A DEEP LEARNING MODEL FOR PROGNOSTICS AND SYSTEM HEALTH MANAGEMENT

Rashid Ramadhan BWAMBALE

PhD Dissertation

Department of Computer Engineering

Programme in Computer Science

Supervisor: Asst. Prof. Dr. Cahit PERKGÖZ

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

October 2022

FINAL APPROVAL FOR THESIS

This thesis titled A DEEP LEARNING MODEL FOR PROGNOSTICS AND SYSTEM HEALTH MANAGEMENT has been prepared and submitted by Rashid Ramadhan BWAMBALE in partial fulfillment of the requirements in "Eskişehir Technical University Directive on Graduate Education and Examination" for the Degree of PhD in Computer Engineering Department has been examined and approved on 26/10/2022.

Committee Members		**Title, Name and Surname**	**Signature**
Member	:	Prof. Dr. Cihan KALELİ	
Member	:	Assoc. Prof. Dr. Tansu FİLİK	
Member	:	Assoc. Prof. Dr. Alper Kürşat UYSAL	
Member	:	Assoc. Prof. Dr. Ahmet ARSLAN	
Member	:	Asst. Prof. Dr. Efnan Şora GÜNAL	

Prof. Dr. Murat TANIŞLI
Director of the Institute of Graduate Programs

26/10/2022

SUPERVISOR APPROVAL

PhD student Rashid Ramadhan BWAMBALE, whom I supervise, has completed this thesis titled A DEEP LEARNING MODEL FOR PROGNOSTICS AND SYSTEM HEALTH MANAGEMENT. According to my inspections, the work is scientifically and ethically appropriate for the student to the thesis defense exam.

Supervisor
Asst. Prof. Dr. Cahit PERKGÖZ

ABSTRACT

A DEEP LEARNİNG MODEL FOR PROGNOSTİCS AND SYSTEM HEALTH MANAGEMENT

Rashid Ramadhan BWAMBALE

Department of Computer Engineering

Programme in Computer Science

Eskişehir Technical University, Institute of Graduate Programs, October 2022

Supervisor: Asst. Prof. Dr. Cahit PERKGÖZ

Through health condition monitoring, Prognostics and system Health Management (PHM) techniques facilitate the prediction of the occurrence of current and future failures and the estimation of the future state of engineering systems. These predictions enable machine operators to objectively manage and take timely maintenance decisions. PHM techniques analyze data generated from machines by extracting insights that facilitate the prognostics and diagnostics process in engineering systems. However, due to the nature of these engineering systems, it is extremely difficult to generate and acquire run-to-failure data from real engineering systems because of critical risks that might emerge during the data collection process. Recurrent Neural Networks and the variants have been used extensively in PHM. Nevertheless, because of the impediments known in such models, non-convergence during training occurs frequently. As a result, models were built shallow notwithstanding the knowledge that the depth of a model significantly contributes to the models' optimization and accuracy of results. It is upon this that this study proposes a deep residual sequence-to-sequence model with LSTM neurons intending to overcome the drawbacks faced by deep models. The model was trained and validated by PHMs' publicly available vibration signal data. The data was preprocessed such that it is compatible with the proposed model. Time to Start Prediction was detected objectively by setting a threshold on Pearson's Correlation Coefficients then the remaining useful life was estimated. Lastly, the model was evaluated with Cumulative Relative Accuracy and found to be 17.87% more accurate than the other models in comparison

Keywords: Prognostics and System Health Management, Deep Residual Sequence to Sequence Model, LSTM, Engineering Systems.

ÖZET

PROGNOSTİK VE SİSTEM SAĞLIĞI YÖNETİMİ İÇİN DERİN BİR ÖĞRENME MODELİ

Rashid Ramadhan BWAMBALE

Bilgisayar Mühendislik Anabilim Dalı

Bilgisayar Bilimleri Bilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ekim 2022

Danışman: Dr. Öğr. Üyesi Cahit PERKGÖZ

Sağlık durumu izleme yoluyla, Prognostik ve Sistem Sağlık Yönetimi (PSSY) teknikleri, mevcut ve gelecekteki hataların oluşumunun ve mühendislik sistemlerinin gelecekteki durumunun tahminini kolaylaştırır. Bu tahminler, makine operatörlerinin bakım kararlarını objektif olarak yönetmesine ve zamanında almasına olanak tanır. PSSY teknikleri, mühendislik sistemlerinde prognostik ve teşhis sürecini kolaylaştıran içgörüler çıkararak makinelerden üretilen verileri analiz eder. Ancak, bu mühendislik sistemlerinin doğası gereği, veri toplama sürecinde ortaya çıkabilecek kritik riskler nedeniyle gerçek mühendislik sistemlerinden arızaya kadar veri üretmek ve elde etmek son derece zordur. Tekrarlayan Sinir Ağları ve varyantları, PSSY'de yaygın olarak kullanılmıştır. Bununla birlikte, modellerde bilinen engeller nedeniyle, eğitim sırasında yakınsama olmaması muhtemeldir. Sonuç olarak, bir modelin derinliğinin, model optimizasyonuna ve sonuçların doğruluğuna önemli ölçüde katkıda bulunduğu bilgisine rağmen, modeller sığ olarak inşa edilmiştir. Bu çalışma, derin modellerin karşılaştığı dezavantajların üstesinden gelmek amacıyla uzun-kısa dönem hafıza nöronları ile modele derin bir artık dizi önermektedir. Model, PSSY'nin araştırmacılara açık titreşim sinyali verileri tarafından eğitilmiş ve doğrulanmıştır. Veriler, önerilen modelle uyumlu olacak şekilde işlenmiştir. Pearson Korelasyon Katsayıları üzerinde bir eşik ayarlanarak, Başlangıç Zamanı Tahmini objektif olarak tespit edilmiş ve kalan faydalı ömür tahmin edilmiştir. Önerilen model, kümülatif bağıl doğruluk değerlendirme metriği ile değerlendirilmiş ve önceki çalışmalara göre %17,87 daha doğru ve güvenilir sonuçlar üretmiştir

Anahtar Sözcükler: Prognostik ve Sistem Sağlık Yönetimi, Derin Artık Diziden Dizi Modeli, Uzun-Kısa Dönem Hafıza, Mühendislik Sistemleri.

ACKNOWLEDGEMENTS

First and foremost, I would like to convey my sincere gratitude to my supervisor, Asst. Professor Cahit PERKGÖZ for not only unstintingly guiding me through the course of doing this research, but also for supporting me morally, spiritually, and financially. I would also like to forward my appreciations to the thesis steering committee members, Assoc. Prof. Dr. Ahmet ARSLAN and Assoc. Prof. Dr. Tansu FİLİK for their guidance towards achieving this milestone without forgetting Prof. Dr. Nihan Kosku PERKGÖZ and Prof. Dr. Cem SEVİK for the computer resources that were used to run and train the model proposed in this research. I can affirm that, without their guidance and concerted efforts, it would have been exceedingly difficult to complete this research. Lastly, I would like to sincerely thank my family for exercising patience for the time I have been away from them. Thank you so much and God bless you all.

Rashid Ramadhan BWAMBALE

26/10/2022

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Eskişehir Technical University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Rashid Ramadhan BWAMBALE

CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF IMAGES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

AI	:	Artificial Intelligence
PHM	:	Prognostics and system Health Management
RNN	:	Recurrent Neural Networks
LSTM	:	Long Short-Term Memory
CNN	:	Convolutional Neural Networks
DLM	:	Deep Learning Models
PdM	:	Predictive Maintenance
PvM	:	Preventive Maintenance
TSP	:	Time to Start Predicting
EOL	:	End of Life
MLM	:	Machine Learning Models
LSMT-A	:	Long Short-Term Memory with Amplified weights
ESN	:	Echo System Networks
GRU	:	Gated Recurrent Units
PCC	:	Pearson’s Correlation Coefficient
PA	:	Prediction Accuracy
CRA	:	Cumulative Relative Accuracy
CVG	:	Matrix Convergence
e_i	:	Error for i^{th} Batch
4IR	:	Fourth Industrial Revolution
PPM	:	Prediction Precision Metrics
ψ	:	Psi PCC value
4IR	:	The Fourth Industrial Revolution
δ	:	Delta A symbol for Sigmoid Function

1. INTRODUCTION

1.1. Problem Overview

Engineering systems have been a fundamental integral in humans' daily lives for a long time, this has been so even before the beginning of the industrial revolutions. Machines play a significant role in aiding humans to conduct activities that could have been difficult to carry out without them. During the stone age, humans used stones as a form of an engineering system to ease and enhance their daily essential activities. Activities like farming, transportation, and hunting all needed the use of machines to be accomplished. This period continued until the use of metal was discovered. The revelation of metals was a catalyst for the growth of Industrial Revolutions. However, the impediment to the use of machines has been that after a certain period of operation time, they tend to degenerate and eventually break down. Every engineering system wears out and breaks down and therefore requires repair or replacement as it reaches the apex of its life span.

Figure 1.1 demonstrates the statistical life span of machines from their installation to total failure. In theory and from figure 1.1, it can be deduced that there is a remarkably high probability of a newly installed machine showing seemingly failure behavior (*m*) which is known as a false failure. This false failure is mainly caused by installation glitches and other inception mechanical faults and difficulties. These glitches are experienced during the first few weeks of operation for normal machines.

After the machine stabilizes, the normal operation condition starts, and statistically, the probability of the machine failing during this period is reasonably low. The machine will again show signs of failure as it approaches its End of Life (EOL) which is shown as (*w*) in figure 1.1. At this stage, engineering systems demonstrate sharper failure known as a true failure compared to the failure demonstrated at the inception stage (*m*). At this time the operators should worry because the machine is experiencing true failures (Khamoudj, Benbouzid-Si Tayeb, Benatchba, Benbouzid, & Djaafri, 2020; Mobley, 2002).

Planning for Maintenance should begin when signs of actual failure start to appear. Depending on the current health condition of a failed component, maintenance can be either repairment or replacement or sometimes replacement of the entire engineering system. However, if maintenance is not professionally managed, it might result in mild or even most of the tines catastrophic consequences. The mild consequences include simple delays in manufacturing or delivery, anxiety, and inventory mismanagement. While the catastrophic consequences include causing a company shutdown, impacting heavy losses to an organization, causing environmental catastrophes like fires and water source contamination, or even causing loss of animal and human lives. Therefore, to avoid such events to happen, managing maintenance becomes an important and integral activity in every place where engineering systems operate.

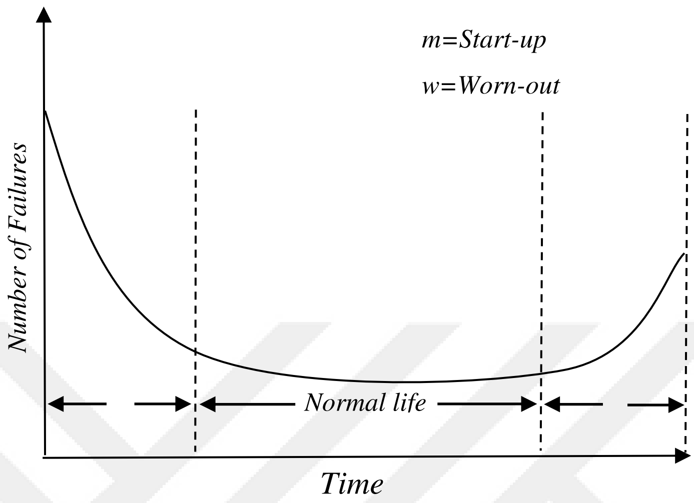


Figure 1.1. *Machine Statistical Life Span Bathtub Curve, Adapted from (Mobley, 2002)*

Maintenance is an activity that should be given the priority it deserves in any industry. Ordinarily, depending on the engineering activities in an industry, the cost of maintenance is in the range of 15 to 60% of the total cost of the products. Beverage industries are found to have the lowest maintenance costs, which is around 15%, this is due to the absence of many industrial activities. In industries with more activities like the steel and iron industries, the cost of maintenance increase exponentially. Such industries take the extreme, i.e., 60% of the average production costs. Other industries that should prioritize maintenance are those with dire consequences if the machine failure happens while in operation, for example, the aerospace and nuclear-related industries (Mobley, 2002).

The transition of the 3.0 to 4.0 industrial revolution (4IR) has come along with challenges in assessing the efficiency of Artificial Intelligence (AI) and its applications. According to studies, AI has not only taken the challenge head-on but also possibly demanded the advent of industry 5.0. The main aim of industry 4.0 is autonomation rather than automation. The autonomation of machines ranges from applications in AI-driven vehicles to self-monitoring of health conditions and self-prognostics (Wenjin Yu, Dillon, Mostafa, Rahayu, & Liu, 2019). Prognostics and system Health Management (PHM) engineers the paradigm shift of machine behavior from automation to autonomation. PHM of machines is realized by using Predictive Maintenance (PdM) techniques (Javed, Gouriveau, Zerhouni, & Nectoux, 2014).

1.2. Problem Background

To understand and appreciate the purpose of PHM, we must understand the traditional maintenance management that was there before PHM was invented. Before the advent of PHM, maintenance ordinarily was of two strategies, reactive as shown in figure 1.2 and preventive maintenance in figure 1.3.

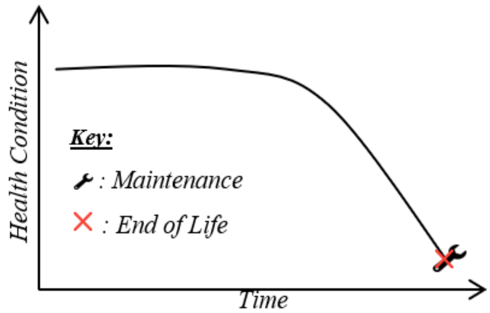


Figure 1.2. *Reactive Maintenance*

1.2.1. Reactive maintenance

This is the most used maintenance strategy in humans' daily lives. The logic behind Reactive Maintenance (RcM) in handling maintenance is "if the machine is broken, fix it, if it hasn't broken, do not fix it" (W. Zhang, Yang, & Wang, 2019). It therefore means, maintenance is performed only when the machine totally fails to perform its required duties as shown in figure 1.2. In most cases, there are no maintenance costs in this type of approach. Maintenance under this category is mostly replacement rather than fixing.

Although RcM is the most common type, it works better and makes much more sense on inexpensive and insensitive engineering systems like fixing a light bulb in a house. However, for expensive machines, it is very risky to wait until the failure of machines occurs, as it might be extremely costly to repair overly expensive damaged parts of the equipment. It is also very risky to employ RcM maintenance strategy t0 machines whose failure is life-threatening like aerial transportation industries. Therefore, this approach is good in some types of engineering systems, and it is rendered useless in others.

1.2.2. Preventive Maintenance

Preventive Maintenance (PvM) is a situation when the equipment is prevented from failing by doing regular checks and quick replacements as shown in figure 1.3. However, the challenge in such a case is to know when a fault will occur and the remaining useful time the engineering system still has to reach its end of life. This type of maintenance is applied in sensitive engineering systems. In such scenarios of executing maintenance within sensitive engineering systems, to predict the time of the replacement, guesswork and subjective prediction are applied which are based on available user manuals or prior knowledge.

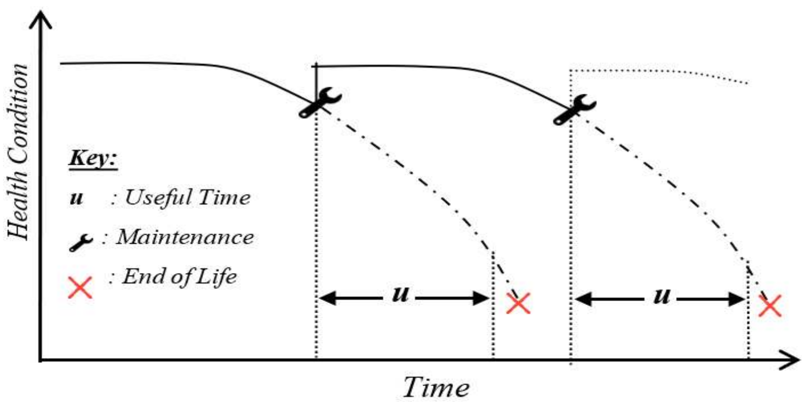


Figure 1.3. *Preventive Maintenance*

This is all done for the purpose of averting dire consequences that might arise if any failure occurs while the machine is functioning. In the case of highly sensitive and expensive mechanical elements, quick replacements are done, as shown in figure 1.3. This could have been a suitable type of maintenance for highly sensitive engineering systems however, it is wasteful as maintenance is done while the mechanical elements still have a significant useful life, as useful life *u* shown in figure 1.3 (Jimenez, Schwartz, Vingerhoeds, Grabot, & Salaün, 2020; W. Zhang et al., 2019)

1.2.3. Predictive Maintenance

Preventive Maintenance (PdM) is performed by applying scientific methods to objectively determine and estimate the time at which the machine is likely to fail as demonstrated in figure 1.4. When the Estimated Time to Failure which is also known as Remaining Useful Life (RUL), is figured out, the optimum time to perform maintenance is scheduled. Unlike PvM, PdM delays maintenance to take advantage of the useful time *u* until the machine closely approached the predicted EOL. PdM is a subset of PHM which enforces the estimation of the equipment's RUL and performs diagnostics (Javed, Gouriveau, & Zerhouni, 2017).

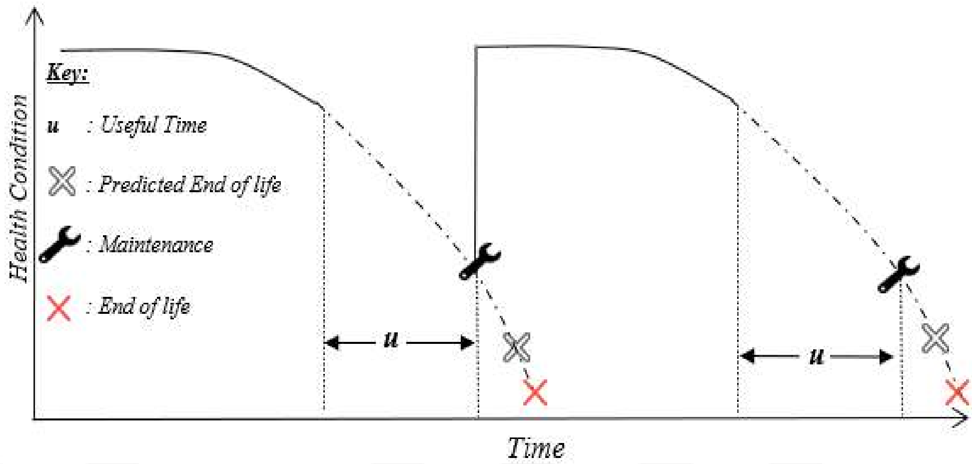


Figure 1.4. *Predictive Maintenance*

To avoid abrupt shutdowns and support continuity, maintenance ought to be planned appropriately. Therefore, in the bid to benefit from minimized maintenance costs, remove or lower human and environmental uncertainties, utilize the useful life of machines, avoid sudden halts, support high safety, and increase reliability, PHM methods are employed. (Tobon-Mejia, Medjaher, & Zerhouni, 2010).

1.3. Problem Statement

Since machines have proven to be a significant integral in humans' daily lives and play a recommendable role, however, it should be known that unfortunately they absolutely reach a point of wear out and eventually they break down. If the downtime of these machines is not well managed, it might be catastrophic and may even cause loss of lives.

PHM engineers the change in basic assumptions of managing and monitoring engineering systems' health conditions. The concept shift has been the introduction of a new form of managing engineering systems behavior from automation and promoted to autonomation. This enables machines with the help of AI to have the capacity of self-health condition monitoring with less or no human intercessions meeting the objective set for industry 4.0. Referring to figure 1.4, studies have shown that PHM is categorized into two main sections, which are Prognostics and Diagnostics.

Prognostics estimates the time to failure which is known as RUL (Tobon-Mejia et al., 2010), while under Diagnostics there are four main activities and these are health/condition monitoring, fault detection, fault classification, and fault severity classification (Dong, Wen, Lei, & Zhang, 2021; Khorram, Khalooei, & Rezghi, 2021; Schwendemann, Amjad, & Sikora, 2021).

PHM that supports the process of PdM has been applied in engineering systems using different methods including machine Learning and Deep Neural Networks. Deep Learning models (DLM) that are proposed for PHM are mostly integrated with other Machine Learning (ML) techniques to support the process, a case in point of Convolutional Neural Networks (CNN) have been used together with Support Vector Machine (Dong et al., 2021) and CNN has also been used with Long Short-Term Memory (LSTM). This integration makes the proposed models complex and difficult to implement (Khorram et al., 2021).

Recurrent Neural Networks (RNN) and the variants have also been used in PHM but in most cases, they have been shallow for the purpose of preventing complexity in models and avoiding exploding or vanishing gradients that might be a reason for model's non-convergence (Chung, Gulcehre, Cho, & Bengio, 2014; W.-C. Huang, Hayashi, Wu, Kameoka, & Toda, 2021; Khan & Yairi, 2018; Pirhooshyaran & Snyder, 2020; Sehovac & Grolinger, 2020; Skomski et al., 2020; Tang, Pang, & Liu, 2020; Zaremba, Sutskever, & Vinyals, 2014). In addition to that, in the most available literature, the Time to Start Prediction (TSP) has been subjectively picked based on prior knowledge of bearings contrary to the main goal of the 4.0 industry revolution of making machines autonomous.

