-to-failure, PdM is a proactive maintenance strategy. The core of PdM is to predict the next failure time of equipment in order to maximize the service life of equipment without increasing the risk of failure. If the maintenance can be scheduled before a critical failure occurs, it can lower the maintenance cost and downtime. The prediction of the next failure of an automobile can bring tangible benefits to maintenance management.

Recently, data-driven approaches have been widely explored in PdM. Among these studies, condition-based PdM has gained increasing attention. With the development of the IIoT and CPS, an increasing amount of data relevant to the health condition of equipment can be collected with the aim to implement PdM (Edwards et al., 2017). Real-time sensor data offers dynamic information on the asset's health condition, which can be used for the asset remaining useful life (RUL) modelling (Civerchia et al., 2017, Nuñez and Borsato, 2018). However, the deployment of IIoT requires extra cost, which cannot be afforded by a part of industrial companies.

For an automobile fleet management company, telematics data that collected from IIoT such as speed and load can be obtained and stored in the intelligent device of an automobile. However, it is challenging to transmit these data to the cloud for further modelling. Two main reasons make this issue challenging. Firstly, the real-time telemetric data is high in volume, which results in heavy data transmission burden. Secondly, automobile as an asset working in a large area, its data connection to the cloud can be unstable. Because of that, realising real-time monitoring and condition-based PdM for an automobile is challenging to deploy in the actual fleet management. Besides modelling asset RUL based on sensor data, another type of data called historical maintenance data can also be used for automobile failure time prediction. Historical maintenance data is the automobile maintenance records collected from a garage. Different from the collection of sensor data which requires extra cost, the collection of historical maintenance data is relatively easy. The PdM studies based on sensor data aims at RUL modelling, while the studies based on historical maintenance data can be used for automobile TBF modelling. Under the scenario that sensor data tends to be challenging to obtain due to the extra cost, exploring automobile TBF modelling can be another way for the next failure prediction of the automobile.

Moreover, the existing PdM research only considers modelling based on single data sources such as sensor data or historical maintenance data. However, both types of data can be used in automobile failure prediction. With the integration of historical maintenance data, the needed volume and features from sensor data can be decreased, which can lower the barrier towards condition-based PdM. Hence, there is a need to study the PdM based on historical maintenance data under the context of Industry 4.0.

Recently, deep learning has been prevailing in big data analysis (LeCun et al., 2015). It was originated from the multilayer perceptron (MLP), and the prevailing structures such as convolutional neural network (CNN), recurrent neural network (RNN) and autoencoder has been widely used in the data analytics for industrial big data (Wang et al., 2018b). As a group of machine learning algorithms, deep learning is good at learning the hidden patterns within data (LeCun et al., 2015) and is prevailing in the multi-source data integration (Gao et al., 2020). In PdM, deep learning has been investigated in recent years (Carvalho et al., 2019), which mainly focused on the condition-based PdM. Since the historical maintenance data also can be high in volume and complex in composition, deep learning can be a useful tool in modelling based on historical maintenance data.

1.3 Research Questions and Objectives

Following the background and motivations, this research aims to investigate automobile PdM using deep learning under the context of Industry 4.0. To achieve the aim of this research, the following research questions have been formulated:

1. *The emerging technologies such as IIoT, CPS and AI have greatly boosted the development of the industry. Under this context, what is a suitable framework for automobile PdM based on the understanding of Industry 4.0?*

2. *The prediction of TBF is important in automobile PdM. Since deep learning is a prevailing tool in big data analysis and it has been widely used in the RUL prediction. However, TBF modelling is also important in automobile PdM. How*

can deep learning be used in automobile TBF modelling based on historical maintenance data?

3. *Deep learning is a type of data 'hunger' algorithm. When the labelled data is insufficient, how can the performance of deep learning not be significantly jeopardised in the study of automobile TBF modelling?*

4. *It is understood that the automobile lifecycle is relevant to the surrounding factors such as weather, traffic, and terrain. However, due to the heterogeneous nature of these data, how to integrate them with historical maintenance data for automobile TBF modelling?*

With the identification of the research questions, the research objectives following these research questions are listed below:

1. *To propose a framework for automobile TBF modelling, prediction, and decision support based on the multi-source data.*

2. *To investigate an approach that integrates deep learning and reliability analysis for automobile TBF modelling based on historical maintenance data.*

3. *To propose a semi-supervised learning algorithm that can enable deep learning to be effective when the labelled data is insufficient.*

4. *To study an approach to integrate the surrounding data collected from GIS into automobile TBF modelling.*

The details of these research will be reported in Chapters 3, 4, 5 and 6.

1.4 Thesis Outline

Chapter 1 aims to provide broader contexts and background as to the research motivation and significance in this thesis.

Chapter 2 provides a comprehensive literature review of the existing body of literature. It is divided into four main parts: (1) the condition-based PdM and the statistical-based PdM; (2) several prevailing deep learning algorithms and the application of deep learning in the industry, (3) semi-supervised learning and its application in industry, and (4) the studies of GIS using machine learning.

In **Chapter 3**, a framework is proposed for automobile TBF modelling, prediction, and decision support based on the industrial big data. There are five layers designed in this framework, which are data collection, cloud data transmission and storage, data mapping, pre-processing and integration, deep learning for automobile TBF modelling and decision support for PdM.

Chapter 4 reports a new approach called CoxPHDL, which can tackle the issues of data sparsity and data censoring that are common in the analysis of historical maintenance data. In this integrated approach, an autoencoder is adopted to convert the nominal data into a robust representation. Then, a Cox PHM is researched to estimate the TBF of the censored data. Finally, with the consideration of sequential patterns in historical maintenance data, a long-short-term memory (LSTM) network is established to train the TBF prediction model based on the pre-processed data. The experimental results revealed the effectiveness of CoxPHDL.

Chapter 5 aims to propose a semi-supervised learning algorithm for automobile TBF modelling using deep learning called DLeSSL to enable deep learning still can reach decent performance when the labelled data is insufficient. Based on DLeSSL, the label of unlabelled data can be estimated using a semi-supervised learning approach that has a deep learning technique embedded so to expand the labelled dataset. Experimental results derived using the proposed method reveal that deep learning (DLeSSL based) outperforms the deep learning (supervised) and deep learning (label propagation based) when the labelled data is limited. In addition, the effect on performance due to the size of labelled data and unlabelled data is also reported.

In **Chapter 6**, another integrated approach is proposed to collect, pre-process heterogeneous GIS data and then these data are mapped with historical maintenance data. Meanwhile, a deep learning architecture called M-LSTM network is designed for the TBF modelling based on the heterogeneous data. The experimental results show the merits of M-LSTM network and reveal that the introduction of GIS data can be helpful to automobile TBF modelling.

Chapter 7 concludes this thesis and presents the achievement of this thesis. The restrictions and future works are reported. Last but not least, the main research contributions to the body of knowledge resulting from this research are summarised.

1.5 Research contributions

This thesis makes several contributions to the wider body of knowledge.

1. A research framework is important to support the research of automobile PdM. A contribution is made within automobile PdM with the framework to achieve Industry 4.0. This framework focuses on automobile TBF modelling, prediction, and decision support based on industrial big data. It outlines a technical path towards Industry 4.0 levelled automobile PdM.

2. TBF prediction is important to fleet management. Deep learning is a prevailing technique that has been used in PdM, while how to bring deep learning into TBF modelling has not been investigated. In order to establish a TBF prediction model, a data-driven approach is proposed to integrate autoencoder, Cox PHM and LSTM network techniques.

3. Deep learning is a type of data ‘hunger’ algorithm. Without a large number of labelled data, it would be challenging for deep learning to obtain satisfactory algorithm performance. Owing to the fact that the collection of labelled data is expensive, and the unlabelled data is relatively easy to obtain, a novel semi-supervised learning algorithm called DLeSSL is proposed.

4. Since the surrounding factors such as weather, traffic and terrain also can have an impact on automobile lifecycle, while it is challenging to integrate these data into TBF modelling. An approach is proposed to integrate these multi-source data into automobile TBF modelling. In this approach, the GIS data collection, pre-processing, mapping are reported. A new deep learning architecture called M-LSTM is designed for TBF modelling based on the multi-source data.

Chapter 2 Literature Review

2.1 Introduction

This chapter examines the related works and previous relevant research concerning the three main sections, Predictive Maintenance, deep learning, semi-supervised learning, and the studies of GIS using machine learning. Predictive maintenance can be classified as two main groups which are condition-based predictive maintenance and statistical-based predictive maintenance. The relevant studies of both approaches were reviewed in Section 2.2. Several prevailing deep learning algorithms and recent applications of deep learning in the industry were reviewed in Section 2.3. Section 2.4 reviewed two types of prevailing methods in semi-supervised learning and its application in industry. The relevant studies of GIS using machine learning were reviewed in Section 2.5, and Section 2.6 summarises this chapter.

2.2 Predictive Maintenance

2.2.1 Condition-based Predictive Maintenance

With the boosting volume of sensor data, the research of condition-based predictive maintenance has become prevailing in recent years. Machine learning can be a useful tool in the big data analysis for a large amount of sensor data. Machine learning is a subset of artificial intelligence, which is used to learn the hidden patterns in data. Recently, Machine learning model has been widely used in PdM and achieved satisfactory performance.

The studies of PdM using NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset have gained increasing attention in recent years. This dataset consists of hundreds of cases of multivariate sensor data relevant to turbine engine in a run-to-failure manner. Various deep learning algorithms such as deep convolutional neural network (DCNN) (Li et al., 2018), dilated convolution neural network (Xu et al., 2020) and bi-directional LSTM network (Zhang et al., 2018b) were proposed and validated using this dataset. Li et al. (2018) presented a DCNN that only used 1D convolutional layers as hidden layers. The structure of DCNN is simple, while it can achieve satisfactory performance and low computational cost. Zhang et al. (2018b) presented a bi-directional LSTM network for RUL modelling. In their approach, they used a simple FCNN to construct the health index of the aero engine. Yuan et al. (2016) proposed a single layer LSTM network model to predict the RUL and failure probability of aero engine. All the above deep learning structures have only one training path. Al-Dulaimi et al. (2019) proposed a noisy and hybrid LSTM network which contains a CNN path and bi-directional LSTM path, where the inputs of both networks are the same. Right-censored data represents the maintenance was scheduled before an asset completely failed. The maintenance time located arbitrarily before the failure. In order to model the asset degradation, a method called Relative Entropy Weibull-SAX was proposed using HI and HS degradation modelling method for multivariate asset data. A HI of asset can be constructed using relative entropy. The experimental results based on C-MAPSS show that this method was able to represent the health stage of observed engine (Aremu et al., 2019).

Zhao et al. (2017a) proposed a deep belief network (DBN) based method to predict the health condition of bearing in the rotating machine. DBN is a deep learning model with a hierarchical structure, and it consists of multiple stacked Restricted Boltzmann Machines. The proposed model is good at predicting the short-term health condition of bearing, and it does not rely on prognostic expertise. RNN is good at processing time-series data. Malhi et al. (2011) proposed a method based on competitive learning to predict long-term machine health status. The vibration data collected from rolling bearing was first pre-processed using continuous wavelet transform method. The features from raw data and the transformed data were then jointly used as the input of an RNN. As traditional RNN is poor in studying long-term patterns of data, a more advanced algorithm called LSTM network was designed to catch and store both long-term and short-term patterns of data (Hochreiter and Schmidhuber, 1997). Feature extraction is of importance to PdM. In order to reduce the effect of time-lagged correlations on the feature extraction, Zhang et al. (2018c) deployed a curve-registration method to evaluate the time lags among sensors. After the time lags of sensors were adjusted, the data was then used to establish an LSTM network model to predict upcoming failure.

Besides deep learning techniques, other prevailing machine learning algorithms also have been widely investigated in PdM. Wei et al. (2013) proposed a dynamic particle filter-support vector regression (PF-SVR) model to predict system reliability based on time series data. Parameter selection is a critical part of training a support vector regression model. In this study, a particle filter was used to learn measurement sequence of data so to estimate the parameters for SVR. Nieto et al. (2015) proposed a hybrid particle swarm optimisation support vector machine (PSO-SVM) to predict the RUL for aircraft engines. PSO was used to optimise the SVM kernel parameters in the model training process. PSO-SVM does not require the information previous operation stage but only use the returned engine information for modelling, which is of advantage in the application. With the combination of the optimisation algorithm, SVM model can be more robust and applicable in PdM. Lee and Pan (2019) proposed an approach to evaluate the reliability of a complex system. In this approach, discrete-time Markov chains were deployed to predict the health state of the components in a complex system. Then a Bayesian network was used for modelling the reliability of the complex system. Another study on PdM for the multi-component system was proposed by Liang and Parlikad (2020) where a model was developed for the PdM of multi-system multi-components networks. In this model, analytical and numerical techniques were combined to optimize the maintenance policy. Furthermore, a genetic algorithm with the agglomerative mutation was adopted to effectively determine the maintenance policy.

Prytz et al. (2015) proposed a data-driven approach that can predict the upcoming failures of vehicles based on the historical maintenance data and the data collected onboard the vehicles. In this approach, the random forest algorithm was used in classification modelling to identify whether the RUL of vehicles was longer or shorter than the planned interval. In order to model the degradation of manufacturing systems, a hidden Markov model with auto-correlated observation was proposed. The current state of this model depends on both the corresponding hidden system state and the previous observations. Besides, EM (expectation maximum) was adopted to estimate the unknown parameters. Based on the prediction of RUL, an optimised maintenance policy was developed (Chen et al., 2017). Wang et al. (2014) proposed a predictive maintenance approach for a machine in manufacturing with deteriorating quality states, with the consideration of imperfect minor maintenance and perfect major repair. In this approach, a hidden semi-Markov decision process is used to estimate the system state. Then a reinforcement learning algorithm called Q-P learning is used to obtain the optimal policy.

Zhou and Feng (2017) proposed a novel deep learning algorithm called deep forest (gcForest), which is a decision tree ensemble approach that requires fewer parameters and can generate decent algorithm performance in comparison with the deep neural networks. The gcForest also has been investigated in engineering, especially in fault diagnosis. Wang et al. (2018a) proposed an approach that combines a deep Boltzmann machine and a multi-grained forest for fault diagnosis. The deep Boltzmann machine was first used to transform the raw data into a binary representation. Then, the gcForest was deployed for modelling based on the pre-processed data. Wang et al. (2018a) proposed a combined method for fault diagnosis. A feature selection approach based on Spearman's correlation was first deployed to remove redundant features. Then, the *k*-means algorithm was adopted to determine the label of data, which is the degradation of a machine. Finally, a gcForest was used to build a classifier based on the processed data. Liu et al. (2018b) deployed a gcForest for fault recognition of rolling bearing based on the sensor data. Liu et al. (2019) proposed a gcForest-based end-to-end intelligent fault diagnosis method for hydraulic turbine fault diagnosis based on sensor data. In this study, the robustness of gcForest to noise was tested and revealed.

Digital twin is a hot topic in smart manufacturing. Ding et al. (2019) proposed a digital twin approach for modelling the remaining useful life (RUL) of shearer key parts. Firstly, the simulation was implemented based on the real-world data to obtain more data such as position, posture, trajectory, which can be used for qualitative analysis. Then, autoencoder and bi-directional gated recurrent unit were used to construct an RUL prediction model based on the data collected from real-world and quantitative analysis.

A summary for the most recent papers for condition-based PdM is shown in Table 2.1. It can be seen that the most recent research has focused on the PdM of system or sub-system level. Deep learning, as a group of neural network-based algorithms, has been extensively investigated in the area of condition-based PdM.

Table 2.1 A summary for the most recent papers for condition-based PdM

Reference	Data Analytics Techniques	Target
(Li et al., 2018), (Xu et al., 2020) and (Zhang et al., 2018b)	Deep learning	Turbine engine
(Liang and Parlikad, 2020)	genetic algorithm+ agglomerative mutation	Multi-component system
(Lee and Pan, 2019)	discrete-time Markov chains+ Bayesian network	Simulated system and components
(Liu et al., 2019)	gcForest	Hydraulic turbine
(Zhang et al., 2018c)	LSTM network+ curve-registration	power-generation equipment
(Chen et al., 2017)	expectation maximum	manufacturing system
(Prytz et al., 2015)	random forest	Vehicles
(Nieto et al., 2015)	particle swarm optimization+ support vector machine	Engine
(Wang et al., 2014)	Q-P learning	A machine in the manufacturing process
(Wei et al., 2013)	particle filter+ support vector machine	submarine diesel engine, turbochargers, and car engines

2.2.2 Statistical-based Predictive Maintenance

Statistical models have been investigated for decades. In some of the cases above, the datasets used in these studies are typically in small size. When the data size increases, statistical models might lack the ability to learn the hidden patterns in data due to the growing impurity and noise. The parametric model assumes time-to-failure or TBF follows a specific statistical distribution such as Weibull (Xie and Lai, 1996). The parametric models are useful when the sample follows a distribution, and the model parameters need to estimate accurately. However, in the actual cases, the algorithm performance tends to be compromised if the distribution is not specified correctly. Xie and Lai (1996) studied a Weibull model to estimate the lifetime distribution for electrical and mechanical components. The model is based on two Weibull survival functions, and a graphical estimation approach was adopted to estimate the parameters for the Weibull model. Mettas (2000) proposed a versatile accelerate failure time model to investigate the accelerate life data. The algorithm combines the life-stress relationships for one or two types of stresses with a model formulated by different distribution such as Weibull and Lognormal. The proposed model can be used to generate the relationship of product life with multi-stresses-types, while traditional AFT models can only generate the relationship of product life with single stresses-type. It can be seen from both cases that the data used for modelling was in small size. The robustness of the models in actual cases needs to be further investigated.

Wang and Shi (2013) presented the estimation of parameters of hazard functions for a class of an exponential family. The record values used in this case followed the exponential distribution. Maximum likelihood and interval estimation were deployed to determine the model parameters. Furthermore, symmetric and asymmetric loss functions were used to obtain the Bayes estimators of reliability performance. Zimmer et al. (1998) proposed Burr XII distribution, which is similar to log-normal distribution, for reliability analysis. The Burr XII distribution can effectively simplify the computation of the likelihood of censored data. Moreover, it also shows merits in representing failure data since the distribution has algebraic tails. Saldanha et al. (2001) presented an application of the non-homogenous Poisson point process to the study of the rates of occurrence of failures for the repairable component. In this approach, maximum likelihood parameter estimation and linear regression analysis are deployed. It was deemed as a useful tool to evaluate the maintenance policy for repairable components.

Besides parametric models, another type of statistical model is semi-parametric. Cox PHM and its variants have been widely used in PdM due to its flexibility and ability in modelling based on uncensored and censored data (Anderson and Senthilselvan, 1982, Kay, 1977, Kumar and Klefsjö, 1994). It was used to analyse the relationship between time-independent covariates and hazard function (Cox, 1992). As the standard Cox PHM is only suitable the time-independent covariates, researchers have proposed the variants of Cox PHM to consider the time-dependent covariates Anderson and Senthilselvan (1982) proposed a two-steps PHM for the time-dependent coefficient. The proposed approach allows the varying covariates to be used in PHM. Conditional likelihood estimation was used to determine the regression covariates in this study. The two-steps PHM showed better performance in a cancer mortality study compared to the performance of standard Cox PHM, while the estimation of its parameters can be further explored.

A proportional intensity model based on the Cox PHM called Prentice, Williams, and Peterson (PWP) was introduced based on the nonhomogeneous Poisson process to deal with recurrent failure event data. A nonhomogeneous Poisson process with power-law intensity function was adopted in this study. The proposed model can achieve better performance based on large data size and increasing failure rate (Landers and Soroudi, 1991). Owing to the fact that Cox PHM requires robust covariates for modelling, while it may not be able to learn the hidden patterns from sparse covariates, Sun et al. (2006) proposed a proportional covariate model (PCM) to tackle the sparsity issue in sensor data. Besides, different from the standard Cox PHM, PCM aims to predict the hazard of a system using covariates caused by the deterioration of a system, which can be used in dynamic system monitoring.

Since the spare part management is a critical issue in the industry, Kian et al. (2019) deployed a mathematical programming model to formulate the failure time of a vessel to optimize the spare part management based on sensor data. Meanwhile, the shortest path dynamic programming formulation is deployed to address the polynomial-time complexity. Cox proportional hazard model is a prevailing model in reliability analysis which can process both censored and uncensored data. Verhagen and De Boer (2018) used both time-independent and time-dependent Cox proportional hazard model to estimate the reliability of aircraft components based on historical operational and maintenance data. In this study, extreme value analysis and maximum difference analysis were adopted to identify operational factors which are relevant to component failure. Since Cox PHM has its limitation in providing the business recommendation straightforwardly, Wanga et al. (2018) combined the Cox PHM and decision tree to a conditional inference tree to conduct reliability analysis.

In this study, raw system log information including time, id and description were extracted and then used to establish an SVM model for failure classification (Patil et al., 2017). Since the faulty data is far less than the normal data in the maintenance dataset, Susto et al. (2015) proposed a Multiple Classifier (MC) predictive maintenance approach to address the labelled unbalanced issues in maintenance datasets that arise in maintenance classification problems. This approach can be deployed in the machine, which carries out the repeated work. It was used to classify the healthy state and faulty state of the machine. Since log data can be beneficial to detect the abnormal event in a largescale network. Kobayashi et al. (2017) proposed a method based on a graph-based algorithm to extract failures and their causes from network system log data. In this method, a graph-based algorithm called PC algorithm is firstly introduced to infer causal structure from event time series efficiently. Then, a data pre- and post-processing methods from a set of log messages was proposed to improve the performance of PC algorithm.

From literature, it is evident that various techniques in statistics and machine learning have been investigated in PdM. In recent years, there is an increasing interest in condition-based PdM. It can also be seen from the existing studies that machine learning, especially deep learning, has been widely employed in PdM in recent years. One of the merits of deep learning is that it can identify essential features and determine model parameters automatically, which is different from other prevailing machine learning algorithms.

2.3 Deep Learning

As a subset of machine learning, deep learning is able to learn the hidden patterns within given data automatically via stacking multiple nonlinear processing layers. According to the literature, the difference between deep learning and conventional machine learning in terms of modelling path was compared and shown in Figure 2.1. It can be seen that data pre-processing is needed for both deep learning and conventional machine learning, while the main difference is the way to process features. Conventional machine learning approaches need efforts on feature engineering such as feature extraction and feature selection. These processes strongly rely on domain knowledge. In strike contrast, deep learning can directly learn the hidden patterns automatically, which is more efficient and effective for the modelling process. However, its drawback is also apparent. The feature abstraction process in deep learning cannot be understood or explained, and therefore it is deemed as a black box. In this section, several main deep learning algorithms are introduced, and the applications of deep learning in the industry are reviewed.

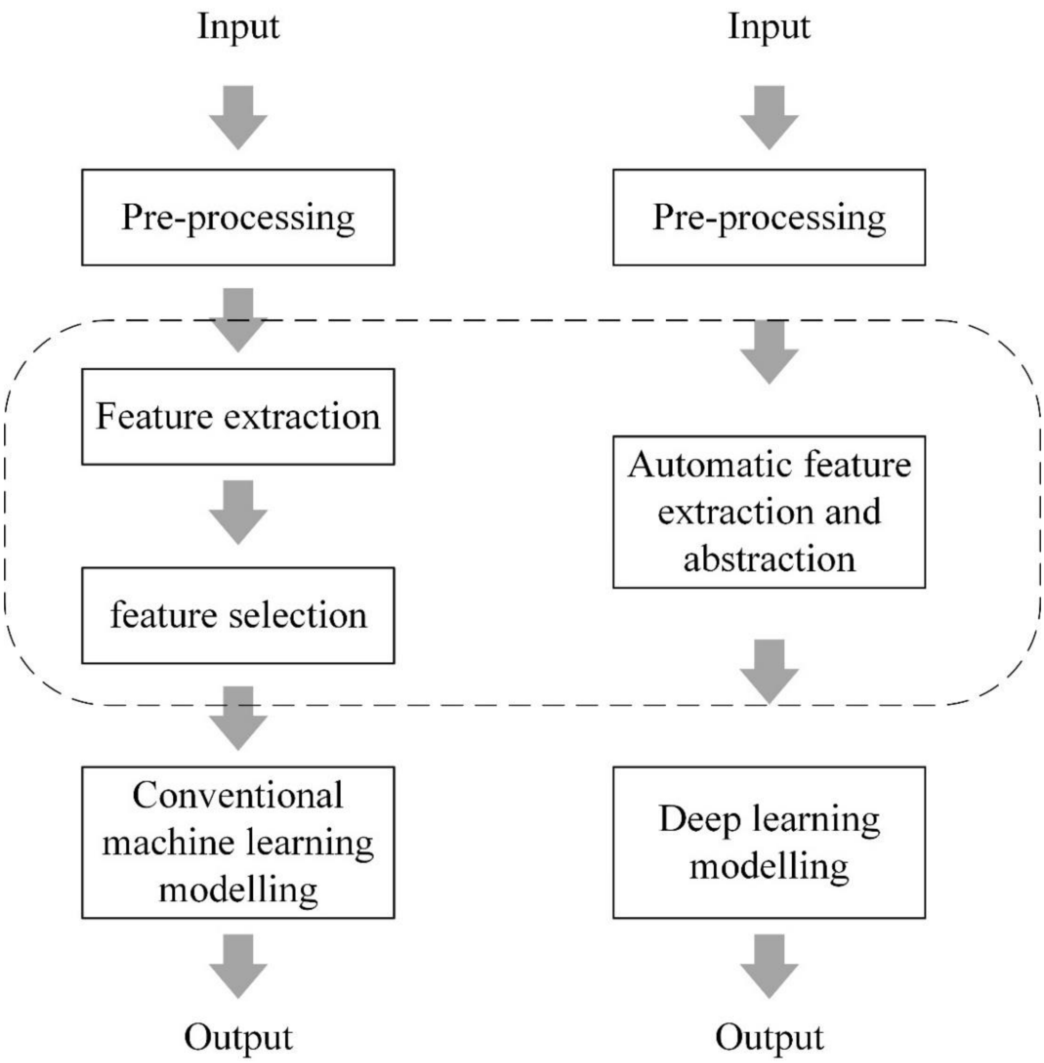

Figure 2.1 The comparison between conventional machine learning and deep learning modelling path

2.3.1 Prevailing Deep Learning Algorithms

- **Fully Connected Neural Network**

The fully connected neural network is the basic type of neural network. It is also known as feedforward neural network or MLP. MLP was proposed by Rumelhart et al. (1986). The simplest MLP has three layers which are input layer, hidden layer and output layer. The input layer is used to process the input data, while the output layer is used to get the result. The hidden layer is used to get the abstract features which are relevant to the output. The gradient explosion and vanish in the back-propagation process leads to the failure of the learning process. Hence, training an MLP with multiple hidden layers is challenging at that time. With the developments of advanced techniques such as Rectified Linear Unit (ReLU), dropout, and optimiser, a deeper neural network becomes achievable. A deeper FCNN has more powerful capability to model the nonlinearity of the input data. Figure 2.2 shows the structure of an FCNN.

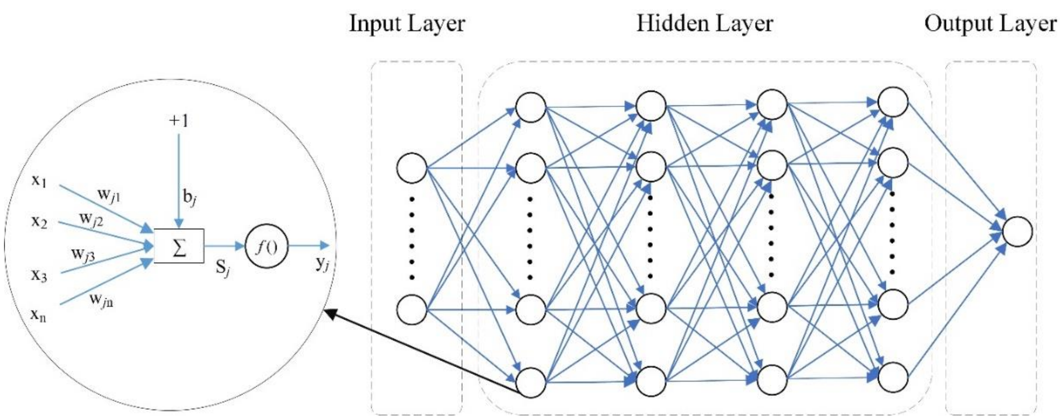

Figure 2.2 The structure of an FCNN

Each layer in an FCNN is composed of different numbers of neurons. Each neuron is a computational unit. The neuron gets the output of the neurons in the previous layer as input (x_1, x_2, ..., x_n). The input will then multiply by the weights (w_{j1}, w_{j2}, ..., w_{jn}) and the products and a bias (b_j) will be added together. The equation is shown below:

$$S_j = \sum_{i=1}^{n} w_{jn} * x_n + b_j \tag{2.1}$$

Then S_j will be sent into the activation function f(). The output y_j from f() is the output of the neuron. The key issue in training the neural network is how to determine the weight (w_{jn}) and bias (b_j) of each neuron. In order to determine the weight and bias, the back-propagation (BP) algorithm will be used. An FCNN is firstly trained feed-forward to get the rough weight and bias. The final output will be compared to the actual value. If the error is large, the model will be trained from the output layer to the input layer in order to fine-tune the weight and bias. Typically, the BP algorithm needs to be implemented for multiple times to fine-tune the weight and bias, which will lead to the long training time (Goodfellow et al., 2016). Besides training the whole network using BP algorithm, a greedy layer-wise training approach was proposed to train a neural network layer by layer, which effectively avoids the issues of gradient explosion and vanish and allows the neural network to be designed with a deep structure (Bengio et al., 2007).

- **Convolutional Neural Network**

CNN is a type of neural network that can process the grid-like structure data such as an image. An image can be deemed as a two-dimensional matrix. CNN has a long history since the 1990s when it was investigated for speech recognition and document reading. After ImageNet (a deep CNN structure) was designed in 2012, CNN has become a mainstream in the area of computer vision. It shows merits in processing a large number of images which contains over a thousand of categories (LeCun et al., 2015). Recently, the variants of CNN, such as ResNet-50 (Xie et al., 2017) and VGG-16 (Simonyan and Zisserman, 2014) have achieved satisfactory performance and have been widely used.

In order to process the grid-like structure data in FCNN, the data needs to be flattened into one-dimension, which will lose part of the patterns in the data. Different from FCNN, CNN adopts the convolution operation to process the two-dimensional data. Convolution process is helpful to the neural network in the following three aspects: sparse interaction, parameter sharing and equivariant representation. Sparse interaction refers to the connection in a convolutional layer is less than fully connected layers with the same neuron size. Parameter sharing indicates that a kernel in a convolutional layer can process the data in different positions of the input, which is also different from the fully connected layer that can only process one position of inputs. Equivariant representation means if the input of a convolutional layer shift, the output will shift in the same way (Goodfellow et al., 2016).

Pooling is another vital component of CNN. In the pooling process, a two-dimension array can be segmented by different grids. A pooling function can downsample a grid with a summary statistic. There are several types of pooling functions such as max pooling and average pooling. Max pooling adopts the maximum values in a grid as the output, while average pooling adopts the mean values of the grid as output. Convolution and pooling operations are used to get the abstract representation within data. Normally, there are more than one convolutional layer and pooling layer in a CNN. After the abstraction and downsampling of the data, a large two dimensions array is transformed into multiple small two dimensions arrays. Then these arrays are flattened and sent to a fully connected layer for further processing (Goodfellow et al., 2016). Figure 2.3 shows the structure of CNN.

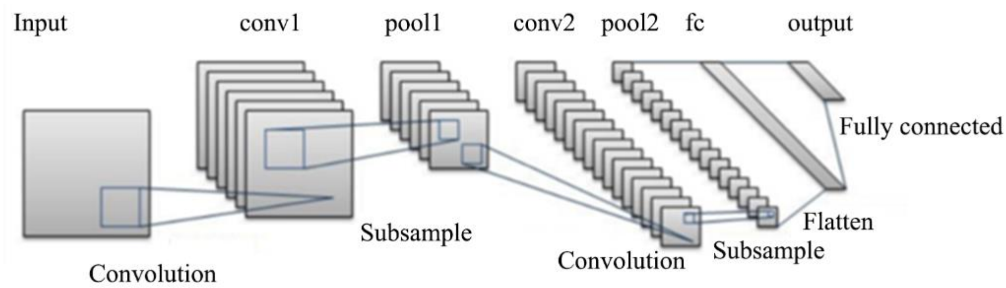

Figure 2.3 The structure of a CNN

- **Recurrent Neural Network**

RNN (Rumelhart et al., 1986) is a prevailing tool to handle the sequential data in deep learning. Since RNN was proposed, it has been widely used in the area of speech recognition and text mining. Different from FCNN that process each instance and update the weights and bias independently, there is a state unit in RNN, which can store the information of the past elements. In other words, the weights in an RNN are shared across different instances of the neurons (Goodfellow et al., 2016).