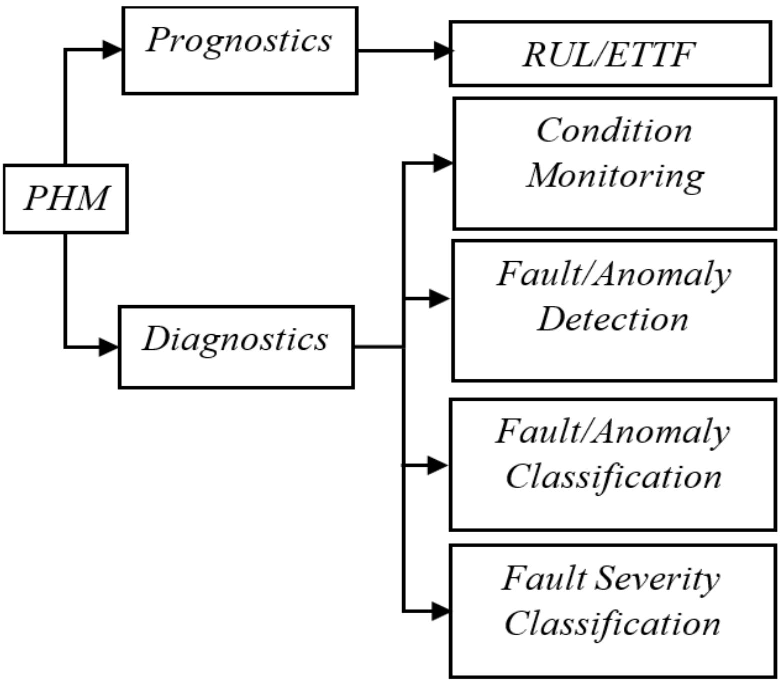


Figure 1.5. *PHM Breakdown and its Subcategories*

1.4. Purpose of The Study

This study aims to propose a deep sequence-to-sequence model with residual blocks of LSTM neurons that can overcome complexity and gradient hindrances. Residual blocks were discovered in a network commonly known as ResNets applied mostly in computer vision and image processing (Hanif & Bilal, 2020; Rao, He, & Zhu, 2017). In the case of the proposed model in this study, residual blocks are customized to be compatible with sequence models that are in the form of Autoencoders. The proposed model was trained and validated with vibration data from mechanical rolling elements in engineering systems that are also known as bearings. Bearings greatly represent the health status of the entire machine, and they have a great impact on both the operation and the life span of engineering systems. (C. Li, Zhang, Peng, & Liu, 2017). In the same regard, this study also proposes a scientific and objective technique for alerting the operator about the occurrence of faults and hence determining the TSP which will subsequently support the estimation of the Remaining Useful Time of the engineering system known as RUL.

1.5. Research Objectives

The research objectives of this study are to.

1. Analyze and preprocess vibration data received from engineering systems so that they become compatible with the model proposed.
2. Use an objective and scientific method to diagnose and detect the occurrence of faults in engineering systems and determine TSP.
3. Study, analyze, and develop a Deep Residual Sequence to Sequence Model (Reseq2seq) that will be able to predict and forecast the future state of engineering systems.
4. Assess and evaluate the model proposed based on a comparative study with results in currently published works.

1.6. Research Contribution

One of the contributions of this research is the proposed technique used in data preprocessing. Data is preprocessed in such a way that if fed into a model, the model is trained to forecast the equivalent of the input data of the same magnitude as that of the proposed lookback value. The preprocessing technique transforms one-dimensional data into dependent and independent variables of three dimensions.

This research also establishes a new scientific and objective method in concurrence with the 4IR that will be applied to detect the point of occurrence of the first fault in the engineering system. This point which is known as TSP is crucial because it supports predicting RUL. Before being converted to 3D, the dependent and independent variables are taken in by the proposed TSP detecting method which produces a scalar variable. This scalar variable is studied, and a threshold is set for detecting TSP.

Lastly, this study also proposes an invented and the novel Deep Learning Model which is trained with the independent and dependent variables acquired from preprocessed vibration signal data to enable it to predict the future state of the engineering system.

1.7. Research Scope and Significance of the Study

This study aims at finding a solution to PHM with DLMs. DLMs and specifically RNNs were studied extensively due to their ability in dealing with time-series data. Vibration data for mechanical rolling elements that are available in public repositories were used in testing and validation mechanisms for the proposed model. TSP detecting methods were also studied aiming to invent a simple, objective, and scientific method for the same purpose.

By the end of this study, it is expected that a mechanism of preprocessing vibration data into a compatible format for RNNs will be realized. An objective and scientific approach to detecting the TSP and the future state of engineering systems will be proposed. For the purpose of predicting the future state, a deep residual sequence-to-sequence model was proposed. This model is expected to be simple and will not be affected by any of the gradient setbacks that were mentioned earlier in this study. This model will aid prognostics and diagnostics processes in engineering systems. The approach proposed in this study is paramount to PHM and the mechanical and engineering systems industry.

1.8. Organization of the Thesis

This thesis is organized into five sections. Section one introduces the problem, section two is about the literature review, section three elaborates on the methodology used in this study, section four throws light on the proposed model, and section five elaborates on the obtained results and discussions.

Section one comprises 9 subsections and is meant to briefly introduce the subject matter in this study. Each distinct subsection gives a detailed description of the problem and the solutions proposed. Subsection 1.3 is about the problem statement; this elaborates on the existence of the problem that this study aims at finding the solution to. The section further elaborated sets the objectives of the study which is a step-by-step flow for solving the mentioned problem. Section one also details the purpose of this research and then describes the research scope by defining clear boundaries of this work. The section in its subsections further mentions the significance of this research and its contribution to knowledge and PHM. Then lastly comes the organization and the structure of this thesis.

Section two is about reviewing relevant and latest literature concerning PHM and is composed of 8 subsections. This section gives the definition of terms used in the subject matter and details some methods available and used in PHM. Explanations of AI-driven solutions to PHM are elaborated, narrowing down to Deep Learning solutions. The section also explains the challenges of applying AI-Based solutions in general and Deep Learning solutions in particular in PHM. Then a brief of the possible ways of overcoming these challenges is elaborated. The last subsection in this section summarizes the section with a review and elaborates on the future and the present of PHM research that gives an insight into the expectations of this research.

Section three is about the proposed research methodology. This section gives a step-by-step approach that elaborates on how the objectives set for this research were achieved. The section is made up of four subsections that start with the proposed research framework which explains the stages set and activities within each stage to realize the objectives of this research. In addition, the abstract of the proposed model is explained diagrammatically starting from preprocessing measures, to model training and validation. The section further explained how the model is tested and assessed, and finally, a summary of all the details in this section is elaborated.

Section four is about the proposed model that was named as Residual Sequence to Sequence (Reseq2seq) Model. The section details the general sequence models and then dives into Sequence-to-Sequence models. Since Long Short-Term Memory (LSTM) is the building block of the proposed model, their operations and computations are elaborated on in this section. Then the proposed Reseq2seq model is explained based on the Sequence-to-Sequence models whose units are those of LSTMs and how it is different from ordinary Sequence to Sequence models. The Encoder and Decoder that make up this model are expounded with mentioning responsibilities of each and how they are merged to form an Autoencoder that is named Reseq2seq in this study. The application of the Reseq2seq model to perform prognostics and diagnostics is further discussed and how the model is evaluated and assessed based on the results it produces.

Section five is about results and discussion and consists of seven subsections. The results obtained from each process proposed in this study, are displayed and discussed in this section. The first subsection highlights the case study and how the data was broken down into training, validation, and testing. It further shows how the model was trained and how it fared with data from different operating conditions. The section then demonstrated the detection of TSP using the proposed method and also shows the evaluation and assessment of this method which is done by carrying out a comparative study. The section then demonstrates how the trained model is used to forecast the future state of the mechanical rolling elements using vibration signal data and how the future state supports the estimation of RUL. The section then demonstrates the assessment and evaluation of the model proposed from which the RULs estimated are used. This evaluation is conducted by a selected evaluation metric and results are compared with published studies. Lastly, the section provides a conclusion that entails the strength of the proposed model and suggests future research in the research area of PHM.

2. LITERATURE REVIEW

2.1. Overview

This section details PHM by explaining its current trends, and existing models meant for prognostics and diagnostics from currently published articles. In addition, the section explains further different published and available solutions for PHM-related problems. To further enlighten the subject matter of this research, this section is broken down into 6 subsections. Subsection 2.1 briefly elaborates on the structure of this section, while subsection 2.2 explains definitions and different types of PHM. Subsection 2.3 further explains the literature available regarding PHM, it clarifies challenges in PHM and the different mechanisms of how the solutions were discovered. Subsection 2.4 discusses the application of deep learning models in PHM, and subsection 2.5 illustrates how PHM models are assessed and evaluated for the purpose of determining how accurate the modes are. Finally, subsection 2.6 is about review and discussion, it gives a summary and conclusion of the whole section and gives an insight into the future research direction of PHM.

2.2. Prognostics and System Health Management

Machines have aided humans in increasing productivity while improving the quality of products and greatly decreasing production time. Even though these machines work as needed, they reach a time of wearing out and they eventually break down. The wearing out is either due to environmental factors or the End of their Lifetime (EOL). It should be noted that, if this time is not well ahead planned for, it might result in unplanned shutdowns and eventually cause a halt in productivity, for overly sensitive engineering systems, it can cause catastrophic eventualities. For this reason, maintenance ought to be planned appropriately. If it is well planned, benefits are enormous which include minimized maintenance costs, removal or lower human and environmental uncertainties, the appropriate utilization of the useful life of machines, avoidance of operations sudden halts, support high safety, and increase reliability (Tobon-Mejia et al., 2010).

Literature available about maintenance shows that maintenance spans from using traditional techniques to scientific mechanisms of determining the appropriate time to perform replacements or repairs. Repairments or replacements are performed on either the affected components of the machine or the complete engineering system if it wears out in totality. As improvements in technology continuously emerge, determining the appropriate time to perform maintenance is transformed from mostly guesswork to the use of user manuals produced by engineering systems manufacturers. Furthermore, with the introduction of industry 4.0 aided by much support from AI research, performing maintenance completely changed from just reading user manuals to autonomation. This paradigm shift is to the discovery of a new way of determining the right time to perform maintenance. The improved maintenance additional component is introduced by the use of AI methods in engineering systems. AI-based solutions are capable of transforming engineering systems into agents that will be able to monitor the health condition and therefore tell the operator how much time the engineering system still has to reach its EOL. All this is done with less input from the external environment. The process of making engineering systems autonomous in the maintenance process is known as Prognostics and systems Health Management (PHM). (W. Zhang et al., 2019).

Based on the type of inputs, there are three approaches to PHM, and these are Model-driven, Knowledge-driven, and Data-driven approaches. Studies have proved that with the emergence of Industrial Wireless Sensor Networks in combination of AI, data-driven models are reliable, and simple to implement and deploy, which is contrary to the other two approaches (Cheng, Chen, Chen, & Wang, 2020; Jimenez et al., 2020; Schwendemann et al., 2021; W. Zhang et al., 2019). These are discussed as follows: -

2.2.1. Model-Driven PHM

Model Driven PHM techniques employ analytical approaches that use physical and mathematical methods to determine the health conditions of engineering systems. Researchers in this regard have used Kalman Filter, empirical techniques, and particle filters to fulfill the task of PHM. Although these methods have produced commendable results, they are limited to prior knowledge of each engineering system which, as a result, makes this approach not suitable in arbitrary situations (Suh, Lukowicz, & Lee, 2022).

2.2.2. Knowledge-Driven PHM

Knowledge-driven PHM is an experienced-based approach to PHM. Machines' life condition is learned through heuristics which are a set of strategies for the correct solution for the problem at hand. Algorithms developed in this approach are based on human expert knowledge about engineering systems. This knowledge is then applied to drive PHM systems. The disadvantage here is that each machine has distinctive behavior and therefore requires a special knowledge base. It is also difficult to get experts for every engineering system produced in every place where they are likely to operate. In addition to that, it is also difficult to collect expertise and experts' knowledge needed for each engineering system to be used to accomplish certain responsibilities (García, Montés, & Alacreu, 2018).

2.2.3. Data Driven PHM

With the emergence of Industrial Wireless Sensor Networks in a combination of Artificial Intelligence (AI) and particularly Machine Learning (ML), data-driven models have been found to produce reliable results and they are simple to implement and deploy which is contrary to the other two approaches (Cheng et al., 2020; Javed et al., 2014; Jimenez et al., 2020; Pech, Vrchota, & Bednář, 2021; Schwendemann et al., 2021; Sutrisno, Oh, Vasan, & Pecht, 2012; W. Zhang et al., 2019).

As it is demonstrated in figure 2.1, data-driven models consist of a series of activities that include acquiring data from sensors, temperature, or any other form of data used in the PHM process. Data is then preprocessed to extract hidden Condition Indicators (CI), or Health Indicators (HI) as named in some works of literature. The CIs should demonstrate certain trend which is analyzable for the purpose of retrieving behavior or certain trend that will support PHM performance. These CIs are then fed into an ML model for training. Based on the behavior extracted from CIs, with the support of other techniques, the ML model is capable of predicting and estimating the expected outputs required from a PHM model (Sutrisno et al., 2012).

The trained model is then assessed with data that it has not seen. If the model produces the expected results, the model is deployed to work in real life. However, if the model does not produce the required results, the process goes back to the sensors to ascertain whether suitable data was collected and investigates why expected results were not achieved. Then the process goes through a similar process as its predecessor through a similar activity as elaborated above. This iteration will continue until the process is optimized to attain the required results with acceptable accuracy.

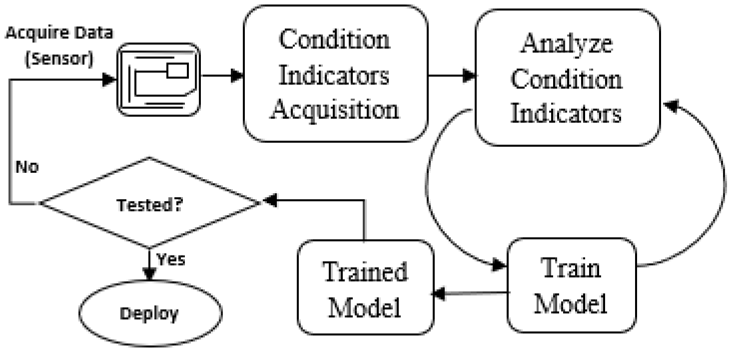


Figure 2.1. *Data-driven PHM Training and Deployment*

The capacity available today that enables the possibility of collecting large data from engineering systems that emerged with the advent of wireless sensors, significantly changed the known assumptions and techniques applied in PHM from a top-down to a bottom-up approach. The previous top-down approach was hectic and tiresome because it involved the traditional physics of machines. This approach also involves the requirement of knowing the structure and the whole architecture of the engineering system before the process of building a model to perform PHM starts. It should also be noted that models were engineering component-specific, in other words, any model built will be build specific for that particular equipment. On the other hand, the bottom-up approach involves collecting data from the equipment first, then the data is analyzed to extract any insights about the equipment's health condition degradation. The insights are then utilized to aid in making decisions about how maintenance should be performed. The bottom-up approach has gained popularity in PHM research because of its simplicity and deliverance of reliable and commendable results (Aydemir & Acar, 2020; Carvalho et al., 2019; Khan & Yairi, 2018).

A case in point of the aerospace industry whose products tend to be overly sensitive because of their nature of failures that may result in huge losses and loss of human lives. The aerospace industry has endlessly endeavored in its production line to produce equipment that will be highly dependable, customizable, and available whenever needed. With the emergence of 4IR, the efforts of producing aerospace mechanical components that are self-manageable, and self-reliant with less or no human input have greatly increased over the years and gained much support in research. These efforts have succeeded due to the emergence of Predictive Maintenance (PdM) which is supported by PHM techniques realized through the autonomation of engineering systems which happens to be the main objective of 4IR.

Engineering systems like airplanes, high-speed trains, nuclear facilities, wind turbines, etc. that use PHM solutions are highly marketable due to their reliability and maintenance effectiveness with low costs. PHM solutions empower these machines in a way that they can inform their operators about how much time is remaining for them to fail. They are also capable of informing the operators about which part of the equipment is about to fail and therefore, require immediate attention either by performing maintenance if the damage is not severe or replacement if the damage is severe. All this is done with little or without human intervention.

Considering the type of inputs, AI-based solutions have broadly been categorized into two categories, deterministic and probabilistic-based solutions. Deterministic-based solutions are further broken down into two, knowledge-based solutions, and data-based solutions. Although the probabilistic-based solutions may fall under both knowledge and data-based solutions, due to their wide scope and having been widely used in many PHM problems, it has been given an independent category as shown in figure 2.2. The probabilistic approaches use diverse types of methods that require preprocessing of data into a kind of data distribution. Some of the methods used in the probabilistic AI-based solutions category include Hidden Markov, Kalman Filter/ Particle Filter, Naïve Bayes classification algorithm, and lastly the Bayesian Neural Networks (BNN) (Cakir, Guvenc, & Mistikoglu, 2021).

It is also indicated in figure 2.2 that Knowledge-based solutions that apply heuristics and quantitative reasoning to approach PHM problems are broken down into solutions like Model-Based solutions, Rule-Based solutions where expert systems are applied, and Ontology-Based solutions. The figure further shows that Data-Based approaches apply Statistical Methods (SM), Machine Learning (ML), and Artificial Neural Networks (ANN) that all fall under the same nomenclature of Machine Learning (ML). ML-Based solutions are further categorized into supervised and unsupervised solutions. There are many solutions under both supervised and unsupervised that have been applied in PHM, among the most prominent ones are Multilayered Perceptron and autoencoders that fall under supervised and unsupervised, respectively, that evolved in Artificial Neural Networks (Khan & Yairi, 2018; Zhu et al., 2022; Zonta et al., 2020).

2.3. Challenges in PHM AI-Based Solution

Most AI-based solutions use previously attained engineering systems experiences represented by knowledge or data that was obtained from similar equipment to create PHM solutions for new systems. The goals of any PHM system are to identify failed components in the engineering system, identify components that are approaching failure by forecasting the future states, and inform the operator of the remaining useful life. These goals have been achieved in most systems even though they are facing some challenges (Khan & Yairi, 2018). Some of the common challenges faced by PHM AI-based solutions include the following:

1. Difficulties in collecting run-to-failure data from real engineering systems due to the considerable risk involved in the process of data gathering. For example, it is hazardous to collect training data from an Airplane from the time it starts until it crushes.
2. Temperature and vibration data are commonly used in PHM but selecting a suitable type of data that will represent machine degradation and its quantification has been a challenge. The type of data that can represent an engineering system as a whole is a big challenge because some elements in the machine may not be able to produce any type of data while showing degradation behavior. This challenge has pushed researchers to concentrate mostly on different machine components to represent the rest of the machine. A case in point, engineering system components like bearings have been used in several PHM research to represent the whole engineering system.
3. There has been a challenge in integrating AI models into already developed engineering systems. Some machines were created with being overly sensitive to any changes and hence difficult to integrate a PHM system.
4. Before reactive systems were developed, deployment of an AI-based solution in the real world was difficult due to its complexity. A model meant to perform prognostics and diagnostics at the same time is evidently complex. The invention of reactive systems solved this challenge by taking in already trained models and just reacting based on the output from the models. Despite all these innovations, the computations that are undertaken behind the scenes are complex and therefore require high computation power.
5. It is difficult to create multitask models that can run on physical systems and perform all tasks under diagnostics and perform prognostics at the same time. Diagnostics which include the classification of faults and detection of the occurrence of faults may not work well in the same system to perform prognostics.
6. The variation of Operating Conditions (OC) is another challenge in PHM. Models meant for a particular operating condition of an engineering system may not be able to work appropriately, or they work with more errors that might not be acceptable in sensitive machines.

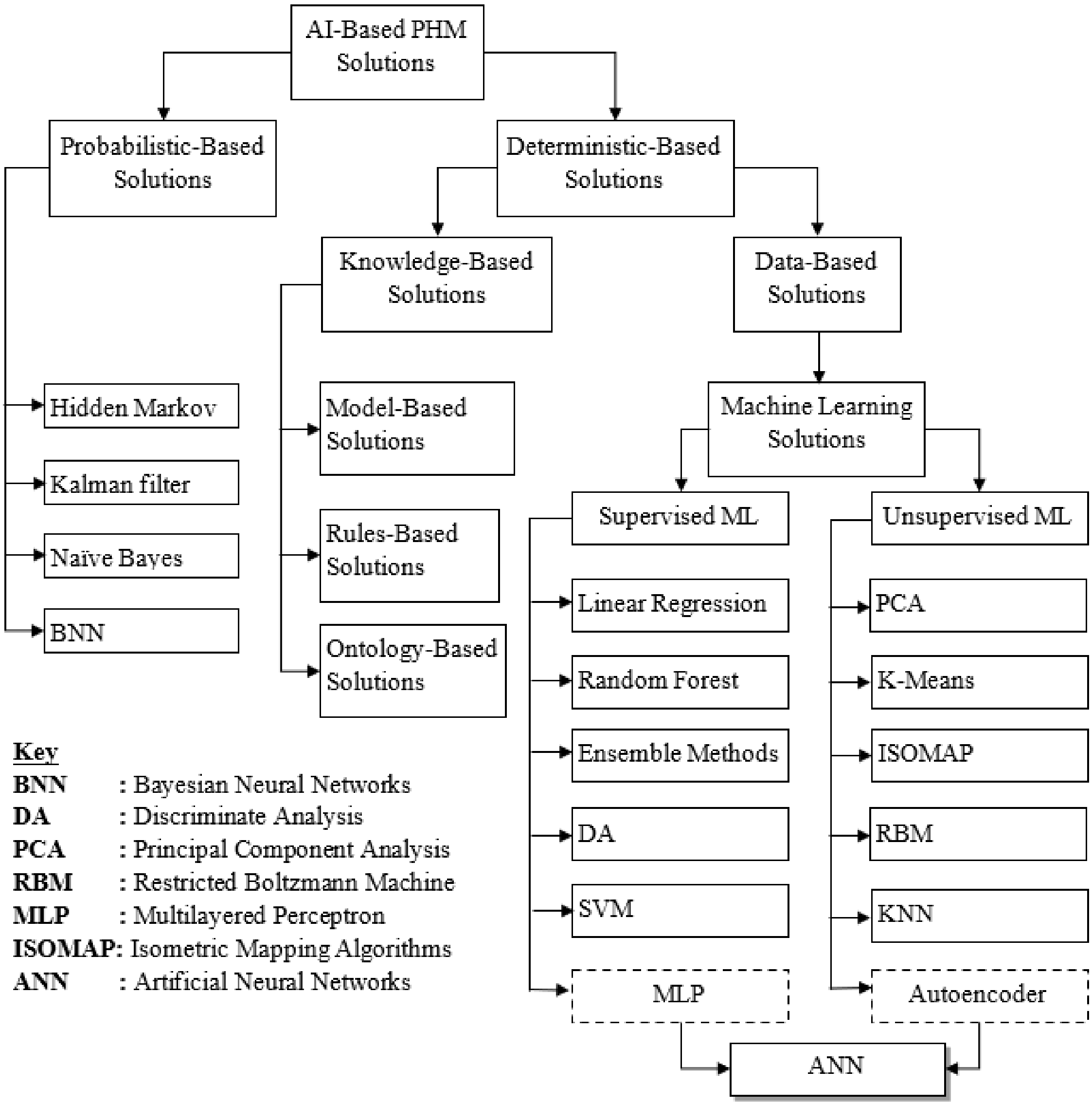


Figure 2.2. *AI Based PHM Solutions*

7. Studies have also discovered noise in sensors during data collection. Any noise precipitated in a sensitive engineering system calls for an alarm. A sensitive engineering system requires that any error in the PHM system should not cause any changes in the operation of the system due to its high sensitivity.
8. There is also the challenge of discriminating one fault from the others. The moment a fault is detected by any AI-based solution, immediately the future state of the machine is forecasted, and RULs computed. The time to discriminate the occurrence of one fault over the other is usually not given the focus after the occurrence of one fault. When one fault occurs detecting another fault in the same sequence is still a challenge for most AI-based solutions in PHM.

9. There is another challenge of false healing and/or false degradation. Some of the data collected from sensors are most likely to demonstrate show natural false healing in data due to the occurrence of the spread of damage and false degradation due to minimal mechanical faults. These cause false alarms that may call for early maintenance when the engineering system still has more useful time and hence taking the entire system back to preventive maintenance.

Despite all these challenges, AI-based solutions have been applied as PHM solutions in engineering systems and they have been able to produce reliable results with some of the challenges mentioned addressed partially or fully solved.

2.4. Literature Available Regarding PHM

Recently, the area of engineering systems maintenance has been widely researched to serve the purpose of reducing the excessive costs of maintenance and at the same time increasing the safety of operators as well as the safety of the systems themselves. The advent of wireless sensors that is capable of collecting diverse types of data from machines has further influenced the increase in the research that uses data to enhance the process of performing maintenance. In a combination of the capability of data collection and AI-based methods, the management of engineering systems maintenance is further boosted through the application of PHM techniques. Many such techniques have been published and they have been found to exhibit reliable results. Table 2.1 shows some important literature published that attempts to overcome PHM problems. The table shows the titles of the articles selected, the model applied the Condition Indicators also known as features used, and the sources of datasets used in the assessment of the proposed models in each article.

The article published by Weiting Zhang (W. Zhang et al., 2019), explains in depth the previous and current trends in data-driven based PHM. The article breaks down the PHM process into; Operational Assessment (OA), Data Acquisition (DA), Feature Engineering (FE), and then Modeling. The process of OA consists of examining the purpose and requirements needed for PHM. OA includes economic and safety purposes, defining possible faults and operation conditions, then assessing how data will be collected from machines. Data collections were categorized into wired, wireless, and hybrid systems. Selecting data collection systems is based on the type and resources available. If the model will go all into the cloud, wireless systems are preferred while if a backup offline might be needed, then a hybrid system is preferred (Ochella, Shafiee, & Dinmohammadi, 2022).

Diverse types of data used in PHM have been mentioned in the article. Based on the type of sensors used to collect can be vibration, acoustic, temperature, electric signal, pressure, or/and rotating speed. This raw data is sampled by using sampling Frequency, sampling time, or/and sampling starts. The raw data itself may not produce the required results if fed into the model in their raw form, this is because the CIs are still hidden in the data and therefore required to be extracted. Before using raw data, CI or features are extracted using different techniques. Models developed for PHM can be trained with several features that can be extracted from data, these include Time Domain, Frequency Domain, and Time-Frequency Domain. Some common features used in PHM are presented in table 2.2. This article also gives a revelation to the trend and the future research area of PHM. The authors explained the process of performing data-driven PHM.

The authors further mentioned that solutions proposed in PHM using ML are categorized into two; traditional ML and Deep Learning (DL) techniques. The article explains the main difference between the methods applied in PHM as categorized in the article. The main difference lies in the fact that MLMs use external techniques for extraction and selection of CI while DL models perform this within the model itself.

The article further illustrates that data acquisition in PHM is through Industrial wireless sensor networks (IWSNs), and industrial cyber-physical systems. It has been proved that data from engineering systems can be gathered using numerous types of sensors with high dependability and in an actual time manner. This enables real-time condition and health monitoring of machines and knowing when most probably the machine is likely to fail.

The efforts concerted from the latest published articles have been on the integration of most of the solutions available in PHM. The merging of anomaly detection in engineering systems, diagnostic and prognostic techniques, and other related functions in PHM is taking the stage in this area of research.

Another case of using ordinary MLMs as a PHM solution is applying a Support Vector Regression Model (SVR). Here two approaches offline and online were put to task to actualize prognostics and diagnostics. The offline model learns the degradation characteristics in the data whereas the online is used to test the trained model. As feature extraction in ordinary MLMs is common, features, in this case, were extracted using wavelet packet decomposition and were reduced by the Isometric Feature Mapping reduction approach. Then the features were fed into an SVR model for the purpose of training. Later the trained model was applied in an online approach. (Benkedjouh, Medjaher, Zerhouni, & Rechak, 2013).

Table 2.1. *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
1	Online detection of bearing incipient fault with semi-supervised architecture and deep feature representation, 18 March 2020	Wentao Mao, Siyu Tian, Jingjing Fanb, Xihui Liang, Ali Safian	stacked denoising auto-encoder and semi-supervised support vector machine (S4VM)	Bearing fault detection	Auto-encoder anomalies output	Three datasets (IEEE PHM Challenge 2012, IMS and XJTU-SY)
2	Constructing a health indicator for roller bearings by using a stacked auto-encoder with an exponential function to eliminate concussion, 23 January 2020	Fan Xu, Zhelin Huang, Fang Yanga, Dong Wang, Kwok Leung Tsui	exponential function is used to eliminate global severe vibration after and Stacked Auto-encoders	Roller bearing	Auto-encoder anomalies output	PRONOSTIA
3	A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders, SEPTEMBER 2020	Aniekan Essien, *Member, IEEE*, and Cinzia Giannetti	Convolutional LSTM Neural Network Auto-encoders	bodymaker machine	RMSE, MAE, sMAPE	Metal packaging plant in the United Kingdom

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
4.	Deep learning models for predictive maintenance: a survey, comparison, challenges and prospect, 7 October 2020	Oscar Serradilla, Ekhi Zugasti, Urko Zurutuza	autoencoders. Deep belief networks. convolutional neural networks (CNN). recurrent neural networks (RNN).	---	----	------
5.	Predictive maintenance using cox proportional hazard deep learning, 22 February 2020	Chong Chen, Ying Liu, et.al	Cox proportional hazard deep learning (autoencoder, Cox proportional hazard model, then LSTM)	Fleet/automobile failure	MCC and RMSE	Fleet maintenance data set provided by a UK fleet company
6.	Review—Deep Learning Methods for Sensor Based Predictive Maintenance and Future Perspectives for Electrochemical Sensors	Srikanth Namuduri et al 2020	autoencoders. Deep belief networks. Convolutional neural networks (CNN). recurrent neural networks (RNN).	---	---	---

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
7	Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoTdata in real-time, 16 January 2021	Serkan Ayvaz, Koray Alpay Serkan. ayvaz@eng.bau.edu.tr (S. Ayvaz), alpay.k@pg.com (K. Alpay).	Random Forest, a bagging ensemble algorithm, and XGBoost, a boosting method	built-in IoT sensors that for detecting properties such as motion, speed, weight, temperature, electrical current, vacuum and air pressure	A total of 101 features, After the mapping and removal of redundant timestamps, the dataset was reduced to 58 features. Principal component analysis (PCA) for dimension reduction, 95% was represented by 17 principal components. *R*2 (R-squared), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Squared Error)	assembly lines producing baby diapers in a real-world production plant

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
8.	A new dynamic predictive maintenance framework using deep learning for failure prognostics, 05 March 2019	Khanh T.P. Nguyen, Kamal Medjaher	LSTM	Turbofan Engine	Min, Max, Mean, Median	Turbofan Engine Degradation Simulation provided by NASA Ames Prognostics Data Repository
9.	Avoiding Environmental Consequences of Equipment Failure via an LSTM-Based Model for Predictive Maintenance	Haiyue Wua, Aihua Huanga, John W. Sutherlanda	LSTM	a motor bearing failure	Time domain features Mean, Variance, RMS, Entropy, Skewness, Kurtosis, Shape factor, Crest factor, Impulse factor, Margin factor, Peak-to-peak. Frequency domain features Mean, Frequency Centre, RMS, Variance, Skewness, Kurtosis, Spectral ratio. Time-frequency domain features Energy ratios of eight frequency sub-bands from haar wavelet package transform	Center for Intelligent Maintenance Systems (IMS) in University of Cincinnati

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
10.	A Hybrid Prognostics Technique for Rolling Element Bearings using Adaptive Predictive Models	Wasim Ahmad, Sheraz Ali Khan, and Jong-Myon Kim	Linear and non-linear regression	Beaaring	RMS	Center for Intelligent Maintenance Systems (IMS) in University of Cincinnati
11.	Industrial Internet of things monitoring solution for advanced predictive maintenance applications, 14 February 2017	Federico Civerchia, Stefano Bocchino, et. Al	NGS-PlantOne	---	33 IoT sensing devices	----
12.	Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model	Mingming Yan, Xingang Wang, Bingxiang Wang, Miaoxin Chang, Isyaku Muhammada	SVM multiclass and hybrid degradation Tracking technique, and linear regression	Bearing	RMS, RRMS and IRRMS	IMS Bearing Dataset and PRONOSTIA dataset

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
13.	A novel bearing intelligent fault diagnosis framework under time-varying working conditions using recurrent neural network, 8 November 2019	Zenghui An, Shunming Li, et. Al: Email: me_anzh@163.com	LSTM bases on infinitesimal method	Bearing	----	Own dataset
14.	A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment, 9 February 2021	Seon Woo Lee, Hyeon Tak Yu, et.al	transfer learning, a deep learning model	Hyper gravity accelerators	Fourier transform (STFT) or Mel-Frequency Cepstral Coefficients (MFCC) spectrogram and converting the input into a 2D image	Pulse 3560C and four accelerometers (B&K 4371) were used to acquire the rotation and vibration data

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
15.	Rolling Element Bearing Feature Extraction and Anomaly Detection Based on Vibration Monitoring	Bin Zhang1, Georgios Georgoulas1, et.al	pdf-based, k-nearest neighbor, and particle-filter-based approaches	Rolling Bearing	characteristic frequencies up to three (3) harmonics, FC1, FC2, FC3 Average of FCs this represents noise level and denoted by SB11 and SB12	Timken LM501310 cup and LM501349 cone
16.	Remaining useful life estimation based on nonlinear feature reduction and support vector regression	T. Benkedjouh, K. Medjaher, et.al	ISOMAP for feature reduction SVR: for Model Construction Regression Prediction: For health assessment of RUL estimation particle-filter-based approaches	Rolling bearing	Mean, RMS, Kurtosis, FFT, WPD, SSE, R-Square, and RMSE	PRONOSTIA

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
17	Railway turnout system RUL prediction based on feature fusion and genetic programming, 21 October 2019	Cong Chen a, Tianhua Xu, et.al	A variant correlation-based feature selection method, Auto-Associative Kernel Regression (AAKR) for prediction, smoothed by using the locally weighted regression, genetic programming (GP) algorithm is employed to predict the RUL	railway turnout systems (RTS) it includes blades, stock rails, and turnout machines	standard deviation (STD), kurtosis (KT), peak to peak value (P2P), skewness (SK), root mean square (RMS), crest factor (CF), mean and variance (Var)	**-----**
18	A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments	K.C. Gryllias, I.A. Antoniadis	SVM	Rolling bearing	Kurtosis, Skewness, Variance, RMS	ball bearing test data set of the Western Reserve University Bearing Data Center Website

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
19.	An adaptive deep transfer learning method for bearing fault diagnosis, 5 November 2019	Zhenghong Wu, Hongkai Jiang, Ke Zhao, Xingqiu Li	An adaptive deep transfer learning method, LSTM and grey wolf optimization algorithm	Bearing	----	Experimental bearing dataset provided by Case Western Reserve University, and the locomotive bearing dataset collected in the practical engineering. For inputting clearer bearing fault information
20.	Automatic bearing fault diagnosis based on one-class v-SVM	Diego Fernández-Francos, David Martínez-Rego, et.al	v-SVM, fast Fourier Transformation	Bearing	RMS, power spectrum through fast Fourier Transformation	IMS dataset
21.	Bearing Fault Diagnosis Based on SVD Feature Extraction and Transfer Learning Classification	Fei Shen, Chao Chen, Ruqiang Yan, et.al	SVD	Bearing	Average curvature, Dimension of fault feature vector	Western Reserve University Bearing Data Center

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
22.	Deep representation clustering-based fault diagnosis method with unsupervised data applied to rotating machinery, 24 March 2020	Xiang Li, Xu Li, Hui Mae,	Auto-Encoder structure, distance metric learning and k-means clustering	Rolling Machinery	Auto-encoder generated features	CWRU dataset and Bogie dataset
23.	Particle swarm-optimized support vector machines and pre-processing techniques for remaining useful life estimation of bearings	Caio Bezerra Souto Maior, Márcio das Chagas Moura, et.al	Swarm-Optimized SVM, Empirical Mode Decomposition (EMD), Wavelet transform	Bearing	Absolute peak amplitude, kurtosis and entropy	PRONOSTIA dataset
24.	Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review	Purushottam Gangsar, Rajiv Tiwari	---	Bearing faults, Rotor related faults	----	----

Table 2.1. (Continued) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
28	Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation, 12 December 2019	Xiang Li, Xiao-Dong Jia, et.al	auto-encoder, and cross-machine adaptation algorithm is adopted for knowledge generalization,	Rolling bearing	----	Bearing Data Center of Case Western Reserve University, Train bogie dataset, and *IMS dataset*
29	Averaged Bi-LSTM networks for RUL prognostics with non-life cycle labeled dataset, 17 March 2020	Yong Yu, Changhua Hu, et.al	Bi-LSTM	Graphite/LiCoO2 battery	MAE	CS2 battery degradation datasets of CALCE Battery Research Group
30	Deep multi-scale convolutional transfer learning network: A novel method for intelligent fault diagnosis of rolling bearings under variable working conditions and domains, 14 May 2020	Bo Zhao, Xianmin Zhang, et.al	Deep multi-scale convolutional neural network (MSCNN)	Rolling bearing	--	experimental rolling baring dataset, which is from Case Western Reserve University Lab

Table 2.1. (Continues) *Important Publications Related to the Study*

No.	Title of the article and the year published	Name/s of the author/s	Model applied	Case study	Features used	Dataset used
31	Fault detection and identification of rolling element bearings with Attentive Dense CNN, 16 May 2020	Spyridon Plakias, Yiannis S. Boutalis	Attentive Dense Convolutional Neural Network (ADCNN)	Rolling bearing	---	IMS bearing dataset and Case Western Reserve University (CWRU) bearing dataset
32	Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions	Bo Zhao, Xianmin Zhang, et.al	Normalized convolutional neural network	Rolling Bearing	---	Case Western Reserve University Lab

Table 2.2. *Common Statistical Features Used in PHM*

Feature Name	Equation	Feature Name	Equation	Feature name	Equation
Wavelet Energy	$\sum_{i=1}^{n}\frac{\omega t_{\varphi}^{2}(i)}{n}$	Spectral Power	$\sum_{i=1}^{n}(f_i)^3 S(f_i)$	Root Mean Square Frequency	$\sqrt{\frac{\sum_{i=1}^{n}(f_i{}^2 X_i)}{\sum_{i=1}^{n}X_i}}$
Standard Deviation (STD)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu(x))^2}$	Spectral Kurtosis	$\sum_{i=1}^{n}\left(\frac{f_i-MF}{STDF}\right)^4 S(f_i)$	Spectral Skewness	$\sum_{i=1}^{n}\left(\frac{f_i-MF}{STDF}\right)^3 S(f_i)$
Standard Deviation Frequency (STDF)	$\sqrt{\frac{\sum_{i=1}^{n}((f_i-MF)X_i)}{\sum_{i=1}^{n}X_i}}$	Variance Frequency	$\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu(x))$	Mean Frequency (MF)	$\frac{1}{n}\sum_{i=1}^{n}f_i$
Mean Square (MS)	$\frac{1}{n}\sum_{i=1}^{n}x_i{}^2$	Root Mean Square	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i{}^2}$	Variance	$\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu(x))^2$
Skewness Value	$\frac{1}{n}\sum_{i=1}^{n}\frac{(x_i-\mu(x))^3}{STD^3}$	Kurtosis value	$\frac{1}{n}\sum_{i=1}^{n}\frac{(x_i-\mu(x))^4}{STD^4}$	Skewness factor	$\frac{\frac{1}{n}\sum_{i=1}^{n}x_i{}^3}{(\sqrt{MS})^3}$

In one of the articles that were authored by Wasim Ahmad, Sheraz Ali Khan, and Jong-Myon Kim, simple linear and quadratic regressions were used to solve the problem of machine faults prediction and computing the Remaining Useful Life (RUL) of bearings. The authors suggest that using complicated algorithms like deep learning models for PHM-related problems was unnecessary for problems that can be solved with simple machine learning models like linear regression This was suggested, however, the significance of high accuracy achieved when DL models are applied in overly sensitive, expensive and risky engineering systems was not considered.

One of the main contributions of Wasim's article was to develop a new method that uniquely determines the expected time to start predicting RUL which was named the Time to Start Predicting (TSP). This time was established by the application of the growth rate of the condition indicators by revealing the exact time when the degradation in a bearing's health condition will emerge. An accurately detected TSP improves the predictive performance of prognostic methods, especially when the degradation of the engineering system is in its initial stages and has not entered the severe degradation stage. Another contribution of this article was the introduction of a technique called linear rectification technique (LRT) that was proposed to manage spurious fluctuations of vibration data in the health indicator that, as a result, might cause a false healing appearance and hence affect the final results. The purpose of LRT was to ensure that the health indicators are all the time in a monotonically nondecreasing mode, which makes improvements in the accuracy of the regressive model and in turn, it improves the accuracy of the RULs to be predicted. Finally, this article proposes a new method that vigorously determines the failure upper limit of a bearing using the slope of the health indicator's curve, which improves the predictive performance of the proposed method in the article.

In another case regarding the application of ordinary MLMs in PHM, a Support Vector Regression Model (SVR) was applied. Here the offline and online approaches were put to task to actualize prognostics and diagnostics. The offline model learns the degradation characteristics in the run-to-failure data as a wholesome whereas the online model is used to assess the trained model based on data collected in a specified unit of time. As feature extraction in ordinary MLMs is common, features, in this case, were extracted using wavelet packet decomposition and were reduced by the Isometric Feature Mapping reduction approach. Then the features were fed into an SVR model for the purpose of training. Later the trained model was applied in an online approach, and according to the study, reliable results were achieved (Benkedjouh et al., 2013). There is also another study that suggested a digital twine in PHM (Aivaliotis, Georgoulias, & Chryssolouris, 2019).

2.5. Deep Learning Models applied in PHM

In the case of applying DLMs in PHM, the DLMs in most cases have been integrated with other MLMs, the integration is either a hybrid of DLMs or an integration of DLMs with ordinary MLMs. A case in point of integrating DLMS with ordinary MLMS includes the integration of Convolutional Neural Networks (CNN) with Support Vector Machine (SVM) (Dong et al., 2021) and hybrids of DLMs in the same regard have also been proposed in many studies like for the case of the integration of CNN with Long Short-Term Memory (LSTM) (Khorram et al., 2021). All the integrations and merging of different techniques are efforts aimed at attaining better and accurate results in implementing PHM in machines due to the possible sensitivity of the engineering system (C.-G. Huang, Huang, Li, & Peng, 2021; Liang & Parlikad, 2020; R. Lin, Yu, Wang, Che, & Ni, 2022; T. Lin, Wang, Guo, Wang, & Song, 2022).