Figure 2.4 shows the time unfolded graph of an RNN. An RNN is considered as a dynamic system. There are three types of connections in an RNN, which is shown as *U*, *V* and *W* in the figure. These three connections are also the parameters of an RNN. *U* is the connection of input and hidden neuron. *V* is the connection of hidden neurons to output, and *W* is the connection of two hidden neurons. In the left side of the figure, RNN is demonstrated as a circuit. When it is unfolded, we can see the mechanism of RNN. The neuron at time *t* not just gets the input X_t, but also get the memory from time *t-1*. The memory is the output of *S* at time *t-1*. By using this connection method, the RNN is able to learn the sequential patterns in [X_{t-1}, X_t, X_{t+1}] (LeCun et al., 2015).

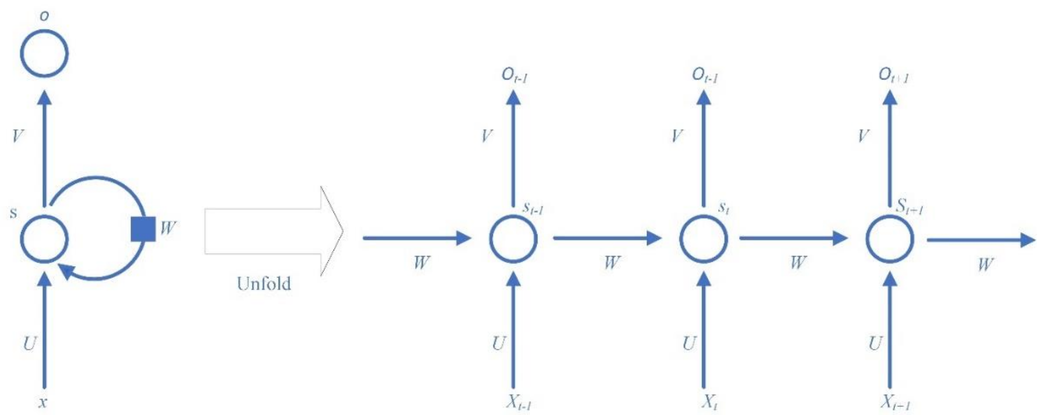

Figure 2.4 The time unfolded graph of an RNN

RNN can be deemed as an FCNN which all the hidden layers share the same weights. RNN was designed to tackle the issue that long-term dependency is hard to learn by a neural network. However, theoretical and empirical evidence indicates that RNN cannot store long-term memory (Bengio et al., 1994). In order to address this issue, long-short term memory (LSTM) network, as a variant of the standard RNN, was proposed (Hochreiter and Schmidhuber, 1997). In comparison with standard RNN, LSTM performs better in long-dependency learning, while it requires higher computational load. After that, another variant of RNN called gate recurrent unit (GRU) network (Cho et al., 2014) was proposed to lower the computational load without sacrifice the algorithm performance in comparison with LSTM network.

- **Autoencoder**

Autoencoder is a deep learning structure for unsupervised learning. The input and output of autoencoder are the same. An autoencoder consists of two parts: encoder and decoder. Normally, the encoder and decoder share the same hidden layer, which is the middle layer of the neural network. The input layer and the first half of the hidden layer constitute the encoder, which aims to compress the input. In order to compress the input to a more dense and complex representation, the number of neurons in the layers of encoder needs to be decreased layer-wise. The dense and complex representation is known as code. In contrast, in order to construct the input data from the code, the decoder needs to be set symmetrically to the encoder. Hence, an autoencoder has a sand clock-like structure, where the output of the bottleneck (i.e. middle layer) is the code (Goodfellow et al., 2016).

Autoencoder can be used for two primary purposes: dimension reduction and denoising. Traditional dimension reduction approaches, such as primary component analysis and linear discriminant analysis, has been widely used in dimension reduction. However, the loss of information is not always negligible. Moreover, feature engineering is always needed in dimension reduction. In contrast, autoencoder can automatically get a robust representation. Autoencoder can detect noise during the training process. That noise shows different patterns with other normal data. The weight of the noise will be very low in the middle layer, and therefore the ratio of noise in code will be lower (Goodfellow et al., 2016).

There are several variants of autoencoder that has been widely used in machine learning fields. The layer of the autoencoder is not limited to a fully connected layer. Convolution layer and recurrent layer can also be used in autoencoder (Ghasedi Dizaji et al., 2017, Gensler et al., 2016), which enables autoencoder to process different types of data such as image and sequential data. Besides, a deep autoencoder tends to obtain more robust representation. Training a deep autoencoder could be challenging due to the gradient explosion and vanish issues. In order to address this problem, stacked autoencoder (Vincent et al., 2010) was proposed, which adopted greedy layer-wise training (Bengio et al., 2007).

2.3.2 The Applications of Deep Learning in Industry

With the dramatic increase deployment of the Internet of things, the data available in the industry has significantly increased. In this context, deep learning as a group of prevailing machine learning algorithms has gained increasing attention due to its capability in handling industrial big data (Wang et al., 2018b). As a particular type of machine learning, deep learning is popularly used to identify objects in images, transcribe speech into text, and select the required data from databases (Schmidhuber, 2015).

Fault defect detection is the main focus of CNN in the industry. CNN is well known for its capability in the image data processing. It is becoming increasingly prevailing in the industry. In order to detect the casting defects using X-ray images, Ferguson et al. (2019) proposed a standardised format for CNN, based on predictive model mark-up language (PMML). Firstly, the pre-trained- Image Net models are converted to PMML format to optimise the distribution of deployment of these models. Then, these models are fine-tuned to get the classification results. Since deep learning is deemed black box due to the low interpretability. Grezmak et al. (2019) bridged the CNN and layer-wise relevance propagation (LRP) algorithm to enhance the interpretability of CNN in machine fault diagnosis. In this approach, CNN is first trained base on the sensor data. The prediction result of CNN and the input signal are then analysed by LRP to enhance the understanding of how CNN identify the linkages between fault types and the input data. Tool wearing is common in manufacturing.

The application of CNN in the industry includes tool wear monitoring, design optimisation for additive manufacturing and human action recognition, etc. In order to predict the status tool wear, Huang et al. (2019) proposed a reshaped time series convolutional neural network (RTSCNN) for multi-sensor data fusion and predicted the wear degree. In this structure, the data collected from multi-sensors are first reshaped into a two-dimension array, which is then passed into the convolutional layer. Williams et al. (2019) investigate the impact of design repository standardisation on the capability of CNN to analyse the geometric data of additive manufacturing. The identification of human action is a key task in human-robot collaboration. Xiong et al. (2020) reported an integrated method based on the optical flow and CNN-based transfer learning to address the issues of low accuracy and robustness of human motions identification and the deficiency of data volume. In this work, the optical flow images which contain the information of human motion are used as the input of a two-stream CNN to predict the human motion. Then transfer learning is introduced to transfer the feature extraction capability from the pre-trained CNN into manufacturing scenarios.

RNN is widely used in processing sensor data in the industry due to its powerful capability in mining the sequential patterns. In all the applications of RNN, LSTM network is the most prevailing RNN structure. In order to model the time-series data in the manufacturing process, Essien and Giannetti (2020) proposed a convolutional LSTM neural network autoencoder structure to learn the hidden representation of time series data. In this structure, convolutional layers are adopted as an encoder, while LSTM layers are used as the decoder. Wind speed forecasting is essential for the energy generation and conversion of the wind power industry. Hu and Chen (2018) presented a combined approach for wind speed forecasting. In this approach, LSTM network and Hysteretic Extreme Learning Machine are combined and used as wind speed prediction.

Meanwhile, the differential evolution algorithm is adopted to optimise the parameters of the LSTM network. In order to accurately simulate the crystal growth process, LSTM network is used to build a thermal field model. Then, the support vector machine algorithm is used to identify model order and lag to determine network input so to improve the algorithm performance of LSTM network Zhang et al. (2019a). Traditional time series forecasting approaches have imposed the challenges of time-consuming and full of complexity. To tackle these issues, a deep LSTM network was proposed to obtain an accurate prediction. The configuration of the proposed deep LSTM network is optimised by genetic algorithm (Sagheer and Kotb, 2019). GRU is another structure of RNN, which is deemed more computationally efficient. Wang et al. (2019) developed an approach that combines deep heterogeneous GRU structure with local feature extraction. Specifically, the local feature extraction method is used to capture the temporal pattern within sequential data. Then the output of local feature extraction is sent to the bi-directional GRU networks with the weighted averaging layer. In the next stage, the output of each GRU networks are processed by fully connected layers. Finally, the data representation is concatenated and passed to an FCNN to get the final prediction. Another study that investigated the GRU network is that a multi-scale dense GRU network was proposed to leverage the feature extraction. In this study, the idea of ensemble learning is also adopted. With the introduction of multi-scale layers and dense layers, the proposed network is able to learn the sequential patterns and ensemble different time-scale patterns (Ren et al., 2019).

Autoencoder also has attracted the researcher's attention in recent years. Sun et al. (2018) presented a deep transfer learning network approach based on autoencoder for machine fault prediction. In this study, three transfer strategies which are weight transfer, hidden feature transfer and weight update, are used to train a sparse autoencoder. The experimental results indicated that the transferred sparse autoencoder can achieve similar performance in comparison with the supervised learning approach when the historical failure data is limited. Another study that combines transfer learning with autoencoder is that Wen et al. (2017) proposed a deep transfer learning method using a sparse autoencoder to extract features. In this method, the maximum mean discrepancy term is used to minimize the discrepancy penalty between the features from training data and testing data. Yu et al. (2019) proposed a stacked denoising autoencoder for the improvement of the performance of process pattern recognition in manufacturing processes. The proposed stacked denoising autoencoder is used to learn the effective features within sensor data, which can be helpful for the fault diagnosis. Anomaly detection is essential to a manufacturing system. Liu et al. (2018a) developed a time delay autoencoder with the structure constructed from the event ordering relationship to detect anomaly in the manufacturing processes. In this approach, the event ordering relationship-based neural network structuring process is adopted to identify the important neuron connections and the weight initialisation of the neural network, which can lower the network complexity and promote the algorithm performance. Besides fault and anomaly detection, autoencoder also has been investigated in bearing RUL prediction. In order to improve the RUL prediction accuracy of deep neural network, (Ren et al., 2018) presented a deep autoencoder based approach that can abstract the features from time and frequency domains. The abstracted features are then sent to a deep neural network jointly with the original time and frequency domains features.

The adoption of deep learning can effectively avoid the complex feature engineering and can be trained in an end-to-end learning manner, which can be implemented by adding layers to map the raw data and data label. With the development of IoT, more and more data such as image and vibration signals are available. In order to handle these industrial big data, deep learning can be a useful tool.

2.4 The Studies of GIS

GIS is a powerful tool which has been widely used in spatial analysis. The knowledge obtained from GIS can be beneficial to decision making (Rikalovic et al., 2014). Recently, researchers have focused on introducing machine learning techniques into GIS. Pham et al. (2017) combined ensemble several methods with multiple perceptron neural Networks to establish a landslide classification model. GIS relevant features such as slope angle, slope aspect, elevation, curvature and plan curvature were adopted for modelling. Aiming to identify the contribution of the features to the landslide, a feature selection method called Relief-F method was used.

Tehrany et al. (2014) proposed a flood susceptibility mapping approach based on the data collected from the records of flood occurrence. The terrain features used in this study included flood inventory, slope, stream power index, topographic wetness index, and altitude, etc. The weight-of-evidence method was applied to measure each relevant factor’s weight. Then, these factors were reclassified using the acquired weights and entered the support vector machine model to evaluate the correlation between flood occurrence and each conditioning factor. Four types of the kernel (linear, polynomial, radial basis function and sigmoid) based SVM was used for modelling. The results indicate the RBF kernel based SVM can achieve the best performance. In order to detect the flash-flood occurrence, Costache et al. (2019) proposed a hybrid approach that combines multilayer perceptron and certainty factor method to predict the distributions of torrential valleys in an area of Romania. The flash-flood potential index was identified with the consideration of landscapes with the potential for torrential flow phenomena. Certainty factor is used to determine the susceptibility to flash-flood, which can be further used to obtain the class label of the data. Multilayer perceptron is then adopted to build a classifier based on the geographical input includes slope, land use, and lithology, etc.

Naghibi et al. (2016) investigated groundwater potential mapping using tree-based algorithms. The main objective of this study was to produce groundwater spring potential maps in Koohrang, Iran. These GIS relevant factors include slope degree, slope aspect, altitude, topographic wetness index, lithology, and land use, etc. The groundwater spring potential was modelled and mapped using classification and regression tree, random forest, and boosted regression tree algorithms. Another study of groundwater potential mapping is that Rahmati et al. (2016) deployed random forest and maximum entropy models for groundwater potential mapping is investigated at Mehran Region, Iran.

Massawe et al. (2018b) presented a mapping approach for soil taxa mapping based on heterogeneous data, which was collected from different sources including satellite image, digital elevation map and digital soil map. The collected features include soil classes, effects of living organisms (vegetation), terrain parameters and spatial location. Random forest and J48 algorithms were used to train the soil profile classification model separately.

In order to build a classifier for different elements using spectral and spatial data, a spectral-spatial feature-based classification (SSFC) framework was proposed to lower the dimension of spectral data and extract the features from spatial data. In this framework, a dimension reduction method called balanced local discriminant embedding (BLDE) was proposed to lower the dimension of hyperspectral images (spectral data with high resolution). CNN was used to extract abstract features from spatial images. The features obtained from BLDE and CNN were then combined and used to train a classifier (Zhao and Du, 2016).

In order to improve soil information for decision making, a predictive soil map was developed using digital soil mapping techniques. The soil profile data was collected, and a numeric classification was performed on the collected data to obtain soil taxa. Then, the soil taxa were spatially predicted and mapped using two machine learning algorithms, which are random forest and J48. Results indicated that random forest shows merits in modelling in comparison with J48 (Massawe et al., 2018a).

In the engineering field, Miles and Ho (1999) proposed several examples of GIS modelling application in civil engineering. The benefits of using GIS in engineering modelling was summarized. Lee (2005) proposed a logistic regression model to evaluate the hazard of landslides using GIS and remote sensing data. Several terrain features such as slope, curvature, and distance from drainage were selected to establish a logistic regression model.

The rapid development of electric vehicles can significantly alleviate environmental problems and energy tension. Zhang et al. (2019b) proposed a multi-objective optimization model based on particle swarm optimization to plan the placement of the charging station of the electric vehicle. GIS was used in this study to identify the intersection of power system and traffic system maps. The intersections of the maps are the candidates of the charging stations. Machine learning techniques have been widely used in the study based on GIS relevant data. Owing to the fact that the lifecycle of assets can be affected by GIS factors such as weather and terrain, it is worthwhile to introduce GIS data into the study of PdM. However, to the best of our knowledge, such studies have not been found in the literature.

2.5 Semi-supervised Learning

Semi-supervised learning is a branch of machine learning, which takes the advantages of both supervised and unsupervised learning. In machine learning, supervised learning is prevailing, while it requires labelled data which a data instance consists of input x and output y. Supervised learning aims at mapping the relationship between input and output of a given labelled dataset. In contrast, unsupervised learning is implemented based on the unlabelled data, which is relatively easy to obtain. In the real world, the collection of a large sum of labelled data is relatively challenging. The performance of supervised learning algorithms is relevant to the data size. When the data size is limited, the algorithm performance of those prevailing machine learning algorithms such as deep learning tends to be compromised. In semi-supervised learning, a small sum of labelled data and a large sum of unlabelled data is utilised to build a supervised learning model. In other words, the unlabelled data is used to leverage the performance of supervised learning when the size of labelled data is limited.

There are three necessary assumptions for semi-supervised learning, which are smoothness assumption, cluster assumption and manifold assumption:

- **Smoothness Assumption:** If two data instances close to each other and located in a dense area, they possibly have the same label. In contrast, if two data instances are apart from each other in a sparse area, their labels tend to be different (Chapelle et al., 2006).
- **Low-density Assumption:** The classifier boundary should pass through the low-density area. The data instances in high density should not be separated into different classes (Chapelle et al., 2009).
- **Manifold Assumption:** The data space is composed of multiple low-dimensional manifolds, where the data instances locating in. The data instances within a manifold have the same label (Ben-David et al., 2008).

The satisfaction of these assumptions is critical for semi-supervised learning. If the assumptions cannot be satisfied, the unlabelled data cannot be helpful to leverage the algorithm performance.

In the research of semi-supervised learning, there are two main types: inductive and transductive given labelled data an unlabelled data. Inductive semi-supervised learning uses labelled data and part of the unlabelled data for training and predicts the label of the unseen unlabelled data. Transductive semi-supervised learning uses the labelled data for training to predict the label of the unlabelled data. Its assumption is the unlabelled data is the testing data. The purpose is to get the best generalisation ability in these unlabelled data (Van Engelen and Hoos, 2020).

2.5.1 Inductive Methods

Inductive methods aim to construct a classifier or regressor that can predict the labelled for unseen data. It can be deemed as the extension of supervised learning which includes the unlabelled data. According to the ways of incorporation unlabelled data, the inductive methods can be classified as three types which are wrapper methods, unsupervised pre-processing methods and intrinsically semi-supervised methods (Van Engelen and Hoos, 2020).

Wrapper methods are the most well-known approaches in semi-supervised learning (Zhu, 2005). The concept of wrapper methods is to train one or more learners using the labelled data and then predict the label of unlabelled data. The unlabelled data with the predicted label is then appended to the labelled dataset. In the next stage, one or more new learners are trained using the updated labelled dataset and predict the label of a new unlabelled data. This procedure is repeatedly implemented to increase the data size of labelled data (Triguero et al., 2015). There are three popular types of wrappers methods which are self-training, co-training and boosting.

Self-training was proposed in 1995 for text mining (Yarowsky, 1995). In the training process, it first trains a model based on a supervised learning algorithm to predict the label of unlabelled data. The unlabelled data instance with the most confident prediction is then added into the labelled dataset. Then re-train the model based on the updated labelled dataset and evaluate the performance. This process is repeatedly implemented until the algorithm performance converges. Self-training has been investigated in different areas such as object detection (Rosenberg et al., 2005), hyperspectral image classification (Dópido et al., 2013), and fake product reviews detection (Wu et al., 2012).

Co-training is originated from self-training. It adopts multiple algorithms to iteratively train models based on labelled data. Assuming there are two trained models which are denoted as A and B. The trained models are used to predict the label of unlabelled data instances, respectively. The unlabelled data with the most confident prediction from A is then added into the training dataset of B, while the unlabelled data with the most confident prediction from B is then added into the training dataset of A. The prediction of an unseen data instance is the mean value of the predictions from model A and B (Zhou and Li, 2005). A critical condition for co-training is that the diversity between different models need to be evident so that the learners are able to learn the exchanged information (Zhou and Li, 2010). The application of co-training is mainly focused on image recognition areas, such as road detection (Caltagirone et al., 2019), multi-organ segmentation (Zhou et al., 2018), and hyperspectral classification (Romaszewski et al., 2016).

Boosting is a type of ensemble learning, which aims to get the prediction by aggregating the predictions of different basic learners. The idea of boosting is to use several weak learners to obtain a powerful learner. Given a dataset, a group of subsets can be sampled from the dataset. Different algorithms can be used to train different learners based on subsets. The prediction of each learner may not be reliable. In the boosting framework, the predictions of these learners are given different powers and then aggregated (Schapire and Freund, 2013). Boosting can be combined with semi-supervised learning. By introducing the pseudo-labelled data into the subsets to increase the algorithm performance. Semi-supervised boosting methods have been investigated for a long time. Grandvalet et al. (2001) proposed a semi-supervised boosting algorithm which is based on the Adaboost algorithm. (Mallapragada et al., 2008) presented an algorithm called SemiBoost to tackle the issue of data points selection for training set construction.

2.5.2 Transductive Methods

Transductive methods is another subset of semi-supervised learning. Inductive methods can generate predictors during the training process, and it has a clear training and testing stage. Different from inductive methods, it is difficult to identify the training and testing stage of transductive methods. The purpose of inductive methods is to predict the label for unseen data. In contrast, transductive methods only concern obtaining the label for the given unlabelled data. In other words, the generalisation ability of the algorithm is not considered. In transductive methods, a graph needs to be defined over all the given labelled and unlabelled data instances (Van Engelen and Hoos, 2020). Then identify the similarity between labelled and unlabelled data instances in the graph. An objective function is defined to match the most similar labelled and unlabelled data instances so to propagate the label (Zhu, 2005).

There are three stages of graph-based semi-supervised learning: graph creation, graph weighting, and inference (Liu et al., 2012). In the first stage, all the data instances in the graph are fully connected. Then the edges are weighted. In the inference stage, an objective function is used to predict the label of unlabelled data. The graph-based methods not only consider the similarity between labelled data instance and unlabelled data instance, but it also considers the similarity between different unlabelled data instances. In other words, the label can be propagated from labelled data instance to unlabelled data instance, and then propagate to another unlabelled data instance (Van Engelen and Hoos, 2020). Transductive methods have been investigated in different fields. Baluja et al. (2008) applied graph-based semi-supervised learning approach to the video recommendation system. Rahman et al. (2019) proposed a transductive learning zero-shot object detection approach to reduces the domain-shift and model-bias against unseen classes. Wang et al. (2017) presented a progressive graph-based transductive learning approach for neurodegenerative disease classification based on multi-model imaging data.

2.5.3 The Application of Semi-supervised Learning in Industry

Due to the obtain of labelled data is costly in the industry, researchers have studied how to bring semi-supervised learning into industrial machine learning modelling to address this issue. When the available labelled data is limited, establishing a classification model with decent performance tends to be hard. Ge et al. (2016) proposed a kernel-driven semi-supervised Fisher discriminant analysis (FDA) model for nonlinear fault classification. The proposed method uses a semi-supervised data matrix and FDA technique to extract the discriminant information, before *k*-nearest neighbours and Bayesian techniques were used for classification. The size of labelled data used in this case was 400, and there were 41 attributes in the dataset, with a ratio of labelled data to unlabelled data is 1:1.

Zhou et al. (2014) proposed a semi-supervised probabilistic latent variable regression method to improve the performance monitoring of variations in the process and the features relevant to the product quality. The probabilistic latent variable regression approach was used for modelling, and an EM (Expectation Maximisation) algorithm was used to estimate the parameters of the semi-supervised model. In this case, 50 labelled instances and 450 unlabelled instances were utilised for model building. The performance of a probabilistic latent variable regression model is the sole benchmark in this case, which makes it challenging to reveal the advantages of the proposed method compared to other approaches.

Zhao et al. (2017b) proposed a semi-supervised model with a capped $l_{2,1}$-norm regularisation. A loss term was used to measure the inconsistency between the prediction and the original labels to the labelled dataset. A global regression regularised term was developed to train a classification model which can achieve better performance. However, the size of the labelled and unlabelled data was not mentioned in this case.

Kang et al. (2016b) proposed a semi-supervised support vector regression (SS-SVR) method based on self-training and applied it to virtual metrology in a semiconductor manufacturing context. The distribution of labels for the unlabelled data was estimated by a probabilistic regression model, and the support vector machine algorithm was used to build the regression model. The data of semi-conductor manufacturing was collected from the sensors embedded in process equipment and previous metrology values. SS-SVR can achieve better prediction accuracy and efficiency compared to the conventional support vector regression. The combined quantity of the labelled and unlabelled data was over 60,000 in this study, and the labelled data make up approximately 6% in this total.

Since the high-quality labelled image data for the deep learning modelling is costly to collect, Sayah et al. (2020) presented a semi-supervised learning approach for additive manufacturing quality identification. In this approach, the labelled image data is first distorted to generate new data samples. Gaussian noise is then added to the increase the model robustness. Several different loss functions are designed to achieve unsupervised group consistency and diversity. In the experimental study, 135 images were adopted to generate 5805 training samples in total. In additive manufacturing, semi-supervised learning has also been applied for the fault detection for laser powder-bed fusion (Okaro et al., 2019) and in-situ video monitoring of selective laser melting (Yuan et al., 2019). Such semi-supervised cases mentioned above have demonstrated that a better modelling performance can be achieved with the help offered by unlabelled data. However, none of the presented studies has assessed the impact of the different ratios of labelled and unlabelled data on performance.

2.6 Summary

In summary, two main types of PdM were reviewed in Section 2.1. With the development of IoT, the sensor data which contains the patterns strongly relevant to the lifecycle of component or asset is available. Hence, condition-based PdM has gained increasing attention in recent years. However, the deployment of such methods requires extra IoT equipment, which increases the burden of companies. The study of statistical PdM is based on the historical maintenance data, which is relatively easy to obtain. It deserves more attention. In Section 2.2, several deep learning algorithms were introduced. With the capability in processing various types of data, it can be seen that deep learning has become prevailing in industrial big data analytics. Section 2.3 reviewed two main types of semi-supervised learning and their application in industry. In Section 2.4, the studies of GIS were reviewed. It can be seen that GIS data have not gained sufficient attention in the industry. From the literature, it is evident that PdM is an important topic under Industry 4.0. Automobile is essential to the industry. However, very few existing studies have focused on automobile PdM. Deep learning as a prevailing tool is helpful in big data analytics of PdM. Since deep learning is a data 'hunger' algorithm, semi-supervised learning is a useful tool when labelled data is limited. Finally, the integration of GIS data is helpful to leverage the prediction accuracy of TBF. In order to investigate deep learning, reliability analysis and semi-supervised learning in automobile PdM, several assumptions need to be made according to different techniques. Firstly, deep learning is sensitive to label accuracy. Due to the historical maintenance data are collected from the real world, the data with inaccurate label is deemed as the minority. Secondly, in order to implement reliability analysis, the failure time of automobile in historical maintenance data is deemed following a specific distribution, such as Weibull. Thirdly, there are three assumptions for semi-supervised learning mentioned in Section 2.6. For the historical maintenance data, the data have similar attributes deemed to have a similar label.

Chapter 3 A Framework for Automobile Predictive Maintenance under Industry 4.0

3.1 Introduction

Industry 4.0 refers to the fourth revolution of the industry, which mainly focuses on intelligence and automation (Qin et al., 2016). The emerging technologies such as AI, cloud computing and IoT has boosted the development of Industry 4.0. Owing to the fact that run-to-failure and preventive maintenance are the main strategies in most companies in the industry, PdM has gained increasing attention under the context of Industry 4.0 since it can significantly lower the maintenance cost and asset downtime. The state-of-the-art of PdM was reviewed in Section 2.2. However, there are no existing study reports on how to model automobile TBF under the context of Industry 4.0. In order to leverage the automobile PdM into next generation, there is a need to investigate a PdM framework that can be helpful to the industry.

In this chapter, a framework is designed for automobile TBF modelling, prediction, and decision support based on the industrial big data. The framework is designed by following the data lifecycle standards, which includes the sections of data collection, transmission, storage, pre-processing, filtering, analysis, and mining, etc (Siddiqa et al., 2016). The data and technique relevant to the proposed framework were reviewed in Section 3.2.

3.2 A Framework for Automobile Maintenance Modelling under Industry 4.0

According to the literature review in section 2, the state-of-the-art of PdM mainly focuses on modelling based on sensor data or historical maintenance data, while the automobile lifecycle can be affected by various factors such as traffic or driving behaviour. With the development of IIoT technique, it is possible to collect these data which are relevant to automobile lifecycle. Hence, there is a need to introduce multi-source data which is relevant to automobile lifecycle and then integrate for automobile maintenance modelling. Meanwhile, deep learning, as a prevailing tool in industrial big data analysis (Mohammadi et al., 2018), has been widely used in PdM and showed merits. Other techniques such as statistical model can also be helpful in PdM. Exploring how to combine different techniques in automobile maintenance modelling also needs to be considered in the framework. The proposed framework is illustrated in Figure 3.1.

The proposed framework includes the following stages: data collection stage, cloud data transmission and storage stage, data mapping pre-processing and integration stage, deep learning for automobile TBF modelling stage, and decision support for PdM stage. The framework was designed referred to the concept of data mining, which was detailed in Appendix A1. The multi-source data is first collected and then uploaded to the cloud. Then the multi-source data is mapped, pre-processed and integrated before it is used for modelling. In the modelling stage, deep learning in conjunction with semi-supervised learning and reliability analysis is adopted to establish a TBF prediction model. Finally, the predicted TBF of an automobile is used to optimise maintenance planning, spare parts management, and job scheduling.

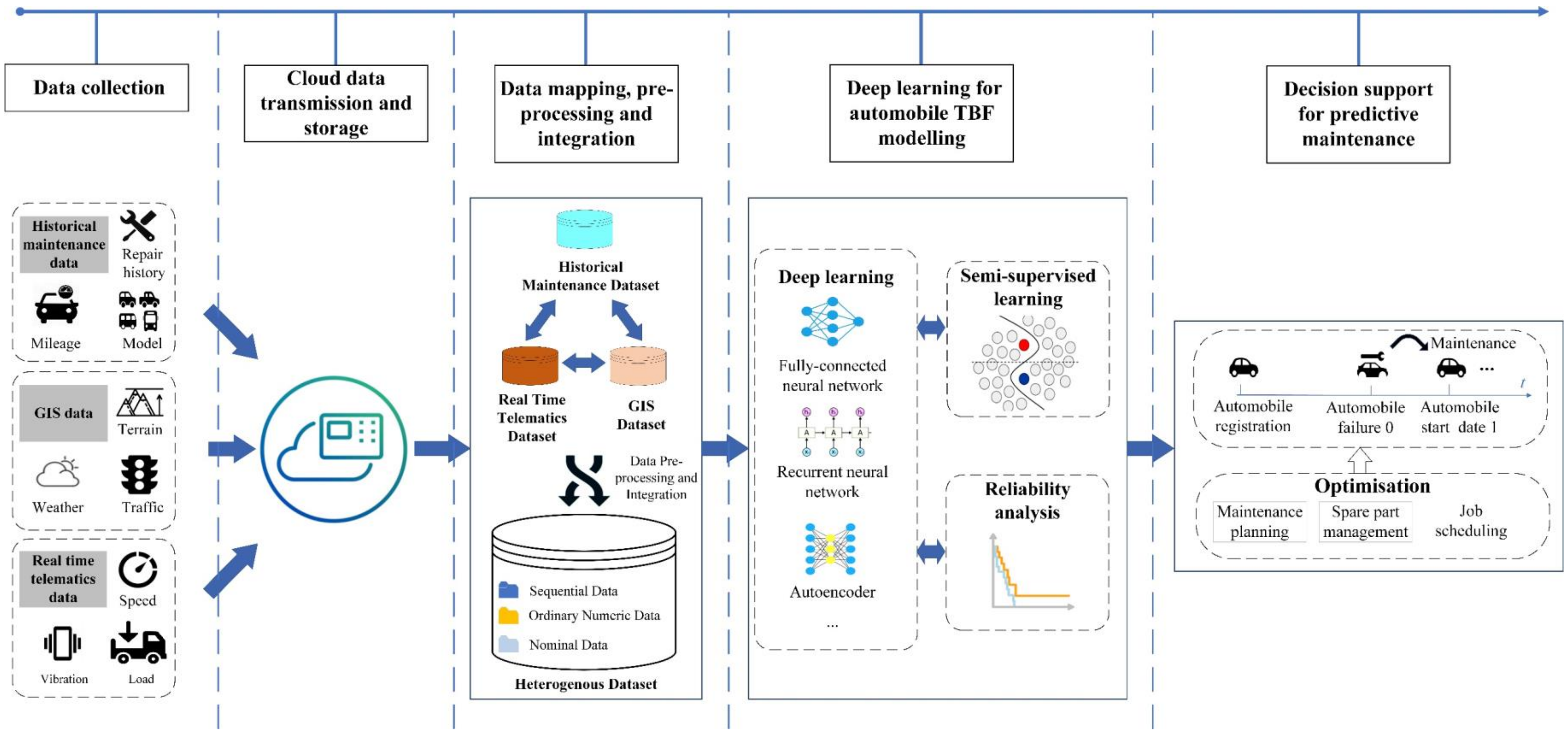

Figure 3.1 A Framework for Automobile Maintenance Modelling under Industry 4.0

3.2.1 Data Collection

There are two main types of data widely used in PdM, which are sensor data and historical maintenance data. In automobile PdM, there are multiple types of data relevant to automobile lifecycle. Figure 3.2 shows the automobile lifecycle with different types of data. An automobile starts its service after registration, it will experience multiple times failures before the end of the lifecycle. After its first failure at t_1, the historical maintenance data can be collected. There are three types of features in historical maintenance data. The manufacturing and geographical features are obtained when the automobile is registered, while the maintenance features can be captured after the faulty automobile is repaired at t_2. With the wide deployment of sensors in automobile, the real-time telematics data such as speed, vibration and load can be collected during the working time of automobile. The frequent speed change could accelerate the failure of the engine, while the increase of vibration may indicate the component is in anomaly working stage. These real-time telematics data can directly reveal the health condition change of the components in an automobile. A challenge towards online monitoring and failure prediction can be how to transmit the large volume of sensor data to the cloud for modelling and prediction.

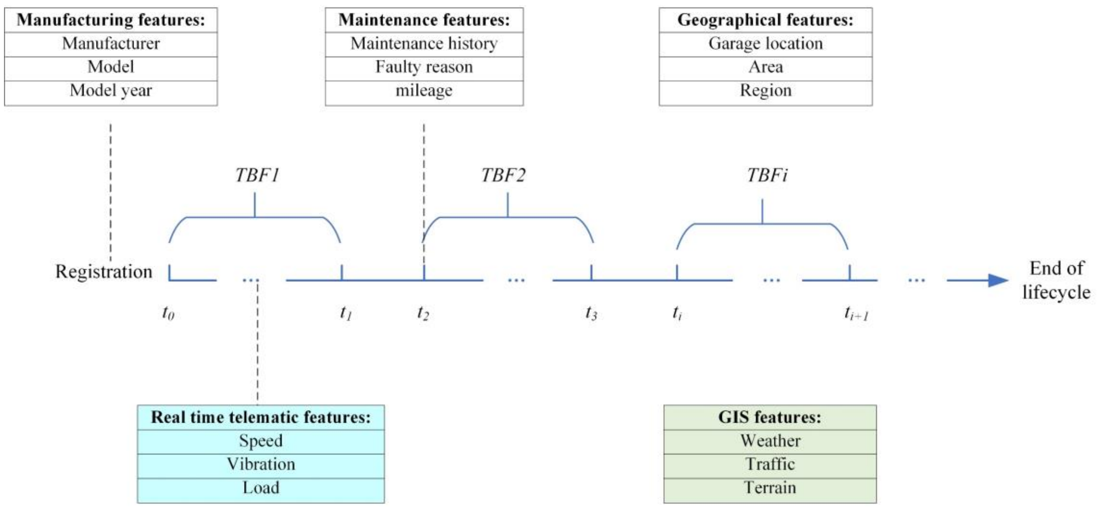

Figure 3.2 The automobile lifecycle with various types of data

Historical maintenance data is the maintenance record collected from the garage of a fleet management company. The features in historical maintenance data include repair history, mileage, and automobile model, etc. An existing study has identified several features, including automobile age and maintenance frequency, etc., which are relevant to automobile TBF (Wang et al., 2018c). In comparison with sensor data, historical maintenance data is relatively easy to obtain. Normally, there are both numeric and nominal features in historical maintenance dataset.

Besides sensor data and historical maintenance data, other data such as weather, terrain and traffic can also be relevant to the automobile lifecycle. Owing to the fact that collecting these data via IIoT remains challenging, it needs to be collected via the link with the outer data source. The real-time GPS (Global Positioning System) information of automobile can be easily obtained from automobile, and it can then be used to bridge the automobile with other data. The weather data, traffic data and terrain data of an automobile can be collected from the outer data source and processed using GIS. For a fleet management company which processes a large number of automobiles, the automobiles may work in different areas. The geographical relevant factors which may affect automobile lifecycle such as weather, terrain, and traffic are different from area to area. Hence, introducing these factors into the study of PdM can bring tangible benefits to the fleet management company.

3.2.2 Cloud Data Transmission and Storage

Under the context of Industry 4.0, the industry has been transformed to digital ecosystem (Finance, 2015). During the transformation, cloud technology has gained increasing attention since it opens new horizons in conjunction with other technologies such as IoT and Cyber-Physical Systems towards industrial digitalisation (Mourtzis and Vlachou, 2018). With the connectivity of the assets in the industry, the data is generated at high speed. Cloud technology enables the integration of hardware, software, network and other resources to realise the calculation, storage, processing and sharing of data (Sabahi, 2012).

After the sensor data was collected in the automobile, the data needs to be transmitted to the cloud via remote transmission technique such as 5G. However, since the sensor data is collected in high frequency which leads to a large data volume. Real-time transmission may lead to unaffordable transmission load. Hence, a technique like edge computing (Shi et al., 2016) can be deployed to address this issue. Furthermore, the historical maintenance data is low in data volume. The garage can be linked to the cloud, and then the historical maintenance data can be directly uploaded to the cloud. Meanwhile, the GPS information from the cloud can be obtained from the automobile. After the cloud receives the GPS information, it can link it with the outer database such as weather, traffic and terrain. These data are then collected to the cloud and then be processed by GIS. All these multi-source data can be stored in the cloud. When the data is needed for modelling, it can be extracted and modelled via cloud computing.

3.2.3 Data Mapping, Pre-processing and Integration

After the data is stored in the cloud, it can be used for modelling. The multi-source data are different in data types and granularities. In order to use multi-source data for modelling, the first step is data mapping. The sensor data is collected in high frequency which sampling rate can up to the scale of a millisecond, while the GIS data such as weather and traffic, the sampling rate can be minutes. In strike contrast, historical maintenance data can only be collected when a failure occurs, which may take years. Hence, the data need to be mapped into the same granularity before it is used for modelling.

Then the multi-source data needs to be pre-processed. In this stage, the extreme and missing values in the dataset need to be removed or replaced. Then, the data needs to be normalised to increase data integrity (Codd, 1970). Finally, the data is collected chronologically, which may contain some local patterns. These patterns may not be representative in the whole dataset and therefore may be harmful to the algorithm performance in the modelling stage. Hence, the dataset needs to be reshuffled. After the multi-source data is pre-processed, it is then integrated into a heterogeneous dataset which contains different types of data including time series, ordinary numeric data, and nominal data, etc.

3.2.4 Deep Learning for Automobile TBF Modelling

When the pre-processed data is obtained from the previous section, it can be used for modelling. Deep learning is a class of machine learning algorithms which are based on ANN (Deng and Yu, 2014, Bengio, 2009). Every neural network is a computational system which can process information through their dynamic state responses to external input. There are many interconnected and straightforward processing elements referred to as neurons. Traditional ANN normally has a shallow structure (the number of hidden layers is lower than 3). When there are too many hidden layers in ANN, the training process will be complicated due to the vanishing gradient problem. In recent years, as the development of optimizers, activation functions, and layers with various functions has continued, training of neural networks has been significantly improved. Current advances in the deep learning field, have developed neural networks with more complex structures, which can offer more powerful learning abilities.

A neural network consists of different layers of three main types: the input layer, the hidden layers and the output layer. The design of a neural network needs to determine the necessary elements, which are the layers, activation function, loss function and optimizer. In order to build the prediction model, all the elements need to be well considered according to their properties and data characteristics. Moreover, the parameters of the deep learning model need to be well considered to optimize the outcome.

Currently, in the deep learning field, there are various types of neural networks which consist of different types of hidden layers. These deep learning models, such as convolutional neural networks and recurrent neural networks are widely used in image recognition and text mining, respectively (LeCun et al., 2015). Each neuron is a tiny computational unit which uses an activation function to produce an output passed to the next layer. When different values are transmitted to a neural network, the input will be given a unique weight. The product of different values and their weight will be added together with a bias. Then, the sum will be imported into the activation function to get the output which will be transmitted to the neurons in the next layer.

In order to design a deep learning model, the following methodology is adopted. Firstly, the task and data type need to be identified. Different tasks and data types require different model components. Secondly, the structural components, the layer, activation function, optimizer and loss function, need to be determined according to the data type and the aims of the modelling. Different layers are suitable for different types of input data. For instance, convolutional layers are used to deal with image data, and fully connected layers are used to deal with numeric data. The activation function is used to calculate the output of the neurons. The optimizer is a component which is used to optimize the gradient descent during the neural network training process. The loss function is used for measuring the compatibility between the actual value and the predicted value. The output of the loss function is referred to as loss. The next step is to set the parameters of deep learning. There are several significant parameters which need to be well considered. These are the number of hidden layers, the number of neurons, batch size, learning rate, and epoch. If the number of neurons and hidden layers is insufficient, the neural network will not be capable of learning the hidden patterns in the data. If the number of neurons and hidden layers is too large, it will cause an expensive computation load. About the number of neurons in each hidden layer, the same size for all hidden layers generally worked better or the same as using decreasing size (Bergstra and Bengio, 2012). Batch size is the number of training instances that need to be fed into the neural network. The smaller batch size enables the faster training speed of the neural network, while it will sacrifice the estimating accuracy of the gradient. Learning rate is a significant parameter which defines the step size in the optimization stage. If the learning rate is too large, the optimization may not converge. If the learning rate is too low, the training process may take a long time. In the neural network field, an epoch contains two phases which are feedforward training and back-propagation. A larger number of epochs can obtain a better training performance but will result in longer training time. The fourth step is to train a deep learning model. The training of deep learning models is always performed by a back-propagation algorithm. In the forward training stage, the error of the entire network is calculated at the output. It is impossible to adjust the weight and bias of the layers only depend on the final error. Back-propagation aims to trace the error of each layer and make the corresponding adjustment so as to lower the final error (Nielsen, 2015). The final step is to validate the deep learning model. Validation is essential because it will indicate whether the model is well designed. If the results are disappointing, then the components and parameters need to be tuned. After the deep learning model is designed, the historical maintenance data can be fed into the neural network to train a TBF prediction model.

Deep learning is a type of algorithms which requires a large number of labelled data to tune the parameters. However, collecting a large sum of data in the industry tends to be challenging as it requires extra cost, and the unlabelled data is relatively easy to obtain. Semi-supervised learning is a technique that can be used to build a regression or classification model based on a small number of labelled data and a large number of unlabelled data (Chapelle et al., 2009). Hence, it can be a useful tool in TBF modelling using deep learning when the labelled data is limited. Reliability analysis is a useful tool in processing event data, which can be used to obtain a hazard function which can reveal the relationship between failure probability and running time. With such a hazard function, the change of the failure probability alone with the running time can be studied. However, such a hazard function cannot indicate the specific failure time of an asset (Leitch, 1995). Some reliability analysis models such as Weibull (Xie and Lai, 1996) and Cox PHM (Cox, 1992) can be used for modelling based on censored data. Data censoring is a specific issue in reliability analysis. The label of the censored data is not accurate. However, the label accuracy is essential for machine learning modelling. With the higher label accuracy, a more accurate machine learning model can be obtained (Cortes et al., 1994). In the state-of-the-art, there is no study that considers data censoring issue in modelling based on historical maintenance data.

Hence, bridging reliability analysis with deep learning can promote the prediction accuracy of TBF.

3.2.5 Decision Support for Predictive Maintenance

After the TBF prediction model is established, it can be used to predict the TBF for an in-use automobile. With a prediction of TBF, the automobile can be returned to the garage before the upcoming failure and therefore, the crucial failure can be prevented. If the maintenance can be scheduled in the early stage of the failure, the maintenance time and cost can be dramatically decreased (Patil et al., 2017). Moreover, the number of failure cars can be estimated with the support of the prediction model. Then the number of spare parts in the garage can be controlled at a reasonable level. Furthermore, the prediction of TBF can also be beneficial to job scheduling. For example, for those automobiles which TBF prediction is short, they can be allocated with short distance and non-heavy tasks so to extend their useful life.

3.3 Summary

With the leverage of the connectivity and intelligence in the era of Industry 4.0, PdM has become an essential key. In the data-rich environment, the surrounding data relevant to the automobile lifecycle such as sensor data, historical maintenance data and GIS data can all be collected and used for automobile TBF modelling. To achieve this target, a framework for automobile TBF modelling under industry 4.0 was proposed in this chapter. In this framework, the multi-source data is first collected and transmitted to the cloud for data mapping, pre-processing and integration. In the next stage, deep learning is used to establish a TBF prediction model. Finally, the TBF prediction of automobiles can be used to leverage the decision support in automobile PdM.

Chapter 4 Predictive Maintenance Using Cox Proportional Hazard Deep Learning

4.1 Introduction

Recently, machine learning, as a subset of artificial intelligence, has been widely used in different areas of industry such as energy consumption prediction (Qin et al., 2020), fault diagnosis (Wan et al., 2018), and adaptive control optimisation (Kruger et al., 2011). Meanwhile, it also has been used in PdM (Prytz et al., 2015, Wei et al., 2013, Nieto et al., 2015, Lee and Pan, 2017, Zhao et al., 2017a, Li et al., 2018, Malhi et al., 2011, Yuan et al., 2016, Zhang et al., 2018a). Among various machine learning methods, deep learning has gained considerable attention in PdM (Zhao et al., 2017a, Li et al., 2018, Malhi et al., 2011, Yuan et al., 2016, Zhang et al., 2018a). Deep learning, as a group of machine learning techniques, has shown its merits in modelling based on high dimensional and large size data (LeCun et al., 2015). LSTM network, as one of the deep learning techniques, has a specialized structure in processing sequential data (Hochreiter and Schmidhuber, 1997). Because the next TBF of an automobile is highly relevant to its previous failure and maintenance information, LSTM network can be employed in the modelling of automobile TBF.

The algorithm performance of deep learning relies on the quality of the data label (Chen and Lin, 2014). Without an accurate label, deep learning may be challenging to learn the hidden patterns within data. In historical maintenance data, the label of the censored data is not sufficiently accurate. Hence, the data censoring problem needs to be addressed. This study aims to propose an approach called CoxPHDL to build a TBF prediction model based on historical maintenance data. The main contributions of this chapter are: (1) Different from most of the existing studies aiming at RUL modelling based on sensor data, an automobile TBF modelling approach based on historical maintenance data is proposed in this study; (2) Due to the data sparsity might damage the algorithm performance of LSTM network and Cox PHM, autoencoder is introduced to convert the sparse data into a robust representation; (3) Because accurate data label is important for deep learning modelling, Cox PHM is introduced to estimate the correct label of censored data so as to improve the algorithm performance of LSTM network. The rest of this chapter is organised as follows: Section 4.2 introduces Cox proportional hazard deep learning. An experimental study is demonstrated in Section 4.3, and its results are demonstrated in Section 4.4. Finally, Section 4.5 discusses the results obtained, and Section 4.6 summarises this chapter.

4.2 Method: Cox Proportional Hazard Deep Learning

A new modelling approach called CoxPHDL is proposed to establish the TBF prediction model based on the historical maintenance data, which contains both numeric and nominal features. The approach consists of three stages. The first stage is nominal data processing. Traditionally, the nominal data is converted to binary attributes for modelling. If the categories in nominal data are numerous, the dataset of binary attributes tends to be sparse. In order to lower the dimension of the sparse binary data without damaging the information dramatically, autoencoder, as a type of deep learning model which is good at extracting significant features, is introduced in this approach. The nominal data is first converted to binary data using one-hot encoding approach. Then autoencoder is used to further process the binary data. The details of the autoencoder are demonstrated in Section 4.2.1.

The second stage is the censored data processing. Censored data is common in historical maintenance dataset. Cox PHM is a statistical model which is used to process censored and uncensored data (Cox, 1992). The data obtained from autoencoder is combined with the numeric data and yield a new dataset. The features in the new dataset are then used as covariates. The new dataset is used to build a Cox PHM, which can reveal the relationship between survival time and reliability. The difference in reliability between corrective maintenance and preventive maintenance is case dependent, which needs to be determined in the actual case. With the difference in reliability, the difference in TBF between corrective maintenance and preventive maintenance can be estimated, and the censored data is compensated. The compensated labels of preventive maintenance instances are closer to their actual TBF, compared to their original label. The compensated censored data is used jointly with the uncensored data to train an LSTM model in the next stage. The details of Cox PHM are introduced in Section 4.2.2.

Finally, after the data is pre-processed, LSTM network, a deep learning model, which is specialized in processing the sequential data, is used to train a prediction model. LSTM network is used to predict the next TBF based on the previous failure information. The details of the LSTM network will be introduced in Section 4.2.3. The flow chart of the proposed method is shown in Figure 4.1.

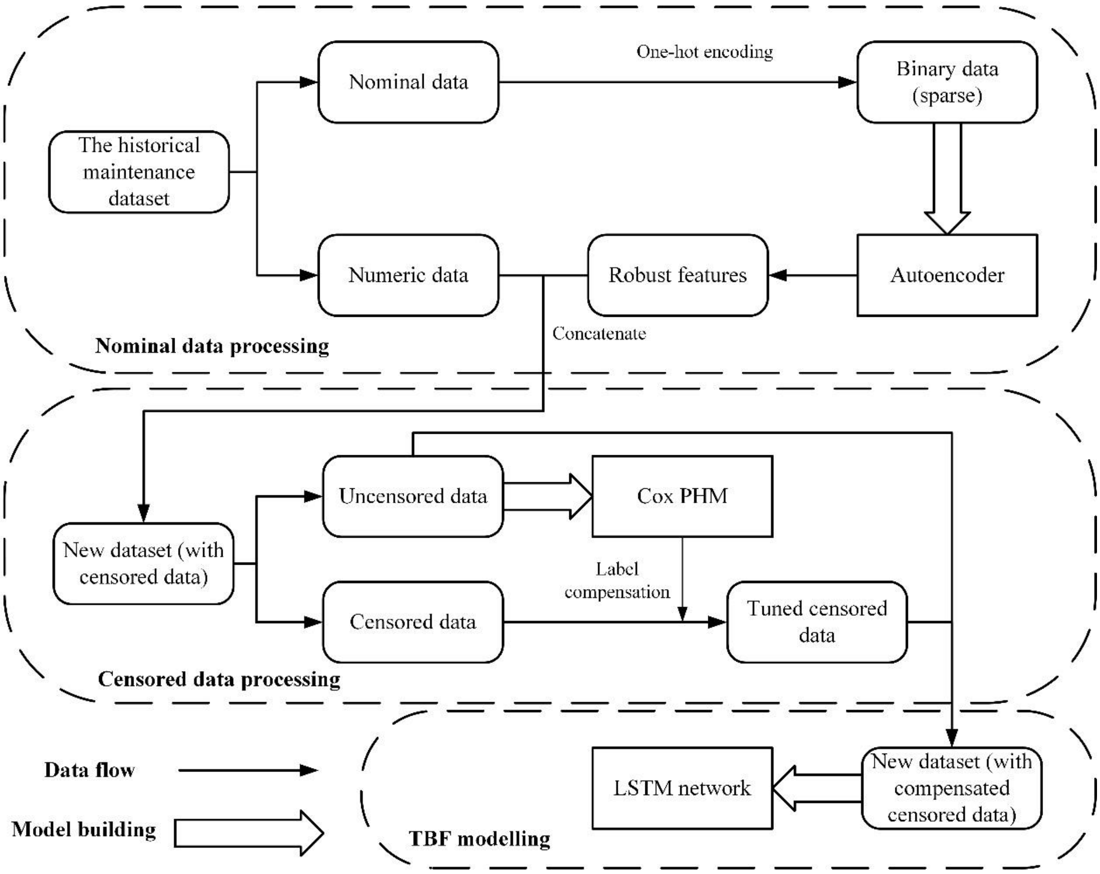

Figure 4.1 The flow chart of the proposed approach

4.2.1 Autoencoder

Autoencoder is an unsupervised learning technique, which has shown merits in feature extraction. It aims to learn the most significant features from data. The learned features are expected to reconstruct the original input completely (Hong et al., 2015). An autoencoder consists of two parts which are encoder and decoder. An autoencoder can be described as a multi-layer neural network. The input layer and the first half of the hidden layers constitute the encoder, and the second half of the hidden layers and the output layer constitute the decoder. The number of nodes in each hidden layer is less than the number of nodes in the input layer and the output layer.

The input vector of the autoencoder is denoted as x. The features learned by the encoder, also known as code, is denoted as z.

The relation between x and z can be denoted as:

$$z = f(Wx + b) \quad (4.1)$$

where W is the weight matrix between the input layer and the hidden layer, b is the bias, and the $f()$ is the activation function.

The features z learned from the hidden layer is then used to construct a vector x' which is expected the same as vector x. The relationship between x' and z can be represented as:

$$x' = f[W'z + b'] \quad (4.2)$$

where $W'$is the weight matrix between the input layer and the hidden layer, b' is the bias, and the $f()$ is the activation function.

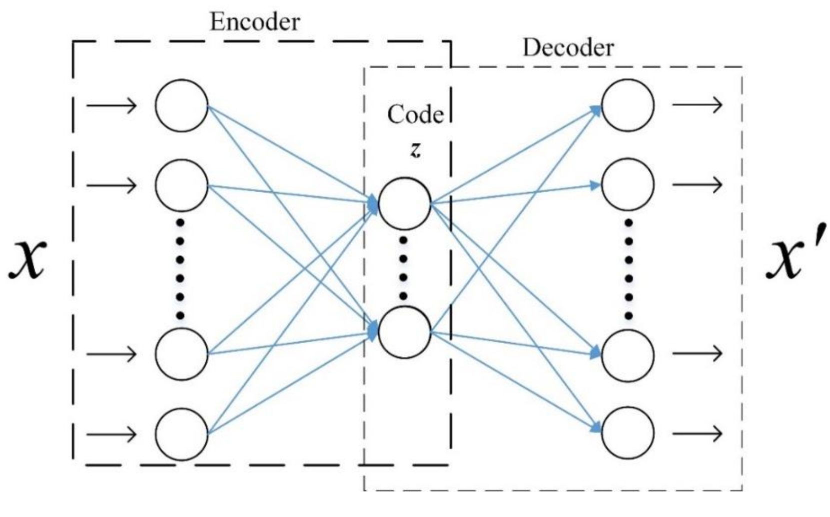

Figure 4.2 The structure of autoencoder

As autoencoder is a type of neural network, the parameters of (W, b) and (W', b') can be trained via back-propagation algorithm. However, in the actual modelling, the vector x' cannot be completely the same as vector x. The difference between vector x and vector x' can be measured by a loss function. The adoption of the loss function is data-dependent. The structure of a three-layer autoencoder is shown in Figure 4.2.

4.2.2 Cox PHM

Cox PHM is a statistical model which aims to analyse the relationship between time-independent covariates and hazard function (Cox, 1992). The baseline hazard function is denoted as $h_0(t)$. The covariate is denoted as β_p and the input vector is denoted as X_p. The Cox PHM is denoted as:

$$h(t, X) = h_0(t)\, exp(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p) \tag{4.3}$$

With the different adoption of $h_0(t)$, Cox PHM can be parametric or non-parametric. A widely used function for the $h_0(t)$ is the maximum likelihood estimator proposed by Breslow (Breslow, 1975).

Figure 4.3 shows a reliability curve generated by Cox PHM. The actual TBF of the censored data needs to be estimated. The difference in reliability ΔR between corrective maintenance and preventive maintenance is assumed the same for all the equipment possessed by the same company. Hence, once ΔR can be determined, it can be used to determine the difference of TBF ΔT between corrective maintenance and preventive maintenance. ΔR strongly depends on the strategy of the fleet management company, which is hard to be determined. The method used to determine ΔR needs to be determined in the actual case. ΔR is used to yield ΔT for the censored data. Then ΔT is used to compensate the TBF of the censored data. The censored data with compensated TBF is then used to modelling jointly with the uncensored data. The performance is noted and compared to determine the suitable value of ΔR.

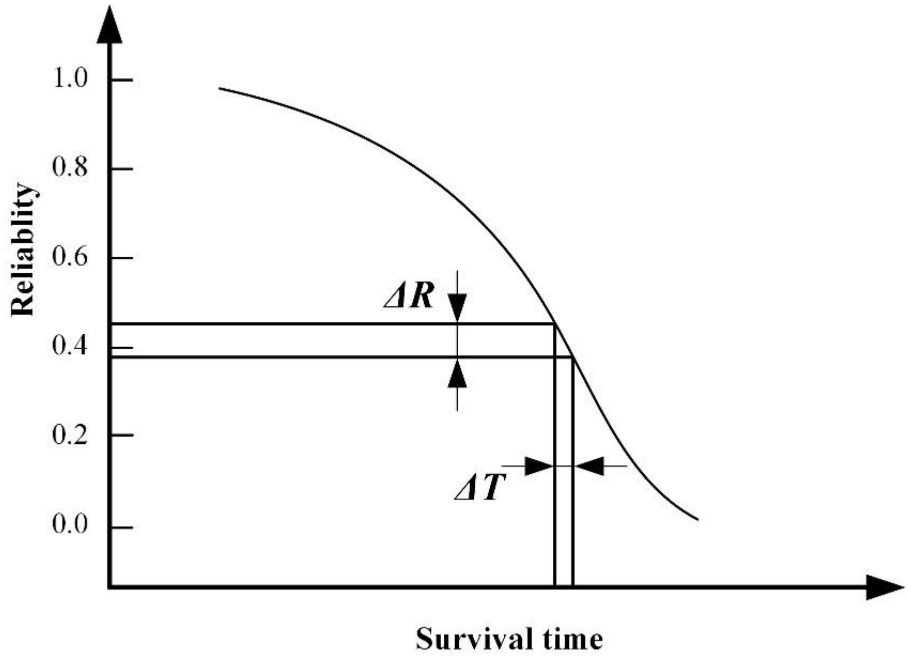

Figure 4.3 The reliability curve generated by Cox PHM

4.2.3 Long-short-term Memory Network

LSTM network is a type of deep learning model which is well known for processing sequential data. An LSTM layer has numerous cells. There are three gates, which are forgotten gate, input gate and output gate, used to control memory in each cell. The structure of an LSTM cell is shown in Figure 4.4.

The c_i and b_i is the input and output of the LSTM memory cell. When c_i is transmitted into the LSTM memory cell, it is first processed by an activation function. The output of the activation function is then multiplied by b_ι. Secondly, the activation function output of the previous time step is multiplied by b_Φ. The product is added to the memory. Finally, the output of the memory is multiplied by b_ω and then transmitted to another activation function to produce b_i. The factors $b_\iota, b_\Phi, b_\omega$ are represented by three white circles in Figure 4.4. These three factors are determined by the input gate, forget gate, and output gate, respectively.

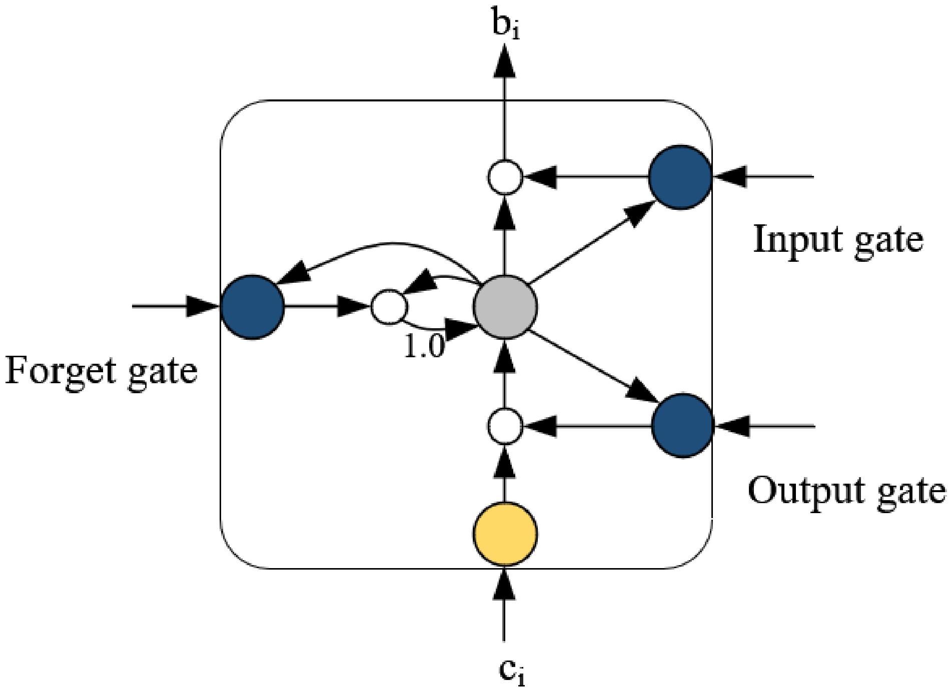

Figure 4.4 The structure of an LSTM cell

The TBF tends to be shorter with the increase of maintenance frequency. Our previous study has demonstrated that the TBF after n^{th} maintenance can be predicted using the automobile information collected in n^{th} maintenance (Chen et al., 2018). However, the TBF of an automobile after n^{th} maintenance is not only relevant to the automobile information collected in n^{th} maintenance, while the previous maintenance information (before n^{th}) also can be relevant to the TBF. Therefore, the information of the previous TBF and the previous maintenance information can be used for TBF modelling. Figure 5 illustrates an LSTM network for TBF modelling. The TBF after n^{th} maintenance is denoted as y_{n+1}, the maintenance information collected in n^{th} maintenance is denoted as x_n, the previous maintenance information $X = [x_1, x_2, x_3, \dots, x_n]$ is expected to be used for TBF modelling. An LSTM network is denoted as $f()$, the TBF modelling using LSTM network can be expressed as follow:

$$y_{n+1} = f(x_1, x_2, x_3, \dots, x_n) \tag{4.4}$$

In an LSTM network model, there are various parameters need to be determined, including the size of the network, optimiser, loss function, and learning rate, etc. Different parameters setting might result in different performance. The parameter setting is case dependent, which needs to be well selected. The performance of the proposed approach is evaluated in the following sections. The LSTM network for TBF modelling is shown in Figure 4.5.

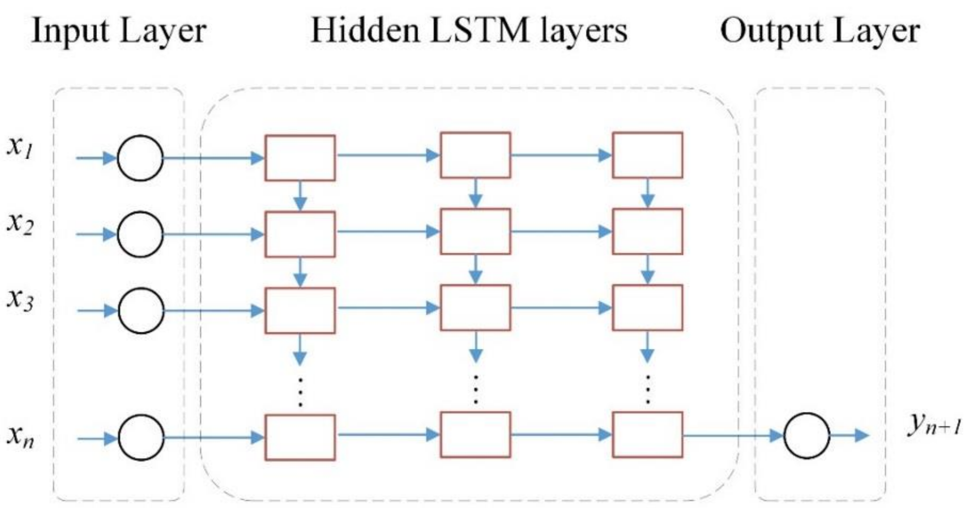

Figure 4.5 The diagram of an LSTM network for TBF modelling

4.3 Experimental Setup

The historical maintenance dataset used in this case was provided by our industrial collaborator, which is a sizable fleet service company in the UK. The company has a keen interest in the TBF prediction of automobiles. An accurate prediction of TBF can offer insights for the fleet maintenance and further help the inventory management of replacement parts.

Firstly, it is worthwhile to provide general information on the company background. This company processes a large number of automobiles including various sizes of vans, personal cars, 4 by 4 vehicles. There are two types of maintenance management in the company. The first maintenance management type is run-to-failure (corrective maintenance), whereby an automobile is sent back to the workshop for maintenance when it breaks down. Workshop records the actual date of automobile failure, and therefore the actual TBF can be calculated. Another maintenance management type is preventive maintenance, whereby soon-to-failure is determined in the scheduled check. If an automobile is deemed to be failed in the near future during the scheduled check, the preventive maintenance is then carried out. The automobiles which experienced preventive maintenance is also recorded. However, the maintenance date in the dataset is earlier than the automobile's actual failure date, which will result in the calculated TBF is shorter than the actual TBF. Thus, the TBF of the preventive maintenance record is right-censored.

Secondly, we focused on the procedure of data processing. With the domain knowledge, feature selection and data pre-processing can be implemented efficiently and accurately. Feature selection and data pre-processing are introduced in Section 4.2. Thirdly, the metrics used to reveal performance and the validation method need to be considered.

Finally, in order to get comprehensive results from modelling, *k*-fold cross-validation was adopted. In this study, the value of *k* was set at 10.

4.3.1 Data

The data contains the maintenance record of the automobile engine. Each instance in the dataset represents one instance of a maintenance record. The data collection period had lasted for nearly nine years, from 2009 to 2017. There are over 12 thousand instances in the dataset. The quantity of censored data is 2,352, which takes 19.2% in the dataset. It can be seen that 40.9% of the automobile engines failed within 500 days. The average of the TBF of all the instances in the dataset is 850.64 days.

Table 4.1 The original features relevant to TBF

Numeric Feature	Note	Nominal Feature	Note
nRepair	The times of engine experienced maintenance	Model	The model of automobile
PAge	The age of automobile engine	Garage	The garage of automobile
VAge	The age of the automobile	Area	The area of automobile
CumM	The cumulative miles when a failure occurs		
Model_Year	The year of the first production		
Seq	A time index for automobile		
Regions	Four binary attributes		
isSch	A label whether a failure is found in schedule check		

The features relevant to automobile engine's lifecycle have been extracted from the dataset, which was used to build the TBF prediction model. The features are shown in Table 4.1. Among all the features, three of them are nominal and the rest are numeric. Due to the limitation of the number of features, all the features were selected for modelling. The numeric data can be directly used for modelling except for *isSch* because this feature cannot be determined before failure occurs. It is used to distinguish the right-censored data and the uncensored data in this case. Meanwhile, the nominal features, which are Model, Garage, and Area, are deemed highly relevant to the TBF according to the domain knowledge. One-hot encoding and autoencoder were used to further process the nominal data. The features mentioned above were selected to establish a TBF prediction model by adopting different machine learning algorithms. Figure 4.6 shows the TBF prediction model with numeric and nominal features.