Recurrent Neural Networks (RNN) and its variants that are known for delivering commendable results in timeseries and sequence related data analysis have been applied extensively in PHM. The RNNs delivered as expected since data used to train models for PHM related solutions are in sequence form because they are collected over a specific period of time and this time is what is known as start-to-failure interval time. It should be noted however that models built with ordinary RNN or its variants that were meant to perform PHM were deliberately made shallow aiming to avoid complexity and the gradient setbacks that include the exploding or vanishing gradients. A case in point of a simple Vanilla LSTM model that was applied in different models for the purpose of overcoming complexity in models used in PHM(Chung et al., 2014; W.-C. Huang et al., 2021; Khan & Yairi, 2018; Pirhooshyaran & Snyder, 2020; Sehovac & Grolinger, 2020; Skomski et al., 2020; Song, Liu, & Zio, 2022; Tang et al., 2020; Wu, Yuan, Dong, Lin, & Liu, 2018; Zaremba et al., 2014).

When DLMs are applied in PHM, published studies have demonstrated that data is preprocessed into shapes and forms compatible with the model to be utilized. Data has been converted into image format using techniques like wavelet transformation, short-time Fourier, and Mel-frequency Cepstral Coefficient Spectrogram (Lee et al., 2021;

Shao, McAleer, Yan, & Baldi, 2018). As for transfer learning, a one-dimensional CNN was used in extracting required condition indicators, then correlational alignment lessened the marginal distribution inconsistencies between the dependent and independent variables (J. He et al., 2021).

Denoising, Sparse, and Variational Autoencoders have also been applied in PHM with notable results. For these Autoencoders, in contrast with other DLMs, feature extraction from the data before being fed into the models was performed. Frequency Spectra that are applied in time series analysis for extraction of condition indicators were used. The features were then fed into a Stacked Denoising Autoencoder (Jia, Lei, Lin, Zhou, & Lu, 2016; Miao, Yu, & Zhao, 2022; Zhao, Feng, & Wang, 2022). In another research paper, Winner-take-it-all Autoencoder was used to denoise the data with white Gaussian noise to examine the detection of noise in the data (C. Li et al., 2017). Others have also used low pass filtering to clean the vibration data to be used in quadratic function-based CNN autoencoder (Chen, Qin, Wang, & Zhou, 2021; S. Zhang, Ye, Wang, & Habetler, 2020).

Probabilistic models have also been used in PHM. A semi-supervised Variational Autoencoder-based generative model that manages the problem of data labeling was proposed. The study solved the problem of shortage and correctness of labeled data for supervised learning in PHM since it is cumbersome to collect such kind of data in real life (S. Zhang et al., 2020).

Putting RNNs to test, an LSTM vanilla model was used to estimate the RUL of engineered systems where convoluted tasks, model degradation, noise, and working conditions of these systems were considered. In addition to that, the use of dynamic differential technology to obtain inter-frame data, and perception capability about model degradation processes were also examined (Wu et al., 2018). In another study, bidirectional LSTM blocks in models were implemented for the purpose of extracting features based on the past and future expected data in PHM, and reliable results were achieved (Elsheikh, Yacout, & Ouali, 2019; Jiujian Wang, Wen, Yang, & Liu, 2018). A combination of CNN based on attention mechanism and Bidirectional LSTM and a convolutional long short-term memory has also been applied in estimating RUL in PHM (Elsheikh et al., 2019; Luo & Zhang, 2022; Wan et al., 2022; W. Wang, Lei, Yan, Li, & Nandi, 2022).

Another RNN variant that is not widely used like LSTMs and GRUs is the Echo System Networks (ESN) variant. ESNs are different from other variants because of their unique feature of having neurons that are sparsely connected and have recurrent loops through their synaptic sequence of neurons. A one-step method of ensembles of ESNs were developed to deal with the problem of data assembled at unequal timesteps (M. Xu, Baraldi, Al-Dahidi, & Zio, 2020).

In the same regard, a new breed of LSTM model was developed with an aim of increasing accuracy in PHM. A Long Short-Term Memory that comprises weights amplification (LSTMP-A) was proposed for the purpose of forecasting the future state and estimating the RUL of a gear. Unlike the commonly known LSTM, LSTMP-A boosts parameters in the hidden layers by attention schemes regarding the type of the input data. Data was first prepossessed using M-band flexible wavelet denoising and the reduced using Isometric Mapping Algorithm, then fed into the model for training (S. Xiang, Qin, Zhu, Wang, & Chen, 2020). The transfer learning mechanism of model training with a supervised metric for RUL that works in several operating conditions was also proposed (J. He et al., 2021; Zhuang, Jia, & Zhao, 2022).

Most models that apply LSTM neurons have been deterministic not considering the uncertainties and bias that might probably be inculcated into the industrial data collected. However, some probabilistic models have also been proposed in the same regard. Ensembles of probabilistic LSTM predictors and correctors were applied to the PHM of bearings. This was done in line with ISO standards for determining the failure threshold (ISO 10816). A two-stage LSTM model ensemble was suggested, consisting of a predictor step for forecasting and a corrector stage for counteracting the RUL prediction. The ensemble of all the various LSTM models provides the epistemic ambiguity in the RUL prediction, and each LSTM model within it is tailored to include a Gaussian layer that captures the aleatoric uncertainty in the forecasted parameter (Nemani, Lu, Thelen, Hu, & Zimmerman, 2021).

2.5.1. Limitations in Using Deep Learning Based PHM Solutions

Deep Learning Models (DLM) main focus has been mostly on computer vision and Natural Language Processing. However, researchers have discovered ways of using DLM as a solution to PHM problems. Recent DLMs have proved the capacity to learn insights from PHM data and to deliver reliable and trusted results. Even though DLMs have delivered reliable results in PHM, some limitations were discovered in the same regard, some of these limitations are as follows:

1. The complex structure of computation in deep learning models required bigger storage capacity and high computation power. The fact that deep learning models take in raw data and extract health condition indicators within themselves and give results make the mathematics within the models more complex. It is therefore upon this to find a balance between applying external methods to extract features and condition indicators before feeding DLMs with these features and using DLMs to do all this work as it is indicated in figure 2.3 option 1 and option 2, respectively. If the accuracy from both options remains constant or with insignificant differences, among the two options, the one that will utilize less computation power is preferred over the other.
2. The other challenge of DLM in PHM has been the vanishing and/or exploding gradient problem. Due to the long dependence of data used in PHM, during training and optimization, if the model is deep enough, there are higher chances of having an exponentially decreasing gradient if the initial parameters were set to zero and or an exponentially increasing gradient if the initial parameters were set to 1. This problem significantly affects the process of training the model and may result in data underfitting (Fink et al., 2020).
3. Deep learning models require huge data for better training and optimization of the models, the more the data you give to a DLM the more the accurate results it will produce. PHM data tend to have a small number of labeled data in comparison with unlabeled data, this is due to the complexity involved in collecting labeled data. Since it is exceedingly difficult to obtain labeled data, the solution to this could be finding a way of combining data with experts' knowledge. Experts' knowledge and heuristics are paramount in ensuring the safety of any engineering system. During the training process of DLMs, when the model requires more labeled data, the model should be trained to be able to query and consult experts' knowledge and heuristics about the particular label needed. This might come out to be the best solution to the problem of lack of sufficient labeled data in PHM (Cao, Zhang, & Tang, 2018).

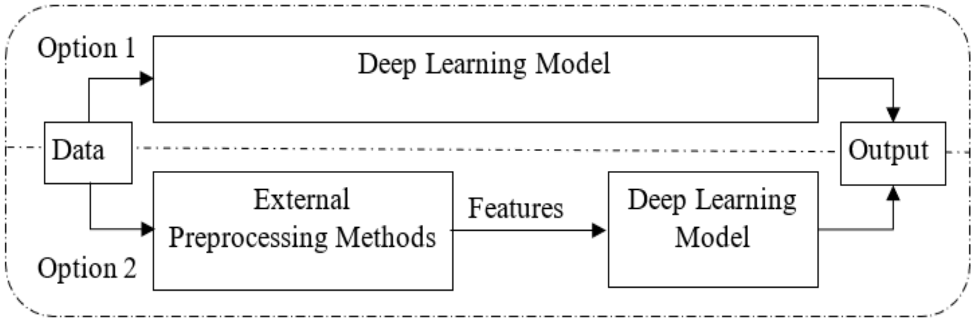


Figure 2.3. *A Two-way Option of Applying Deep Learning Models in PHM*

2.6. Data Used in PHM

Data that originate from the actual operating equipment for PHM is exceedingly difficult to obtain. This is because the nature of the actual engineering system from which data is collected is complicated and therefore data acquisition is expensive and might be life-threatening. A case in point here collecting run-to-failure data from an Aeroplan or a high-speed train. It carries considerable risky to continuously collect data from such equipment to the point of failure, this is because this might cause severe damage to the whole engineering system and may cause loss of life to whoever is operating the machine and also to the users.

The advent of simulators and wireless sensors has provided a cheaper and less risky option when collecting PHM data is concerned. This approach has facilitated researchers' capability to make full use of data from the seemingly actual operating equipment rather than from the experimental and synthetic platforms. Because of this, the practical problems in industrial manufacturing have been better solved. A substantial number of published research have applied publicly available datasets provided by some public platforms to validate and test models proposed in PHM. Some of the widely used datasets are collected as shown in table 2.3. The table shows some of the available datasets that contain bearings vibration data (Du, Hou, & Wang, 2022; R. Lin et al., 2022; Miao et al., 2022; S. Zhang et al., 2019).

Table 2.3. *Publicly Available Bearing Datasets Used in PHM*

No.	Provider	Link to the Dataset
1.	FEMTO-ST	http://data-acoustics.com/measurements/bearing-faults/bearing-6/, https://www.femto-st.fr/en
2.	XJTU-SY Bearing Datasets	http://biaowang.tech/xjtu-sy-bearing-datasets/
3.	Acoustics and Vibration Data	http://data-acoustics.com/
4.	PcoE Datasets	https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/
5.	University of Connecticut	https://figshare.com/articles/Gear_Fault_Data/6127874/1
6.	IMS	https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/
7.	Paderborn	https://mb.uni-paderborn.de/kat/forschung/datacenter/bearing-datacenter/
8.	MFPT	http://data-acoustics.com/measurements/bearing-faults/bearing-2/
9.	CWRU	https://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website

2.6.1. FEMTO-ST (PRONOSTIA Dataset)

Franche-Comté Electronics, Mechanics, Thermal Processing, Optics Science and Technology (FEMTO-ST) at the department of Automatic Control and Micro-Mechatronic Systems developed a bearing experimental platform that they called PRONESTIA for testing and validating bearing PHM system as it is demonstrated in image 2.1. The primary objective of this experimental test is to deliver actual run-to-failure data that depicts the degeneration of bearings throughout their operational lifetime from when they start operating until they reach their EOL. This experiment, if conducted in a real engineering system, may take a long time. But with the help of additional equipment (PRONESTIA) the test bed was designed to complete in just a few hours and yet provide the obtained data from the machine as in real life. This allows the operation of a sizable number of experiments in a short period of time (Nectoux et al., 2012).

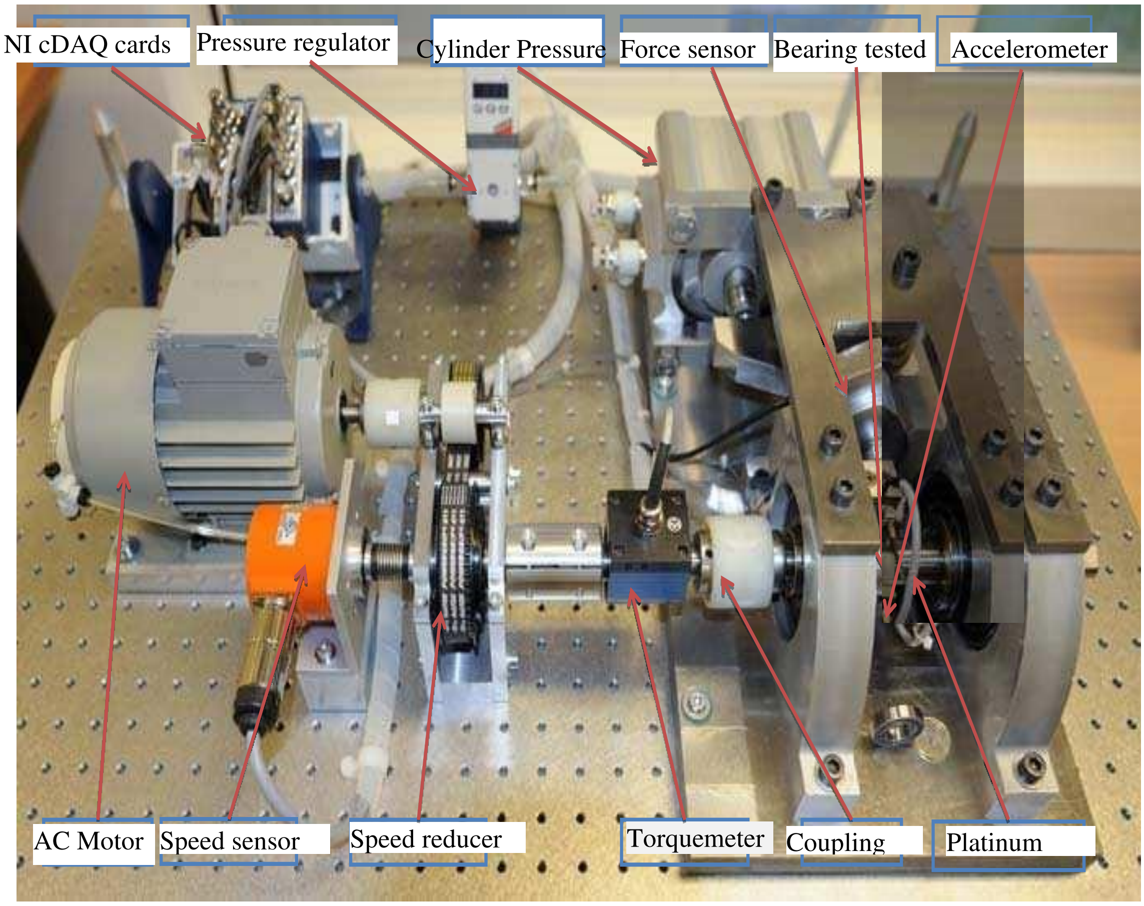


Image 2.1. *The Image Above Demonstrates PRONESTIA Test Bed Platform*

This experiment was developed to produce data for a challenge of PHM that was named the 2021 IEEE International Conference of Prognostics and Health management. As shown in image 2.1, three basic systems make up the PRONOSTIA test bed and these are the rotating system, the generation of the radial force system which is a result of a radial force imposed on the bearing that is being assessed, and lastly the measurement system.

The rotating system is comprised of the asynchronous motor accompanied by the gearbox and the two shafts as shown in image 2.2. The first shaft is fixed closer to the motor while the second is put at the ride side where the incremental encoder is found. Through a system of gearing and other connections, the asynchronous motor acts as the actuator to cause the bearing to revolve. The motor, which has a power output of 250 W, transmits the rotational motion through a gearbox, enabling it to run at its rated speed of 2830 rpm and produce the rated torque while keeping the secondary shaft's speed at less than 2000 rpm. A timing belt that connects two pulleys and is kept in place by a turnbuckle makes up the gearbox, which is hand-made. Shaft couplings, both flexible and rigid, are used to provide connections for transferring the motor's spinning motion to the shaft support bearing.

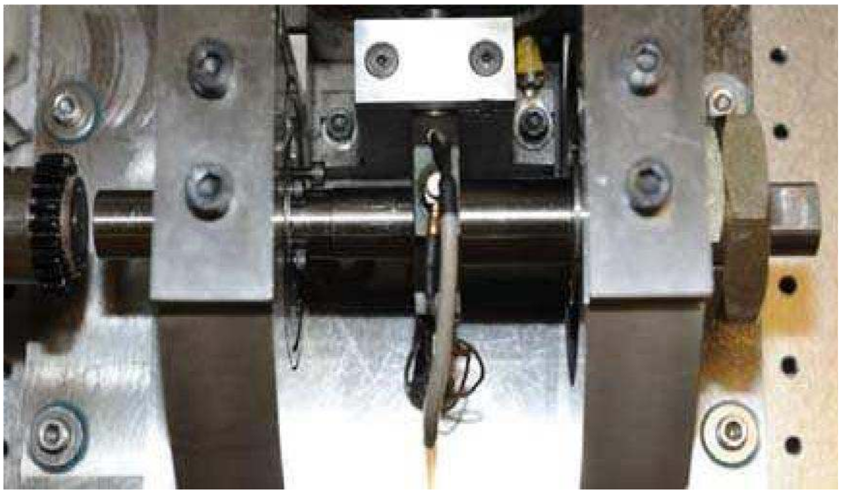


Image 2.2. *The Shafts that Support Bearing in the Experimental Study*

The other system on the test bed is the generation of the radial force system. This system is comprised of an aluminum plate that gives support to the pneumatic jack, a force sensor, a vertical axis with its level arm, a support test bearing shaft, a clamping rind of the experimental bearings, and two pillow blocks with the bearing as it is shown in image 2.3. The center of the complete system is the radial force exerted on the test ball bearing. By increasing its value to the bearing's 4000 N maximum dynamic load, the radial force shortens the life of the bearing. This load is produced by a force actuator that consists of a pneumatic jack, with a digital electro-pneumatic regulator supplying pressure.

On the PRONOSTIA test bed, the measurement system takes measures of the radial force that is exerted on the bearing, the torque inflicted on the bearing, and most importantly the rotation speed of the shaft holding the bearing. All these measurements are acquired at the frequency of 100Hz. The other measures considered in the measurement system are the ones extracted from the bearing that represents its degradation. For the case of the PRONOSTIA test bed, two distinct types of sensors were placed in the measurements system to actuate this task, which are the accelerometers and the temperature sensors as shown in image 2.4. Two miniature accelerometers are used in the case of the PRONOSTIA bed, which are placed at 90^0 to each other to enable them to collect vibration data from the vertical axis and horizontal axis. These accelerometers are positioned centrifugally at the external race of the bearing being assessed. The other sensor is the temperature sensor, the Resistance Temperature Detector sensor platinum PT100 (1/3 DIN class) probe is used in the experiment. This is positioned inside the hole near the ring of the external bearing. The vibration data are sampled at 25.6 kHz and the temperature at 10Hz.

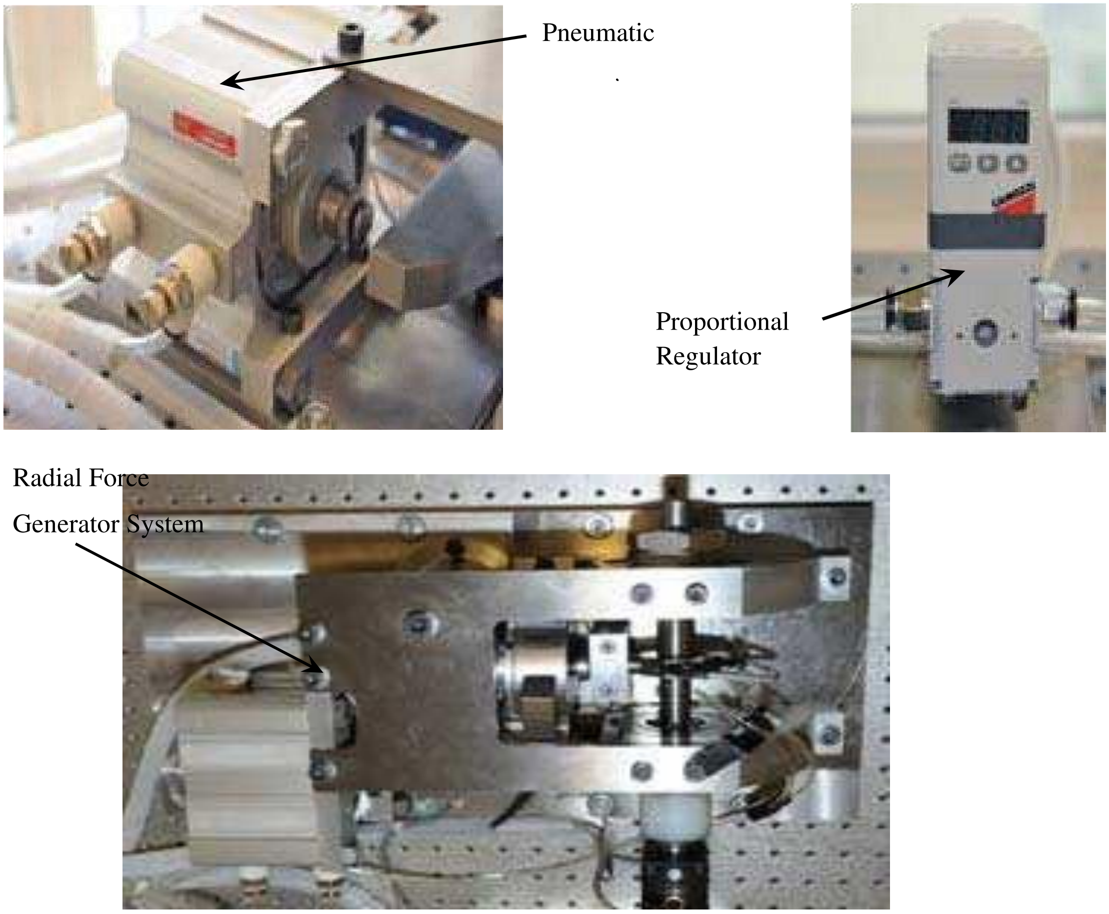


Image 2.3. *The Pneumatic Jack, the Pressure Regulator, and the Radial Force Generation System*

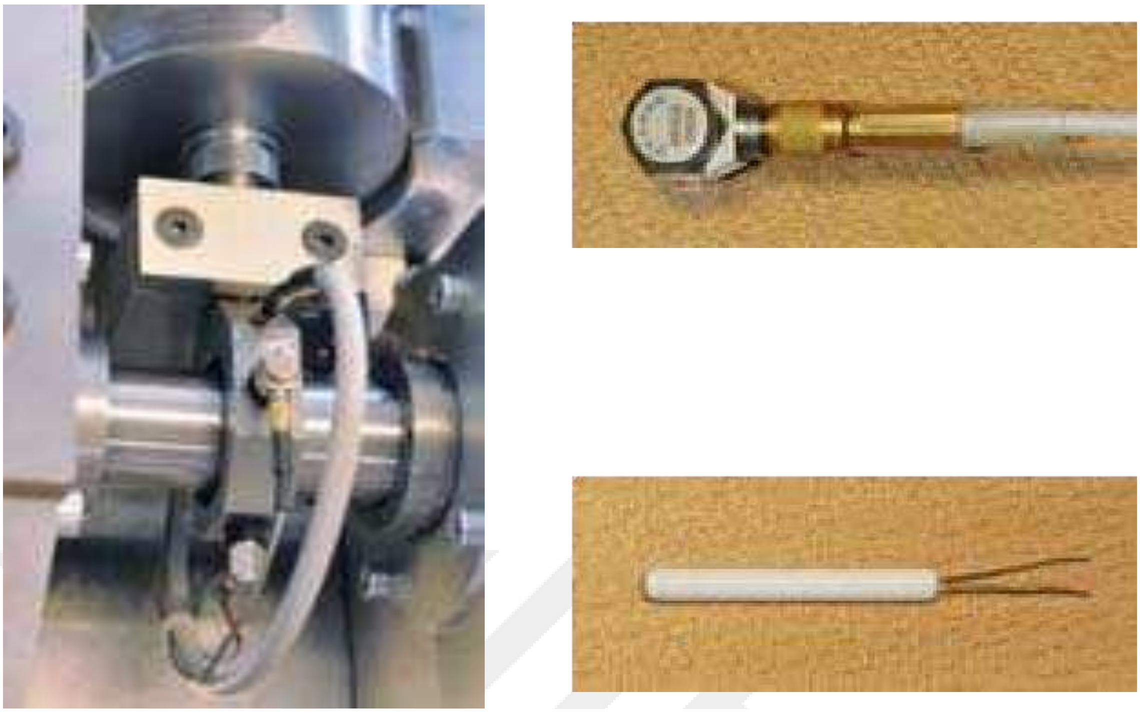


Image 2.4. *The Measurement System Made up of the Accelerometers and Temperature Sensors*

During the operation of the experiment test, the run-to-fail data is determined from the time the experiment is started to the time the bearings show signs of damage and malfunctioning. The damages displayed on the bearings are usually on the balls, inner or outer races, and the cage. Different bearings displayed different damages however, the causes of one damage over the other are not investigated. Some of the damages incurred on bearings after running the bearing in the experiment are as shown in image 2.5.

The raw vibration signal data collected as displayed in figure 2.4 shows a pattern degradation behavior in a non-linear manner. The figure indicates that as the health condition of the bearing worsens; the vibration signals show a monotonic behavior. This behavior is the focal point of prognostics and diagnostics.

To give a sense of operating in different environments, vibration signal data were collected based on three operation conditions which are 1800 rpm and 4000N, 1640 rpm and 4200N, and 1500 rpm and 5000N. The experiment was comprised of 6 run-to-failure different bearing vibration datasets for the purpose of model training, and 11 bearings datasets that are truncated such that the trained models can predict the RUL of corresponding testing categories.

Image 2.5. *Showing a Normal and Damaged Bearing Before and After Operations on PRONOSTIA*

These datasets have been widely used in published research as a validation and testing factor for many proposed models. The scoring function evaluation metric proposed in the prognostic competition has also been widely used in assessing newly developed models. (G. Ding, Wang, & Zhao, 2022; Y. Ding, Ding, Zhao, Cao, & Jia, 2022; C.-G. Huang et al., 2021; Luo & Zhang, 2022; Mao, Chen, Liu, & Liang, 2022; Nectoux et al., 2012; Nieves Avendano et al., 2022; Suh et al., 2022; Xiao, Liu, Zhang, Zheng, & Cheng, 2020).

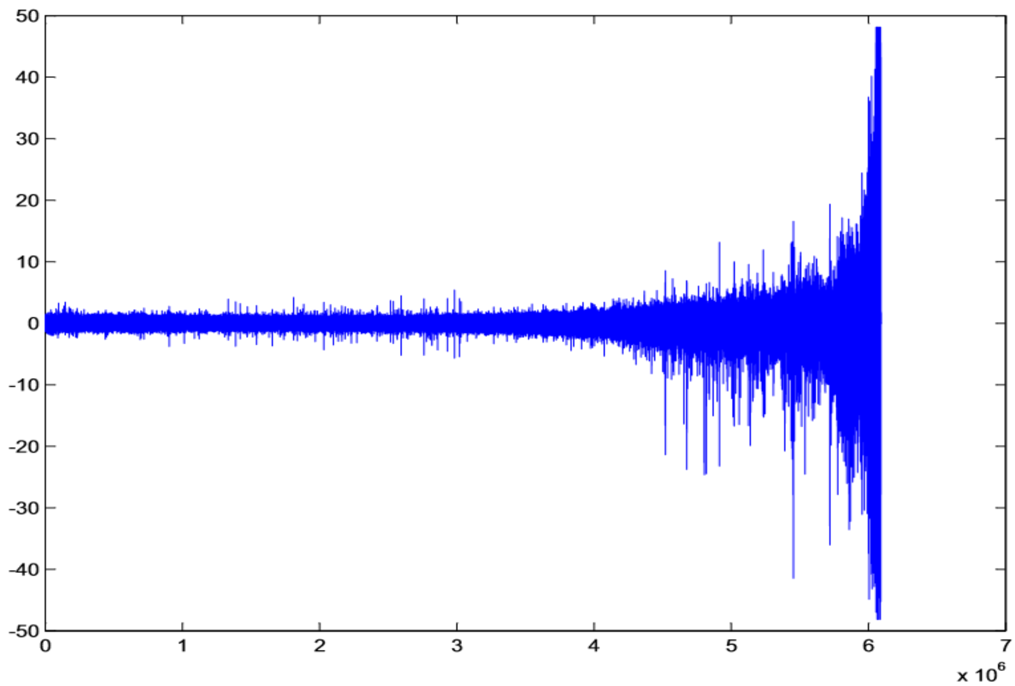


Figure 2.4. S*hows Raw Run-to-Failure Vibration Signals from One of the Bearings Tested*

Table 2.4. *Shows the Breakdown of Experimental Bearing in PRONOSTIA*

Data sets	**Operating Conditions**		
	Conditions 1	**Conditions 2**	**Conditions 3**
Learning set	Bearing1_1	Bearing2_1	Bearing3_1
	Bearing1_2	Bearing2_2	Bearing3_2
Test set	Bearing1_3	Bearing2_3	
	Bearing1_4	Bearing2_4	
	Bearing1_5	Bearing2_5	Bearing3_3
	Bearing1_6	Bearing2_6	
	Bearing1_7	Bearing2_7	

2.6.2. IMS Dataset

IMS vibration datasets are created by NSF I/UCR Center for Intelligent Systems (IMS) with assistance from Rexnord Corporation. The experiment starts with fixing Rexnord Za-2115 double row test bearings to a shaft and the rotation speed is maintained at 2000rpm by an AC motor that is driving the shaft through the rub belts, this is as demonstrated in image 2.6. During the operation of this experimental study, the vibration data excreted by the bearings were collected by PCB 353B33 High Sensitivity Quartz ICP accelerometers. The two accelerometers are fixed on every bearing, one on the x-axis and the other on the y-axis, making 90^0 between them for dataset 1 and one for every bearing for the other datasets, totally there are 4 bearings for each dataset. The experiment continued for every four sets of bearing until they surpassed the designed end of life that was set at a threshold of 100 million revolutions. By the end of the assigned threshold different damage had been inflicted on the bearing during the operation. Some articles that have used IMS datasets for testing and evaluating the models include (Ahmad, Khan, & Kim, 2017; Kumar et al., 2022).

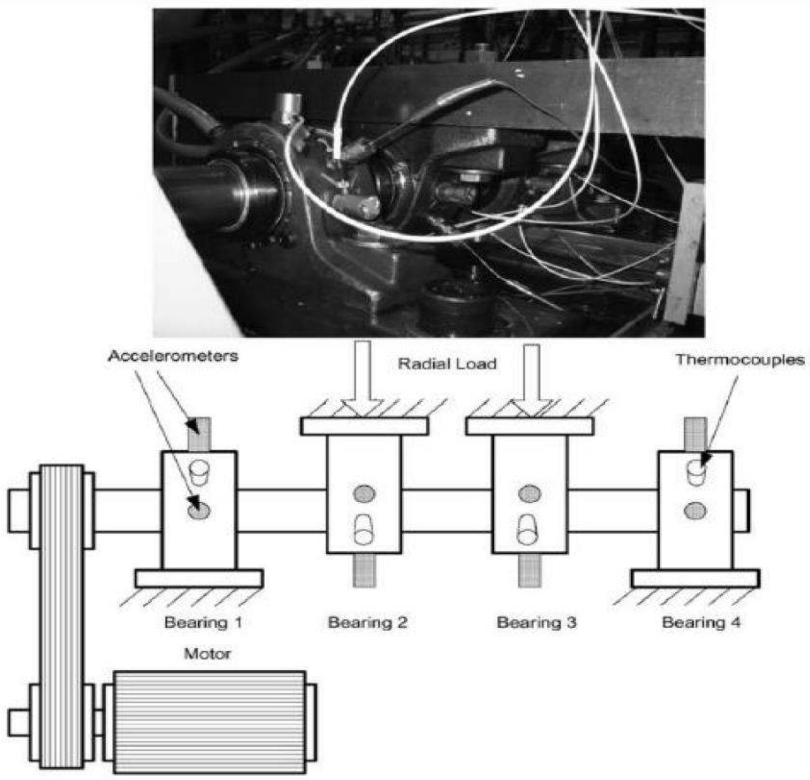


Image 2.6. *Demonstration of IMS Test Bed for Bearing Experimental Test*

2.6.3. XJTU-SY Bearing Datasets

This experimental bearing test study is conducted by the Institute of Design Science and Basic Component at Xian Jiaotong University (XJTU) with support from Shaanxi, PR China, and Changxing Sumyoung Technology Co. LTD. (SY), Zhejiang, P.R. China. The test is performed on 15 LDK UER204 bearings and all datasets comprise the run-to-failure vibration data. Details about the bearings used in this test are given in table 2.5 which contains features of these bearings and their measurements.

Table 2.5. *Demonstrated Features and Measurements for the Bearing in XJTU-SY Datasets*

Parameter	Value	Parameter	Value
Outer race diameter	39.80 mm	Inner race diameter	29.30 mm
Bearing mean diameter	34.55 mm	Ball diameter	7.92 mm
Number of balls	8	Contact angle	0°
Load rating (static)	6.65 kN	Load rating (dynamic)	12.82 kN

The experimental testbed is comprised of an Alternating Current (AC) induction motor, a motor speed controller, a digital force display, two support and thick heavy-duty roller bearings, a support shaft, a hydraulic loading system, the bearing on which tested will run, two accelerometers of type PCB 352C33, one placed horizontally and the other vertically. The experimental testbed is meant to execute the accelerated degradation tests of different bearings under different operation conditions (OC) as shown in table 2.6. The OC is comprised of two parameters, the radial force, and the rotating speed. The rotation speed of the bearing that is being assessed is maintained by the speed controller, which is placed at the AC induction motor, while the radial force is controlled by the hydraulic loading system which is deployed onto the frame of the bearing that is being tested. In this experimental study, the sampling frequency is set to 25.6kHz and produced a total of 32768 datapoints every 60 seconds. The details of the datasets are demonstrated in table 2.6.

In each experiment, the bearing is allowed to run until complete failure. The failure is signaled by exceeding the maximum amplitude that represents the normal operation of the horizontal or vertical vibration signals which is $10A_h$. This allowed the observation of the full run-failure experiment of bearings. The produced datasets have been widely used in many studies as testing and validation data because they contain less noise and demonstrate clear degradation behavior of the bearings. Some studies that have used XJTU-SY datasets include the following references (Du et al., 2022; Kumar et al., 2022; J. Li, Zi, Wang, & Yang, 2022; Mao et al., 2022; Ni, Ji, & Feng, 2022; B. Wang, Lei, Li, & Li, 2018; Y. Wang, Wu, Cheng, Wang, & Hu, 2022; J. Xu, Duan, Chen, Wang, & Fan, 2022)

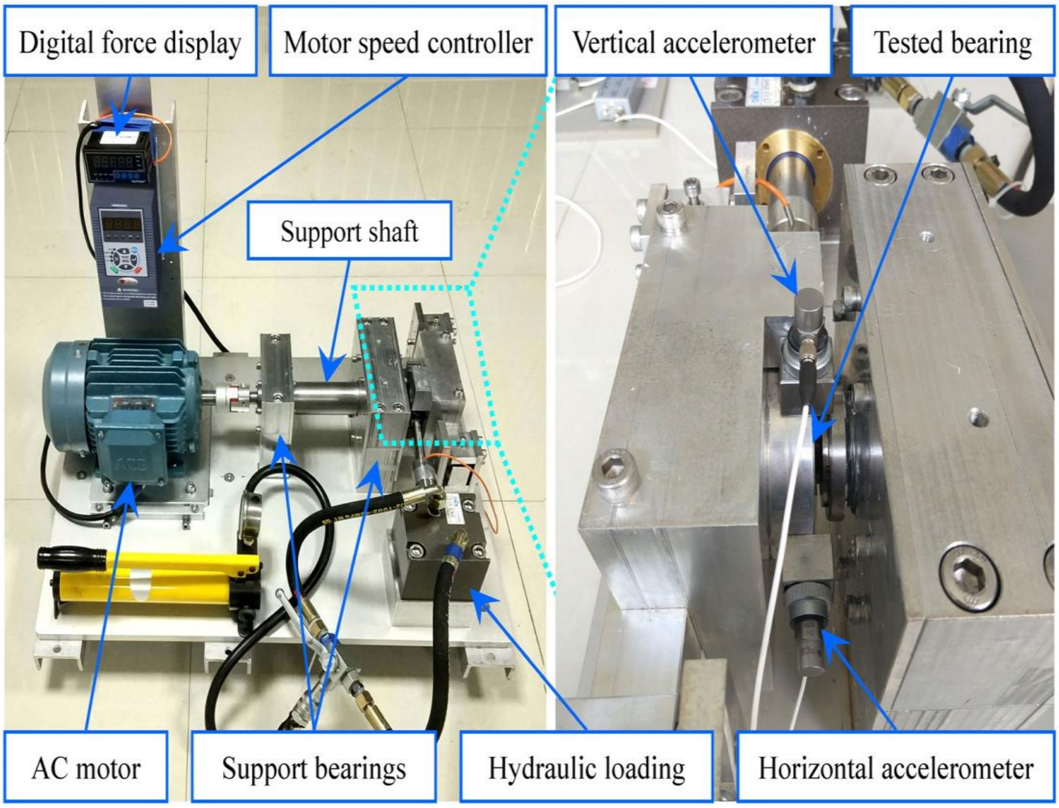


Image 2.7. *Demonstration of Test Best Setup for XJTU-SY Experimental Study*

The test bearings are allowed to run till they are damaged. These damages were experienced on the inner race and the outer race. Some bearings reached a stage of complete rupture as demonstrated in image 2.7. It is visible from image 2.7 that by the end of the operation the failures are a result of different damages on different specimen of the bearings. The raw data was collected in the same way as the data from PRONOSTIA datasets was. Generally, data collected demonstrate a monotonicity behavior as the health of the bearing deteriorates. Figure 5.1: of Raw and absolute vibration data in section 5 demonstrates the monotonicity behavior for a selected bearing.

Table 2.6. *Breakdown of XJTU-SY Bearing Datasets*

Operating condition	Bearing dataset	Number of files	Bearing lifetime	Fault element
Condition 1 (35 Hz/12 kN)	Bearing 1_1	123	2 h 3 min	Outer race
	Bearing 1_2	161	2 h 41 min	Outer race
	Bearing 1_3	158	2 h 38 min	Outer race
	Bearing 1_4	122	2 h 2 min	Cage
	Bearing 1_5	52	52 min	Inner race and outer race
Condition 2 (37.5 Hz/11 kN)	Bearing 2_1	491	8 h 11 min	Inner race
	Bearing 2_2	161	2 h 41 min	Outer race
	Bearing 2_3	533	8 h 53 min	Cage
	Bearing 2_4	42	42 min	Outer race
	Bearing 2_5	339	5 h 39 min	Outer race
Condition 3 (40 Hz/10 kN)	Bearing 3_1	2538	42 h 18 min	Outer race
	Bearing 3_2	2496	41 h 36 min	Inner race, ball, cage and outer race
	Bearing 3_3	371	6 h 11 min	Inner race
	Bearing 3_4	1515	25 h 15 min	Inner race
	Bearing 3_5	114	1 h 54 min	Outer race

2.7. PHM Models Performance Measures and Evaluation Metrics

There are two ways of evaluating the performance of models that are meant for PHM. Models that are meant for classification of faults, or machine health degradation severity identification use accuracy metrics. The classification metrics are meant to tell how many times the model has predicted the right class rightly and how many times it has predicted the right class wrongly. In the case of prognostics, prediction precision is measured to ascertain models' performance. The prediction precision of the results from different PHM models for both TSP and RUL estimation is evaluated by performing a comparison study with already published works applying different prognostic metrics (Zhu et al., 2022). Each prognostic metric has special strength over the others.

(A) (B) (C) (D)

Image 2.8. *Demonstrated Diverse Types of Damages Inflicted on the Bearings During the Experimental Study (A) Inner Race Damage (B) Cage Damage (C) Outer Race Damage (D) Outer Race Fracture*

2.7.1. Accuracy Metrics

Considering bearings as a case study, bearing faults are classified based on the occurrence of damage to various parts of bearing. These faults are classified into four classes, these are inner or out race damage, balls, and cage damage. In the case of severity classification, most literature classified severity into low, middle, high, and extreme deterioration severity conditions (Yan, Wang, Wang, Chang, & Muhammad, 2020). Accuracy metrics evaluate classification models based on instances rightly or wrongly classified. Every instance of a predicted label with its actual label is evaluated independently and thereafter the mean of all the instances is computed for comparison purposes with existing models(R. Lin et al., 2022; T. Lin et al., 2022).

One of the examples of Accuracy Metrics is the Classification Accuracy (CA). CA is described as the percentage of rightly classified labels as computed by equation 2.1, Given actual labels (y_i) and predicted labels ($\hat{y}_i$)

$$CA = \frac{100}{M}\sum_{i=1}^{M} Accurate(y_i, \hat{y}_i) \tag{2.1}$$

Where, $Accurate(y_i, \hat{y}_i) = 1$ if $y_i = \hat{y}_i$ and if $y_i \neq \hat{y}_i$ then, $Accurate(y_i, \hat{y}_i) = 0$,

M= Total number of instances to be evaluated.

CA without discriminating penalizes every model if the number of labels is large, this makes it not a suitable metric for problems with a large number of labels. The alternatives to CA are label-based metrics. As it is for CA, label-based metrics also evaluate instances independently thereafter the mean is computed for the purpose of evaluation. These metrics with the help of the confusion matrix include Precision (P), Recall (R), and the F-Score (F), which are calculated by equations 2.2, 2.3 and 2.4, respectively. It should be noted that, the larger the F-score the more accurate the results produced by the model (Fink et al., 2020; Khan & Yairi, 2018; Zhu et al., 2022).

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{2.2}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{2.3}$$

$$F_i = \left(\frac{P_i^{-1} + R_i^{-1}}{2}\right)^{-1} \tag{2.4}$$

Given; TP = True Positive, FP = False Positive, TN = True Negative and FN = False Negative.

where;

$$TP_i = \sum_{i=1}^{M} Comp(\,y_i = l_i \; and \; \hat{y}_i = l_i)$$

$$FP_i = \sum_{i=1}^{M} Comp(\,y_i \neq l_i \; and \; \hat{y}_i = l_i)$$

$$TN_i = \sum_{i=1}^{M} Comp(\,y_i \neq l_i \; and \; \hat{y}_i \neq l_i)$$

$$FN_i = \sum_{i=1}^{M} Comp(\, y_i = l_i \; and \; \hat{y}_i \neq l_i)$$

Given

$Comp(Paramters) = 1$ if both parameters share similar assignment sign and $Comp(Paramters) = 0$ if both parameters share different assignment sign

2.7.2. Prediction Precision Metrics

Prediction Precision Metrics (PPM) just like Accuracy Metrics are the methods used to measure how accurate the model is compared with other models in existence. PPM is applied to models that perform prognostics i.e., predicting the RUL of machines. One of the PPM methods used in PHM is the Prediction Accuracy (PA) method. PA assesses the percentage of correctly estimated RUL from instances available. PA is confirmed when Error (e_i) is between -13 and 10 i.e., $e_i \in [-13, 10]$. PA is computed by equation 2.5. A particular prediction is found to be False Positive (FP) if $e_i < -13$ and also found to be False Negative if $e_i > 10$ (Wennian Yu, Kim, & Mechefske, 2021).