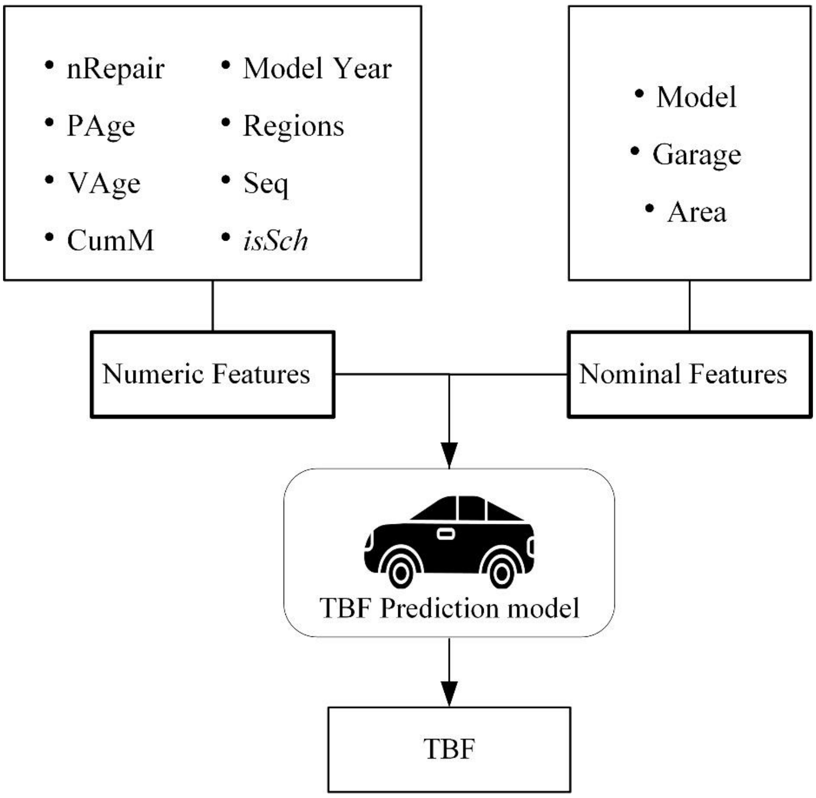

Figure 4.6 TBF prediction model with features

The data was collected from the real world, which contains some impurity and noise in the dataset. The impurity and noise in the dataset were caused by the meter failure and meter misreading. The dataset with noise and impurity may damage the algorithm performance. Hence, the data pre-processing needs to be carried out. Firstly, abnormal and missing values were deleted. For example, there are approximately 30 records where the *CumM* of the automobile is over 4,000,000 miles, significantly exceeding the mean *CumM* of 129,219 miles. These abnormal values were considered as the impurity which may be caused by meters failure or reading mistakes. Secondly, most of the missing values situated in the nominal features, while these missing values are hard to estimate and replace. Hence, the instances which contain the abnormal and missing values in the dataset were removed. Thirdly, *Model Year* is a feature of point-in-time, which is meaningless to be analysed by machine learning algorithms. Hence, the difference between automobile registered year and its model year was used instead because it can represent the age of this automobile model. Finally, the data were normalised in order to improve data integrity and lower data redundancy. Since the nominal data had not been converted in this stage, the normalisation was carried out later.

4.3.2 Model Setup

In the modelling stage, five machine learning algorithms, which are LSTM network, DCNN (Li et al., 2018), RNN, FCNN and support vector machine (SVM), are used for modelling. DCNN is a novel algorithm which shows merits in RUL modelling (Li et al., 2018). RNN and FCNN are two standard deep learning algorithms. Support vector machine is a prevailing machine learning algorithm which has been widely used in data analytics (Ma and Guo, 2014). The technical details of DCNN and SVM have been introduced in Appendix A2. All the algorithms were used to train a TBF prediction model separately based on the pre-processed dataset. We adopted the default settings of the prevailing machine learning algorithm. The deep learning models were designed and established using Python Keras package (Chollet, 2015). The aim of modelling is to build a prediction model based on historical maintenance data. When designing a deep learning model, there are several issues that need to be considered.

Firstly, the structure of deep learning model needs to be determined. The factors such as the type of layer, the number of layers, and the number of nodes in each layer directly affect the performance of the deep learning. If a model is designed extremely deep and large, it is able to predict TBF accurately. However, the computational cost will be extremely large as well. Hence, it is vital to balance the model complexity and computational cost. After several trials, three LSTM layer, one fully connected layer and three dropout layers were adopted as hidden layers in our LSTM network model. The size of the hidden nodes in LSTM and fully connected layers were set at 1000. Drop out layer is used to prevent overfitting by disconnecting a certain percentage of nodes in the training process. In this case, the percentage was set at 20%. The deep learning model designed in this study is an LSTM network model.

Secondly, the components and parameters of the LSTM network model, such as optimiser and loss function need to determined according to the data type and the aim of the study. The optimizer is used to optimise the learning process of the LSTM network model. RMSprop is an optimizer which is suitable for the LSTM network, and therefore it was adopted. The loss function is set to be the mean squared difference between the actual value and prediction value.

The parameters relevant to the training process are the number of lookbacks, learning rate, batch size, and epochs. The number of lookbacks is the number of states that the LSTM network is considered at the same time. In this case, it was set at two, which means the variables in time *t-1* and time *t-2* are used to predict the output of time ***t*** due to the failure times of approximately 85.3% instances are lower than 2. The learning rate is the stride of the training process. In order to enable the LSTM network to sophisticatedly learn the hidden patterns of data, the learning rate was set at 0.001. The batch size was set at 150, which means 150 instances were fed into the LSTM network each time. The number of epochs was set at 45, which means the back-propagation process was repeated for 45 times to tune the parameters of the LSTM network. Meanwhile, the configuration of the FCNN model in this study was basically the same as the LSTM model, except all the layers of the FCNN are fully connected layers.

In order to tackle the data sparsity issue of one-hot encoded data, autoencoder, another deep learning algorithm, was introduced in this study. Autoencoder is a neural network which can extract the most significant features from sparse data (Hong et al., 2015). The autoencoder designed in this case was a three-layer neural network comprising of an input layer, hidden layer, and output layer. The number of nodes of input and output layers was set at 160, which equals the size of one-hot encoding features. The number of nodes in the hidden layer is equal to the expected dimensions, which will be determined in the actual case. $\ell 1$-norm is a term that can be used to improve the prediction quality and its interpretability of modelling based on sparse data. It can be embedded in autoencoder to enhance its capability (Han et al., 2017). Because the one-hot encoding data is sparse, $\ell 1$-norm was introduced in the autoencoder in this case.

The parameter setting of the autoencoder was different from the deep learning algorithms used for modelling. Firstly, due to the output of the autoencoder is binary, and therefore Adadelta, a prevailing optimiser was adopted. Secondly, binary-cross-entropy was chosen as the loss function. Thirdly, the learning rate was set at the default value in Adadelta, which is 1. The batch size was set at three, and the number of epochs was set at 30.

In this study, three scenarios were introduced. In scenario 1, prevailing machine learning and LSTM network were used for modelling based on the sparse data in conjunction with numeric data in historical maintenance dataset. In scenario 2, autoencoder was introduced to convert the one-hot encoding data to low dimension and robust data. Then the prevailing deep learning algorithms were used for modelling based on the robust data in conjunction with numeric data in historical maintenance dataset. Meanwhile, the relation between the algorithm performance and the number of the robust data dimension was explored. In scenario 3, based on the techniques in scenario 2, Cox PHM was introduced to tackle data censoring. In order to explore the impact of data sparsity on the algorithm performance of Cox PHM, two control experiments were set in scenario 3. In the first experiment, data compensation was based on a Cox PHM, which was trained by sparse data in conjunction with the numeric data in historical maintenance dataset. In the second experiment, data compensation was based on Cox PHM, which was trained by robust data in conjunction with the numeric data in historical maintenance dataset. After the censored data was compensated, the compensated censored data and uncensored data was jointly used for modelling using different algorithms. Also, the relation between the difference in reliability and the algorithm performance was explored in this scenario.

4.3.3 Performance Evaluation

Different metrics are needed to evaluate the algorithm performance from different perspectives. In this study, two metrics called Model correlation coefficient (MCC) and root-mean-square-error (RMSE) were chosen to evaluate the performance of algorithms. Both metrics have been widely used to evaluate the results of the regression. They can reveal the algorithm performance in different perspectives. MCC is used to measure the correlation between two variables, and can be expressed mathematically as:

$$\text{MCC} = \frac{S_{PA}}{\sqrt{S_P S_A}}, \tag{4.5}$$

where,

$$S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1};\ S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1};$$

$$S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1};$$

p_i is the predicted value and $\bar{p}$ is the average of the predicted value. a_i is the actual value and the $\bar{a}$ is the average actual value. n is the number of training data.

It is a scale-dependent metric which measures the difference between the prediction value and the actual value. RMSE is 0 if the prediction value equals to the actual values. The expression of RMSE is:

$$\text{RMSE}=\sqrt{\frac{\sum_i (p_i - a_i)^2}{n}} \tag{4.6}$$

4.4 Experimental Results

4.4.1 Scenario 1: Prevailing Machine Learning Algorithms VS. LSTM Network

In this scenario, four deep learning and one prevailing machine learning algorithms were used for modelling based on numeric and one-hot encoding data. One-hot encoding was used to convert the nominal data to binary data, which can be processed by machine learning algorithms. There are three nominal features in the dataset which can be converted to 160 different categories using one-hot encoding technique. The one-hot dataset was then concatenated with the other numeric data in historical maintenance dataset. One-hot encoding enables the nominal data to be converted to numeric form without any information sacrifice. However, with the one-hot encoding features, there was a large number of 0 in the dataset, which leads to significant sparsity in the dataset. After 10-fold cross-validation, the mean and standard deviation (STD) of MCC and RMSE were compared to reveal the algorithm performance. All tests were conducted on an Intel i5-6500 3.20Ghz PC with Nvidia GeForce GTX 1060 graphics card. The training time for each algorithm was marked and used to reveal the computational cost.

The modelling results based on the one-hot encoding data are shown in Table 4.2, which indicate that the LSTM network achieved the highest MCC which is 0.8248 and the lowest RMSE which is 379.8 days. SVM shows the worst performance in this scenario, which RMSE is 432.4 days and MCC is 0.7738. The algorithm performance of LSTM network in terms of MCC and RMSE are better than other algorithms. Moreover, the STD of MCC and RMSE of DCNN are the lowest in this scenario. Although the algorithm performance of LSTM network in terms of MCC and RMSE are better than the benchmarking algorithms, it also requires the longest training time.

Table 4.2 The results of machine learning modelling based on one-hot encoding data

	LSTM Network	RNN	FCNN	DCNN	SVM
MCC_Mean	**0.8248**	0.8221	0.8240	0.8240	0.7738
MCC_STD	0.0122	0.0136	0.0101	**0.0097**	0.0173
RMSE_ Mean (days)	**379.8**	382.1	387.2	387.2	432.4
RMSE_ STD (days)	14.78	13.27	16.62	**12.59**	15.91
Modelling time (s)	259.2	107.5	34.65	43.15	**7.263**

4.4.2 Scenario 2: Modelling Based on Features Converted by Autoencoder

After the nominal data was converted using autoencoder, the one-hot encoding data was then combined with numeric data in historical maintenance dataset to generate a new dataset. The relation between the number of converted features and the algorithm performance in terms of MCC and RMSE are shown in Figure 4.7 and Figure 4.8.

It can be seen that the algorithm performance of all deep learning algorithms in terms of MCC and RMSE fluctuated in the beginning, and then become worse along with the larger number of converted features. In contrast, the algorithm performance in terms of MCC and RMSE of SVM is relatively stable. The algorithm performance in terms of MCC and RMSE of all algorithms reached their lowest points when the number of converted features ranges from 10 to 20. With the consideration of computational cost in TBF modelling stage and algorithm performance, the number of converted features is set at 16 in the following tests.

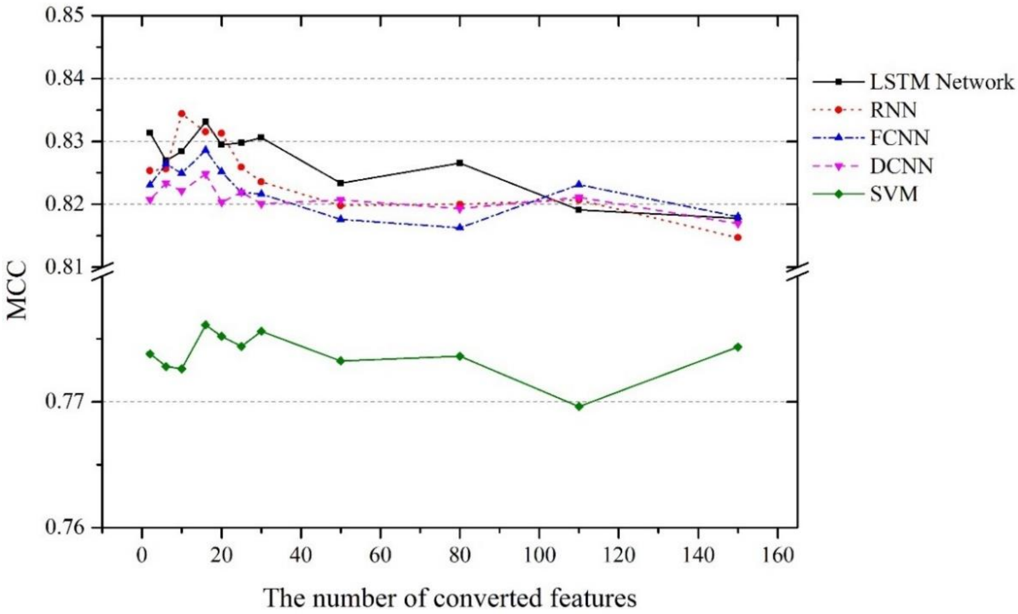

Figure 4.7 The relation between the number of converted features and the algorithm performance in terms of MCC

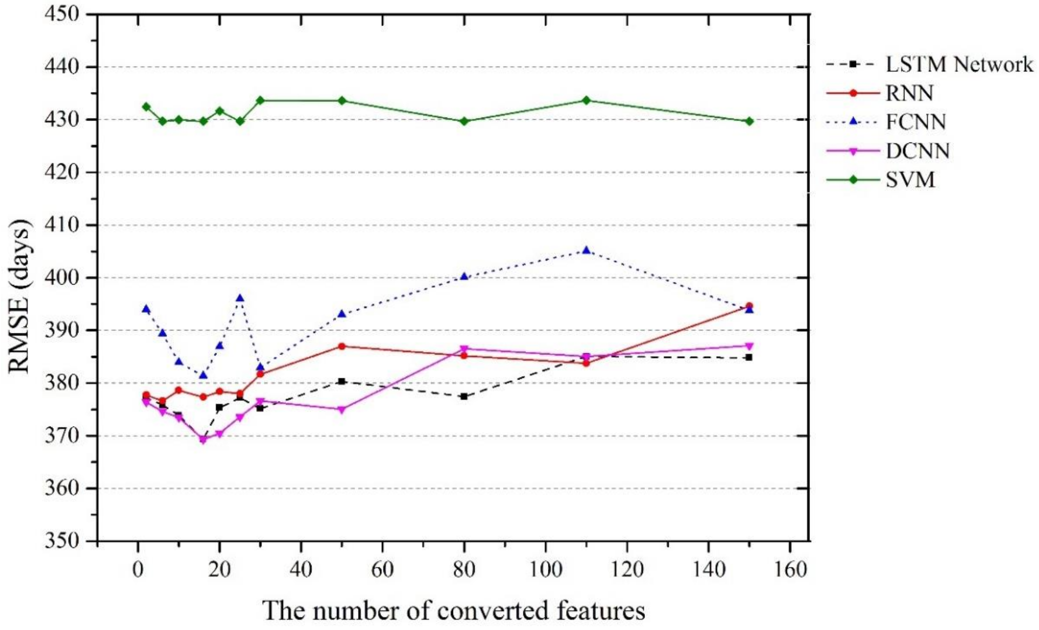

Figure 4.8 The relation between the number of converted features and the algorithm performance in terms of RMSE

After the number of converted features was determined. The converted data was concatenated with the numeric data in historical maintenance dataset and used for modelling. The results of the comparison between the modelling based on one-hot encoding and autoencoding are shown in Figure 4.9 and Figure 4.10. With the help of autoencoder, the algorithm performance in terms of RMSE of all the algorithms witnessed a decrease. LSTM network has still achieved the highest MCC which is 0.8348 and the lowest RMSE which is 369.3 days. When the autoencoder was introduced to convert the one-hot encoding data, the improvement of MCC and the decline of RMSE of the LSTM network are 0.91% and 1.84% respectively. The performance of the RNN in terms of RMSE increased by 1.13% and 2.10%, respectively. Meanwhile, FCNN experienced the largest RMSE reduction in this study, which is 2.23%. In contrast, the performance of SVM in terms of MCC and RMSE are merely increased with the help of autoencoder, while the STD of both MCC and RMSE are decreased dramatically. Finally, the STD of all the algorithms declined with the help of autoencoder. Hence, the results demonstrated that the robust features generated by autoencoder are helpful to improve performance in terms of MCC, RMSE and the stability to all the algorithms used in this study.

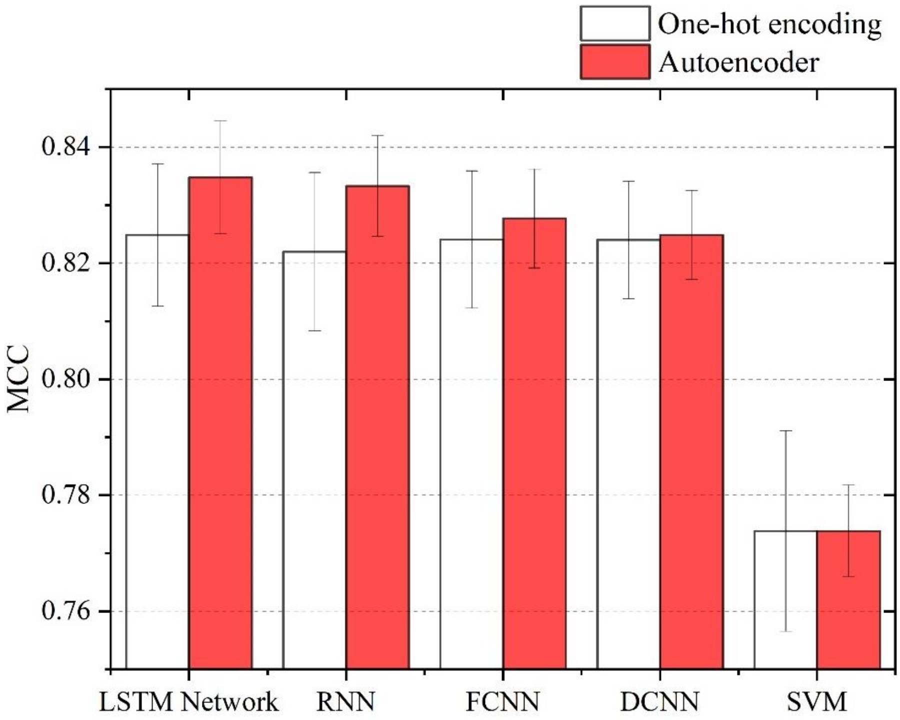

Figure 4.9 The algorithm performance comparison between one-hot encoding-based modelling and autoencoder based modelling in terms of MCC

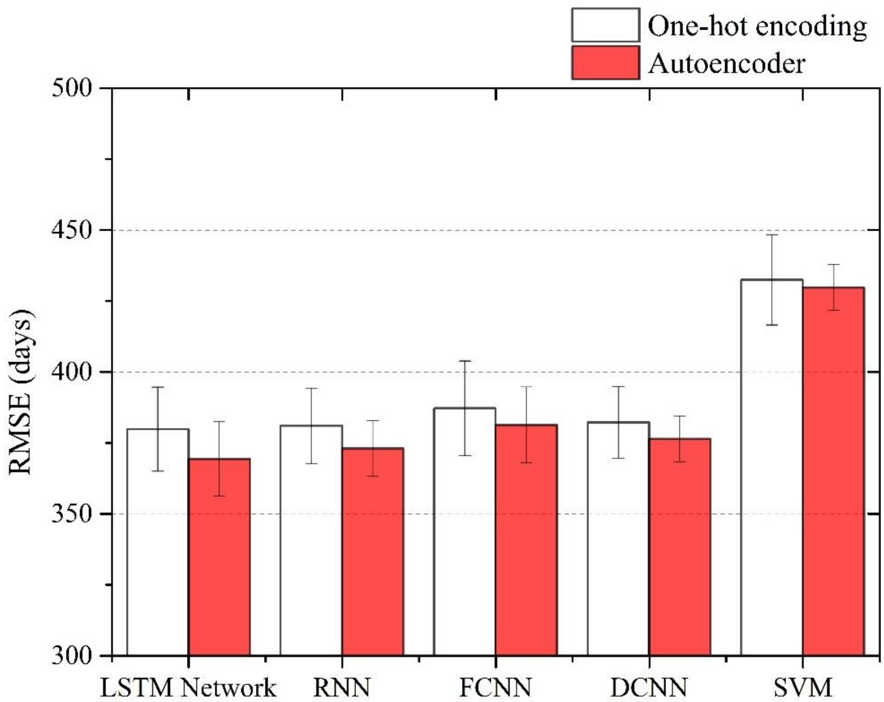

Figure 4.10 The algorithm performance comparison between one-hot encoding based modelling and autoencoder based modelling in terms of RMSE

4.4.3 Scenario 3: Modelling Based on Cox Proportional Hazard Deep Learning

Data censoring leads to the inaccurate label of data. If the censored data is directly used for modelling, it may jeopardise the algorithm performance. If an appropriate compensation for the censored data can be estimated, the algorithm performance could be promoted. There are 2,352 censored instances in this study, which takes 19.2% in the dataset. In order to estimate the actual TBF for the censored data, Cox PHM was introduced into this study. Uncensored data was used to build a Cox PHM.

Appropriate compensation based on the difference in reliability can be beneficial to performance. With Cox PHM, the relationship between reliability and TBF can be estimated. However, when the difference in reliability is too large, the algorithm performance tends to be damaged due to the tuned TBF is inaccurate. The difference in reliability was used to generate compensation for the censored data according to Cox PHM of each censored instance. When a difference in reliability is set, then the compensation of each censored instance can be determined. The censored data with compensation was then used for modelling jointly with the uncensored data. In this case, the ideal difference in reliability needs to be determined. *The* difference in reliability was first set in the range of 0% to 5%. If the algorithm performance can be promoted when the difference in reliability increases from this range, it would then be expended to find the optimal point. Two control experiments were conducted in this scenario to reveal the impact of data sparsity on the algorithm performance of Cox PHM. The results of the first experiment are shown in Figure 4.11 and Figure 4.12.

A Cox PHM was trained by sparse data in conjunction with the numeric data in historical maintenance dataset, which was used for label compensation. It can be seen from Figure 4.11 that the fluctuation of all the deep learning algorithms is considerable. The algorithm performance of SVM in terms of MCC reached its peak when the difference in reliability is 2%, following by a monotonous fall. Also obvious is that, with suitable label compensation, the algorithm performance of different algorithms in terms of MCC were promoted. LSTM network achieved the highest MCC is this experiment, which is 0.8383. With the help of label compensation, the maximum MCC improvement of LSTM network, RNN, FCNN, DCNN and SVM are 0.51%,0.55%, 0.31%, 0.19% and 0.37%, respectively. From Figure 4.12, it is evident that the algorithm performance in terms of RMSE for all the algorithms fell to a low point before a monotonous increase. LSTM network achieved the lowest RMSE in this figure which is 364.5 days. RNN witnessed the largest algorithm performance improvement in terms of RMSE, which is 1.80%.

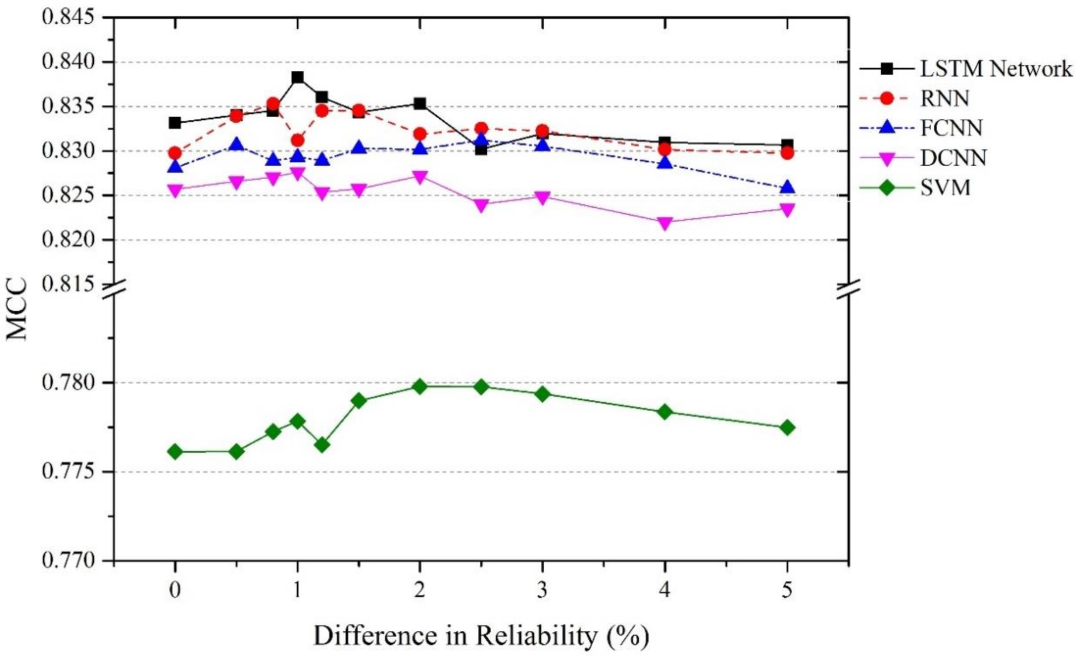

Figure 4.11 The MCC of modelling based on Cox PHM trained by sparse data in conjunction with numeric data in historical maintenance dataset

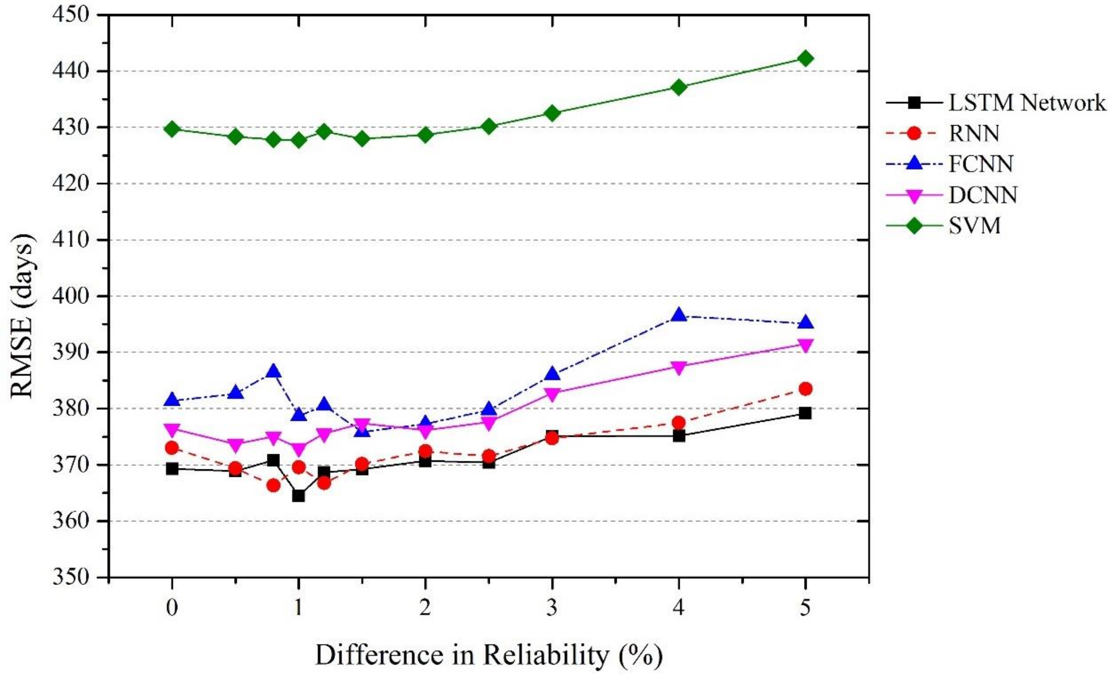

Figure 4.12 The RMSE of modelling based on Cox PHM trained by sparse data in conjunction with numeric data in historical maintenance dataset

With the robust features getting from scenario 2, a reliable Cox PHM trained by the robust data in conjunction with the numeric data in historical maintenance dataset was established. With the consideration of the impact of data sparsity on the algorithm performance of Cox PHM, the robust data in conjunction with the numeric data was used to train a reliable Cox PHM for label compensation. The modelling results are shown in Figure 4.13 and Figure 4.14. It can be seen from Figure 4.13 that the MCC of all algorithms are leveraged when the difference in reliability is chosen appropriately. The MCC of SVM has been increased to the highest point when the difference in reliability grows from 0% to 1.2%. In contrast, the performance of all deep learning algorithms in terms of MCC fluctuated slightly in all stages. The MCC of all deep learning algorithms firstly increased and then reached their peaks when the difference in reliability ranged from 0.5% and 2%, followed by a period of continuously decrease. LSTM network achieved the highest MCC in this scenario, which is 0.8395. The MCC of the LSTM network is higher than that of other algorithms in all the stage. It is also clear in Figure 4.14 that all the algorithms experienced a similar trend when the difference in reliability grows from 0% to 5%. The RMSE of LSTM network, RNN, FCNN and CNN reach their lowest point when the difference in reliability is 0.8% or 1%. The lowest RMSE in this scenario is 359.1 days, which is achieved by LSTM network. Meanwhile, the performance of SVM in terms of RMSE become better when the difference in reliability ranges from 0.3% to 1.2%. The maximum RMSE decline in terms of RMSE of LSTM network, network, RNN, FCNN, DCNN and SVM is 2.75%, 2.56 %, 1.78%, 1.61% and 0.96% respectively. Hence, in terms of MCC and RMSE, the performance of all the algorithms was promoted with the help of the Cox PHM trained by in conjunction with the numeric data in historical maintenance dataset.

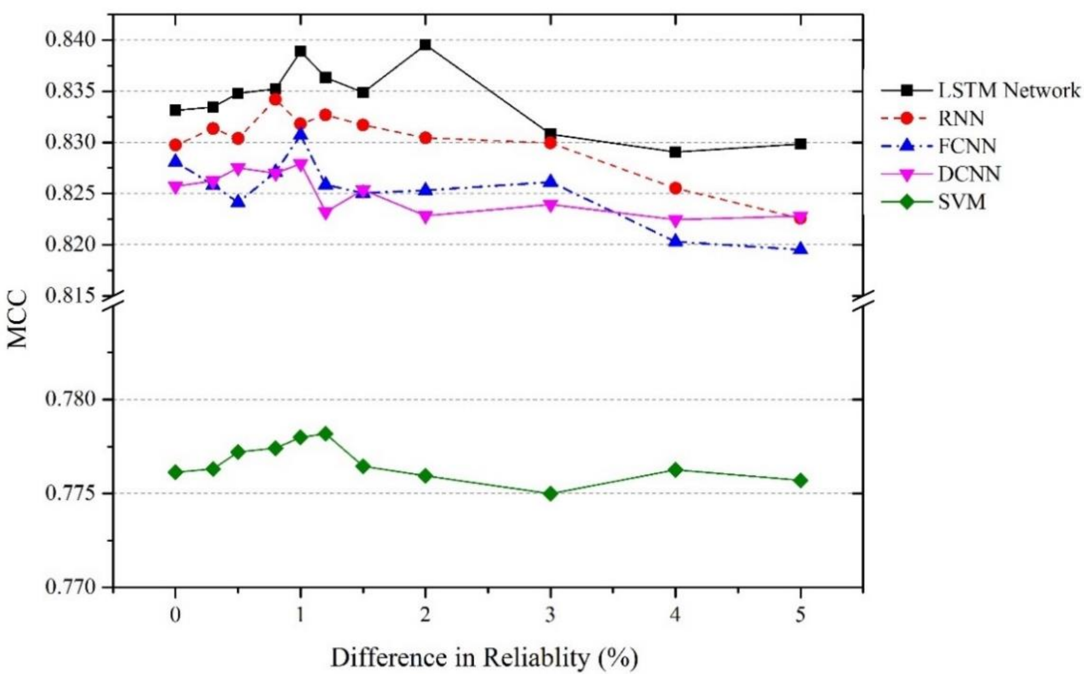

Figure 4.13 The MCC of modelling based on Cox PHM trained by robust data in conjunction with numeric data in historical maintenance dataset

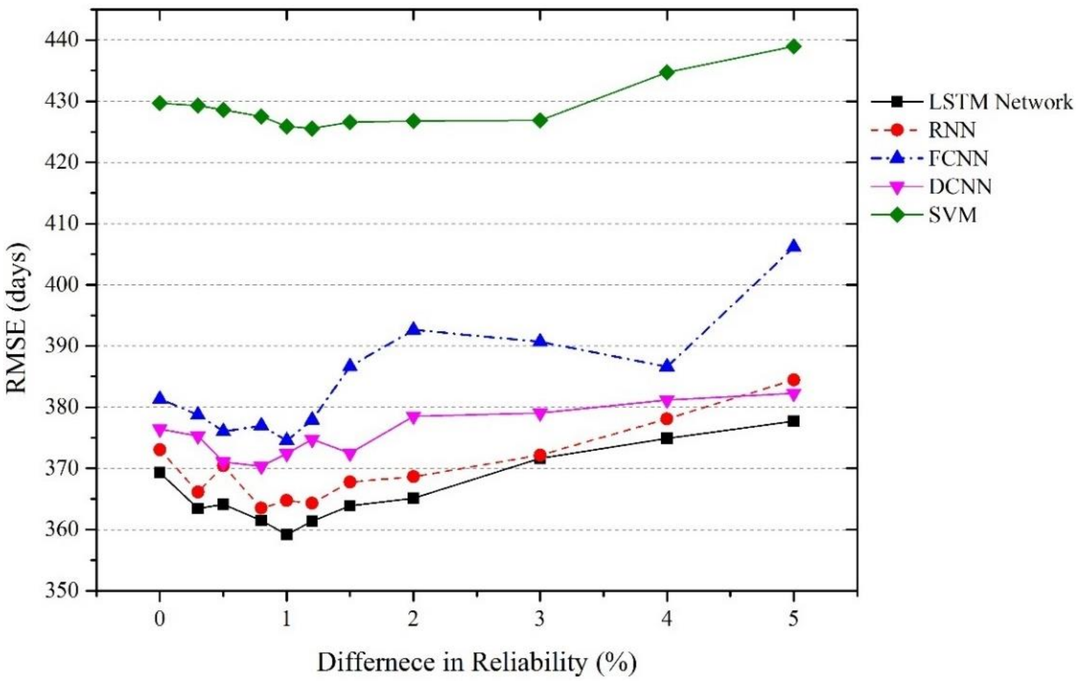

Figure 4.14 The RMSE of modelling based on Cox PHM trained by robust data in conjunction with numeric data in historical maintenance dataset

Figure 4.11-4.14 demonstrated that the algorithm performance in terms of MCC and RMSE of all the algorithms were benefited more from the Cox PHM trained by the robust data in conjunction with the numeric data, which indicates that the data sparsity jeopardises the algorithm performance of Cox PHM. Since LSTM achieved the best algorithm performance in terms of MCC and RMSE, a further algorithm comparison is shown in Figure 4.15. It can be seen that with the solutions of data sparsity and data censoring, the algorithm performance of LSTM network in terms of MCC was promoted by 1.8% and RMSE was reduced by 5.4%. Besides, the standard deviation of MCC and RMSE in 10-fold cross-validation was shrunk with the help of autoencoder and Cox PHM.

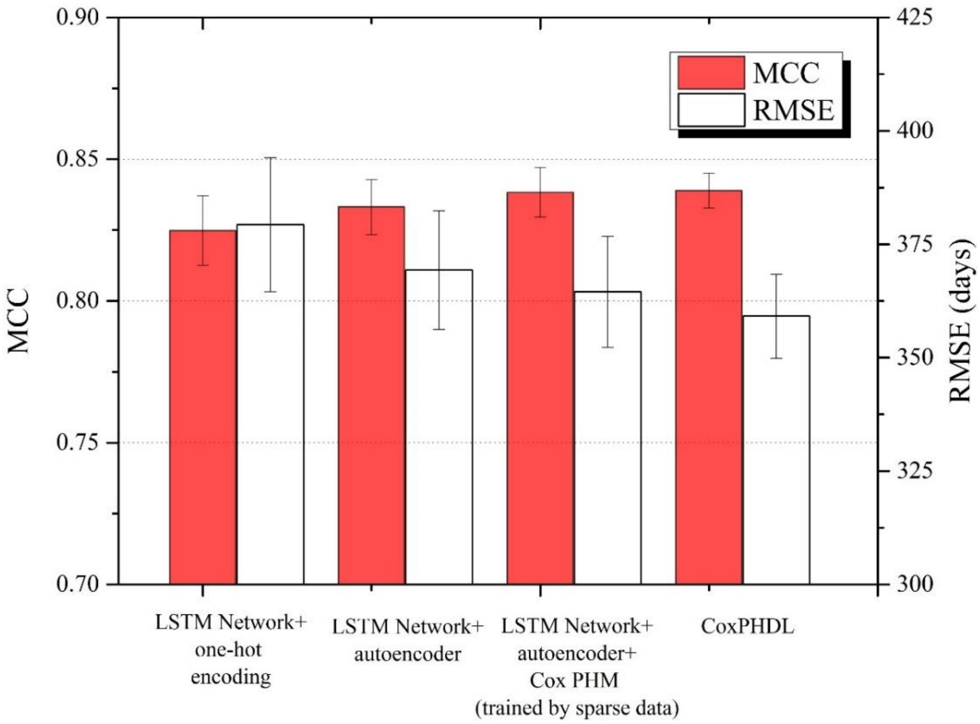

Figure 4.15 The algorithm performance of LSTM in different schemes in terms of MCC and RMSE

4.5 Discussion

PdM is of importance to the industry. The issues of maintenance planning, job scheduling and spare parts inventory management have long been concerned by various industries such as fleet management. With the accurate prediction of TBF, better fleet management can be achieved. Most research in PdM has been conducted using sensor data. However, sensor data collection requires extra expenditure, which is unaffordable for some companies. Different from the existing research, this study focuses on PdM based on the historical maintenance data, which is relatively easy to obtain in the industry.

LSTM network shows merits in TBF modelling. Another existing study investigating LSTM network in RUL modelling also indicates that LSTM shows merits in RUL modelling based on sensor data in comparison with RNN, FCNN and SVM (Zhang et al., 2018b), which indicate that the LSTM network can be a useful tool in PdM. The results derived from this study cannot be compared with the results in the existing study since this is the first time that TBF modelling approach was proposed. The proposed TBF modelling approach will be further validated when another historical maintenance dataset is available.

The results of scenario 1 indicate that the algorithm performance is positive to the computational cost in this study. The LSTM networks achieved the highest MCC and lowest RMSE in scenario 1, while it took the longest training time in comparison with the prevailing machine learning algorithms. What is noticeable in the results of scenario 2 is that the robust data representation converted by autoencoder is useful to improve the algorithm performance of all the algorithms used in this study, which indicates that data sparsity damages algorithm performance in terms of the MCC and RMSE of all the algorithms used in this study. The algorithm performance in terms of MCC and RMSE worsen along with the larger number of converted features. With this knowledge, the autoencoder can be better deployed in future studies.

The results of scenario 3 firstly indicate that the data sparsity also has a negative impact on the algorithm performance of Cox PHM. With the robust features obtained from autoencoder, the effect of label compensation using Cox PHM was enhanced. After the conversion of the categorical variables using autoencoder and the estimation of the label of censored data using CoxPHM, the algorithm performance of the standard LSTM network in terms of MCC and RMSE are generally better than all other algorithms, which indicates it is more sensitive to the data sparsity and data censoring issue than other algorithms used in this study. The proposed technique can improve the algorithm performance in terms of MCC and RMSE of LSTM network, which is 1.8% and 5.4% respectively. The relation between the difference in reliability and algorithm performance was revealed in this study. The ideal difference in reliability in this study ranges from 0.8% to 1.5%, which can offer some insights into the application of CoxPHDL.

4.6 Summary

The prediction of TBF can bring tangible benefits to the industry so as to achieve better maintenance planning, job scheduling and spare parts inventory management. In this chapter, the focus was on the modelling and prediction of TBF based on historical maintenance data. Based on autoencoder, Cox PHM, and LSTM network, we have proposed a new approach called Cox proportional hazard deep learning (CoxPHDL) to predict TBF based on historical maintenance data. In this approach, autoencoder is used to convert the nominal data for Cox PHM and LSTM network. Cox PHM is used to estimate the label of censored data. After the data is pre-processed, LSTM network algorithm is used to build a TBF prediction model. An experimental study was carried out based on real-world automobile historical maintenance data. There are two key findings in our study. Firstly, LSTM network shows merits in TBF modelling in comparison with several prevailing machine learning algorithms, but it leads to a higher computation cost. Secondly, data sparsity shows a negative impact on the algorithm performance of Cox PHM and LSTM network. With the consideration of this issue, autoencoder was deployed to address this issue and promote the algorithm performance of LSTM network and most of the machine learning algorithms. Then, with the help of Cox PHM, the algorithm performance of LSTM network and most of the machine learning algorithms can be further leveraged. In the actual fleet management scenario, this is deemed very useful to improve the job scheduling, automobile maintenance planning and the inventory management of spare parts.

Chapter 5 Deep Learning Embedded Semi-supervised Learning (DLeSSL) for Automobile TBF Modelling

5.1 Introduction

The existing methods used for automobile TBF modelling rely on the quantity of labelled data. In Chapter 4, a large amount of real-world historical maintenance data was used for modelling. However, if the labelled historical maintenance data is limited, its underlying process of modelling and prediction tends to be difficult. In data mining, a technique called semi-supervised learning aims to make full use of the unlabelled data for a supervised or unsupervised learning task (Hady and Schwenker, 2013). In the real world, labelled data is often expensive to be collected as it requires efforts to categorise the data. One of the critical factors which affect the performance of the supervised learning algorithm is the size of the labelled data. In supervised learning, in order to train an accurate model, a certain amount of labelled data is required. When the labelled data is insufficient, machine learning algorithm tends to be challenging in mining the hidden patterns within the data. However, with the help of unlabelled data, the performance of a supervised learning algorithm could be improved (Hady and Schwenker, 2013). In semi-supervised learning, firstly, the unlabelled data was assigned a pseudo label via unsupervised learning. The data with a pseudo label is then used to enrich the labelled dataset. Then the supervised learning approach is used to build a classification or regression model based on the enriched dataset. With larger data size, The enriched dataset can promote the performance of supervised learning

(Zhu, 2011). Deep learning has gained increasing attention and widely used for supervised learning. Owing to the fact that the deep neural networks have a large number of parameters which need to be tuned during the training process, it requires high volume labelled dataset. Hence, it is worthwhile to study how the semi-supervised learning technique could enable deep learning to be applicable when the labelled dataset is small.

The purpose of this study is to establish an automobile TBF prediction model through a big data-driven approach. Since the labelled historical maintenance data is often limited and expensive to obtain, while unlabelled data is abundant in the real-world industry, a semi-supervised learning approach, i.e., deep learning embedded semi-supervised learning (DLeSSL), is proposed to tackle the issue. Based on DLeSSL, unlabelled data can be labelled and compensated using a semi-supervised learning approach that has a deep learning technique embedded so to expand the labelled dataset. An experimental study using a large amount of historical maintenance data shows the merits of the proposed approach. Results derived using the proposed method reveal that LSTM network (DLeSSL based) outperforms the LSTM network (supervised) and LSTM network (label propagation based) when the labelled data is limited. Besides, the effect on performance due to the size of labelled data and unlabelled data is also reported.

This chapter proposed a semi-supervised approach for automobile TBF modelling based on insufficient labelled data. In Section 5.2, the algorithm details of DLeSSL are introduced. Section 5.3 reports the experimental setup and the experimental results are demonstrated in Section 5.4. Finally, Section 5.5 summarises this chapter.

5.2 Deep Learning Embedded Semi-supervised Learning

We assume that there are a labelled dataset and an unlabelled dataset. The size of the unlabelled dataset is significantly larger than that of the labelled dataset. If the labelled data is insufficient to train a model with a decent performance, exploiting the unlabelled data emerges as an available option to improve the algorithm performance. Different kinds of supervised learning algorithms have been introduced for TBF modelling. Such tools show the advantages of building the prediction model based on labelled data. However, when the labelled data is insufficient, the performance of supervised learning approach tends to be unsatisfactory. In order to improve the algorithm performance when the labelled data is limited DLeSSL was proposed. The general flow of DLeSSL is shown in Figure 5.1. The proposed approach can be separated into two stages i.e. the label finding stage and the label compensating stage. The label finding stage is originated from the label propagation algorithm (Zhu and Ghahramani, 2002). In the label finding stage, new labels of the unlabelled data are determined. In the label compensating stage, the new labels are compensated using deep learning technique. When the compensated labels of the unlabelled data are obtained, the labelled data and the unlabelled data with the compensated label are combined as an enriched dataset. The enriched dataset obtained from DLeSSL is used to train a new deep learning model, which is used to predict TBF.

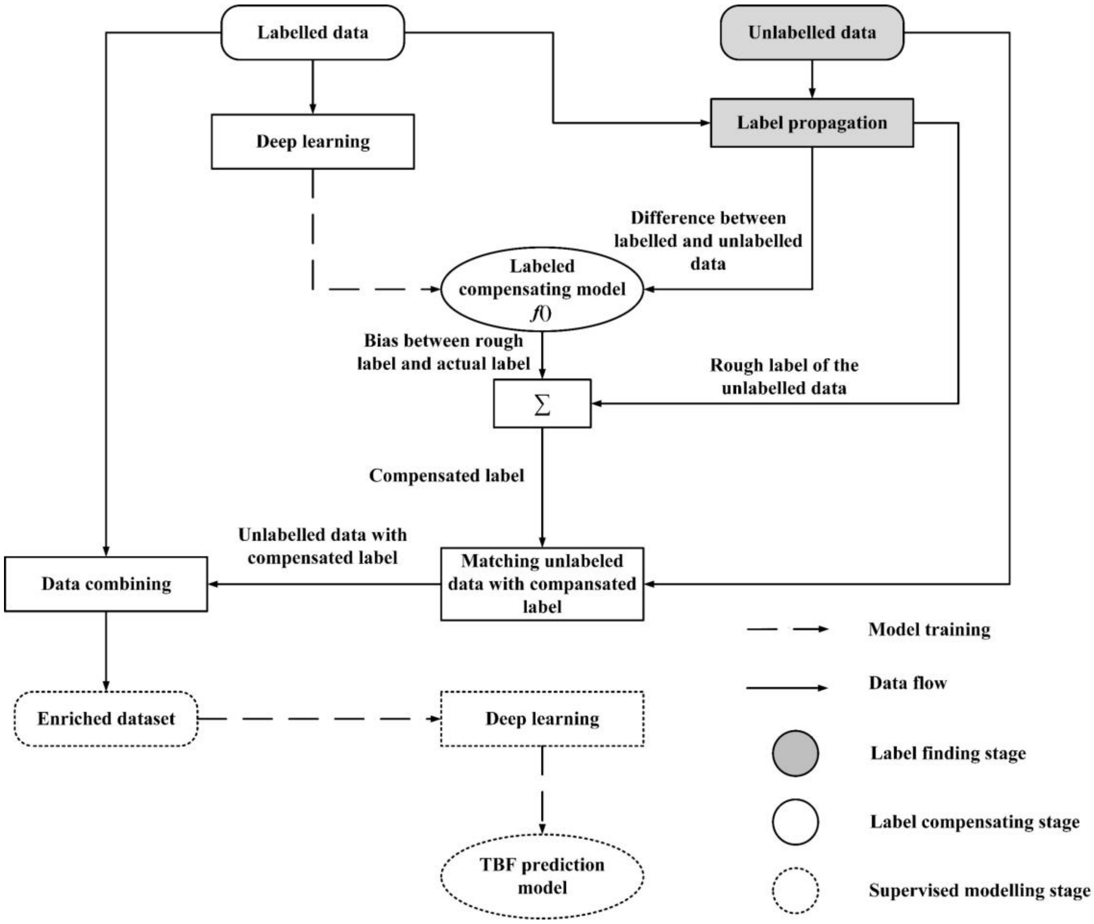

Figure 5.1 The flow chart of DLeSSL

The label finding stage aims to identify the labels for the unlabelled data. Firstly, the most similar labelled instances for the unlabelled data is determined. In this case, we have adopted radial basis function (RBF) to determine the similarity between two instances base on its performance and convenience (Vert et al., 2004). The RBF is used to represent the similarity or distance between l_i and u_j. When the $RBF\ (l_i, u_j)$ close to zero, it indicates the similarity between the label instance l_i and the unlabelled instance u_j is considerably high. An RBF is denoted as:

$$RBF\ (l_i, u_j) = exp\ \left(-\frac{d(l_{i,}\, u_j)^2}{2\sigma^2}\right) \tag{5.1}$$

where σ is a parameter of RBF and d is the Euclidean distance.

Secondly, label propagation is carried out based on the $RBF\ (l_i, u_j)$, which means the most similar instance $\hat{l}_i$ for u_j will be found. Then, the label of $\hat{l}_i$ is propagated to u_j. The details of the label propagation algorithm can be found in the work of (Zhu and Ghahramani, 2002). Basically, when y_i is known and y_j is unidentified, y_i can be considered approximately equal to y_j . The label propagation process can be represented as:

$$(\hat{l}_i, y_i) \rightarrow (u_j) \Rightarrow (l_j, y_i) = (u_j, y_j) \tag{5.2}$$

However, even $\hat{l}_i$ is the most similar instance for u_j in the labelled dataset, it does not mean the new label y_j is equal to the actual label of u_j. There is still a slight difference between y_j and u_j. In order to get close to the actual label of u_j, compensation to y_j need to be carried out. In this stage, u_j is unknown and the bias Δy_j between new label y_j and the actual label of u_j is estimated in the next stage.

In the label compensating stage, we model the relationship between Δl_i and Δy_i by utilising deep learning technique due to its excellent performance in data analytics (LeCun et al., 2015). The number of Δl_i is $N = \frac{I(I-1)}{2}$, which grows rapidly when the number of label data increases. The model which is used to represent the relationship between Δl_i and Δy_i can be modelled using deep learning algorithms. It can be denoted as：

$$\Delta y_i = f(\Delta l_i) \tag{5.3}$$

where $f()$ is the label compensating model training deep learning algorithm.

After the label compensating model $f()$ is obtained, it can be used to determine the bias Δy_j. In the label compensating model $f()$, Δu_j is used as the input of $f()$ to yield Δy_j, which can be denoted as:

$$\Delta y_j = f(\Delta u_j) \tag{5.4}$$

Finally, with the new label y_j and the bias Δy_j, the compensated label $\hat{y}_j$ is denoted as:

$$\hat{y}_j = y_j + \Delta y_j \tag{5.5}$$

When the new label y_j is obtained, the unlabelled data and the label data is combined as an enriched dataset. The enriched dataset yielded by DLeSSL is then used to build a deep learning model to predict TBF.

5.3 Experimental Setup

5.3.1 Data

The data used in this experiment is the same as the data used in Chapter 4. In the dataset, there are three nominal features in the historical maintenance dataset were converted to numeric data using one-hot encoding and autoencoder techniques proposed in Chapter 4. Hence, both techniques were deployed in this study again to convert the nominal data to numeric data. After the nominal data was converted, the maintenance dataset was further pre-processed using Cox PHM to address the data censoring issues of the dataset. Besides, in the data pre-processing stage, the abnormal values in the dataset were then removed and the dataset was normalized. Then, the abnormal data entries were removed. The TBF of some data entries is too low which were deemed as noisy data. Hence, the data entries with TBF lower than 30 days were removed. After this stage, there are 28 features and over 10 thousand data entries in the dataset. Finally, the dataset was reshuffled to yield comprehensive results.

5.3.2 Model Setup

In Chapter 4, LSTM network shows merits in TBF modelling based on historical maintenance data. Hence, it was adopted again in TBF modelling in this experimental study. In the label compensating stage of DLeSSL, a deep learning model is needed to get the compensation values of the actual label. The data used in the label compensating stage is the difference between data entries, which is ordinary numeric data. Hence, an FCNN is adopted in this stage to build a label compensating model. The algorithm details of the FCNN and LSTM network were demonstrated in Table 5.1. Besides the main components mentioned in Table 5.1, other advanced deep learning techniques such as dropout (Srivastava et al., 2014) and regularisation (Cortes et al., 2012) were also adopted to prevent overfitting.

Table 5.1 The network parameters of the deep learning algorithms

Network parameters	FCNN	LSTM Networks
Layer type	Fully connected layer	LSTM layer+ Fully connected layer
Number of layers	4	4
Number of neurons	400	500,500
Optimizer	Adam (Kingma and Ba, 2014)	RMSprop (Bengio and CA, 2015)
Loss function	Mean squared error	Mean squared error
Learning rate	0.05	0.001
Batch size	500	150
Training epochs	40	45

The deep learning models were designed and established using Python Keras package (Chollet, 2015). The targets of modelling are to establish a TBF prediction model when labelled data is limited and examine the impact of labelled data size on algorithm performance.

DLeSSL is composed of two stages: label finding stage and label compensating stage. As deep learning is used in DLeSSL, its parameters need to be well considered. The setting of deep learning used in label compensating stage is basically the same as the FCNN model mentioned in Section 5.3, except the activation function and the size of the nodes in each layer. After trails, linear function was adopted as the activation function in label compensating model as there are both positive and negative values in input data. Meanwhile, the number of nodes was reduced to 400. Because LSTM network achieved the best performance in terms of MCC and RMSE in TBF modelling in our previous study, both metrics adopted in this section. Meanwhile, 5-fold cross-validation was implemented.

In this study, two scenarios were designed to evaluate the performance of DLeSSL. In scenario 1, only a small part of the instances in the dataset were selected as labelled data. The size of labelled instances was set at different values in the range from 50 to 1000. The rest instances in the dataset were used as unlabelled data, which labels were removed. In the modelling stage, after the compensated labels of the unlabelled data were obtained, the labelled data was combined with the unlabelled data with the compensated labels to an enriched dataset, which was used to train a TBF prediction model using deep learning. In order to reveal the performance of DLeSSL, a deep learning algorithm based on label propagation algorithm was adopted as a baseline. Meanwhile, a deep learning algorithm using a small size of labelled data was also adopted as a baseline. It is denoted as deep learning (supervised). Label propagation algorithm is a semi-supervised learning algorithm that used to determine the label of unlabelled data (Zhu and Ghahramani, 2002).

In scenario 2, whether DLeSSL can boost the algorithm performance of LSTM network in terms of MCC and RMSE was investigated. The labelled data is sufficient to train a LSTM network with satisfactory performance in this scenario. The label of the extra unlabelled data was determined using DLeSSL and added into the training dataset. The relationship between different amounts of unlabelled data and the algorithm performance was revealed.

5.4 Experimental Results

In this stage, DLeSSL was introduced into our study to label the unlabelled data. There are two purposes in this section. On the one hand, in what extent DLeSSL can be beneficial to the algorithm performance using a small size of labelled data and a large size of unlabelled data needs to be determined. On the other hand, what is the suitable range for the ratio of labelled data and unlabelled data for DLeSSL needs to be investigated. LSTM network was used for modelling. There are three different scenarios in this section, which are DLeSSL, label propagation and supervised learning.

Figure. 5.2 shows the relation between the number of labelled data and the MCC of LSTM network based on different algorithms, where the x-axis represents the number of labelled data and the y-axis represents MCC. It can be seen that the performance of the LSTM network (DLeSSL based) in terms of MCC is the highest before the data size reach 1800. Starting at 0.372, there is a dramatic increase in performance when the number of labelled data increased from 10 before the growth slow down after 1000.

In the meantime, the performance of the LSTM network (label propagation based) in terms of MCC are lower than that of the LSTM network (DLeSSL based). The trend of the performance of the LSTM network (label propagation) in terms of MCC is similar to that of the LSTM network (DLeSSL based). The performance of the LSTM network (label propagation based) in terms of MCC is obviously lower than that of LSTM network (DLeSSL based) when the data size locates in the range from 500 to 800. When the data size reaches 1000, the merit of the LSTM network (DLeSSL based) is not obvious anymore.

In this study, the LSTM network (supervised learning based) was only trained by labelled data. It can be seen that when the data size is limited, the algorithm performance of LSTM network (supervised learning based) in terms of MCC fluctuates at -0.1. When the data size reaches 1200, the algorithm performance of LSTM network (supervised learning based) in terms of MCC rises significantly. After the data size reaches 2000, the algorithm performance of LSTM network (supervised learning based) surpasses the MCC of both semi-supervised learning approach.

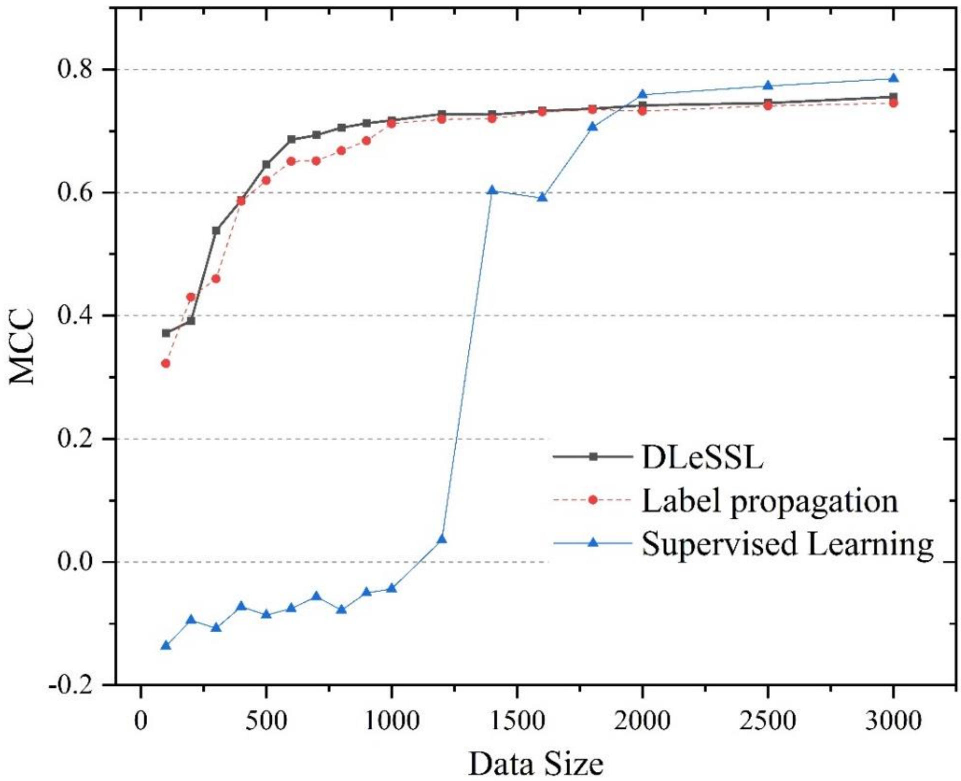

Figure 5.2 The relation between the number of the labelled data and the MCC of LSTM network based on different algorithms

Figure. 5.3 shows the relation between the number of labelled data and the RMSE of the LSTM network based on different algorithms. It can be seen that the algorithm performance of both LSTM network (DLeSSL based) and LSTM network (label propagation based) in terms of RMSE experienced rapid decrease before it tends to converge. The difference of RMSE in most stages between LSTM network (DLeSSL based) and LSTM network (label propagation based) is approximately 20 days. Different from the results of MCC, the algorithm performance of LSTM network (supervised learning based) in terms of RMSE decreased gradually. When the data size reaches 2500, the algorithm performance of LSTM network (supervised learning based) in terms of RMSE becomes the lowest in comparison with both semi-supervised learning algorithms.

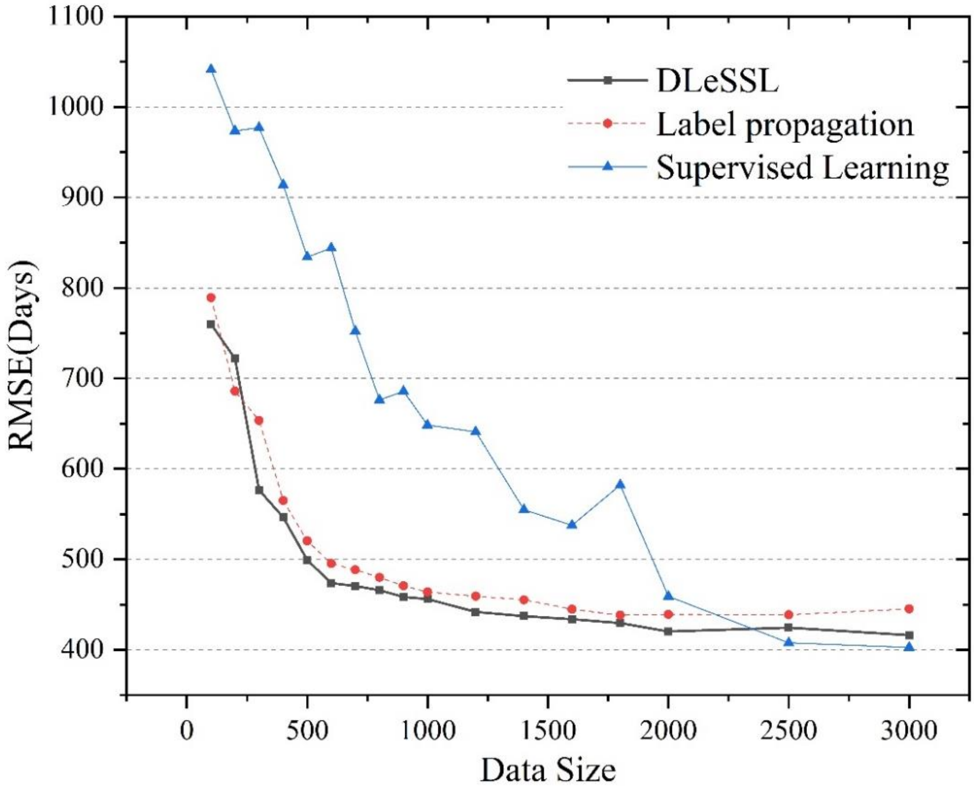

Figure 5.3 The relation between the number of labelled data and the RMSE of LSTM network based on different algorithms

It also can be seen from both figures that when the number of labelled data is over 600, the performance of the LSTM network (DLeSSL based) in terms of MCC and RMSE tends to converge. Moreover, when the number of labelled data range from 300 to 1000, LSTM network (DLeSSL based) shows merits. When the labelled data increases to 1000, the difference in performance in terms of MCC between deep learning (DLeSSL based) and deep learning (label propagation based) is not considerable. Furthermore, when the data size reaches 2500, the algorithm performance of LSTM network (supervised learning based) exceeds both semi-supervised learning algorithms.

Hence, the range of labelled data between 300 to 1000 can be deemed as a suitable range for DLeSSL. The data size of labelled and unlabelled data in total is 8600. The suitable range for the ratio of labelled data and unlabelled data is 3.6% to 13.2% for DLeSSL algorithm.

5.5 Discussion

It is noticeable from the results that when the data size is insufficient, LSTM network shows poor performance in terms of MCC and RMSE. Moreover, the performance of the LSTM network tends to be unstable when the data size is small, which can be seen from the fluctuated trend of LSTM network (supervised). The experimental results demonstrate that the performance of the LSTM network (DLeSSL based) is better than the LSTM network (label propagation based) and LSTM network (supervised) when labelled data is limited. Hence, the experimental results indicate that DLeSSL can be a useful tool in automobile TBF modelling. In this case, the algorithm performance of the LSTM network (DLeSSL based) in terms of MCC and RMSE is 0.717 and 456.1 days when the labelled data size is 1000. In order to achieve similar performance, LSTM network (supervised learning based) needs 2000 labelled data samples. With the help of unlabelled data, the requirement of the labelled data can be decreased by approximately 50%.

When the labelled data size reached 1500, the algorithm performance of the LSTM network (DLeSSL based) in terms of MCC and RMSE tends to converge. The best results of MCC and RMSE that the LSTM network (DLeSSL based) achieved in this experiment were worsen than the results obtained from Chapter 4, which are 0.8395 and 359.1 days. The reason can be that the assumption mentioned in Section 2.5 has not been sufficiently fulfilled. In the historical maintenance dataset used in this experiment, some data entries that have similar features while the TBF varies dramatically. This is because the features in the historical maintenance data have not covered the features relevant to the automobile TBF. Hence, the smooth assumption cannot be perfectly fulfilled. When more features relevant to automobile TBF can be collected, the algorithm performance of the LSTM network (DLeSSL based) is likely to be further promoted.