PA considers all errors incurred across the process of prediction equal. It does not have the capacity to penalize a model that has a specific behavior at a specific time. Because of this, PA might not be a good metric for prognostics. For this purpose, therefore, researchers have proposed other better metrics. Among the most applied evaluation metrics in the literature available include the Score Function (SF), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Health Degree, Mean Absolute Percentage Error (MAPE), Expected value range, and the lambda performance index between Actual RUL and Predicted RUL (R. Lin et al., 2022; Luo & Zhang, 2022; Miao et al., 2022; Song et al., 2022; Y. Wang et al., 2022; Wennian Yu et al., 2021).

$$PA = \frac{100}{M} \sum_{i=1}^{M} Accurate(e_i) \qquad (2.5)$$

Where $Accurate(e_i) = 1$ if $-13 < e_i < 10$ and otherwise $Accurate(e_i) = 0$

During the IEEE PHM 2012, Prognostics Challenge organized by IEEE Reliability Society and FEMTO-ST Institute, an experimental platform for bearings accelerated degradation tests called PRONOSTIA was conducted. The score used to evaluate competitors in this experiment was the SF. Although different studies have defined SF differently from the one proposed in the competition, SF has gained popularity due to its capacity to penalize overprediction more and penalize underprediction less, this is because of safety purposes.

The difference between the SF used in the PRONOSTIA experiment (Nectoux et al., 2012) and the SF used in many other references (Elsheikh et al., 2019) is the computation of Error (e_i) and the final score. The former computes e_i as percentage error (Equation 2.6) and the final score as the mean of all scores (Equation 2.7), while the latter computes error as only normal error (equation 2.8) and the score as the sum of all scores (equation 2.9). SF penalizes each PRUL with a negative score when it is greater than ARUL i.e., $e_i \leq 0$. The best performance should be PRUL that is equal to or not much less than ARUL, i.e., $e_i > 0$, this is rewarded with a positive score. With this, the accuracy of the model can be evaluated and determined better.

$$e_i = 100 \times \frac{ARUL_i - PRUL_i}{ARUL_i} \tag{2.6}$$

$$SF = \frac{1}{M}\sum_{i=0}^{M} S_i \tag{2.7}$$

$$e_i = ARUL_i - PRUL_i \tag{2.8}$$

$$SF = \sum_{i=0}^{M} S_i \tag{2.9}$$

where;

$$S_i = \begin{cases} e^{-\ln(0.5).\left(\frac{e_i}{5}\right)}, & e_i \leq 0 \\ e^{+\ln(0.5).\left(\frac{e_i}{20}\right)}, & e_i > 0 \end{cases}$$

ARUL = Actual RUL and PRUL = Predicted RUL

Another important metric is the Cumulative Relative Accuracy (CRA) which penalizes models with errors based on the time the prediction is made. It penalizes Errors incurred during the early predictions differently from those errors incurred during late predictions. This is computed by equation 2.10. CRA puts the accuracy close to 1, the closer to 1 the more accurate the model is, e_i is the Relative Accuracy (Ahmad et al., 2017; Du et al., 2022).

$$CRA = \sum_{i=1}^{i} w_i e_i \tag{2.10}$$

where;

$$w_i = \frac{i}{\sum_{i=1}^{i} i}$$

and

$$e_i = 1 - (\frac{|ARUL_i - PRUL_i|}{ARUL_i})$$

Another metric widely used in PHM model evaluation is the RMSE. The smaller value indicates better estimation than a bigger value, calculated by equation 2.11 given that $\varphi^{<t_m>} = \Delta(RUL^T, RUL^{<t_m>})$ where $RUL^T \; and \; RUL^{<t_m>}$ are ARUL and PRUL respectively. The lower the RMSE and the higher the SF the better the model.

$$RMSE = \sqrt{\frac{1}{m}\sum_{m=0}^{m} (\varphi^{<t_m>})^2} \tag{2.11}$$

Unlike RMSE which computes the expected error between ARUL and PRU, like SF, MAPE (Equation 2.12) considers PRUL that is less than ARUL than the one that is greater, which considers the fact that more accuracy should be attained towards the EOL of the machine, it also gives less score to overestimation than underestimation (Nieves Avendano et al., 2022).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{(ARUL_i + c) - (PRUL_i + c)}{ARUL_i + c}\right| \tag{2.12}$$

MAPE is used extensively in forecasting problems, and it has been dependable in the evaluation results, where the only setback is that it suffers from an error of division by zero. To avoid this, there is a small *c* added to each factor as seen in equation 2.12.

The other commonly used metric is Convergence (CVG) which measures how fast the PRUL converges to ARUL. This is a measure of Euclidean distance between TSP and the centroid of the area under the PRUL and the ARUL curve, which is computed by equation 2.13, the lower the convergence value the more the accuracy(Du et al., 2022).

$$CVG = \sqrt{(C_x - TSP)^2 + {C_y}^2} \tag{2.13}$$

(C_x, C_y) defined as centroid of area under PRUL and ARUL

2.8. General Review, Discussion and Conclusion

Considering the literature reviewed in this section, much has been discussed about PHM and its current trends. The section elaborated more on different solutions available for PHM-based problems and discusses challenges within the processes of developing solutions and proposed possible countermeasures to them. The discussion on published solutions goes deeper into some publications, showing the required details of how these solutions were developed and the genre of results they produced. The point of discussion about AI-based solutions in general and in specific deep learning-based solutions was discussed more than the other solutions available. The limitations in developing deep learning-based solutions and how studies are making efforts to overcome these limitations and challenges were elaborated. Data being a crucial component of the training and the testing of models, the sources of data used to validate and test PHM solutions were also articulated. Lastly, evaluation metrics that are meant to assess the viability of models were also clarified in which their strengths and weaknesses were mentioned.

Based on the literature reviewed in this section, this study aims at proposing a deep sequence-to-sequence model with residual blocks made up of LSTM neurons. This model is meant to overcome the gradient and complexity hindrances mentioned in this section as one of the challenges in deep learning models. Residual blocks were first discovered in a network which is commonly known as ResNets that is applied in computer vision (Hanif & Bilal, 2020; Rao et al., 2017). The proposed model was trained and validated with vibration data from rolling elements in machines (bearings). Bearings were selected because they highly impact the operations of machines and their lifespan. Hence, they greatly represent the status of the entire machine (Benkedjouh et al., 2013; C. Li et al., 2017).

In addition to that, in the most available literature, the Time to Start Prediction (TSP) is subjectively picked based on prior knowledge of bearings contrary to the objective of the autonomation of engineering systems. In this regard, this study also proposes an objective and scientific technique for determining TSP.

3. RESEARCH METHODOLOGY

3.1. Overview

The research process and steps that were undertaken to reach the solutions to the discovered problems in DLMs that are meant for PHM are detailed and elaborated on in in this section. The first subsection 3.1 gives a backbone of this section while subsection 3.2 illustrates the research framework that explains the breakdown of the entire research process and phases undertaken to accomplish this research. It breaks down the process into three phases and each phase contains activities and methods employed to accomplish a specific goal. In addition, each phase receives an input, and the input is processed to produce the required output that subsequently becomes the input for the following phase. Subsection 3.3 elaborates on the proposed model in depth which starts with data collection, preprocessing, and feeding this data into the model for the purpose of training. The data is split further into training and validation to control the model from overfitting. Then the model is assessed with vibration data compiled based on different conditions. Lastly, subsection 3.4 is a conclusion that briefly summarizes the section.

3.2. Research Framework

The research framework, as shown in figure 3.1, breaks down the research process into three phases, and these are the initial, discovery, and optimization phases. Each phase is an activity of inputs, processing, and output. As it is indicated in figure 3.1, the major inputs of the first phase which translates to the main inputs of this research, are the latest articles published regarding PHM. While major outputs as it is demonstrated in the last phase of the research framework, are results from diagnostics and prognostics of engineering systems.

3.2.1. Initial Phase

The initial phase is the inception of the research which is dubbed method 1 (M^1) as shown in figure 3.1. To understand this area of research, it was also necessary to implement some of the published articles and compare published results with the results that were obtained in local experimental tests. Activities in this phase were dominated by reading many articles and writing subsequent codes in different programming languages, mainly MATLAB and Python. The output of this phase as expected was the discovery of the research gap which this research is meant to investigate and innovate possible solutions to it.

3.2.2. The Discovery Phase

The Discovery Phase takes the input from the initial phase which is the problem discovered. This phase contained activities related to discovering the optimum solution to the problem. In addition to that, suitable datasets for this problem became one of the inputs as well. The scope is training and testing different DL models to find the one that would be suitable for solving the problem. However, much emphasis was put on timeseries models since vibration signals produced by electromechanical machines are in time-series form. The research started with normal DL models as presented in different articles (Eren, Ince, & Kiranyaz, 2019; Mao, Tian, Fan, Liang, & Safian, 2020; Jiaxing Wang, Wang, Wang, Li, & Song, 2021), then it was zeroed down to Long Short Term Memory(LSTM) and its variant known as Gated Recurrent Unit (GRU) which have been paramount in time-series analysis (Khorram et al., 2021). The final discovered solution that produced reliable results was a sequence-to-sequence model using LSTM, this will be discussed further in section 4. In this same phase, data is collected and thereafter preprocessed to fit in the proposed model.

3.2.3. Optimization Phase

This phase's input is the model that has been tested and proved to give the best results as far as finding the solution to the problem discovered is concerned. In this phase, the model is further trained, validated, and tested with data that the model has not seen. The output here is a trained model that does three main activities, health condition monitoring, diagnostics, and prognostics. With the help of other techniques discovered, the model aids in monitoring the health of the machine whenever it receives new data, detects fault occurrence, and estimates Remaining Useful Life (RUL).

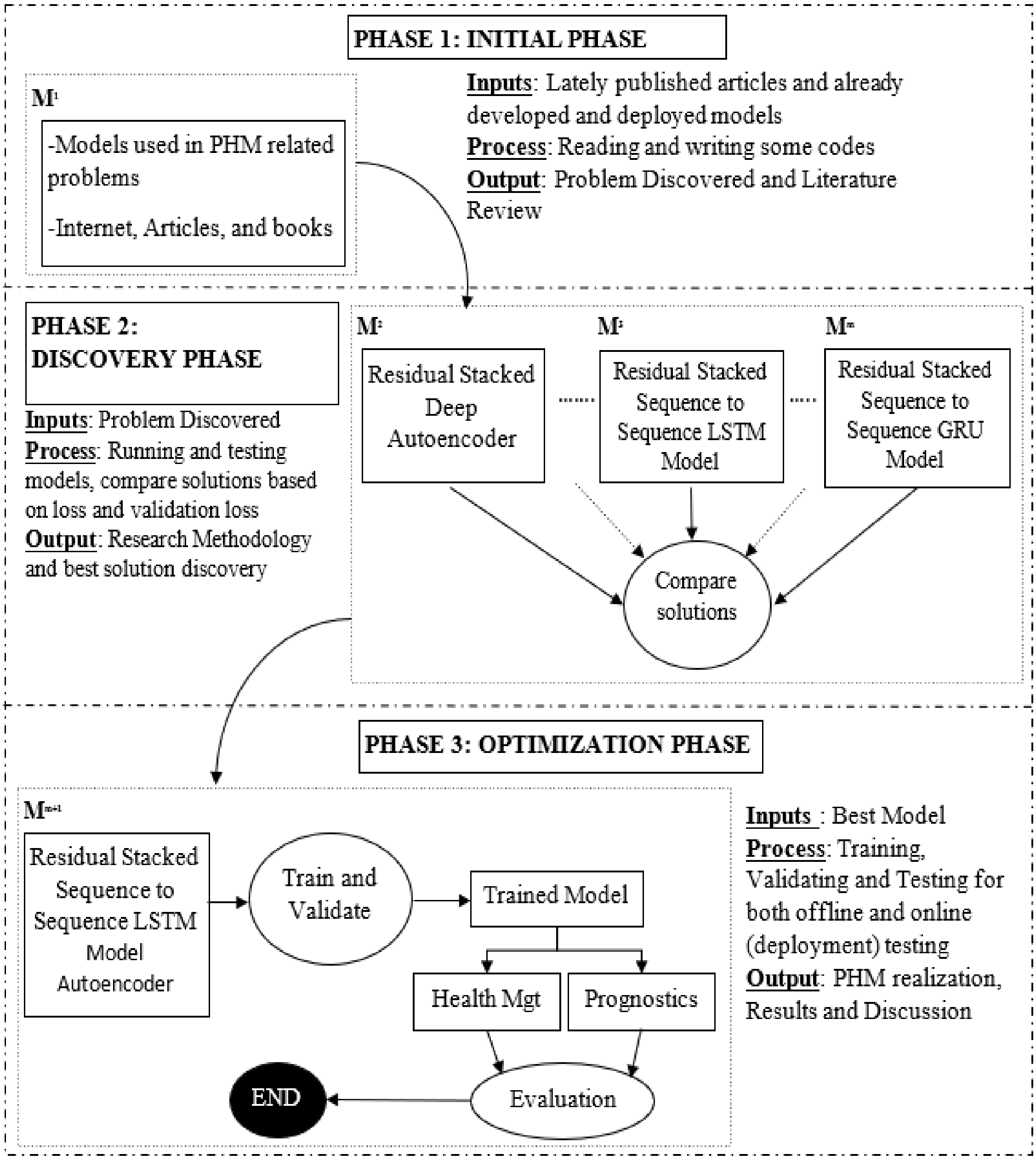


Figure 3.1. *Research Framework*

3.3. Proposed Model

The PHM models were discussed in subsection 2.2. Data-Driven Model (DDM) was found to be more accurate and easier to use and implement as the process is detailed in figure 2.1. This study proposes a novel data-driven deep sequence-to-sequence model that mimics and better implements techniques used in widely published articles. Figure 3.2 shows the proposed model for this study. The proposed model in the study was trained and tested through three main modes, training & validation, testing, and online mode.

These modes are symbiotically related but use data input from separate rolling bearings ran under different operating conditions.

3.3.1. Training and Validation

Training began with the collection of data that was used in this phase. There are different public sources for vibration signals run-to-fail data for mechanical rolling elements (bearing) produced in specified time interval as discussed in section two in subsection 2.3 and detailed in table 2.3. As it has been discussed, the sources of these datasets include public and private repositories.

All these datasets have been used in many studies and published; however some datasets were found to have many outliers that created challenges in training and therefore it was difficult to coming up with suitable models (Mao, Feng, Liu, Zhang, & Liang, 2021; Porotsky & Bluvband, 2012; X. Wang et al., 2020; Zonta et al., 2020). This study chose validating the model with data from XJTU-SY datasets because it is the latest with less outliers as it is discussed in further sections.

The data that was used in this case was first preprocessed into two-dimensional data $\{(T, X) \in R^{n \times m}\}$, where, *X* is the amplitude vibration representation, while *T* are the indices representing the time unit equating to the lifetime of the machine. n=2 depicting the 2D matrix of t_m and x_m that make up the two variables, respectively. The data is further preprocessed into independent (x_m) and dependent (y_m) variables through equations 3.1 and 3.2, respectively. This means that the lookback or timestep chosen for this case is equivalent to the data collected in t_m. This process aids training the model such that the model is able to estimate a batch expected after the input batch and hence predict the future state of the machine. The shape of each batch must be equal to the timestep or lookback value that is selected for the problem. The input data is then reshaped into a 3-D shape (number of samples, timestep, number of features) that is compatible to RNN (Zhu et al., 2022).

$$\{t_m, x_m\}_{m=0}^{m-1} = \{X \in R^m | x^m = (x_1, \dots x_n)^0, (x_{n+1}, \dots, x_i)^2, \dots, \left(x_{k+1}, \dots, x_j\right)^{m-1}\} \tag{3.1}$$

$$\{t_m, y_m\}_{m=1}^{m} = \{X \in R^m | y^m = (x_{n+1}, \dots x_i)^1, (x_{i+1}, \dots, x_a)^3, \dots, (x_{j+1}, \dots, x_v)^m\} \tag{3.2}$$

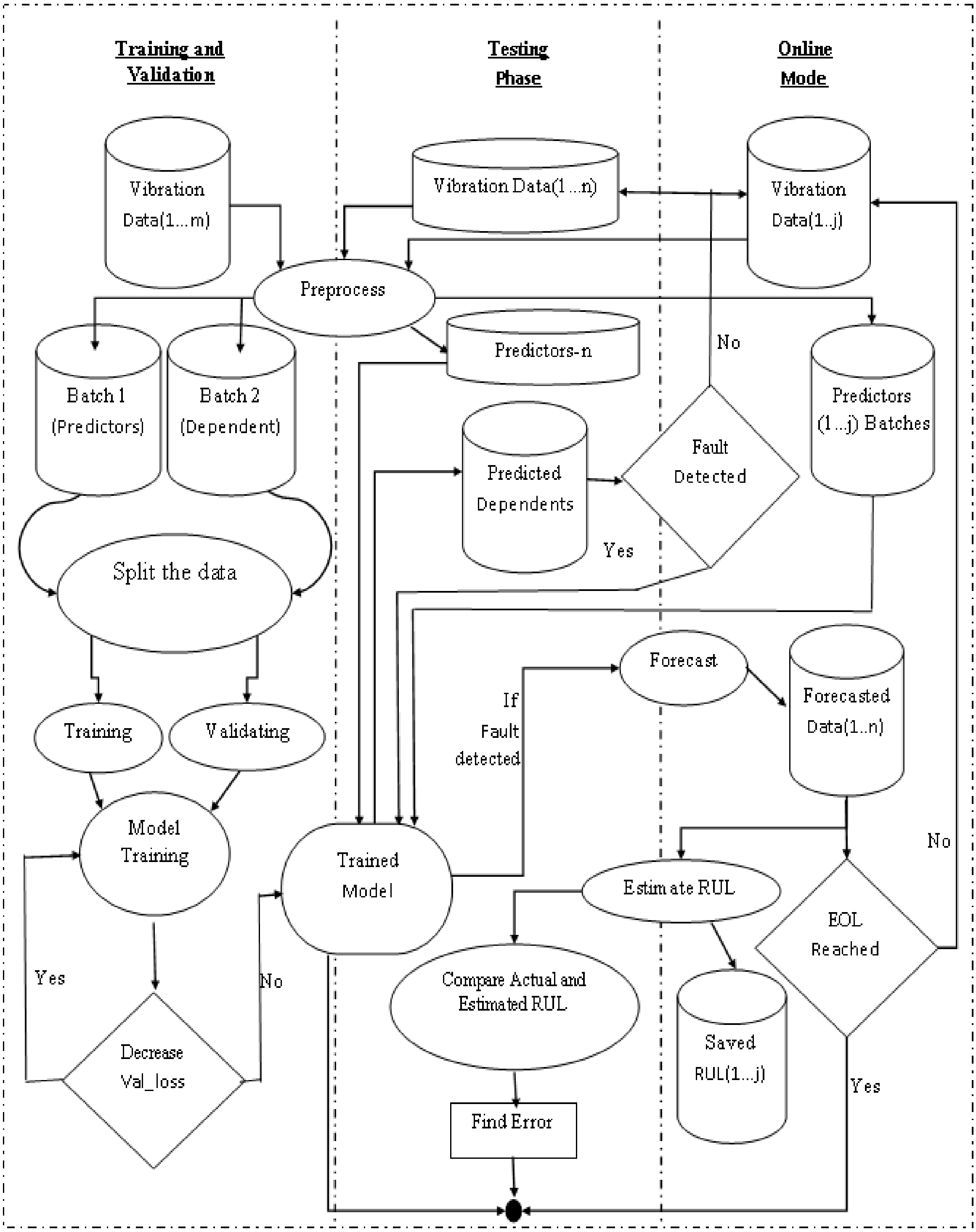


Figure 3.2. *The Proposed Model Structure*

Sample data that is extracted from different bearings are further broken down into training, validation, testing, and online test samples. The purpose of validation samples is to check the occurrence of overfitting while the model is being trained. These samples are then used as inputs into the model as per the required task for the purpose of validating the model. During model training, if the model produces decreasing validation loss, the model is learning and hence this process will continue until the validation loss starts to increase. When the validation loss starts to increase, the model is assumed to have sufficiently trained and hence training is stopped. This model is then promoted to be trained and ready for prediction of the dependents. However, if the model does otherwise, the process of developing this model is repeated by parameters and hyperparameter tuning. This is performed until a decrease in validation data is attained. The model is trained such that when it receives the first batch of data $\{t_m, x_m\}_{m=0}^{m=0}$ it predicts $\{t_m^*, x_m^*\}_{m=0}^{m=0}$ which is expected to be equal or nearly equal to the future state $\{t_m, x_m\}_{m=1}^{m=1}$ based on the same timestep *m* as per equations 3.3 and 3.4. When the trained model is attained, the training phase ends, and the testing phase begins.

$$f_l(T, X), f_l(T^*, X^*) = (Ts, Xs), (Ts^*, Xs^*) \tag{3.3}$$

$$\{t_m^*, x_m^*\}_{m=0}^{m} \cong \{t_m, x_m\}_{m=1}^{m+=1} \tag{3.4}$$

3.3.2. Testing Mode

Data for testing the trained model is selected and picked from several bearing data from different sources as mentioned in subsection 3.3.1. and detailed in section two. In this testing phase, data is serialized as it was in the previous phase of training and validation. The only difference here is that data is broken down and left into data produced in a particular unit of time, i.e., 1 minute as it is for this study. The model is trained to project and forecast data that will possibly be produced in the next same unit of time.