To the best of my knowledge, most of the existing efforts in semi-supervised learning have tended to investigate the algorithm performance. However, the deployment of semi-supervised learning also needs to be studied. The identification of the relationship between the ratio and performance can be beneficial to the deployment of the semi-supervised approach in the actual case. In this case, we determined a suitable range from 3.6% to 13.2%. This insight can be helpful in the deployment of DLeSSL in the actual case. Moreover, DLeSSL is used to find and compensate the label of the unlabelled data. By using this method, unlabelled data can be used to help the supervised learning task. Several existing semi-supervised learning algorithms only use unsupervised learning method to determine the label for unlabelled data. Different from the existing efforts, both unsupervised learning (label finding) and supervised learning (label compensating) are used in DLeSSL to determine the label of unlabelled data. In the future, the difference between the proposed approach and the existing semi-supervised learning algorithms will be studied to further reveal the performance of DLeSSL and find potential ways to improve it.

5.6 Summary

In order to address the issue that labelled data is difficult to obtain in the real world, a new semi-supervised approach called DLeSSL (deep learning embedded semi-supervised learning) for TBF modelling using a small size of labelled data and a large size of unlabelled data was proposed. Different from the existing efforts that use an unsupervised learning algorithm to determine the label of unlabelled data, DLeSSL uses both unsupervised and supervised learning algorithm to label the unlabelled data. This approach consists of two stages: the label finding and the label compensating stage. The aim of DLeSSL is to determine and compensate the labels of unlabelled data using deep learning technique. Thereafter, the new dataset yielded by DLeSSL is used to train a deep learning model to predict the automobile TBF. A case study is carried out using real-world historical maintenance data. The results have shown the merits of the proposed approach. Experimental results also have indicated that with the help of DLeSSL, LSTM network tends to yield better performance when labelled data is scarce. Meanwhile, our finding highlights the suitable range for the ratio of labelled and unlabelled data in this study, which can offer insights to the actual deployment of the proposed approach. In PdM, DLeSSL can be a useful tool in automobile TBF modelling when labelled data is limited.

Chapter 6 M-LSTM Network-based Predictive Maintenance Enriched by GIS Data

6.1 Introduction

In PdM, most research has studied the equipment lifecycle only based on the sensor data or historical maintenance data. However, to the best of our knowledge, there is no research which considers the impact of GIS factor on the product lifecycle. For an automobile, its lifecycle can be affected by various factors such as weather, traffic, and terrain. Introducing these GIS data into the study of automobile PdM can improve the automobile TBF prediction accuracy, which can help a fleet management company to adjust its maintenance management. The previous study in Chapter 4 introduced weather data into automobile TBF modelling using deep learning. The historical maintenance data and weather data were directly concatenated and fed into the neural network for model training. The experimental results showed that prediction accuracy can be promoted with the help of weather data. However, the data integration of historical maintenance data and GIS data can be further investigated.

Firstly, how to integrate historical maintenance data and GIS data needs to be considered. Automobile historical maintenance data is originated from the automobile maintenance record. In comparison with the sensor data which can reveal the automobile health status, historical maintenance data records the automobile information in a maintenance event. Each data entry in historical maintenance data contains the automobile basic information such as mileage, age, last time to repair and model in the automobile start date (i.e. the date automobile after maintenance or first used). The output of each data entry is next TBF. Hence, a historical maintenance data entry is collected in a specific point-in-time. Besides, historical maintenance data can be further classified as two specific types which are sequential data and ordinary numeric data. The data which has sequential property is relevant to the maintenance history such as automobile age, the repaired time and mileage. Because the next TBF of an automobile tends to be shorter than the last TBF, the sequential features in last TBF is relevant to the next TBF (Wang et al., 2018c). The ordinary numeric features in historical maintenance data are constants such as model. In the study of Chapter 4 and Chapter 5, all the ordinary numeric features are considered as sequential data and used to train an LSTM network model. The algorithm performance in terms of MCC and RMSE is better than that of FCNN. However, when the number of ordinary numeric features increase, this algorithm performance of LSTM network could be compromised. A new data integration technique needs to be explored for automobile TBF modelling based on these two types of data.

A historical maintenance data entry can only be collected when maintenance is implemented. There are two types of data in GIS data which are sequential data and ordinary numeric data. If a feature such as temperature is changing alone with time, it is classified as sequential. Otherwise, it is classified as ordinary numeric data. The sequential GIS data can be collected chronologically and in a higher frequency. For example, the weather data can be collected daily, weekly and monthly, etc. If a GIS feature is collected monthly in a year, it is a one-dimension array with 12 samples. If multiple GIS features are collected in a long period according to a specific sampling frequency, a two-dimensions array can be obtained. Figure 6.1 shows the automobile historical maintenance data and GIS data. The features in historical maintenance dataset, which contains both sequential and ordinary numeric features, are a one-dimension array. Because the historical maintenance data and GIS data are complex in data types and granularities, it is necessary to investigate a solution that can introduce GIS data into TBF modelling.

This chapter proposed a merged-LSTM Network for automobile TBF modelling based on historical maintenance data and GIS data. In Section 6.2, the methodology about how to introduce GIS data into TBF modelling and the algorithm details of M-LSTM was introduced. Section 6.3 reports the experimental setup, and the experimental results were demonstrated in Section 6.4. Finally, Section 6.5 discusses the experimental results and Section 6.6 summarises this chapter.

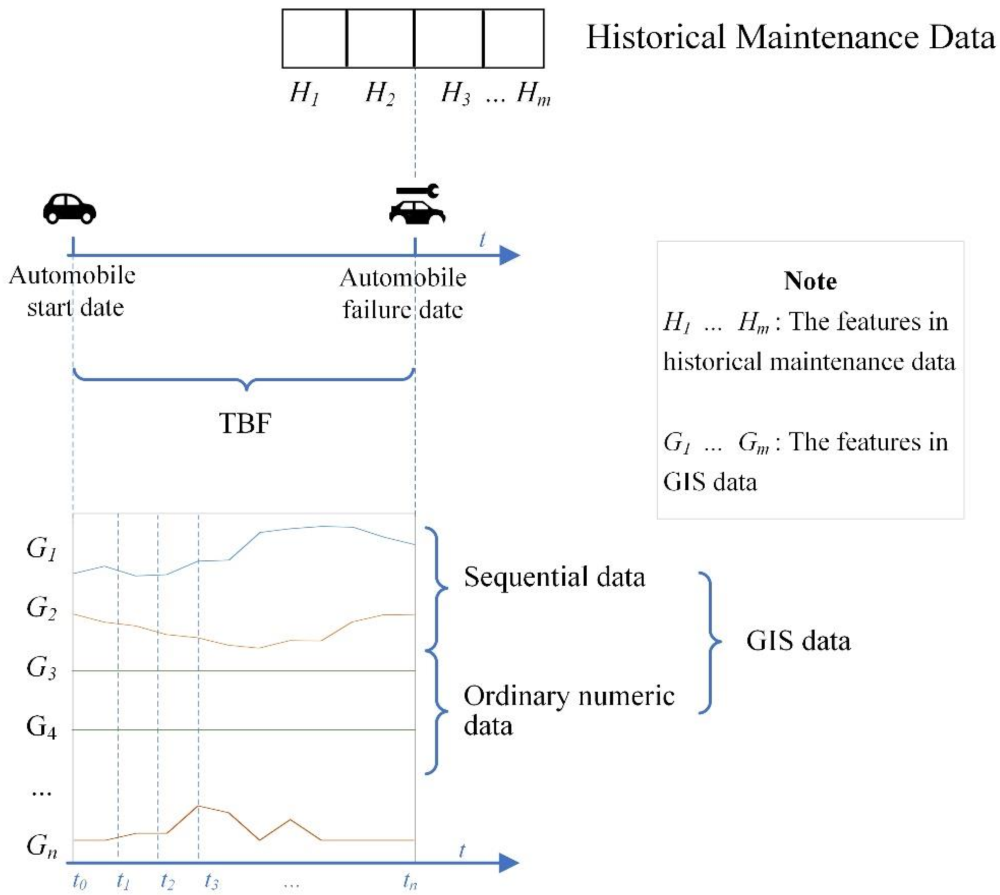

Figure 6.1 The automobile historical maintenance data and GIS data

6.2 Methodology

This study proposed an approach to introduce GIS data into the study of PdM. In the proposed approach, firstly, the maintenance data need to be collected from the garage of a fleet management company, while the raw GIS data need to be collected and then mapping according to the automobile working area to obtain mapped GIS data. Then the mapped GIS data is pre-processed and then sent into an M-LSTM network in conjunction with the sequential data and ordinary numeric data obtained from historical maintenance data. The flow chart of the proposed approach is shown in Figure 6.2.

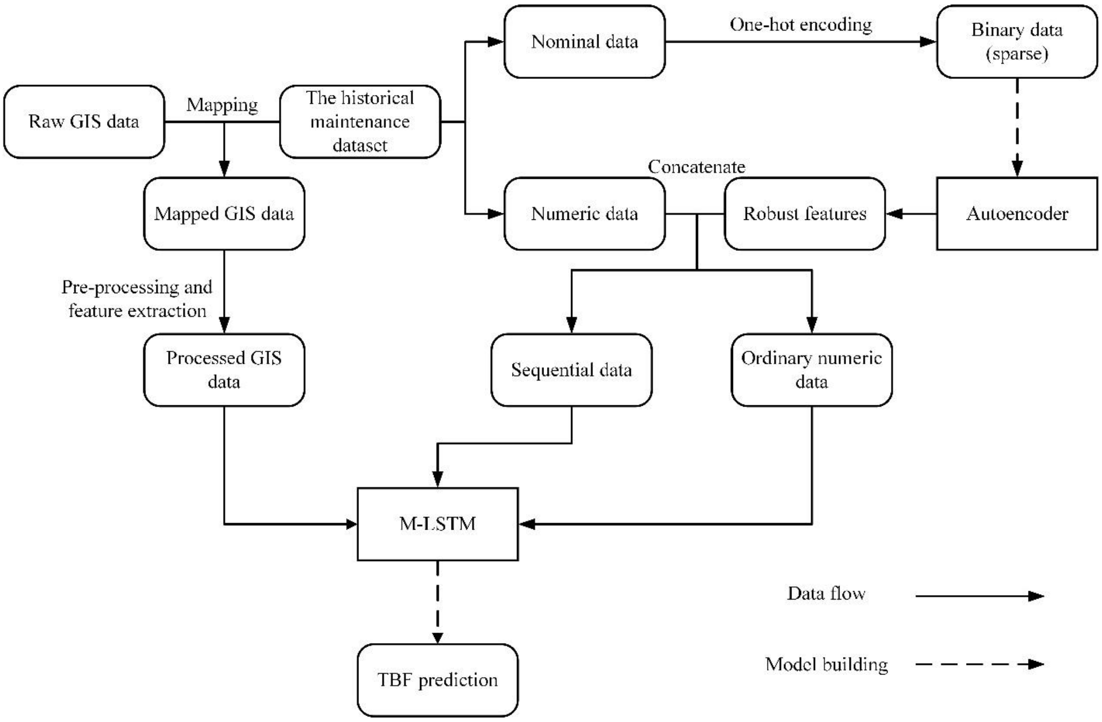

Figure 6.2 The flowchart of the proposed approach

6.2.1 GIS Data Collection and Pre-processing

Automobile lifecycle can be affected by various factors includes weather, traffic and terrain. A fleet management company processes a large number of automobiles which the working environment can be very different in terms of the GIS factors mentioned above. Hence, the GIS data of a specific working area need to be summarised and extracted using GIS software. Temperature and humidity can impact the lifecycle of an asset (Lüttenberg et al., 2018). Weather data such as temperature and rainfall of a specific working area can be obtained from the weather observation stations within the working area. Meanwhile, automobile lifecycle can be affected by the traffic condition. In an area with heavy traffic, the frequency of acceleration and deceleration tends to be higher, which may accelerate the failure of an automobile. Traffic data of a specific working area such as traffic flow statistics can be collected from the traffic department. The terrain is another aspect that can impact automobile life. The mountainous area with a large number of ramps can accelerate the failure of the automobile. The terrain data regarding elevation and slope in a specific working area can be analysed and extracted from the elevation map in GIS software. The taxonomic graph of the historical maintenance data and GIS data is shown in Figure 6.3. It is interesting to note that some GIS factors such as weather and traffic are obviously changing alone with time, while other factors such as terrain are relatively stable in a period. In other words, weather and traffic data can be considered as sequential data, while terrain data can be considered as ordinary numeric data in the study of PdM.

After the GIS data was mapped according to the automobile geographic location, it needs to be pre-processed before it is used for modelling. For the ordinary numeric data such as terrain, it can be directly used for modelling. For the sequential data such as weather and traffic, it would be challenging that using these sequential data for modelling due to the different granularities. Hence, the mean value and standard deviation of the sequential GIS data were extracted and then concatenated with the ordinary numeric GIS data. Finally, the processed GIS data can be obtained.

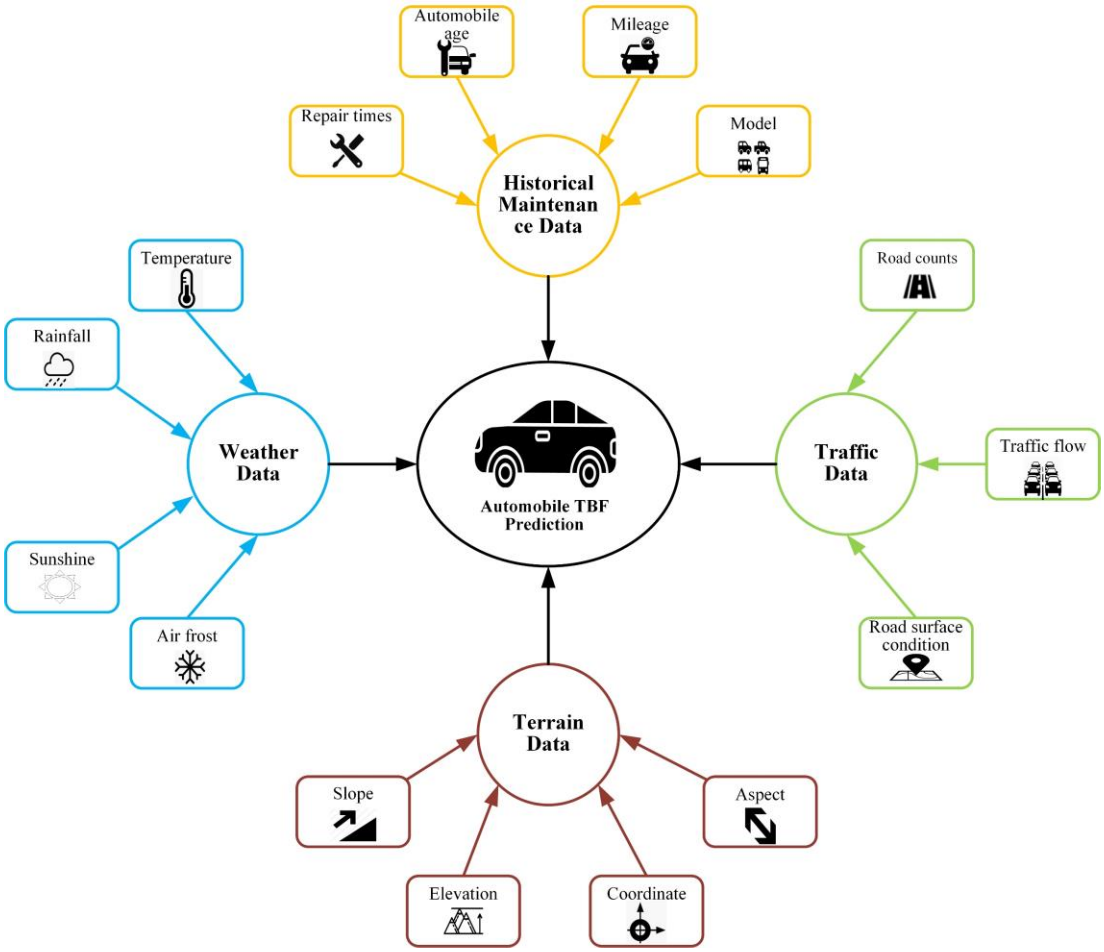

Figure 6.3 The taxonomic graph of the historical maintenance data and GIS data

6.2.2 M-LSTM Network

Since the conventional deep learning algorithms can only process single data type such as ordinary numeric data and sequential data, a novel deep learning structure called M-LSTM is designed for modelling TBF based on both sequential data and ordinary numeric data in historical maintenance dataset and GIS data. Since the ordinary numeric data in historical maintenance dataset and processed GIS data come from different sources, designing two sub-networks for both data can be beneficial in hidden patterns learning. The structure of M-LSTM takes the advantages of LSTM network and fully connected network to handle the sequential data and ordinary numeric data simultaneously. Figure 6.4 shows the structure of M-LSTM. There are four major parts in M-LSTM:

1. **Ordinary numeric data processing path:** Since an ordinary numeric data entry in historical maintenance data is a one-dimensional vector, the ordinary numeric data obtained from the last section can be directly processed by fully connected layers in a neural network. Hence, a two-layers fully connected sub-network is designed in this path. The technical details of fully connected layer can be found in Section 2.3.1.
2. **Sequential data processing path:** LSTM network is expertise in learning the sequential patterns within data. The technical details of the LSTM network can be found in (Hochreiter and Schmidhuber, 1997). The sequential data obtained from historical maintenance data can be further processed to three-dimensional format (i.e. features, data size and time step). In order to learn the hidden patterns within the sequential data, two LSTM layers are deployed in this path. Moreover, a flatten layer is set to transform the output of LSTM layer for further data integration.
3. **GIS data processing path:** Since the processed GIS data is ordinary numeric data, a subnetwork consists of two fully connected layers is designed. The key features of GIS data relevant to automobile lifecycle are expected to be learnt and then passed to the data fusion stage.
4. **Data integration path:** After the abstract representation is learnt by both paths mentioned above, a data fusion path is needed to fuse the representation and implement the regression task to predict the automobile TBF. In this path, a concatenate layer is deployed to concatenate the output from ordinary numeric data processing path and sequential data processing path. Then, two fully-connected layers are employed to further concatenate the representation and learn the hidden patterns relevant to automobile TBF.

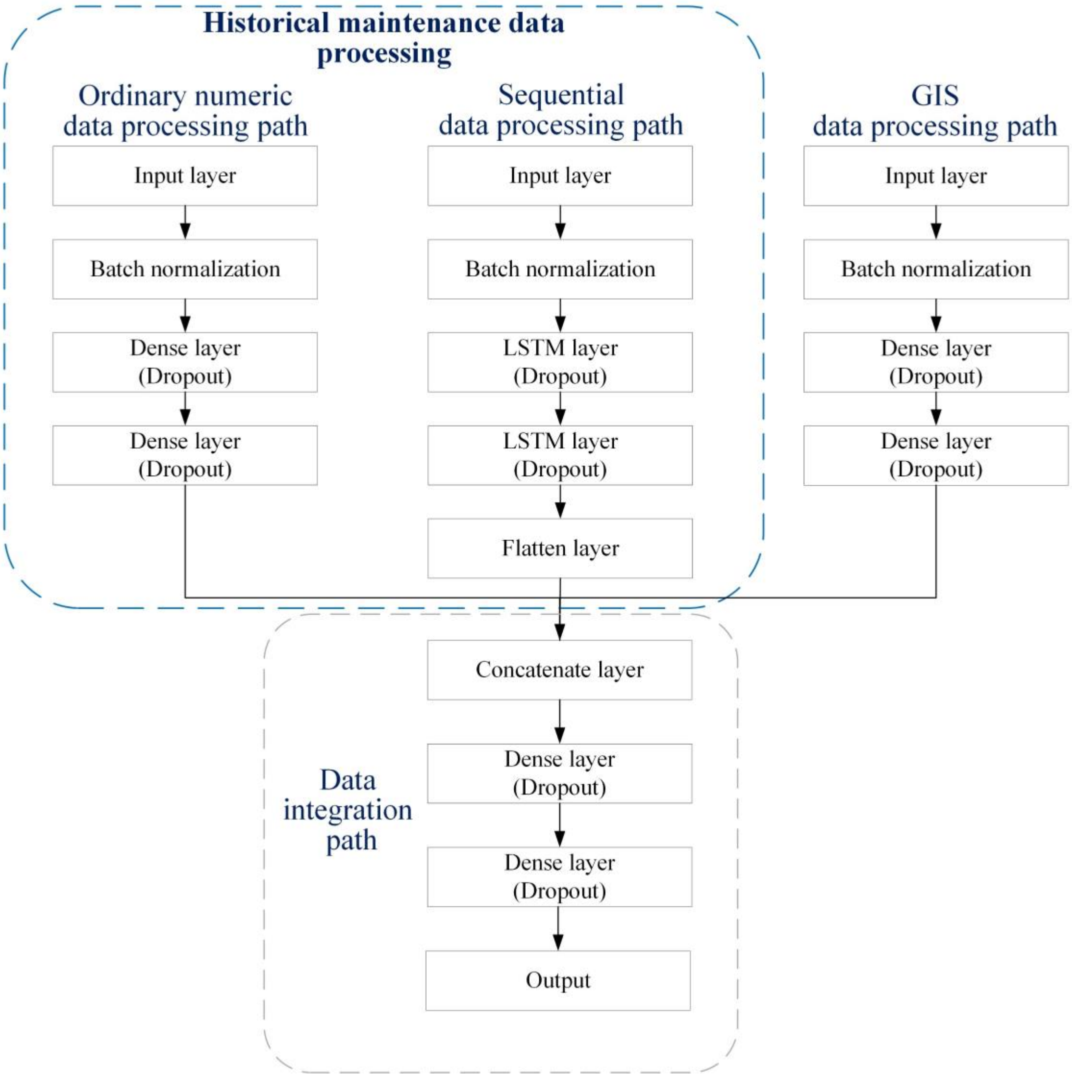

Figure 6.4 The structure of M-LSTM network

There are several essential components and parameters in M-LSTM network, such as optimiser, the number of neurons and batch size, need to be determined in the actual case. Besides, in order to avoid overfitting, batch normalisation, l_2 regularizer and dropout techniques are deployed in M-LSTM network.

6.3 Experimental Setup

6.3.1 Data

The historical maintenance data used in this study is the dataset pre-processed in Chapter 4. The autoencoder and Cox PHM were deployed to address the data censoring and data sparsity issues. The GIS data was collected according to the mobility area and time. Firstly, there are over 60 garages in the fleet management company. The garage location was set as the centre of the mobility area for an automobile. A circular area with a mobility radius of 30 km was set as the mobility area for the automobiles in the same garage. All the garage location and mobility area were plotted in ArcGIS (Price, 2010) software. Figure 6.5 shows two examples of garage and mobility area. Secondly, the GIS data in this area between the automobile start date and failure date was extracted and summarised. There are three types of GIS data, which are weather, traffic and terrain, were introduced in this study.

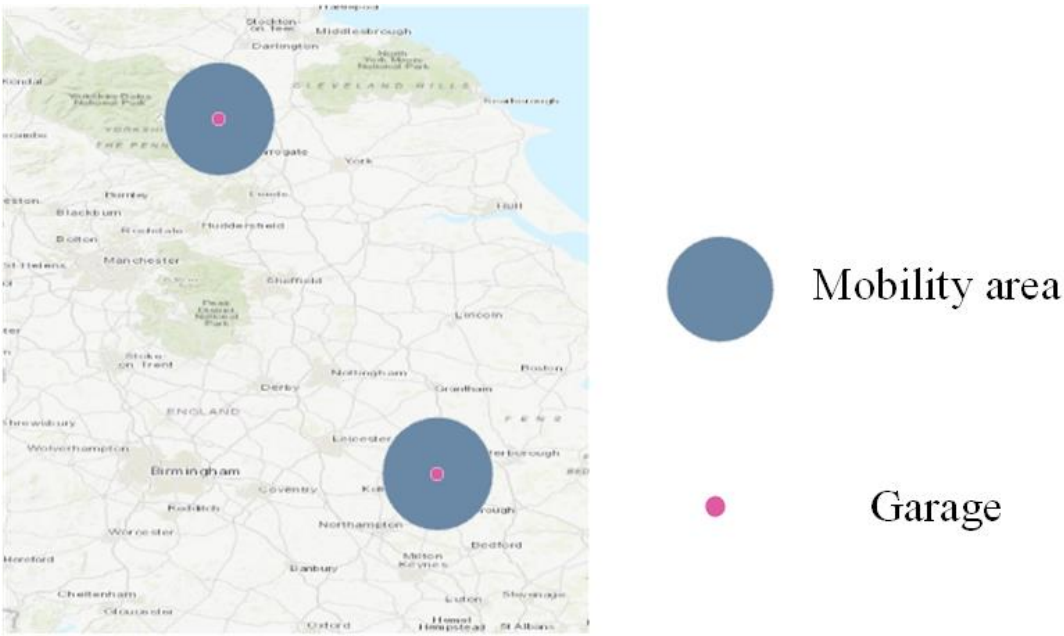

Figure 6.5 Examples of automobile garage and mobility area

Weather condition may affect the performance of an automobile. The weather data was collected from the website of the MET office, UK (2018). There are approximately 40 weather observation stations all around the UK with data collected from 2009 to 2017. Some mobility areas may cover multiple weather observation stations and therefore the mean value of the data from these weather observation stations was adopted. Meanwhile, some other mobility areas do not cover any weather observation station, linear approximation based on the data of several near weather observation stations was implemented to yield the weather data in these mobility areas. The weather data used in this study was sampled monthly and includes the following features: *Name of observation station, Time, Rainfall, Max_temp, Min_temp, Days of air frost,* and *Sunshine*. The weather dataset is shown in Appendix B1. With the name of an observation station, the location of weather observation station can be identified and plotted in ArcGIS software. Then, the weather data in each mobility area was summarised.

Traffic condition is another impact that can affect automobile lifecycle. The traffic data was collected from the public dataset of the department of transport, UK (Department-for-Transport, 2020). The traffic dataset is shown in Appendix B2. The traffic data of over 200 local authorities all around the UK, which was collected from 2009 to 2017, were available. The traffic data was sampled yearly and includes the following features: *Local authority name, Year, Link length (km), Link length, Cars and taxis*, and *All motor vehicles*. In the above features, *Link length (km)* and *Link length (miles)* can be converted to each other. Hence, *link length (km)* was dropped. Similar to the process of weather data, the local authority name is used to identify the locations of the local authority in ArcGIS software, before the traffic data of each mobility area was summarised.

The automobile lifecycle also can be impacted by the terrain condition. The terrain data used in this study was extracted from the elevation map in ArcGIS software. The terrain features of a mobility area can be directly extracted and summarised. The terrain data includes the following features: *Mean elevation, Maximum elevation, Minimum elevation, STD of elevation, Mean slope, Maximum slope, STD of slope, Mean aspect*, *Longitude* and *Latitude*. The terrain dataset is shown in Appendix B3. The summary of all the GIS data is shown in Table 6.1.

Since the weather data and traffic data were collected monthly and yearly. It is challenging to directly used these for TBF modelling such datasets have three dimensions which are features, time and location. In order to integrate these data with historical maintenance dataset, four key statistical features of each weather attribute and traffic attribute were extracted, which are mean value (M), standard deviation (StD), peak to peak (PtP) and skew (S) were extracted.

Table 6.1 The summary of the GIS data

Types	*Sampling frequency*	*Feature*	*Description*	*Feature*	*Description*
Weather	*Monthly*	*Rainfall*	The rainfall (mm)	*Max_temp*	The maximum temperature (°C)
		Min_temp	The minimum temperature (°C)	*Days of air frost*	The days of air frost
		Sunshine	The sunshine hours		
Traffic	*Yearly*	*Link length (km)*	*The total length of each junction to junction link on the major road network*	*Cars and taxis*	The number of cars and taxis
		All motor vehicles	*The amount of all motor vehicles*		
Terrain	*NIL*	*Mean elevation*	The mean elevation of the mobility area	*Maximum elevation*	The maximum elevation of the mobility area
		Minimum elevation	The minimum elevation of mobility area	*STD of elevation*	The Standard deviation of elevation of mobility area
		Mean slope	The mean slope of the mobility area	*Maximum slope*	The maximum slope of the mobility area
		STD of slope	The Standard deviation of slope of mobility area	*Mean aspect*	The mean aspect of mobility area
		Longitude	The latitude of the longitude	*Latitude*	The latitude of the garage

For the features in GIS data, it is likely that there are some features that are highly correlated. The redundant features may damage the algorithm performance. Hence, it is necessary to remove redundant features. Figure 6.6 shows the heatmap of the extracted features of the GIS data. The heatmap was generated using Python Seaborn package (Bisong, 2019). The highly correlated features (correlation coefficient close to 1 or -1) which correlation coefficient is over 0.8 were identified via heatmap and then were removed. In the heat map, there are 42 features in total. After the removal of the GIS features, only 22 features were kept for further modelling.

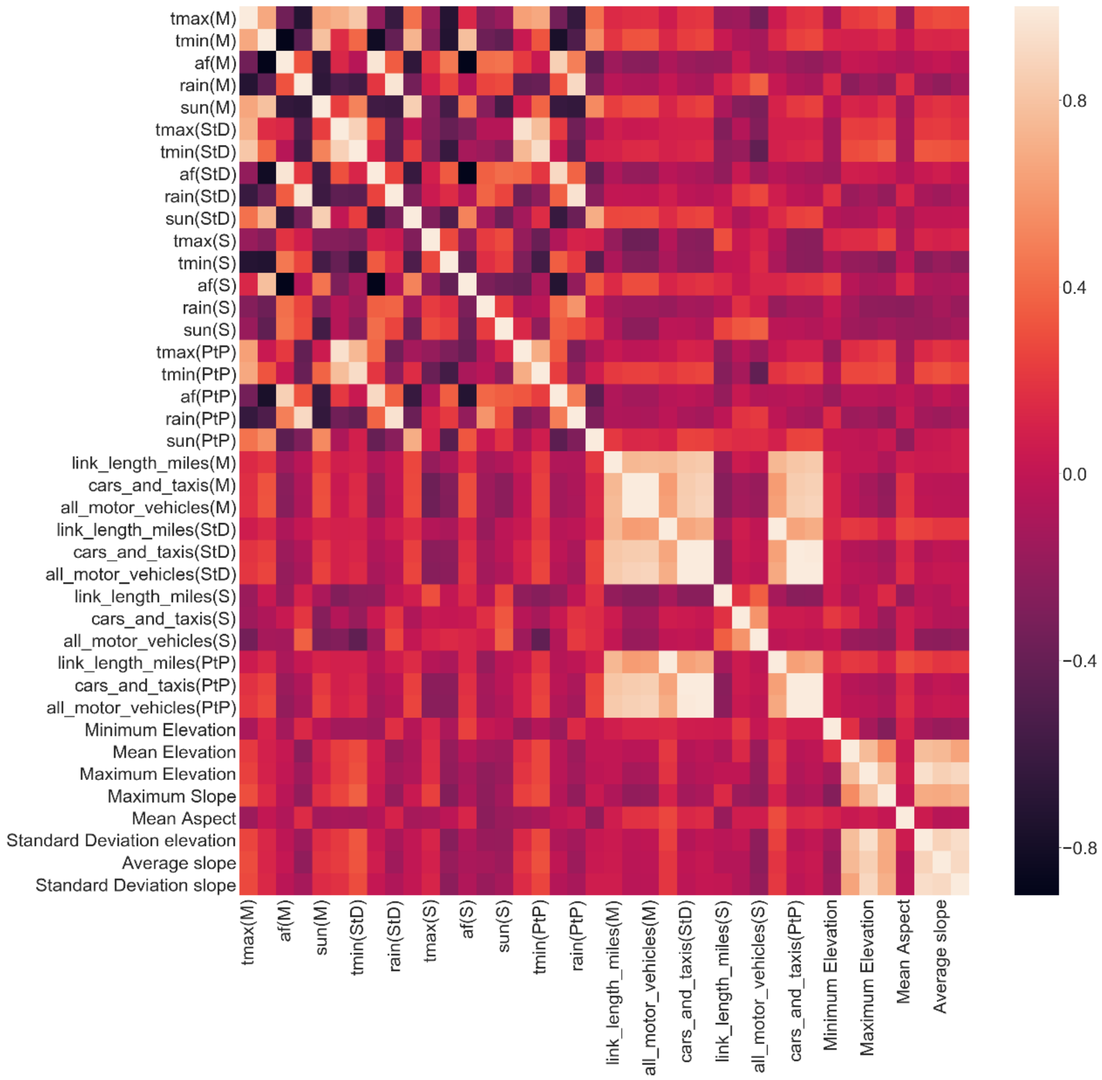

Figure 6.6 The heatmap of the GIS features

6.3.2 Experimental Setup

After the historical data and GIS data is pre-processed. The parameters of the M-LSTM need to be determined. The number of neurons of the layers in the ordinary numeric data processing path, sequential data processing path, GIS data processing path and data integration path are 500, 500, 100 and 1600, respectively. Secondly, the parameters relevant to the training process were determined. RMSprop (Tieleman and Hinton, 2012) was selected as the optimizer with the learning rate set at 0.005. The batch size and training epochs were set at 150 and 10, respectively. Besides, the dropout rate was set at 0.5.

In this study, two scenarios were explored to demonstrate the effectiveness of the proposed method. In scenario 1, the algorithm performance of M-LSTM was revealed. The modelling based on historical maintenance data was implemented. Hence, there are only two subnetworks in M-LSTM network. Several prevailing algorithms, including LSTM, FCNN, and SVM, were adopted as benchmarks. In scenario 2, the GIS data was introduced into modelling. The impact of different types of GIS data was revealed. Different types of GIS data are used for modelling. The algorithm performance changes can be used to reveal the importance of GIS features.

In order to obtain comprehensive results, 10-fold cross-validation was adopted. The evaluation metrics for algorithm performance used in this study are MCC, RMSE and modelling time, which is the same as the metrics used in Chapter 4.

6.4 Experimental Results

6.4.1 Scenario 1: M-LSTM Network VS. Prevailing Machine Learning Algorithms

In order to reveal the algorithm performance of M-LSTM network, in this scenario, four machine learning algorithms, which are M-LSTM network, LSTM network, FCNN and SVM were used for modelling based on historical maintenance data. Since GIS data was not introduced in this stage, the GIS data processing path in M-LSTM network was removed. The results of the benchmarking algorithms were obtained from Scenario 3 in Chapter 4. The specific results comparison is shown in Table 6.2.

Table 6.2 The results of TBF modelling using different algorithms

	M-LSTM Network	LSTM Network	FCNN	SVM
MCC	**0.8426**	0.8389	0.8307	0.7778
RMSE (days)	**355.2**	359.2	364.7	425.8
Modelling time (s)	157.4	259.2	34.65	**7.263**

It can be seen that M-LSTM network achieved the highest MCC and lowest RMSE in this scenario. The algorithm performance of M-LSTM in terms of MCC and RMSE are slightly better than that of LSTM network. The algorithm performance of deep learning in terms of MCC and RMSE are better than SVM, while deep learning algorithms require higher computational load. In comparison with LSTM network, the M-LSTM network not only shows merit in algorithm performance in terms of MCC and RMSE, but it can lower the modelling time by 39.2%. The M-LSTM network shows merits in TBF modelling.

6.4.2 Scenario 2: Modelling based on Historical Maintenance Data and GIS Data

With the introduction of GIS data, it is worth to investigate how the GIS features can impact the algorithm performance of different algorithms adopted in this study. In this scenario, the GIS data includes weather data, traffic data, and terrain data was introduced into modelling. Besides introducing all the GIS features, three different types of GIS data were introduced individually to examine their impact on automobile TBF modelling. Figure 6.7 shows the MCC of modelling based on historical maintenance data and different GIS data. It can be seen that the algorithm performance for M-LSTM network and FCNN in terms of MCC were boosted with the introduction of all the GIS features, while the algorithm performance of LSTM network and SVM in terms of MCC was slightly compromised. When different types of GIS data are used individually for modelling, the impacts on the algorithm performance of different algorithms in terms of MCC were not the same. The algorithm performance of M-LSTM network and FCNN were increased with the introduction of weather data, traffic data, and terrain data. With the help of the introduction of all GIS data, M-LSTM achieved the highest MCC in this scenario, which is 0.8492. It was promoted by 0.79% in comparison with the modelling result without GIS data. When different GIS data was introduced individually, the algorithm performance in terms of MCC of M-LSTM network and FCNN were promoted. In contrast, the introduction of GIS data was not helpful to improve the algorithm performance in terms of MCC of LSTM network and SVM.

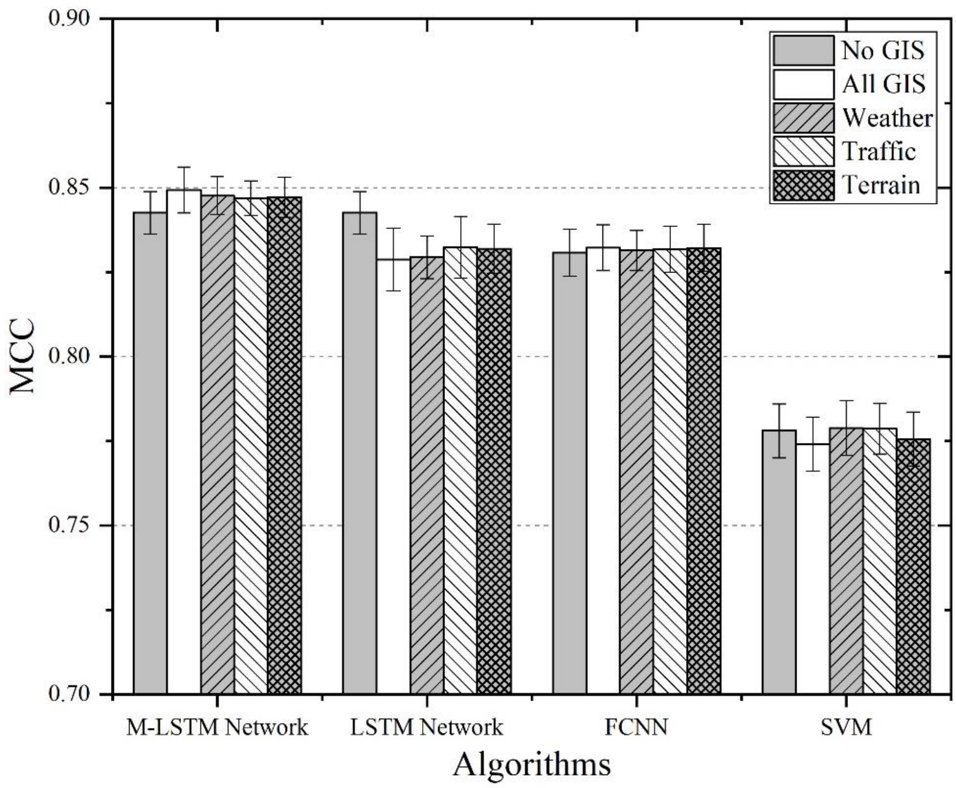

Figure 6.7 The MCC of modelling based on historical maintenance data and different GIS data

Figure 6.8 shows the RMSE of modelling based on historical maintenance data and different GIS data. It can be seen that when all the GIS data is fed into the algorithms, the algorithm performance of M-LSTM network and FCNN were decreased. However, when different types of GIS data were introduced for modelling, the impact on algorithm performance in terms of RMSE is quite different. M-LSTM network achieved the lowest RMSE in this scenario with the help of all the GIS features, which is 348.3 days. The comparison of different types of GIS features shows that weather data has the most considerable impact on the decrease of RMSE for M-LSTM network. For the LSTM network, the introduction of all GIS features increased the RMSE by 10.8 days, while the RMSE of FCNN and SVM were slightly changed with the introduction of GIS features.

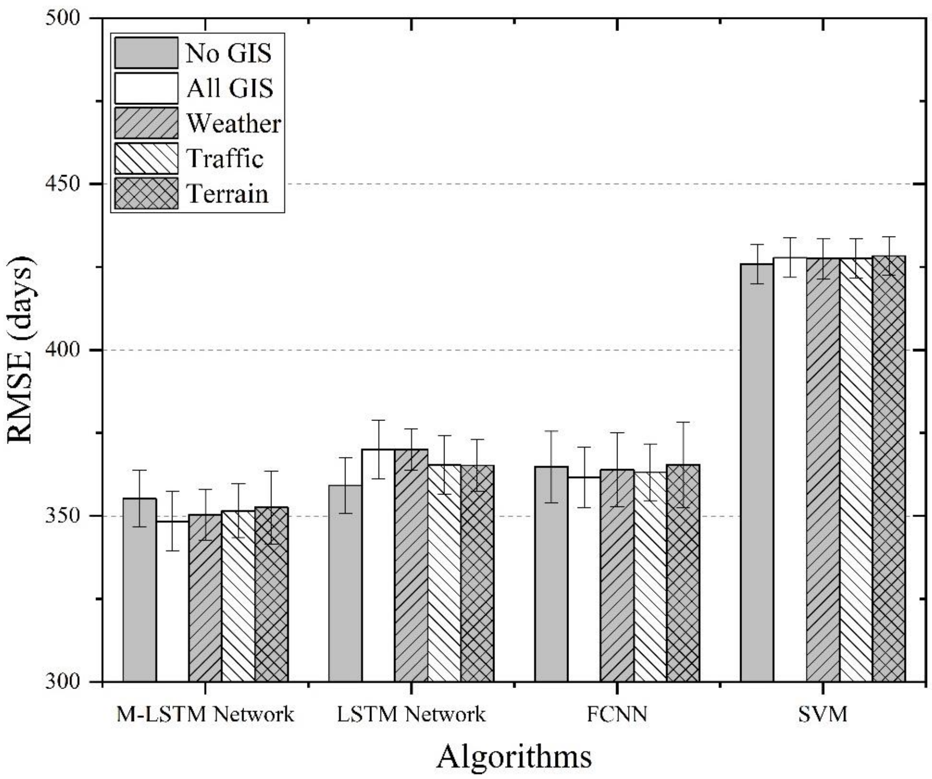

Figure 6.8 The RMSE of modelling based on historical maintenance data and different GIS data

6.5 Discussion

From the results of scenario 1, what evident is that the M-LSTM network shows merits in modelling based on heterogeneous data which contains sequential data and ordinary numeric data. It promoted the algorithm performance in terms of MCC and RMSE, while it requires less computational load in comparison with LSTM network. Hence, it can be a useful tool in heterogeneous data modelling. The results of scenario 2 indicated the introduction of individual groups of GIS features can promote the algorithm performance of M-LSTM network and FCNN in terms of MCC and RMSE. The introduction of the weather data promoted the largest algorithm performance improvement in terms of MCC and RMSE of M-LSTM network, which indicate that weather factors have a higher impact to the automobile lifecycle. However, the introduction of individual groups of GIS features worsens the algorithm performance of LSTM network in terms of MCC and RMSE. LSTM network is mainly used to process sequential data. When the proportion of ordinary numeric features dramatically outweigh than that of sequential features, LSTM does not show merits. The worst results of LSTM network may be caused by the introduction of numerous ordinary numeric features, which jeopardized the algorithm performance. The improvement of the algorithm performance of M-LSTM network is more significant than that of FCNN, which indicate that M-LSTM network is more suitable in modelling based on multi-sources data.

The introduction of GIS data promoted the algorithm performance of M-LSTM and FCNN network in terms of MCC and RMSE, which indicate the automobile TBF is relevant to weather, traffic and terrain conditions. The M-LSTM network witnessed the most considerable performance improvement and decrease in terms of MCC and RMSE with the help of three types of GIS data, which demonstrated that M-LSTM shows merits in modelling with GIS data. However, the improvement of the algorithm performance in terms of MCC and RMSE are still limited. The reason can be the GIS data collected in this study does not match the actual surrounding factors of the automobile mobility area of each automobile. For example, if there are a mountainous area and a plain area in the data collection area, the maximum elevation and the slope tends to be dramatic. However, there is not likely too many roads in such a mountainous area. Hence, the collected terrain data did show a positive impact on automobile TBF modelling. Its actual impact needs to be further investigated.

6.6 Summary

In automobile PdM, the existing studies only focused on modelling using sensor data or historical maintenance data which is directly relevant to the automobile failure. The surrounding factors, which can impact the automobile lifecycle, such as weather, traffic and terrain have not been considered in automobile PdM. These data can be captured and processed using GIS. In order to introduce GIS data into automobile PdM, an approach that can be used to develop an M-LSTM network-based predictive maintenance enriched by GIS data was proposed in this study. How to collect and pre-process the GIS data was investigated. M-LSTM network, as a novel deep learning architecture, was proposed for modelling based on multi-source data. The experimental study validated the effectiveness of the proposed approach. The experimental results indicated that M-LSTM network shows merits in multi-source data modelling. The introduction of the weather and traffic data can be beneficial to promote the algorithm performance in terms of MCC and RMSE.

Chapter 7 Achievements and Conclusions

7.1 Achievements

The key purpose of this research was to model and predict the next TBF for the automobile so to optimise the maintenance management of a fleet management company. This motivation was explored at the beginning of this thesis and is underpinned by a discussion of the background of PdM. Despite the fact that significant media and academia have paid attention to the growing potential of condition-based PdM for asset RUL prediction, there has been little research which focused on TBF modelling based on historical maintenance data, though this can bring tangible benefit to fleet management. This research focused on integrating multi-source data for automobile TBF modelling using deep learning. One of the key concepts in Industry 4.0 is ‘connected everything’ (Qin et al., 2016). In the past, only sensor data or historical maintenance data were adopted in the research of PdM. Under the context of Industry 4.0, other available public data such as GIS data can also be adopted in PdM. Though the real-time telematics data is absent in this research, the multi-source data, which contain historical maintenance data and three types of GIS data, were adopted for this research. In this thesis, there are four research questions proposed in Chapter 1. Based on the work achieved. The answers to the research questions are obtained.

To determine the state-of-the-art research, the literature review was provided with the relevant technologies and relevant research. Firstly, the studies of condition-based PdM and statistical-based PdM were reviewed. In the state-of-the-art, different machine learning and statistical methods have been widely studied in PdM. Deep learning has been widely used in the studies of PdM and shows merits. The prevailing deep learning algorithms and the studies that deploy deep learning in the industry were reviewed. Since the surrounding data can be captured and processed by GIS, the studies of GIS were reviewed. Furthermore, semi-supervised learning is helpful for deep learning modelling when the labelled data is limited. Thus, two main types of semi-supervised learning algorithms and the application of semi-supervised learning in the industry were reviewed.

The first research question is: what is a suitable framework for automobile PdM based on the understanding of Industry 4.0? Following the understanding of the-state-of-the-art, a framework for automobile TBF modelling under industry 4.0 was proposed. By summarising several different perspectives, the main concepts of the automobile PdM under the context of PdM have been identified to inform the research aim. It can be seen that there was a gap between the current automobile PdM and the achievement of the automobile PdM under the context of Industry 4.0 in terms of the data used in the study of PdM. The current research of PdM only considers sensor data or historical maintenance data, rather than comprehensively integrate different types of relevant data to PdM into modelling. In order to fill this gap, there are three types of data included in this framework which are historical maintenance data, GIS data and real time telematics data. In this framework, the stage of data collection, data transmission and storage, data mapping, pre-processing, and integration were specified. After the data is prepared, deep learning as the primary TBF modelling tool is used in conjunction with reliability analysis and semi-supervised techniques for modelling so to obtain an accurate TBF prediction. Finally, the TBF prediction results can be used to leverage the decision support of fleet management.

In order to explore how deep learning can be used in automobile TBF modelling based on historical maintenance data, a new approach called CoxPHDL was proposed. It aims for modelling based on historical maintenance data by integrating deep learning and reliability analysis. There are two issues in the analysis of historical maintenance data, which are data sparsity and data censoring. In this approach, autoencoder and Cox PHM are adopted to address the data sparsity and data censoring issues. With the consideration of the sequential patterns in historical maintenance data, LSTM network was used to build a TBF prediction model based on the pre-processed data. The key findings in this study are: (1) With larger computational loads, LSTM network can achieve better algorithm performance in comparison with those prevailing machine learning algorithms; (2) With the help of autoencoder and Cox PHM, the algorithm performance of most algorithms deployed in this study can be promoted.

After the TBF modelling approach is determined, an issue in TBF modelling using deep learning that needs to be considered is the label data size. With the consideration of the label data size, the third research question was proposed: when the labelled data is insufficient how can the performance of deep learning not be significantly jeopardised in the study of automobile TBF modelling? The sufficiency of labelled data is the prerequisite of satisfactory algorithm performance. Without a large amount of labelled data, the algorithm performance of deep learning will be jeopardised. Considering the unlabelled data is relatively easy to obtain in the real world, a new semi-supervised learning algorithm called DLeSSL was proposed for modelling based on limited labelled data. The experimental results indicate that (1) with the help of unlabelled data, the algorithm performance of deep learning can be boosted when the labelled data is limited; (2) the existing studies have not reported how the labelled data size impacts the algorithm performance, while this study revealed the relationship between the size of labelled data and algorithm performance, which can offer insights to the actual deployment of DLeSSL.

The TBF modelling Chapter 4 and 5 are based on historical maintenance data. Besides, the features in historical maintenance data, various surrounding factors also are relevant to automobile lifecycle. The final research question focuses on investigating how to integrate the heterogeneous data with historical maintenance data for automobile TBF modelling. The data relevant to these factors can be collected and processed by GIS. With the coordinate of the automobile garage, the corresponding GIS data can be obtained. The taxonomy of the GIS data was illustrated and how to process the GIS data for TBF modelling was detailed. A new deep learning architecture called M-LSTM was designed for multi-source data integration and TBF modelling. The experimental results show: (1) M-LSTM network shows better algorithm performance and less computational load in multi-source data modelling based on historical maintenance data in comparison with LSTM network; (2) The impacts of the three groups of GIS features were examined. The weather and traffic features can be helpful in TBF modelling.

7.2 Future Works

This thesis aims to research automobile PdM under Industry 4.0. In this study, due to the unavailability of telematics data, only historical maintenance data and GIS data were adopted in the experiment. The predicted TBF can offer insights to a fleet management company in terms of spare part management, maintenance planning and job scheduling. Firstly, the company can adjust its parts order in the coming year to avoid parts overstock or shortage. Furthermore, with the predicted TBF, the period of the scheduled check can be adjusted dynamically to better implement maintenance. Last but not least, the TBF prediction can help be helpful in job scheduling. For example, the automobile with long TBF prediction can be distributed with a heavy load, while the automobile with short TBF prediction can work in light load and short-haul. In the future, with the enrichment of the telematics data, the features that relevant to the automobile health status can be obtained, and therefore the TBF prediction accuracy can be improved. Meanwhile, the effectiveness of the proposed approach can be further validated in future works when the telematics data is available.

The proposed approaches can be further investigated to achieve better performance. The algorithm performance of CoxPHDL in terms of MCC and RMSE can be continuously improved and decreased in future research. In the historical maintenance dataset used in this study, we have noticed that some automobiles with a similar condition have different TBFs. In other words, there are some data whose features are similar or even the same, while their labels are different. The main reason is that the features relevant to the automobile lifecycle were not sufficiently collected. The historical maintenance data may not contain sufficient patterns relevant to automobile engine degradation, which enable the algorithm to yield a precise prediction model. It is well known that the TBF of the automobile depends on various factors, including geographical environment, driving behaviour, and product design, etc. If these data can be collected and introduced into our study, the algorithm performance is likely to be further promoted. Moreover, the research on these data will also be helpful for the fleet management company to have a better understanding of how these factors impact the automobile lifecycle. The Cox PHM is under the assumption that the difference in reliability between corrective maintenance and preventive maintenance is the same for all automobiles owned by the same company. However, the difference in reliability of automobiles might slightly vary from each other. It can be caused by the following reasons. On the one hand, the data used in this case was collected from different garages of the fleet management company under investigation. The maintenance rules and standards could be slightly different in different garages. On the other hand, the preventive maintenance strategy deployed in the company's garages also depends on the engineer's judgment and experience which could lead to the variation of difference in reliability. If a better estimation of the difference in the reliability of each automobile can be achieved, it is possible to further lift the algorithm performance of LSTM network. Hence, a better approach used to estimate the difference in reliability needs to be investigated further.

DLeSSL shows merits in TBF modelling when the ratio of labelled data and unlabelled data situated in the ranges from 3.6% to 13.2%. However, the best algorithm performance of DLeSSL in terms of MCC and RMSE in this range is worse than the modelling results from Chapter 4. This may be caused by the label accuracy of the data. If the label is not accurate, it is hard to get the accurate estimation of the TBF for DLeSSL. It is common that some data entries in historical maintenance dataset used in this thesis have similar features, while the TBF is dramatically different. The main reason is the missing of telemetric data. With the telemetric data which can indicate the automobile health status. The label determination for DLeSSL can be more accurate. With an accurate estimation of the label of the unlabelled data, the algorithm performance of DLeSSL can be further promoted. Hence, one of the essential future works is to collect the automobile telemetric data and introduce it into automobile TBF modelling.

In the future, with the development of IIoT, the real-time telematics data of automobile can be collected during the in-use period. With the GPS information, the real-time GIS data can be collected from the external data sources and then be transmitted to the cloud. The real-time telematics data, GIS data, and historical maintenance data can be used together to establish a more accurate automobile maintenance prediction model, which can offer real-time health condition monitoring and RUL prediction. Furthermore, the architecture of M-LSTM can be further explored. The M-LSTM designed in this thesis using FCNN and LSTM network for feature extraction. Then the extracted features are sent to concatenated layer for integration. Finally, a two layers FCNN is deployed for final TBF prediction. With the development of machine learning, there are various emerging algorithms proposed recently. It is worthwhile to explore whether the FCNN in the data integration path can be replaced by another machine learning algorithm. Moreover, since the impact of GIS features on automobile TBF is not significant, and therefore the impacts of the group of features on automobile TBF was tested. Since deep learning is a type of black-box algorithm, which cannot indicate how the features affect the final output. It cannot offer too many insights to the fleet management company. If the impact of each GIS feature can be identified, it can bring tangible benefits for the fleet management company to optimise the maintenance management. Hence, the future works also target on developing a new approach to identify the impact of each feature.

7.3 Conclusions

In conclusion, the common theme throughout this research was to promote the intelligence level of automobile PdM in terms of automobile TBF modelling. TBF modelling is essential in automobile PdM. In this thesis, a framework was designed to provide a roadmap for automobile TBF modelling, prediction, and decision support.

A modelling approach called CoxPHDL was proposed for TBF modelling based on historical maintenance data. Furthermore, since labelled data is expensive to obtain in the real world and the deep learning needs a large amount of labelled data to achieve satisfactory performance, a semi-supervised learning algorithm called DLeSSL was proposed. Finally, with the consideration of the impact on automobile lifecycle from the surrounding factors, an approach concerns how to integrate GIS data for TBF modelling was proposed. Meanwhile, a new deep learning architecture called M-LSTM network was designed for multi-source data integration. This thesis has demonstrated how to model and predict automobile TBF based on multi-source data and address the issue that labelled data is limited. In the future, automobile PdM will gain increasing attention. The approaches developed in this thesis will bring tangible benefits to the automobile fleet management.

References

2018. *Weather and Climate Change-Met Office* [Online]. Available: https://www.metoffice.gov.uk/ [Accessed 2018/12/07].

AL-DULAIMI, A., ZABIHI, S., ASIF, A. & MOHAMMED, A. 2019. NBLSTM: Noisy and Hybrid CNN and BLSTM-based Deep Architecture for Remaining Useful Life Estimation. *Journal of Computing and Information Science in Engineering*, 1-14.

ALLAH BUKHSH, Z., SAEED, A., STIPANOVIC, I. & DOREE, A. G. 2019. Predictive maintenance using tree-based classification techniques: A case of railway switches. *Transportation Research Part C: Emerging Technologies,* 101, 35-54.

ANDERSON, J. & SENTHILSELVAN, A. 1982. A two-step regression model for hazard functions. *Applied Statistics*, 44-51.

AREMU, O. O., HYLAND-WOOD, D. & MCAREE, P. R. 2019. A Relative Entropy Weibull-SAX framework for health indices construction and health stage division in degradation modeling of multivariate time series asset data. *Advanced Engineering Informatics,* 40, 121-134.

BALUJA, S., SETH, R., SIVAKUMAR, D., JING, Y., YAGNIK, J., KUMAR, S., RAVICHANDRAN, D. & ALY, M. Video suggestion and discovery for youtube: taking random walks through the view graph. Proceedings of the 17th international conference on World Wide Web, 2008. 895-904.

BARUAH, P. & CHINNAM*, R. B. 2005. HMMs for diagnostics and prognostics in machining processes. *International Journal of Production Research,* 43, 1275-1293.

BEHRENDT, A., MÜLLER, N., ODENWÄLDER, P. & SCHMITZ, C. 2017. Industry 4.0 demystified–lean's next level. *Retrieved March,* 3, 2017.

BEN-DAVID, S., LU, T. & PÁL, D. Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning. COLT, 2008. 33-44.

BENGIO, Y. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning,* 2**,** 1-127.

BENGIO, Y. & CA, M. 2015. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. *corr abs/1502.04390*.

BENGIO, Y., LAMBLIN, P., POPOVICI, D. & LAROCHELLE, H. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 2007. 153-160.

BENGIO, Y., SIMARD, P. & FRASCONI, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks,* 5**,** 157-166.

BERGSTRA, J. & BENGIO, Y. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research,* 13**,** 281-305.

BISONG, E. 2019. Matplotlib and Seaborn. *Building Machine Learning and Deep Learning Models on Google Cloud Platform.* Springer.

BOYES, H., HALLAQ, B., CUNNINGHAM, J. & WATSON, T. 2018. The industrial internet of things (IIoT): An analysis framework. *Computers in industry,* 101**,** 1-12.

BRESLOW, N. E. 1975. Analysis of survival data under the proportional hazards model. *International Statistical Review/Revue Internationale de Statistique***,** 45-57.

CALTAGIRONE, L., SVENSSON, L., WAHDE, M. & SANFRIDSON, M. 2019. Lidar-Camera Co-Training for Semi-Supervised Road Detection. *arXiv preprint arXiv:1911.12597*.

CARVALHO, T. P., SOARES, F. A., VITA, R., FRANCISCO, R. D. P., BASTO, J. P. & ALCALÁ, S. G. 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering,* 137**,** 106024.

CHAPELLE, O., CHI, M. & ZIEN, A. A continuation method for semi-supervised SVMs. Proceedings of the 23rd international conference on Machine learning, 2006. 185-192.

CHAPELLE, O., SCHOLKOPF, B. & ZIEN, A. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks,* 20**,** 542-542.

CHEN, C., LIU, Y., SUN, X., WANG, S., DI CAIRANO-GILFEDDER, C., TITMUS, S. & SYNTETOS, A. A. Reliability Analysis Using Deep Learning. ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2018. American Society of Mechanical Engineers, V01BT02A040-V01BT02A040.

CHEN, C., LIU, Y., WANG, S., SUN, X., DI CAIRANO-GILFEDDER, C., TITMUS, S. & SYNTETOS, A. A. 2020. Predictive maintenance using cox proportional hazard deep learning. *Advanced Engineering Informatics,* 44**,** 101054.

CHEN, X.-W. & LIN, X. 2014. Big data deep learning: challenges and perspectives. *IEEE access,* 2**,** 514-525.

CHEN, Z., LI, Y., XIA, T. & PAN, E. 2017. Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy. *Reliability Engineering & System Safety*.

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. *arXiv preprint arXiv:1406.1078*.

CHOLLET, F. 2015. Keras: Deep learning library for theano and tensorflow. *URL: https://keras. io/k,* 7**,** 8.

CHOUDHARY, A. K., HARDING, J. A. & TIWARI, M. K. 2008. Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing,* 20**,** 501.

CIVERCHIA, F., BOCCHINO, S., SALVADORI, C., ROSSI, E., MAGGIANI, L. & PETRACCA, M. 2017. Industrial Internet of Things monitoring solution for advanced predictive maintenance applications. *Journal of Industrial Information Integration,* 7**,** 4-12.

CLOUDERA 2017. Driving Predictive Maintenance in a Connected World.

CODD, E. F. 1970. A relational model of data for large shared data banks. *Communications of the ACM,* 13**,** 377-387.

CORTES, C., JACKEL, L. D. & CHIANG, W.-P. 1994. Limits on learning machine accuracy imposed by data quality. *Advances in Neural Information Processing Systems,* 7**,** 239-246.

CORTES, C., MOHRI, M. & ROSTAMIZADEH, A. 2012. L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*.

CORTES, C. & VAPNIK, V. 1995. Support-vector networks. *Machine learning,* 20**,** 273-297.

COSTACHE, R., HONG, H. & WANG, Y. 2019. Identification of torrential valleys using GIS and a novel hybrid integration of artificial intelligence, machine learning and bivariate statistics. *Catena,* 183**,** 104179.

COX, D. R. 1992. Regression models and life-tables. *Breakthroughs in statistics.* Springer.

CRESCI, J.-P., LORTIE, P. & BHATTACHARYA, K. 2017. Machine Learning: A Turning Point For Predictive Maintenance?

DENG, L. & YU, D. 2014. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing,* 7**,** 197-387.

DEPARTMENT-FOR-TRANSPORT. 2020. *GB Road Traffic Counts*

[Online]. Available: https://data.gov.uk/dataset/208c0e7b-353f-4e2d-8b7a-1a7118467acc/gb-road-traffic-counts [Accessed 03/01/2021].

DING, H., YANG, L. & YANG, Z. 2019. A predictive maintenance method for shearer key parts based on qualitative and quantitative analysis of monitoring data. *IEEE Access,* 7**,** 108684-108702.

DÓPIDO, I., LI, J., MARPU, P. R., PLAZA, A., DIAS, J. M. B. & BENEDIKTSSON, J. A. 2013. Semisupervised self-learning for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing,* 51**,** 4032-4044.

EDWARDS, T., BAYOUMI, A. & EISNER, M. L. 2017. Internet of Things–A Complete Solution for Aviation's Predictive Maintenance. *Advanced Technologies for Sustainable Systems.* Springer.

ERL, T., KHATTAK, W. & BUHLER, P. 2016. *Big data fundamentals: concepts, drivers & techniques*, Prentice Hall Press.

ESSIEN, A. & GIANNETTI, C. 2020. A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. *IEEE Transactions on Industrial Informatics,* 16**,** 6069-6078.

FERGUSON, M., JEONG, S., LAW, K. H., LEVITAN, S., NARAYANAN, A., BURKHARDT, R., JENA, T. & LEE, Y.-T. T. A Standardized Representation of Convolutional Neural Networks for Reliable Deployment of Machine Learning Models in the Manufacturing Industry. International

Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2019. American Society of Mechanical Engineers, V001T02A005.

FINANCE, A. T. C. C. 2015. Industry 4.0 Challenges and solutions for the digital transformation and use of exponential technologies. *Finance, Audit Tax Consulting Corporate: Zurich, Swiss*.

GAO, J., LI, P., CHEN, Z. & ZHANG, J. 2020. A survey on deep learning for multimodal data fusion. *Neural Computation,* 32**,** 829-864.

GE, Z., ZHONG, S. & ZHANG, Y. 2016. Semisupervised kernel learning for FDA model and its application for fault classification in industrial processes. *IEEE Transactions on Industrial Informatics,* 12**,** 1403-1411.

GENSLER, A., HENZE, J., SICK, B. & RAABE, N. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. 2016 IEEE international conference on systems, man, and cybernetics (SMC), 2016. IEEE, 002858-002865.

GHASEDI DIZAJI, K., HERANDI, A., DENG, C., CAI, W. & HUANG, H. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. Proceedings of the IEEE international conference on computer vision, 2017. 5736-5745.

GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016. *Deep learning*, MIT press.

GRANDVALET, Y., D'ALCHÉ-BUC, F. & AMBROISE, C. Boosting mixture models for semi-supervised learning. International conference on artificial neural networks, 2001. Springer, 41-48.

GREZMAK, J., ZHANG, J., WANG, P., LOPARO, K. A. & GAO, R. X. 2019. Interpretable Convolutional Neural Network Through Layer-wise Relevance Propagation for Machine Fault Diagnosis. *IEEE Sensors Journal*, 20**,** 3172-3181.

HADY, M. F. A. & SCHWENKER, F. 2013. Semi-supervised learning. *Handbook on Neural Information Processing.* Springer.

HAN, K., LI, C. & SHI, X. 2017. Autoencoder Feature Selector. *arXiv preprint arXiv:1710.08310*.

HAND, D. J. & ADAMS, N. M. 2014. Data Mining. *Wiley StatsRef: Statistics Reference Online***,** 1-7.

HOCHREITER, S. & SCHMIDHUBER, J. 1997. Long short-term memory. *Neural computation,* 9**,** 1735-1780.

HONG, C., YU, J., WAN, J., TAO, D. & WANG, M. 2015. Multimodal deep autoencoder for human pose recovery. *IEEE Transactions on Image Processing*, 24**,** 5659-5670.

HU, Y.-L. & CHEN, L. 2018. A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm. *Energy conversion and management,* 173**,** 123-142.

HUANG, Z., ZHU, J., LEI, J., LI, X. & TIAN, F. 2019. Tool Wear Predicting Based on Multisensory Raw Signals Fusion by Reshaped Time Series Convolutional Neural Network in Manufacturing. *IEEE Access,* 7**,** 178640-178651.

KANG, H. S., LEE, J. Y., CHOI, S., KIM, H., PARK, J. H., SON, J. Y., KIM, B. H. & DO NOH, S. 2016a. Smart manufacturing: Past research, present findings, and future directions. *International journal of precision engineering and manufacturing-green technology,* 3**,** 111-128.

KANG, P., KIM, D. & CHO, S. 2016b. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Systems with Applications,* 51**,** 85-106.

KAY, R. 1977. Proportional hazard regression models and the analysis of censored survival data. *Applied Statistics***,** 227-237.

KIAN, R., BEKTAŞ, T. & OUELHADJ, D. 2019. Optimal spare parts management for vessel maintenance scheduling. *Annals of operations research,* 272**,** 323-353.