However, before the model starts predicting the future state of the machine, TSP must be detected and EOL estimated. TSP was detected using Pearson's Correlation Coefficient (PCC) computed by equation 3.5, while EOL was set to 1 since data is first normalized using minimax normalization given in Equation 3.7. At the moment TSP is detected, the trained model is fed with the last data batch to start predicting. Prediction goes on till the threshold (EOL) is attained. Once EOL is detected in any predicted batch, prediction is stopped, and thereafter Remaining Useful life (RUL) is estimated from the projected data with equation 3.6 by computing the change between TSP and EOL.

$$\psi_n = \left|\left(f_{pcc}\{t_m, x_m\}_{m=0}^{m-1}, \{t_m, y_m\}_{m=1}^{m}\right)\right|$$

where; (3.5)

$$f_{pcc} = \frac{\sum[(x_m - \mu_x)(y_m - \mu_y)]}{\sqrt{\sum(x_m - \mu_x)^2 \sum(y_m - \mu_y)^2}}$$

μ_x, μ_y are sample means of x_m and y_m respectively.

$$RUL^{<t_m>} = \Delta(TSP^{<t_m>}, EOL^{<t_m>}) \tag{3.6}$$

Based on these facts, the Predicted RUL (PRUL) is compared with the Actual RUL (ARUL) using different evaluation metrics. In published articles and manuals that are distributed along with publicly available datasets, it is indicated that the computation of errors is based on the ARUL and the PRUL. These errors are evaluation factors that determine how accurate the model is in comparison to the published and existing ones. In many cases, proposed models have successfully produced results that are close to the ARUL hence giving less error and therefore dependable to use in the real world. The proposed model in the testing mode goes through the aforementioned process before the activities are shifted to the online mode.

$$minmax(x_i) = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{3.7}$$

$$minmax \in [0,1]$$

3.3.3. Online Mode

In this mode, the real-world situation is simulated by feeding the trained model with testing data batch-by-batch as the outputs are studied and examined. It is simulated in such a way that the trained model is taking in data as the data is produced from the sensors connected to the engineering system in operation. As it was in the previous phase of testing, as the model takes in data produced by the bearing, it forecasts and projects what would possibly be the next sample of data. This continues till a fault is detected.

The moment a fault is detected, the model will forecast/project all the data until a threshold that represents EOL is reached. This forecasted data represents the future state of the engineering system and aids the computation of RUL. When this point is reached, the projection will be stopped and subsequently, RUL will be estimated and kept for further reference. In this situation, the machine has not reached EOL, but the model has projected data till EOL. When the machine produces the next batch of data, the model again forecasts till EOL and computes another RUL that is expected to be less than the first RUL. This process continues iteratively till RUL is zero. During the process of RUL estimation, warnings are sent to the engineering systems operator about the health situation of the machine.

3.4. Conclusion

This section elaborates on the research methodology which included the step-by-step activities performed to achieve the objectives. The section highlighted the research framework that was broken down into a number of processes. Each process accepts inputs and produces an output that will correspond to the input on the next processes. The proposed research process that is inculcated in the research methodology was put under 3 wide phases, and these are the Initial, Discovery, and Optimization Phases.

This section further described the architecture and the structure of the proposed model. The model is made up of three major activities, which were named modes and include training and validation, offline testing, and online testing modes. It explains how data was collected, mentioning diverse sources of such datasets. It takes the reader through the process of preprocessing data that can fit into the model. It further explains how the model is trained and upon successful training, it goes further to show how it is assessed both online and offline and then deployed. It also shows the differences between offline testing and online mode, and how they are useful in solving the problem this study discovered.

4. DEEP RESIDUAL SEQUENCE TO SEQUENCE MODEL (AUTOENCODER)

4.1. Overview

This section explains the proposed model in detail using the building blocks mentioned and elaborated in section 2. In this section, much of the logic and mathematics behind the proposed model are discussed. This section starts with subsection 4.1 where the breakdown of the body and the structure of this section is presented. Subsection 4.2 in which generic RNNs are explained gives the genesis of RNN and its variants and how they have been successfully used in different sequence-related problems. Subsection 4.3 presents in detail the structure of an LSTM because LSTMs are the building blocks of the proposed model in this study. Subsection 4.4 demonstrates sequence-to-sequence models whose building blocks are LSTM units. Then the proposed residual sequence-to-sequence model is elaborated in subsection 4.5, explaining further the formation of an Autoencoder with Encoders and Decoders that form a Reseq2seq model. The section then goes further into the details of how the model actuates the functions of health monitoring, diagnostics, and prognostics in subsections 4.6 and 4.7. Model evaluations being one of the most important activities in models that are being proposed, subsection 4.8 expounds on how the proposed model in this study was evaluated. Subsection 4.9 gives a summary of the whole section.

4.2. Recurrent Neural Networks

Recurrent Neural Networks (RNN) whose prominent variants are Gated Recurrent Models including the most prominent ones LSTM and GRU as discussed in section 2, have played a major role in improving the regular RNN and have been successfully implemented in sequence and time-series analysis including and not limited to Natural Language Processing, voice recognition and time-series analysis (W.-C. Huang et al., 2021; Qi et al., 2020; Tang et al., 2020; Z. Xiang, Yan, & Demir, 2020; Zheng et al., 2021). These models have also been applied in machine translation with commendable results in return (Guo, Xu, & Chen, 2020). They have also been successfully used as denoising agents in multilingual machine translation models. The process of denoising was achieved by deducing latent variables that lay between the inputs and the outputs of the model (Liu et al., 2020).

Even though RNN performed significantly well in sequence models, deep regular feedforward RNN did not generally perform well due to the nature of sequence data that have long-term also known as long-range dependencies that results in making the tracking of learned insights from long series of data mandatory throughout the depth of the data. In such models, the persisting setback of diminishing and exploding gradients became a hindrance to the optimization of this model. (Bengio, Simard, & Frasconi, 1994; Pascanu, Mikolov, & Bengio, 2013). In the search for solutions, researchers proposed several ways of solving this trouble. RNN variants were proposed and among the prominent ones are the Echo State Network (ESN) (Gallicchio & Micheli, 2017) and the gated recurrent networks (GRU) including LSTMs and GRUs. Gated Recurrent variants have uninterrupted gradient flow and can track artifacts through deep and many timesteps, all have been applied to PHM problems (Long, Zhang, & Li, 2019; Rigamonti et al., 2016; Wu et al., 2018; S. Zhang et al., 2019).

4.3. Long Short-Term Memory

Long Short-Term Memory (LSTM) units were applied as building blocks in this study. They consist of three main gates: the Forget Gate (FG), the Update Gate (UG), and the Output Gate (OG). As shown in figure 4.1, all these gates receive and manipulate input data with the equations designed for each process. The FG (Γ_{fg}) gets rid of irrelevant and stores relevant behavior in data using equation 4.1. This data is forwarded to the next process. The UG (Γ_{up}) discriminatingly updates the cell state using equation 4.2. The OG (Γ_{ot}) returns the sieved cell state for further processing using equation 4.3. All three gates apply the sigmoid function δ to accomplish their assignments.

$$\Gamma_{\boldsymbol{fg}} = \delta(W_{fg}[a^{<t-1>}, x^{<t>}] + b_{fg}) \tag{4.1}$$

$$\Gamma_{\boldsymbol{up}} = \delta(W_{up}[a^{<t-1>}, x^{<t>}] + b_{up}) \tag{4.2}$$

Unlike other variants, LSTM was developed to accept three inputs, the current sequence data inputs, the cell state representative, and the hidden input, which is the output of the previous neuron, refer to figure 4.1. These inputs are denoted as $\boldsymbol{x}^{<t>}$, $\boldsymbol{c}^{<t-1>}$ *and* $\boldsymbol{a}^{<t-1>}$, respectively. In addition, $\check{c}^{<t>}$ represents a variable that is meant to control the update gate before feeding into the current state of the neuron, this variable is computed by equation 4.4.

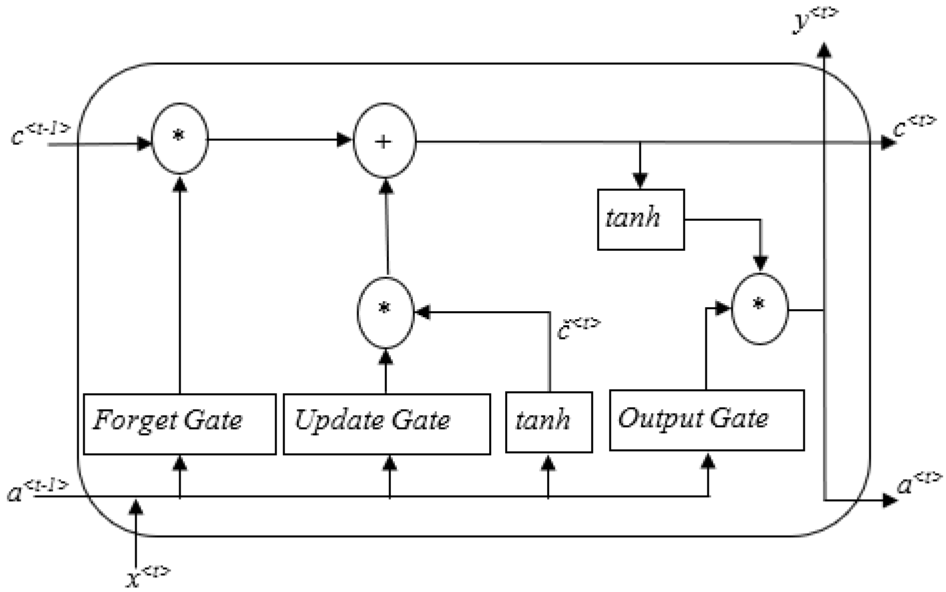


Figure 4.1. *An Ordinary LSTM Neuron*

$$\Gamma_{\boldsymbol{ot}} = \delta(W_{ot}[a^{<t-1>}, x^{<t>}] + b_{ot}) \tag{4.3}$$

$$\check{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \tag{4.4}$$

The current state of the unit ($c^{<t-1>}$) that comes from the unit before the current unit is updated by the update and forget gates which are multiplied elementwise by ($\check{c}^{<t>}$) variable and the current state ($c^{<t-1>}$), respectively, which generates the state of the next unit ($c^{<t>}$) in the sequence of units, this is computation is carried out by equation 4.5. Equation 4.6 computes the overall output of the current unit ($a^{<t>}$) using elementwise multiplication of the output gate and the hyperbolic tangent of the state of the next unit in the sequence ($c^{<t>}$) (Hochreiter & Schmidhuber, 1997; B. Zhang, Zhang, & Li, 2019).

$$c^{<t>} = \Gamma_{up} * \check{c}^{<t>} + \Gamma_{fg} * c^{<t-1>} \tag{4.5}$$

$$a^{<t>} = \Gamma_{ot} * \tanh\,(c^{<t>}) \tag{4.6}$$

4.4. LSTM Sequence to Sequence Models

Unlike other models that use LSTM as building blocks, in the Sequence-to-Sequence model, inputs (X_n) are accepted as a sequence of data during feedforward (ff), and the outputs produced are also a sequence of data. The architecture of the Sequence-to-Sequence model is demonstrated in figure 4.2. This model is made up of the Encoder and Decoder models that are independent and yet in a symbiotic relation. The Encoder is the gate of the input data whose responsibility is to manipulate the input to produce feature representation with the use of learned latent variables (Zhu et al., 2022). The generated features are then loaded into a Decoder and thereafter they are manipulated into the preferred predicted sequence of output ($\hat{y}_n$) in accordance with the latent variables. It finds the independent sequence data input (Y_n) of the shape of the $\hat{y}_n$, and a comparison is made by loss value computation using the preferred loss function of the two vectors. The loss value obtained is investigated and by applying a preferred optimization function, the predicted data is fed back into the decoder for backpropagation all the way to the encoder again. This process continues iteratively until the least mean loss is attained and thereby discovering the weights that map X_n to Y_n (Guo et al., 2020; W.-C. Huang et al., 2021).

The Sequence-to-Sequence plan of LSTMs has been mostly applied in problems related to Natural Language Processing like machine translation and commendable results are yielded. It has also been used in PHM, however, with shallow architecture just to avoid gradient and complexity concerns (Chung et al., 2014; Guo et al., 2020; W.-C. Huang et al., 2021; Tang et al., 2020; D. Wang et al., 2020; Z. Xiang et al., 2020)

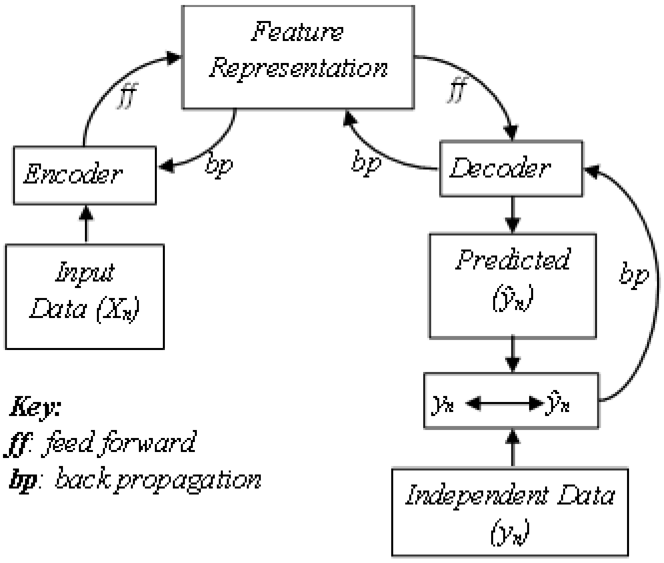


Figure 4.2. *Sequence-to-Sequence Model Representation.*

4.5. Proposed Residual Sequence to Sequence Model (Reseq2seq)

It is a fact that has been proved in different studies that models with more depth facilitate the extraction of more insights and hidden behavior in data in case overfitting and gradient hitches are avoided. The depth of the model is significant as it has been proven that the deeper the architecture the better the model the better the results and therefore, deeper networks are preferred over shallow networks. However, deeper networks come with impediments of overfitting and gradient mismatches including vanishing or exploding gradients. Researchers labored to find solutions to these problems such that the benefits of having deeper networks are achieved. One of the solutions out there was to use residual blocks. ResNets that are applied in computer vision use CNN building units and implemented residual blocks to have deeper networks. The technique of using residual blocks successfully managed to overcome the networks depth hindrances. Because of this technique, ResNets benefitted from the advantages of having deeper networks (K. He, Zhang, Ren, & Sun, 2016). To emulate ResNets, a Residual Sequence to Sequence (Reseq2seq) network is proposed in this study.

Unlike ResNets which are made of stacked model architecture, applying residual blocks in a Reseq2seq whose architecture is that of an autoencoder is quite different. This difference is due to the flow of the number of neurons through the depth of the model. The ordinary Autoencoder consists of a uniform architecture of neuron distribution which translates to differences in data shapes within the model itself. Apart from Autoencoders, other stacked model architectures including the ResNets, are built based on random numbers of neurons for each layer across the model. However, for the Autoencoders neurons are uniformly increasing and decreasing through the depth architecture of the model for encoders and decoders, respectively. This architecture forms the niche of applying residual blocks in an autoencoder. By applying this discovered niche, a Reseq2seq is proposed in this study as demonstrated in the figure. 4.3.

Reseq2seq network, as depicted in figure 4.3, shows residual blocks connected with adder or subtractor units for Encoder and Decoder, respectively for the purpose of overcoming complexity and gradient hindrances. Here each residual block, emulating ResNet, is made up of two layers in both Encoder and Decoder. These layers are in ascending and descending number of units for each block, respectively.

As it is for Sequence-to-Sequence models, the Reseq2seq model has the capacity of accepting inputs of sequence data, then manipulates the data, and transforms them into another sequence of output data for both Encoders and Decoders. It should be noted though that the output and input may or may not be of the same length. The input in the Encoder is transformed into a vector with a definite and fixed length known as a context vector. The context vector is then transformed by a repeat vector layer back into a 3D shape for compatibility purposes, this is because this output will form the input of the

Decoder (Cho et al., 2014; Sutskever, Vinyals, & Le, 2014; Vinyals, Toshev, Bengio, & Erhan, 2015).

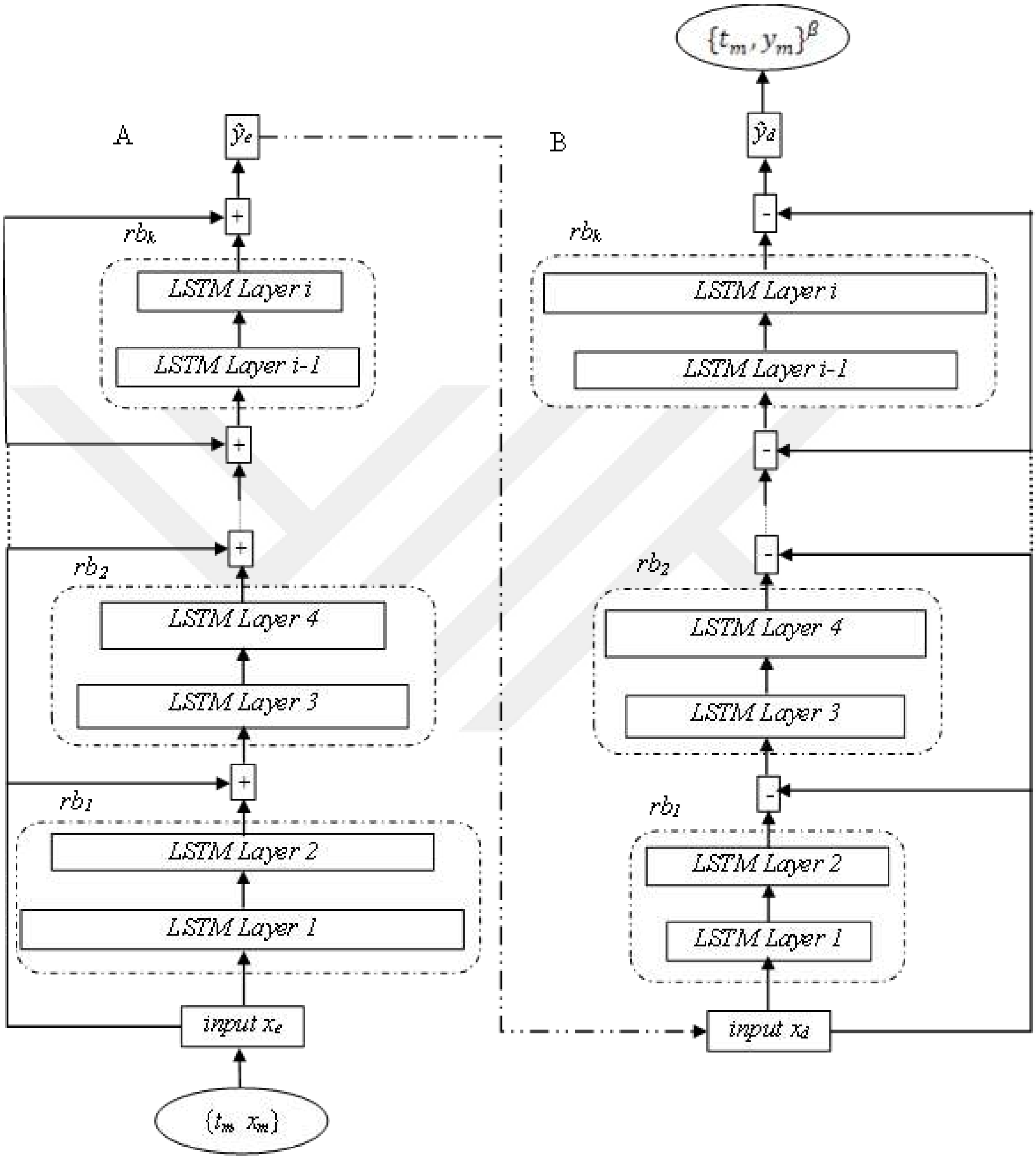


Figure 4.3. *Residual Blocks of* ***A****=Residual Encoder and* ***B****= Residual Decoder Architecture*

The input sequence data are transformed by compression in the Encoder. The context vector from the Encoder is transformed into the input to the Decoder. The Decoder transforms its input by uncompressing it, then it goes through a dense layer with one unit, and finally wrapped with a time-distributed layer. An appropriate loss function and optimizer with an appropriate learning rate are significant to produce reliable results and therefore they should be well selected. The model maps the input to the output through iterations of feedforward and backpropagation processes as shown in figure 4.2, for the purpose of achieving the least mean loss as different parameters including the latent variable, weights, and biases are learned.

4.5.1. Encoder

In the proposed architecture, the Encoder is designed to allow the data inputs in sequence format. Before feeding the Encoder with data, the vibration data is preprocessed in a format compatible with recurrent networks. Data is broken down into blocks or batches meant to be produced by the sensor after every unit of time, refer to section 3 subsection 3.3. The Encoder will create a through-pass of the input data by compression into a single unit of LSTM. This unit then passes the data through a repeat vector layer whose purpose is to transform the output of this model to be compatible with the Decoder's inputs.