KINGMA, D. P. & BA, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

KOBAYASHI, S., FUKUDA, K. & ESAKI, H. Mining causes of network events in log data with causal inference. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017. IEEE, 45-53.

KRUGER, G. H., SHIH, A. J., HATTINGH, D. G. & VAN NIEKERK, T. I. 2011. Intelligent machine agent architecture for adaptive control optimization of manufacturing processes. *Advanced Engineering Informatics,* 25**,** 783-796.

KUMAR, D. & KLEFSJÖ, B. 1994. Proportional hazards model: a review. *Reliability Engineering & System Safety,* 44**,** 177-188.

LANDERS, T. L. & SOROUDI, H. E. 1991. Robustness of a semi-parametric proportional intensity model. *IEEE Transactions on Reliability,* 40**,** 161-164.

LECUN, Y., BENGIO, Y. & HINTON, G. 2015. Deep learning. *Nature,* 521**,** 436-444.

LEE, D. & PAN, R. 2017. Predictive maintenance of complex system with multi-level reliability structure. *International Journal of Production Research,* 55**,** 4785-4801.

LEE, D. & PAN, R. 2019. Evaluating reliability of complex systems for Predictive maintenance. *arXiv preprint arXiv:1902.03495*.

LEE, J., DAVARI, H., SINGH, J. & PANDHARE, V. 2018. Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. *Manufacturing letters,* 18**,** 20-23.

LEE, S. 2005. Application of logistic regression model and its validation for landslide susceptibility mapping using GIS and remote sensing data. *International Journal of Remote Sensing,* 26**,** 1477-1491.

LEITCH, R. D. 1995. Reliability analysis for engineers: an introduction. *OUP Catalogue*.

LI, X., DING, Q. & SUN, J.-Q. 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety,* 172**,** 1-11.

LIANG, Z. & PARLIKAD, A. K. 2020. Predictive group maintenance for multi-system multi-component networks. *Reliability Engineering & System Safety,* 195**,** 106704.

LIU, J., GUO, J., ORLIK, P., SHIBATA, M., NAKAHARA, D., MII, S. & TAKÁČ, M. Anomaly detection in manufacturing systems using structured neural networks. 2018 13th World Congress on Intelligent Control and Automation (WCICA), 2018a. IEEE, 175-180.

LIU, Q., GAO, H., YOU, Z., SONG, H. & ZHANG, L. Gcforest-based fault diagnosis method for rolling bearing. 2018 Prognostics and System Health Management Conference (PHM-Chongqing), 2018b. IEEE, 572-577.

LIU, W., WANG, J. & CHANG, S.-F. 2012. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE,* 100**,** 2624-2638.

LIU, X., TIAN, Y., LEI, X., LIU, M., WEN, X., HUANG, H. & WANG, H. 2019. Deep forest based intelligent fault diagnosis of hydraulic turbine. *Journal of Mechanical Science and Technology,* 33**,** 2049-2058.

LÜTTENBERG, H., BARTELHEIMER, C. & BEVERUNGEN, D. 2018. Designing Predictive Maintenance for Agricultural Machines.

MA, Y. & GUO, G. 2014. *Support vector machines applications*, Springer.

MALHI, A., YAN, R. & GAO, R. X. 2011. Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement,* 60**,** 703-711.

MALLAPRAGADA, P. K., JIN, R., JAIN, A. K. & LIU, Y. 2008. Semiboost: Boosting for semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence,* 31**,** 2000-2014.

MASSAWE, B. H., SUBBURAYALU, S. K., KAAYA, A. K., WINOWIECKI, L. & SLATER, B. K. 2018a. Mapping numerically classified soil taxa in Kilombero Valley, Tanzania using machine learning. *Geoderma,* 311**,** 143-148.

MASSAWE, B. H. J., SUBBURAYALU, S. K., KAAYA, A. K., WINOWIECKI, L. & SLATER, B. K. 2018b. Mapping numerically classified soil taxa in Kilombero Valley, Tanzania using machine learning. *Geoderma,* 311**,** 143-148.

METTAS, A. Modeling and analysis for multiple stress-type accelerated life data. Reliability and Maintainability Symposium, 2000. Proceedings. Annual, 2000. IEEE, 138-143.

MILES, S. B. & HO, C. L. 1999. Applications and issues of GIS as tool for civil engineering modeling. *Journal of computing in civil engineering,* 13**,** 144-152.

MOBLEY, R. K. 2002. *An introduction to predictive maintenance*, Elsevier.

MOHAMMADI, M., AL-FUQAHA, A., SOROUR, S. & GUIZANI, M. 2018. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials,* 20**,** 2923-2960.

MOURTZIS, D. & VLACHOU, E. 2018. A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of Manufacturing Systems,* 47**,** 179-198.

NAGHIBI, S. A., POURGHASEMI, H. R. & DIXON, B. 2016. GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran. *Environmental monitoring and assessment,* 188**,** 44.

NIELSEN, M. A. 2015. *Neural networks and deep learning*, Determination Press.

NIETO, P. G., GARCÍA-GONZALO, E., LASHERAS, F. S. & DE COS JUEZ, F. J. 2015. Hybrid PSO–SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. *Reliability Engineering & System Safety,* 138**,** 219-231.

NUÑEZ, D. L. & BORSATO, M. 2018. OntoProg: An ontology-based model for implementing Prognostics Health Management in mechanical machines. *Advanced Engineering Informatics,* 38**,** 746-759.

OKARO, I. A., JAYASINGHE, S., SUTCLIFFE, C., BLACK, K., PAOLETTI, P. & GREEN, P. L. 2019. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Additive Manufacturing,* 27**,** 42-53.

PATIL, R. B., PATIL, M. A., RAVI, V. & NAIK, S. Predictive modeling for corrective maintenance of imaging devices from machine logs. 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017. IEEE, 1676-1679.

PHAM, B. T., BUI, D. T., PRAKASH, I. & DHOLAKIA, M. 2017. Hybrid integration of Multilayer Perceptron Neural Networks and machine learning ensembles for landslide susceptibility assessment at Himalayan area (India) using GIS. *Catena,* 149**,** 52-63.

PIATESKI, G. & FRAWLEY, W. 1991. *Knowledge discovery in databases*, MIT press.

PRICE, M. H. 2010. *Mastering ArcGIS*, McGraw-Hill.

PRYTZ, R., NOWACZYK, S., RÖGNVALDSSON, T. & BYTTNER, S. 2015. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering applications of artificial intelligence,* 41**,** 139-150.

QIN, J., LIU, Y. & GROSVENOR, R. 2016. A categorical framework of manufacturing for industry 4.0 and beyond. *Procedia cirp,* 52**,** 173-178.

QIN, J., LIU, Y., GROSVENOR, R., LACAN, F. & JIANG, Z. 2020. Deep learning-driven particle swarm optimisation for additive manufacturing energy optimisation. *Journal of Cleaner Production,* 245**,** 118702.

RAHMAN, S., KHAN, S. & BARNES, N. Transductive learning for zero-shot object detection. Proceedings of the IEEE International Conference on Computer Vision, 2019. 6082-6091.

RAHMATI, O., POURGHASEMI, H. R. & MELESSE, A. M. 2016. Application of GIS-based data driven random forest and maximum entropy models for groundwater potential mapping: a case study at Mehran Region, Iran. *Catena,* 137**,** 360-372.

RAUCH, E., LINDER, C. & DALLASEGA, P. 2020. Anthropocentric perspective of production before and within Industry 4.0. *Computers & Industrial Engineering,* 139**,** 105644.

REN, L., CHENG, X., WANG, X., CUI, J. & ZHANG, L. 2019. Multi-scale Dense Gate Recurrent Unit Networks for bearing remaining useful life prediction. *Future Generation Computer Systems,* 94**,** 601-609.

REN, L., SUN, Y., CUI, J. & ZHANG, L. 2018. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. *Journal of Manufacturing Systems,* 48**,** 71-77.

RIKALOVIC, A., COSIC, I. & LAZAREVIC, D. 2014. GIS based multi-criteria analysis for industrial site selection. *Procedia Engineering,* 69**,** 1054-1063.

RØDSETH, H. & SCHJØLBERG, P. Data-driven predictive maintenance for green manufacturing. 6th International Workshop of Advanced Manufacturing and Automation, 2016. Atlantis Press.

ROMASZEWSKI, M., GŁOMB, P. & CHOLEWA, M. 2016. Semi-supervised hyperspectral classification from a small number of training samples using a co-training approach. *ISPRS Journal of Photogrammetry and Remote Sensing,* 121**,** 60-76.

ROSENBERG, C., HEBERT, M. & SCHNEIDERMAN, H. 2005. Semi-supervised self-training of object detection models.

RUMELHART, D. E., HINTON, G. E. & WILLIAMS, R. J. 1986. Learning representations by back-propagating errors. *nature,* 323**,** 533-536.

SABAHI, F. 2012. Secure virtualization for cloud environment using hypervisor-based technology. *International Journal of Machine Learning and Computing,* 2**,** 39.

SAGHEER, A. & KOTB, M. 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing,* 323**,** 203-213.

SALDANHA, P. L., DE SIMONE, E. A. & E MELO, P. F. 2001. An application of non-homogeneous Poisson point processes to the reliability analysis of service water pumps. *Nuclear engineering and design,* 210**,** 125-133.

SAYAH, M., GUEBLI, D., AL MASRY, Z. & ZERHOUNI, N. 2020. Robustness testing framework for RUL prediction Deep LSTM networks. *ISA Transactions*.

SCHAPIRE, R. E. & FREUND, Y. 2013. Boosting: Foundations and algorithms. *Kybernetes*.

SCHMIDHUBER, J. 2015. Deep learning in neural networks: An overview. *Neural Networks,* 61**,** 85-117.

SEZER, E., ROMERO, D., GUEDEA, F., MACCHI, M. & EMMANOUILIDIS, C. An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs. 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 17-20 June 2018 2018. 1-8.

SHI, W., CAO, J., ZHANG, Q., LI, Y. & XU, L. 2016. Edge computing: Vision and challenges. *IEEE internet of things journal,* 3**,** 637-646.

SIDDIQA, A., HASHEM, I. A. T., YAQOOB, I., MARJANI, M., SHAMSHIRBAND, S., GANI, A. & NASARUDDIN, F. 2016. A survey of big data management: Taxonomy and state-of-the-art. *Journal of Network and Computer Applications,* 71**,** 151-166.

SIMONYAN, K. & ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I. & SALAKHUTDINOV, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research,* 15**,** 1929-1958.

STOCK, T. & SELIGER, G. 2016. Opportunities of Sustainable Manufacturing in Industry 4.0. *Procedia CIRP,* 40**,** 536-541.

SUN, C., MA, M., ZHAO, Z., TIAN, S., YAN, R. & CHEN, X. 2018. Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. *IEEE Transactions on Industrial Informatics,* 15**,** 2416-2425.

SUN, Y., MA, L., MATHEW, J., WANG, W. & ZHANG, S. 2006. Mechanical systems hazard estimation using condition monitoring. *Mechanical systems and signal processing,* 20**,** 1189-1201.

SUSTO, G. A., SCHIRRU, A., PAMPURI, S., MCLOONE, S. & BEGHI, A. 2015. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Transactions on Industrial Informatics,* 11**,** 812-820.

TEHRANY, M. S., PRADHAN, B. & JEBUR, M. N. 2014. Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in GIS. *Journal of Hydrology,* 512**,** 332-343.

TIELEMAN, T. & HINTON, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning,* 4**,** 26-31.

TRIGUERO, I., GARCÍA, S. & HERRERA, F. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems,* 42**,** 245-284.

VAIDYA, S., AMBAD, P. & BHOSLE, S. 2018. Industry 4.0 – A Glimpse. *Procedia Manufacturing,* 20**,** 233-238.

VAN ENGELEN, J. E. & HOOS, H. H. 2020. A survey on semi-supervised learning. *Machine Learning,* 109**,** 373-440.

VERHAGEN, W. J. C. & DE BOER, L. W. M. 2018. Predictive maintenance for aircraft components using proportional hazard models. *Journal of Industrial Information Integration,* 12**,** 23-30.

VERT, J.-P., TSUDA, K. & SCHÖLKOPF, B. 2004. A primer on kernel methods. *Kernel Methods in Computational Biology***,** 35-70.

VINCENT, P., LAROCHELLE, H., LAJOIE, I., BENGIO, Y. & MANZAGOL, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research,* 11**,** 3371-3408.

WAN, X.-J., LIU, L., XU, Z., XU, Z., LI, Q. & XU, F. 2018. Fault diagnosis of rolling bearing based on optimized soft competitive learning Fuzzy ART and similarity evaluation technique. *Advanced Engineering Informatics,* 38**,** 91-100.

WANG, C., LU, N., CHENG, Y. & JIANG, B. Deep forest based multivariate classification for diagnostic health monitoring. 2018 Chinese Control And Decision Conference (CCDC), 2018a. IEEE, 6233-6238.

WANG, J., MA, Y., ZHANG, L., GAO, R. X. & WU, D. 2018b. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems,* 48**,** 144-156.

WANG, J., YAN, J., LI, C., GAO, R. X. & ZHAO, R. 2019. Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Computers in Industry,* 111**,** 1-14.

WANG, L. & SHI, Y. 2013. Reliability analysis of a class of exponential distribution under record values. *Journal of Computational and Applied Mathematics,* 239**,** 367-379.

WANG, S., LIU, Y., CAIRANO-GILFEDDER, C. D., TITMUS, S., NAIM, M. M. & SYNTETOS, A. A. 2018c. Reliability Analysis for Automobile Engines: Conditional Inference Trees. *Procedia CIRP,* 72**,** 1392-1397.

WANG, X., WANG, H., QI, C. & SIVAKUMAR, A. I. 2014. Reinforcement learning based predictive maintenance for a machine with multiple deteriorating yield levels.

WANG, Z., ZHU, X., ADELI, E., ZHU, Y., NIE, F., MUNSELL, B. & WU, G. 2017. Multi-modal classification of neurodegenerative disease by progressive graph-based transductive learning. *Medical Image Analysis,* 39**,** 218-230.

WANGA, S., LIUA, Y., DI CAIRANO-GILFEDDERB, C., TITMUSC, S., NAIMA, M. M. & SYNTETOSA, A. A. 2018. Reliability Analysis for Automobile Engines: Conditional Inference Trees. *Procedia CIRP,* 72**,** 1392-1397.

WEI, Z., TAO, T., ZHUOSHU, D. & ZIO, E. 2013. A dynamic particle filter-support vector regression method for reliability prediction. *Reliability Engineering & System Safety,* 119**,** 109-116.

WEN, L., GAO, L. & LI, X. 2017. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 49**,** 136-144.

WILLIAMS, G., MEISEL, N. A., SIMPSON, T. W. & MCCOMB, C. 2019. Design Repository Effectiveness for 3D Convolutional Neural Networks: Application to Additive Manufacturing. *Journal of Mechanical Design,* 141.

WIRTH, R. & HIPP, J. CRISP-DM: Towards a standard process model for data mining. Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, 2000. Citeseer, 29-39.

WU, Z., WU, J., CAO, J. & TAO, D. HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012. 985-993.

XIE, M. & LAI, C. D. 1996. Reliability analysis using an additive Weibull model with bathtub-shaped failure rate function. *Reliability Engineering & System Safety,* 52**,** 87-93.

XIE, S., GIRSHICK, R., DOLLÁR, P., TU, Z. & HE, K. Aggregated residual transformations for deep neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 1492-1500.

XIONG, Q., ZHANG, J., WANG, P., LIU, D. & GAO, R. X. 2020. Transferable two-stream convolutional neural network for human action recognition. *Journal of Manufacturing Systems*.

XU, X., WU, Q., LI, X. & HUANG, B. 2020. Dilated Convolution Neural Network for Remaining Useful Life Prediction. *Journal of Computing and Information Science in Engineering,* 20.

YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. 33rd annual meeting of the association for computational linguistics, 1995. 189-196.

YU, J., ZHENG, X. & WANG, S. 2019. A deep autoencoder feature learning method for process pattern recognition. *Journal of Process Control,* 79**,** 1-15.

YUAN, B., GIERA, B., GUSS, G., MATTHEWS, I. & MCMAINS, S. Semi-Supervised Convolutional Neural Networks for In-Situ Video Monitoring of Selective Laser Melting. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 7-11 Jan. 2019 2019. 744-753.

YUAN, M., WU, Y. & LIN, L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. Aircraft Utility Systems (AUS), IEEE International Conference on, 2016. IEEE, 135-140.

ZHANG, J., TANG, Q. & LIU, D. 2019a. Research Into the LSTM Neural Network-Based Crystal Growth Process Model Identification. *IEEE Transactions on Semiconductor Manufacturing,* 32**,** 220-225.

ZHANG, J., WANG, P., YAN, R. & GAO, R. X. 2018a. Deep Learning for Improved System Remaining Life Prediction. *Procedia CIRP,* 72**,** 1033-1038.

ZHANG, J., WANG, P., YAN, R. & GAO, R. X. 2018b. Long short-term memory for machine remaining life prediction. *Journal of Manufacturing Systems,* 48**,** 78-86.

ZHANG, S., LIU, C., SU, S., HAN, Y. & LI, X. 2018c. A feature extraction method for predictive maintenance with time‐lagged correlation–based curve‐registration model. *International Journal of Network Management,* 28**,** e2025.

ZHANG, Y., GUO, Z., LV, J. & LIU, Y. 2018d. A framework for smart production-logistics systems based on CPS and industrial IoT. *IEEE Transactions on Industrial Informatics,* 14**,** 4019-4032.

ZHANG, Y. & LIU, Q. 2002. Reliability-based design of automobile components. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* 216**,** 455-471.

ZHANG, Y., ZHANG, Q., FARNOOSH, A., CHEN, S. & LI, Y. 2019b. GIS-Based Multi-Objective Particle Swarm Optimization of charging stations for electric vehicles. *Energy,* 169**,** 844-853.

ZHAO, G., LIU, X., ZHANG, B., ZHANG, G., NIU, G. & HU, C. 2017a. Bearing Health Condition Prediction Using Deep Belief Network. *Annual Conference of the Prognostics and Health Management Society*.

ZHAO, M., CHOW, T. W., ZHANG, H. & LI, Y. Rolling fault diagnosis via robust semi-supervised model with capped l 2, 1-norm regularization. Industrial Technology (ICIT), 2017 IEEE International Conference on, 2017b. IEEE, 1064-1069.

ZHAO, W. & DU, S. 2016. Spectral–Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. *IEEE Transactions on Geoscience and Remote Sensing,* 54**,** 4544-4554.

ZHOU, L., SONG, Z., CHEN, J., GE, Z. & LI, Z. 2014. Process-Quality Monitoring Using Semi-supervised Probability Latent Variable Regression Models. *IFAC Proceedings Volumes,* 47**,** 8272-8277.

ZHOU, Y., WANG, Y., TANG, P., SHEN, W., FISHMAN, E. K. & YUILLE, A. L. 2018. Semi-supervised multi-organ segmentation via multi-planar co-training. *arXiv preprint arXiv:1804.02586*.

ZHOU, Z.-H. & FENG, J. 2017. Deep forest. *arXiv preprint arXiv:1702.08835*.

ZHOU, Z.-H. & LI, M. Semi-Supervised Regression with Co-Training. IJCAI, 2005. 908-913.

ZHOU, Z.-H. & LI, M. 2010. Semi-supervised learning by disagreement. *Knowledge and Information Systems,* 24**,** 415-439.

ZHU, X. 2011. Semi-supervised learning. *Encyclopedia of machine learning.* Springer.

ZHU, X. & GHAHRAMANI, Z. 2002. Learning from labeled and unlabeled data with label propagation.

ZHU, X. J. 2005. Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences.

ZIMMER, W. J., KEATS, J. B. & WANG, F. 1998. The Burr XII distribution in reliability analysis. *Journal of quality technology,* 30**,** 386-394.

Appendix A.Advanced Data Analytics Technologies

A1 Data mining

In the last decades, data mining has attracted significant attention from the information industry, mainly because there is a large amount of data which can be widely used. There is an urgent need to convert this data into useful information and knowledge. The information and knowledge obtained from data mining can be widely used in a variety of applications including business management, production control, market analysis, engineering design, and scientific exploration, etc. (Hand and Adams, 2014) Data mining is the "mining" of knowledge from large amounts of data. In other domain, such as database, it is also known as ‘knowledge discovery in databases (KDD) (Piateski and Frawley, 1991). It also can be deemed as a stage for knowledge discovering. In 2000, a standard process model for data mining was proposed which name is Cross Industry Standard Process for Data Mining (CRISP-DM). There are six stages in CRISP-DM which are Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment phase. The entire process is iterative and adaptive (Wirth and Hipp, 2000). In order to better deploy data mining in the industry, the steps of Domain understand, Feedback& valuable information, and Knowledge obtaining & storage (Choudhary et al., 2008). In connection with automobile PdM, the whole process is described below:

- **Domain Understanding:** The general process of automobile lifecycle and maintenance need to be understood. Meanwhile, the method to collect the data needs to be identified.

- **Data Collection:** Collecting the data relevant to automobile lifecycle.

- **Data Cleaning & Transformation:** The data is collected from the real world, which means it may contain some impurities. Furthermore, the data may need to be transformed before it is used for modelling.

- **Modelling & Testing:** The TBF predicting model is built based on the cleaned dataset by using deep learning or other machine learning algorithms. Then the model is evaluating using different metrics.

- **Operating:** After the model is built and evaluated, it can be deployed in the actual fleet management.

- **Feedback& Valuable Information:** During the deployment, the pros and cons of the model can be surfaced, which can be used as feedback.

- **Knowledge Obtaining & Storage:** By analysing the information obtained from the last stage, the knowledge can be extracted and used to adjust the previous stages.

A2 Machine learning

Big data analytics is a hot topic in both academia and industry, and it has been widely used in various industries such as business and engineering. Due to the high complexity, dimensions, and variability of the big data, how to implement data mining to extract the knowledge from big data needs to be guided by machine learning techniques. A new challenge for those conventional machine learning algorithms is how to deal with big data. In recent years, prevailing machine learning algorithms such as deep learning has become a useful tool to implement big data analytics. Two benchmarking algorithms which are DCNN and SVM have not been detailed. In this subsection, the algorithm details of both algorithms are introduced.

- **DCNN** (Li et al., 2018) is a novel deep learning algorithm. An input signal is denoted as $X_i = [x_1, x_2, \dots, x_N]$. Firstly, multiple 1D signals are concatenated to a 2D array, which can be expressed by:

$$X_{i:i+k} = \boldsymbol{X_i} \oplus X_{i+1} \oplus \dots \oplus \boldsymbol{X_{i+k}} \quad \text{(A1)}$$

where k is the number of the signal and $\oplus$ is the symbol of concatenate. The convolutional operation is defined as:

$$Z_i = f(W^T X_{i:i+k} + b) \quad \text{(A2)}$$

where $*^T$ denotes the transpose of a matrix and b represent the bias term a. *f()* and *W* represent the activation function and filter kernel. By sliding the filter window from the beginning to the end in the sample data, the feature map of the j^{th} filter can be obtained, which is denoted as:

$$Z_j = [z_j^1, z_j^2, \dots, z_j^m] \quad \text{(A3)}$$

where j is the j^{th} filter kernel.

The four convolutional layers deployed in DCNN is used to extract and abstract the features within the data. Then two FC layers are deployed for RUL prediction. The structure of a DCNN is shown in figure A1.

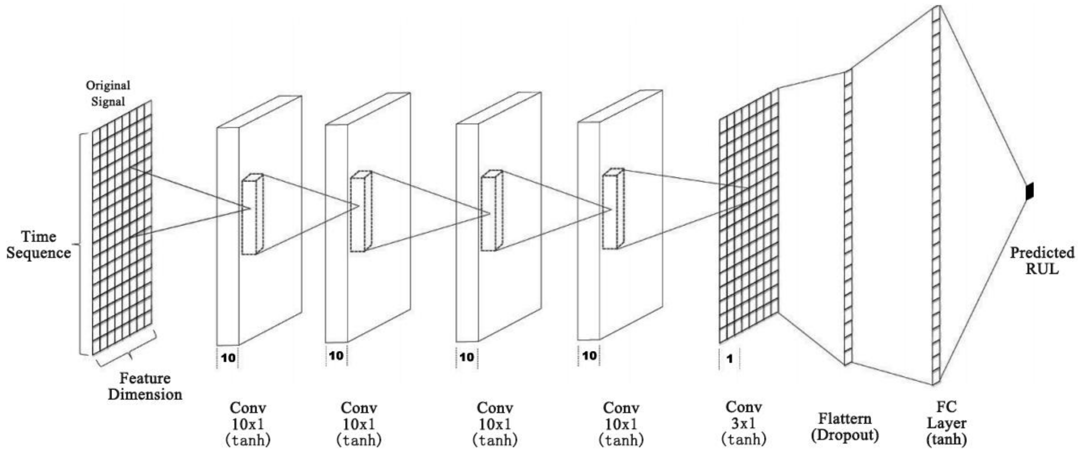

Figure A. 1 The structure of DCNN

- **SVM (Cortes and Vapnik, 1995)** is a supervised learning algorithm which can be used for classification and regression mission. In order to model the TBF, support vector regressor was adopted in this thesis. Assuming the prediction with bias less than ε is deemed as correct prediction, the incorrect prediction results in a high penalty to the algorithm. One of the main components within SVM is kernel function, which is used for pattern analysis. x_i and x_j are two vectors. There are four types of prevailing kernel functions which are widely used in SVM:

$$\text{Linear kernel: } K(x_i, x_j) = x_i^T x_j \tag{A4}$$

$$\text{Polynomial kernel: } K(x_i, x_j) = (\gamma x_i^T x_j + b)^d \tag{A5}$$

$$\text{RBF kernel: } K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2) \tag{A6}$$

$$\text{Sigmoid Kernel: } K(x_i, x_j) = tanh(\gamma x_i^T x_j + b) \tag{A7}$$

where γ, b, d are parameters that need to be set

.

Appendix B. Datasets used in this thesis

B1 Weather dataset (Selected region)

year	month	tmax	tmin	af	rain	sun
2005	1	8.6	3.5	5	36.2	75.4
2005	2	6.5	2.1	3	57.8	54.6
2005	3	9.7	4.2	0	35.9	68.5
2005	4	12.9	5.9	0	35.4	156.1
2005	5	15.4	8.4	0	60.1	225.8
2005	6	19	11.5	0	38.2	191.7
2005	7	19.9	13.7	0	79.6	132.7
2005	8	20	12.5	0	27.7	181.4
2005	9	19.7	13	0	75.1	151.4
2005	10	17	12.1	5	66.3	115.6
2005	11	10.3	4.1	5	48.3	80.3
2005	12	7	2.1	4	63.8	49.7
2006	1	6.5	2.9	4	9.9	49.6
2006	2	6.3	2.2	12	69.7	58.1
2006	3	7.9	2	1	33.2	112.2
2006	4	12.6	5	0	30.4	163.7
2006	5	15.9	9	0	73.4	154.8
2006	6	18.5	11.4	0	16.7	246
2006	7	23.7	16	0	23.2	290.7
2006	8	20	12.5	0	27.7	181.4
2006	9	19.7	13	0	75.1	151.4
2006	10	16.9	11.8	0	74.9	113.7
2006	11	12.1	6	0	57.3	101.4
2006	12	9.2	4.3	0	46	37.7
2007	1	10	4.8	7	57.8	60.1
2007	2	8.8	3.9	3	79.9	58.1
2007	3	11.2	4.6	6	56.2	171

(Continued)

2007	4	15.2	7.3	1	0.2	225.7
2007	5	16.2	9.4	0	150	156.1
2007	6	18.7	12.3	0	102.7	130
2007	7	20.8	12.9	0	70.9	190
2007	8	21.4	14.1	0	72.4	153
2007	9	18.5	11.2	0	34.1	152
2007	10	14.7	8.4	0	48.7	95.2
2007	11	10.4	4.5	2	54.6	67.4
2007	12	8.1	3.4	4	44.5	38.8
2008	1	9.5	4.4	0	80.2	56.35
2008	2	9.55	2.25	7	22	136.1
2008	3	9.35	3.65	2.5	89.2	103.75
2008	4	12.4	4.9	2.5	52.25	178.55

B2 Traffic dataset (Selected region)

year	link_length_km	link_length_miles	cars_and_taxis	all_motor_vehicles
2008	6975.02	4334.08	2.77E+09	3.66E+09
2009	6958.6	4323.87	2.74E+09	3.58E+09
2010	6966.4	4328.72	2.74E+09	3.6E+09
2011	6970.7	4331.39	2.75E+09	3.6E+09
2012	6975.892	4334.62	2.76E+09	3.6E+09
2013	6981.044	4337.82	2.78E+09	3.63E+09
2014	6979.19	4336.67	2.85E+09	3.74E+09
2015	7024.6	4364.88	2.92E+09	3.87E+09
2016	7044.63	4377.33	3.05E+09	4.04E+09
2017	7039.56	4374.18	3.16E+09	4.21E+09

B3 Terrain dataset

LATITUDE	LONGITUDE	Minimum Elevation	Mean Elevation	Maximum Elevation	Maximum Slope	Mean Aspect	Standard Deviation elevation	Average slope	Standard Deviation slope
53.92738	-1.49768	10	162.848879	556	34.98472	266.828243	117.839922	1.33824	1.308976
56.09272	-3.1981	-1	85.180126	281	23.87554	105.601905	44.463102	0.994441	0.660883
50.88191	-3.36686	-27	88.073675	255	20.90952	151.929833	48.399748	0.560512	0.411216
51.4138	-0.54744	-4	164.799476	716	47.77212	111.983092	134.054638	2.018861	2.028791
51.60353	0.169115	10	162.848879	556	34.98472	266.828243	117.839922	1.33824	1.308976
53.37617	-3.9853	14	112.805937	419	38.39471	126.276316	47.874533	1.188044	0.939421
52.21439	1.239178	4	71.297489	285	21.12251	44.532626	43.860767	0.743768	0.700304
51.64711	-2.50283	10	162.848879	556	34.98472	266.828243	117.839922	1.33824	1.308976
51.64006	-0.4684	-4	80.477258	249	30.12485	242.434443	39.842607	0.877658	0.493944
51.81204	-0.06294	-1	85.180126	281	23.87554	105.601905	44.463102	0.994441	0.660883
51.03549	0.535454	-4	164.799476	716	47.77212	111.983092	134.054638	2.018861	2.028791
53.64141	-2.87032	61	167.537891	507	32.18109	204.826791	77.543299	1.023938	0.764644
51.8758	0.436507	14	112.805937	419	38.39471	126.276316	47.874533	1.188044	0.939421
53.3522	-2.8278	14	112.805937	419	38.39471	126.276316	47.874533	1.188044	0.939421
54.8381	-5.80529	-30	157.56912	511	33.93518	84.910715	92.311651	1.295866	0.97198
57.29001	-2.0112	4	71.297489	285	21.12251	44.532626	43.860767	0.743768	0.700304
52.64879	-1.86357	-7	65.309057	489	36.33088	44.084447	66.43052	0.734891	0.696947
52.22644	-2.63851	25	89.433561	235	21.36498	50.576892	43.617429	0.75664	0.543808

(Continued)

51.95541	-4.8397	-13	97.116069	318	23.05055	71.748543	63.18256	1.003636	0.799793
52.01298	-2.17364	-1	85.180126	281	23.87554	105.601905	44.463102	0.994441	0.660883
51.97507	-0.80328	4	71.297489	285	21.12251	44.532626	43.860767	0.743768	0.700304
53.39452	-0.49603	38	121.638969	521	42.29375	67.035419	72.258605	0.89393	0.948026
51.68053	-3.13403	10	162.848879	556	34.98472	266.828243	117.839922	1.33824	1.308976
51.70434	0.150828	2	90.974725	407	32.02382	179.048589	77.372908	1.001826	1.026038
50.9901	-0.18257	-30	157.56912	511	33.93518	84.910715	92.311651	1.295866	0.97198
55.64773	-4.46546	4	71.297489	285	21.12251	44.532626	43.860767	0.743768	0.700304
52.80394	-1.11025	10	162.848879	556	34.98472	266.828243	117.839922	1.33824	1.308976
56.05612	-4.12153	-1	85.180126	281	23.87554	105.601905	44.463102	0.994441	0.660883
55.06354	-2.86111	63	105.796532	252	29.54619	141.438916	49.251535	0.645147	0.578769
53.13766	-0.02924	10	162.848879	556	34.98472	266.828243	117.839922	1.33824	1.308976
57.66015	-4.08767	20	90.103698	179	16.69722	155.666465	39.94855	0.557367	0.496341
51.52195	-0.28562	14	112.805937	419	38.39471	126.276316	47.874533	1.188044	0.939421
52.05341	0.886179	14	112.805937	419	38.39471	126.276316	47.874533	1.188044	0.939421
53.90387	-2.41362	-16	47.953734	191	40.66471	119.591047	34.463993	0.782938	0.521384
52.05358	-0.43561	-20	25.473349	79	20.4001	73.997967	16.316471	0.289462	0.199926
51.67207	-0.28094	-1	85.180126	281	23.87554	105.601905	44.463102	0.994441	0.660883