It is known that the encoder outputs a two-dimensional matrix that is defined by the number of neurons in the last layer of the decoder. Since recurrent models, including LSTM, expect a three-dimensional input of shape (number of samples, timestep/lookback, and number of features), a kind of adapter known as a repeat vector layer is used to transform the 2D matrix to a 3D matrix compatible with recurrent models. This repeat vector layer is constructed to replicate the context vector step by step for each lookback, in otherwise its number of neurons is considered as the number of timestep/lookback induced for a particular case

4.5.2. Decoder

The Decoder is another independent model proposed in this study. Like the Encoder, the Decoder being a recurrent model, it expects input data compatible with the RNNs. The Decoder does the reverse of what the Encoder does, it decompresses its input data into a sequence form of data whose magnitude is that of the lookback value selected during data preprocessing. The model works well with a number of neurons of powers of two in each layer for both Encoder and Decoders. A one-unit normal dense layer is used as the output of the Decoder, and a wrapper of time distributed layer wraps this layer's output to enable it to output the results in batches of the timesteps or a lookback value predetermined during data preprocessing.

4.5.3. Autoencoder

The Encoder and the decoder which are independent separated residual models are integrated to enable the production of a residual sequence-to-sequence model that is also dubbed an autoencoder. This will enable the model to map a sequence of inputs to a sequence of outputs based on the timestep inculcated into the model. This integration is achieved by making the input of the Reseq2seq network the input of the Encoder, the output of the Encoder be the input of the Decoder, and the output of the network the output of the Decoder.

The deep Reseq2seq network is expected to retrieve the latent variable or the relationship available between that maps the dependent and dependent variables through model training and validation. It is this latent variable that is learned through model training that maps between the independent and dependent variables and will be utilized within the to forecast the next set/batch of data for every input. The model is set to output data of unit datapoints equivalent to the predetermined lookback value meant for a unit time for every data input. The model however can be tuned to accept an undefined size of input less or more than the required output, and this is one of the strengths of the proposed model.

The proposed model expects 3D dependent and independent data, the preprocessing of this data to a format compatible with the proposed model has been discussed in section 3 subsection 3.3.1, refer to this section for more details.

Reseq2seq network is made up of an Encoder (A) and a Decoder(B) as depicted in figure 4.3. A is the recipient of the overall input data (x_e) while B outputs the overall predicted output ($\hat{y}_d$). The two models are made up of an equal number of residual blocks (rb_k) which are comprised of two layers of neurons of ascending and descending order for A and B respectively that are applied in each rb. The outputs of each rb ($\hat{y}_e^{<i>}$ *and* $\hat{y}_d^{<i>}$) that use the output gate ($\Gamma_{ot}^{<i/j>}$), as from equation 4.3, are added and subtracted from the input (x_e and x_d) for models A and B with equations 4.9 and 4.10, respectively. $W_{ot}^{<i/j>}$ *and* $b_{ot}^{<i/j>}$ are weights and biases, and i for the Encoder and j for the Decoder, respectively.

$$\hat{y}_e^{<i>} = \Gamma_{ot}^{<i>} * \tanh\left(c^{<t><i>}\right) + x_e \tag{4.9}$$

where,

$$\Gamma_{ot}^{<i>} = \delta(W_{ot}^{<i>}\left[a_e^{<t-1><i>}, x_e^{<n>}\right] + b_{ot}^{<i>})$$

$$\hat{y}_d^{<i>} = \Gamma_{ot}^{<j>} * \tanh\left(c^{<t><j>}\right) - x_d \tag{4.10}$$

where:

$$\Gamma_{ot}^{<j>} = \delta(W_{ot}^{<j>}\left[a_d^{<t-1><j>}, x_d^{<n>}\right] + b_{ot}^{<j>})$$

It can also be noted that after every even number of layers there is an addition or subtraction layer for encoders and decoders respectively as expressed in cases of equations 4.11 and 4.12. This is what makes residual blocks in the proposed model.

$$\hat{y}_e^{<i>} = \begin{cases} \{\Gamma_{ot}^{<i>} * \tanh\left(c^{<t><i>}\right), c^{<t><i>}\}, & if\ \ layer\%2 = 1 \\ \{\Gamma_{ot}^{<i>} * \tanh\left(c^{<t><i>}\right) + x_e, c^{<t><i>}\}, & if\ layer\%2 = 0 \end{cases} \tag{4.11}$$

$$\hat{y}_d^{<i>} = \begin{cases} \{\Gamma_{ot}^{<j>} * \tanh\left(c^{<t><j>}\right), c^{<t><j>}\}, & if\ \ layer\%2 = 1 \\ \{\Gamma_{ot}^{<j>} * \tanh\left(c^{<t><j>}\right) - x_d, c^{<t><j>}\}, & if\ layer\%2 = 0 \end{cases} \tag{4.12}$$

To make a Reseq2seq, the output of the residual encoder is equated to the input of the decoder as shawm in equations 4.13 and 4.14. Equation 4.15 resulting in equation 4.16 evidently shows that the main input of the model x_e has a direct contribution towards attaining the output of the decoder $\hat{y}_d$, in other words, if the output gate is small or equal to zero the absolute of the two variables would be nearly equal as shown in equation 4.16, and if they are both large, one will decrease the other by subtraction as in equation 4.15. As a result of this, the model's complexity is significantly reduced, and it completely discards the possibilities of vanishing/exploding gradient even if the model's depth is significant.

$$x_d = \hat{y}_e \tag{4.13}$$

which implies,

$$x_d = \Gamma_{ot}^{<i>} * \tanh\left(c^{<t><i>}\right) + x_e \tag{4.14}$$

and,

$$\hat{y}_d = \Gamma_{ot}^{<j>} * \tanh\left(c^{<t><j>}\right) - (\Gamma_{ot}^{<i>} * \tanh\left(c^{<t><i>}\right) + x_e) \tag{4.15}$$

therefore

$$\hat{y}_d = \cancel{\Gamma_{ot}^{<j>} * \tanh\left(c^{<t><j>}\right)} - \left(\cancel{\Gamma_{ot}^{<i>} * \tanh\left(c^{<t><i>}\right)} + x_e\right)$$

$$\hat{y}_d \approx |x_e| \tag{4.16}$$

It should be noted that the expected output from the Encoder $\hat{y}_e^{<i>}$ is in the 2D shape of (number of samples, features). To make this compatible to act as the input of the Decoder, it is reshaped to a 3D shape. In addition to that, the shape of the overall output from the Decoder $\hat{y}_d$ is also set based on the timestep set in the training data that will be fed into the model. The completion of training is signaled by the increase in validation loss that is attained between the validation dependent data y_d and the validation predicted data $\hat{y}_d$, at this point, it is assumed that the optimal latent variable has been attained.

The overall training and validation losses are computed by equation 4.17 and summed up to obtain an overall loss from each neuron and each residual block and computed by equation 4.18. When the optimal latent value is reached, the model is trained enough to be able to predict the future state of the machine $\{(T^*, X^*)\varepsilon R^{n\times m}\}$ based on the timestep the model was trained on, refer to section 3.

$$\mathcal{L}_{e/d}^{<t>}(\hat{y}_{e/d}^{<t>}, y_d^{<t>}) = -y_d^{<t>} \log \hat{y}_{e/d}^{<t>} - (1 - y_d^{<t>}) \log(1 - \hat{y}_{e/d}^{<t>}) \tag{4.17}$$

$$\mathcal{L}(\hat{y}_d, y_d) = \sum_{t=1}^{t} [\mathcal{L}_{e/d}^{<t>}(\hat{y}_{e/d}^{<t>}, y_d^{<t>})] \tag{4.18}$$

Vibration data used in the model consists of random fluctuation vibration signal amplitudes obtained from sensors during the experimental study. The maximum amplitude attained denotes the End of Life (EOL) of the machine. This point was used in this study as a point to alert the model to stop predicting the future state of the machine because it is assumed that the machine has reached its EOL. EOL is selected either by picking the maximum amplitude collected during the experimental study or by using the maximum range of a scaling or normalization function used in data preprocessing. Performing diagnosis and prognosis depends entirely on the estimated future state of the machine and when it will reach its EOL, this is further discussed in subsection 4.6.

4.6. Performing Diagnostics

The model proposed in this study is intended to play a role of an engine that aids PHM by supporting the performance of diagnostics and prognostics of machinery rotating elements. When the proposed model estimates the future state of the machine, this becomes a cornerstone for prognosis and diagnosis. The main objective of the model is to support the inspection of the health condition of the machine as its health deteriorates as well as inform the machine user of how much useful time is still available for the machine to totally fail.

Engineering systems' health condition is monitored through critical assessment of the vibration data that is collected from sensors attached to them. The run-to-fail data collected $\{(T, X)\varepsilon R^{n\times m}\}$ show fluctuation of vibration signal amplitudes but generally demonstrates monotonicity as the health condition deteriorates and yet nonlinear behavior as indicated in figure 5.1. The health condition or normal operation represents signals with less random fluctuation, while a deteriorating machine will emit signals with high random fluctuation. When the health condition begins to deteriorate, the amplitude of the vibration signals increases significantly. The first time the machine's vibration signal amplitude significantly increases, it is deduced that it has experienced its first fault. At the point of experiencing the first fault, the process of estimating RUL starts, this time is known as the Time to Start Predicting (TSP) or First Predict Time (FPT). This behavior is the fulcrum of the analysis of PHM in many published works (Khan & Yairi, 2018; B. Wang et al., 2018; Zhu et al., 2022).

Predicting the future state of engineering systems aids prognostics and diagnostics. When the future state of a mechanical rotating element also known as a bearing, is predicted, it represents much of the future state of the whole engineering system. When the future state is known, it makes it possible to estimate what could happen to the engineering system in the future and enables the preparation of maintenance before the occurrence of the estimated fault. However, before the future state is predicted, the model must be trained to conduct this task. Since the model proposed falls under data-driven models, data is therefore required for this process. It has been discussed already that vibration signals-based run-to-failure data shall be used in this case. An analysis of the data reveals that the signals with less random fluctuation represent the normal health condition while the increased amplitude fluctuation corresponds to a deteriorating health condition of the bearing (Khan & Yairi, 2018; B. Wang et al., 2018; Zhu et al., 2022). Before the data is used to train the model, it goes through a series of preprocessing activities as elaborated in Algorithm 1 for data preprocessing.

Algorithm 1 consists of a pseudocode of the logical implementation of the activities of preprocessing the vibration signals provided for this study. The process starts with accepting raw vibration signals that have been merged with their respective index based on the position of each datapoint in a data file, and the signal data that are represented by T and X respectively. The data is then broken down into two batches $\{t_m, x_m\}$ and $\{t_m, y_m\}$ based on the look back value n as indicated in the loop from line 3 to line 10 and explained further with equations 3.1 and 3.2 in section 3. These two batches will later form independent and dependent variables after they are converted to 3D shapes by the function on lines 11 to 13. The 3D shapes should be compatible with the model proposed in this study as shown and computed with the function. With these activities of preprocessing, the data is ready to train the model.

After the models are trained, the process of predicting the future state of the engineering system begins. However, before predicting the future state of engineering systems, the time to start this process should first be determined. This time is known as the Time to Start Prediction (TSP) which is detected when an engineering system experiences its first fault. To determine TSP, this study proposed an objective and a scientific method of computing the absolute value of the Pearson Correlation Coefficient (PCC) of the two batches of $\{t_m, x_m\}, and \{t_m, y_m\}$. This process is performed by equation 3.5 in section 3 and demonstrated in Algorithm 2. The two batches are used before they are converted to 3D to become dependent and independent variables. The result is a scalar value ψ for each pair of batches. A threshold for the scalar output is set, and the algorithm will accept inputs iteratively in the processing of reaching the threshold. The moment the threshold is reached, it indicates that the first fault has occurred and therefore TSP is detected, this is evidently demonstrated in figure 5.3 in section 5.

Algorithm 1: Data Preprocessing

Input : $\{X \in R^m\} = \{(T, X) \in R^{nxm}\}$

Output: Independent Variable ⇐ ***indep***

Dependent Variable ⇐ ***dep***

$\{t_m, x_m\}, \{t_m, y_m\} \Leftarrow [\ \ ], [\ \ ]$

count ⇐ 0

while $count \leq n$:

  $\{t_m, x_m\}.append\{t_m, x_m\}^{count}$

  $count \Leftarrow$ count + 1

  $\{t_m, y_m\}.append\{t_m, y_m\}^{count}$

  If $count \Leftarrow n$:

    Break from loop

  End of if condition

End of loop

function convertTo3D($data$):

  return $data.reshape[data.shape[0], data.shape[1], 1]$

End of function

$\boldsymbol{indep} \Leftarrow convertTo3D(\{t_m, x_m\})$

$\boldsymbol{dep} \Leftarrow convertTo3D(\{t_m, y_m\})$

It should be noted that this process of detecting TSP does not demonstrate monotonicity behavior as shown in figure 5.3. In other words, if iteration of computing TSP continues, fluctuations of ψ are expected. There might be some values of ψ that might be less than or greater than the threshold after the predefined threshold is detected. The fluctuations are expected, and this does not imply self-healing but rather false self-healing that is caused by the spread of damages in the rotating machine. When the damage is inflicted on the machines' rotating elements in operation, the damage is expected to spread. When the damage spreads, the vibration signals tend to depict normal health condition behavior (Yan et al., 2020). However, this rare behavior does not last for long before it again increases beyond the set threshold. This is because the spread damages do not persist for long until they develop other severe damages. Because of these fluctuations, TSP is detected the first time ψ crosses the threshold, and therefore whatever happens after this, it will not be considered even if ψ goes below or above the threshold.

The outputs of this process are the TSP itself and the latest batch of vibration data that was used to trigger the detection of TSP.

Algorithm 2: TSP Detection

Input : $\{t_m, x_m\}^n, \{t_m, y_m\}^n$

Output: $\psi_n \Leftarrow Pcc\ Value, \{t_m, x_m\}^*$ # The last window after TSP detection

count $\Leftarrow 0$

while $count \leq n$:

  $\text{var1}^{count} \Leftarrow \{t_m, x_m\}^{count} - mean(\{t_m, x_m\}^{count})$

  $var2^{count} \Leftarrow \{t_m, y_m\}^{count} - mean(\{t_m, y_m\}^{count})$

  $$\psi_n = \frac{\sum(\text{var1}^{count} \times var2^{count})}{\sqrt{((\text{var1}^{count})^2 \times (var2^{count})^2)}}$$

  If $\psi_n \Leftarrow threshold$:

    TSP detected

    Break from loop

  End of if condition

End of loop

$\{t_m, x_m\}^*$ # The last window after TSP detection

Unlike a substantial number of research that estimate TSP subjectively, this study does it scientifically by using Pearson's Coefficient Correlation of the dependent and independent variables with a certain threshold. When this threshold is crossed, TSP is determined, at this point, the trained model is fed with the latest batch of data to predict the future state of the machine.

4.7. Prognostics Performance

Using the last batch received after TSP, the trained Reseq2seq network is then used to forecast the future state of the machine until the EOL of the bearing is attained. RUL is then computed by computing the change in TSP and EOL using equation 3.7. However, for this study, the change in TSP and EOL was realized by the multiplication of unit time for each data point with the number of data points predicted as the future state as demonstrated in line 17 of algorithm 3.

The detection of TSP triggers the future state estimation of the engineering systems as well as RUL computation. The process of estimating the future state of an engineering system is performed by a trained Residual Sequence to Sequence network, and then the estimated datapoints are used to compute RUL as it is depicted in Algorithm 3. The Algorithms inputs are the last batch from which TSP was determined that is from Algorithm 2, the predetermined duration t_m from which a complete batch is collected, the number of datapoints that are in a single batch $dpoints_m$ and the predetermined EOL point. The outputs expected from the algorithm are the estimated data that represents the future state of the bearing $\{t_m, y_m\}^{\beta}$ as computed in lines 7 and 11 in the algorithm which in turn are used to compute the predicted RUL (PRUL) as it is defined in line 17 of the algorithm.

The model is trained to predict the data points amounting to the equivalency of the lookback value predetermined while performing data preprocessing for every iteration. In every iteration, the algorithm checks from the predicted datapoints whether the EOL is reached, if not the predicted batch is appended to the $\{t_m, y_m\}^{\beta}$, then the algorithm calls for more prediction from the trained model. The batch from which EOL point is determined, the only datapoints that are less than EOL are appended to $\{t_m, y_m\}^{\beta}$ which forms data representing the complete future state of the bearing. Computing RUL using the representative data of the future state of the machine is by determining how many data points are in $\{t_m, y_m\}^{\beta}$.This number is multiplied by the time determined representing each datapoint. This time is in seconds, if needed in minutes, it is divided by 60 as it is depicted in Algorithm 3 and therefore this process determines the PRUL.

The process of detecting TSP before embarking on future state estimation is the standard used in computing RUL and it has been applied in most published works that propose models that will perform prognostics (Ahmad et al., 2017; Tobon-Mejia et al., 2010).

Algorithm 3: Future State Estimation & RUL Prediction

Input : $\{t_m, x_m\}^*$ # Last window of data after TSP detection

: $t_m,\ dpoints_m$ # Batch time in seconds, and batch data points

: EOL # End of Life Value

Output: $\{t_m, y_m\}^\beta$ #estimated future state

: PRUL #Predicted RUL

1: $\{t_m, x_m\}^\beta \Leftarrow [\quad]$

2: count $\Leftarrow$ 0

3: **while** $true$:

4: $\{t_m, y_m\}^{count} \Leftarrow \text{trainedModel.predict}(\{t_m, x_m\}^*)$

5: $\boldsymbol{for}\ i\ in\ \{t_m, y_m\}^{count}$:

6: $\boldsymbol{if}\ i \geq EOL$: # EOL is end of life of

7: $\{t_m, x_m\}^\beta.append(\{t_m, y_m\}^{count})[0{:}\,t_i]$ $\#t_i\ end\ point\ after\ EOL$

8: **Break from the loops**

9: **End of if Condition**

10: **End of for loop**

11: $\{t_m, x_m\}^\beta.append(\{t_m, y_m\}^{count})$

12: $\{t_m, y_m\}^{count} \Leftarrow \{t_m, x_m\}^*)$ #Making the predicted sequence dependent value

13: $count \Leftarrow \text{count} + 1$

14: **End of while loop**

15: $dpts \Leftarrow count(\{t_m, x_m\}^\beta)$ # Number of data points estimated

16: tdpoint $\Leftarrow \left(\dfrac{dpoints_m}{t_m}\right)$ #time is seconds for each data point

17: PRU = $\left(\dfrac{\text{tdpoint} \times dpoints}{60}\right)$ # PRUL in minutes

When TSP is detected, the proposed process goes two ways, which are the offline test and online modes refer to figure 3.1. In these modes, the model forecasts the possible remaining vibration data until the point of EOL is reached as indicated in algorithm 2 line 7. EOL is predetermined based on the time that the machine completely experienced failure and the scale or normalization technique that was applied. Estimating the time to failure serves as a warning to the mechanics and the engineering system users to proactively plan for maintenance as well as maximally and correctly use the lifespan of the machine.

It should be noted that there were two testing sessions named offline and online. The offline session predicted a single RUL when it receives its first batch of data after TSP is detected. During the live testing session as demonstrated in the methodology, an engineering system in operation is simulated. TSP detection algorithm will accept processed input data as it monitors the health status of the engineering system in operation. The moment TSP is detected, the last batch received is sent to the trained Reseq2seq model to predict the future state. This in turn will enable the computing of RUL using Algorithm 3. The results obtained here will inform the operator of the machine about RUL and the health condition of the machine to enable them to make informed decisions about maintenance. This process continues iteratively. It produces RUL at every iteration until the machine experiences its end of life (EOL). These RULs should be in decreasing order for every iteration for any appropriate model to perform PHM.

4.8. Model Evaluation

Evaluation metrics used in PHM as discussed in section 3 are a way of investigating and assessing the performance of models. Reseq2seq model was evaluated using the Cumulative Relative Accuracy (CRA) metric. CRA was chosen because it penalizes less for errors incurred during early predictions as TSP is detected and penalizes more for errors incurred during late predictions as the machine approaches EOL. The machine is in a critical condition as EOL is approached and any error at this stage might be life-threatening. Therefore, such errors should be severely penalized. Models that obtained less error between Actual RULs (ARUL) and Predicted RULs (PRUL) as EOL of machines is approached are penalized less by CRA and favored in real-life applications. CRA imposes a weight w_i to Relative Accuracy (RA) for the aforementioned penalty. The closer the results to 1 the more they are accurate (Du et al., 2022; B. Wang et al., 2018; M. Xu et al., 2020).

4.9. Conclusion

The Deep Residual Sequence to Sequence model proposed in this study consists of two independent models, the Encoder, and the Decoder. The Encoder acts as the input gate of sequence data and outputs another sequence of data that will be converted into input data of the Decoder to form a Reseq2seq model. The proposed model is trained with vibration signal data acquired from a public repository. This data is preprocessed for the purpose of making it compatible with the model proposed. Before the model forecasts the future state of the machine, TSP is detected. RUL is computed using the estimated future state data from the model. Cumulative Relative Accuracy was chosen as an evaluation metric because it penalizes errors in late predictions more than those incurred in early predictions. The model has demonstrated recommendable results working with vibration data for bearings as it is in detail in section 5. The model is then evaluated using Cumulative Relative Accuracy.

5. RESULTS DISCUSSION AND CONCLUSION

5.1. Overview

This section illustrates the way the proposed Reseq2seq model is applied to perform prognostics and diagnostics. Subsection 5.2 demonstrates the case study and further elaborates on the data used in the study and how it was split into training, validation, and online testing. Subsection 5.3 explains how the model was trained and the obtained results during this process. Subsection 5.4 elaborates on the diagnostics approach, how TSP was determined, and shows all the TSPs determined for all the bearings from the case study while matching the results with TSPs of similar bearing in published works. It further demonstrates the parameters and hyperparameters used in the model development process. Subsection 5.5 demonstrates how the trained model forecasted the future state of the bearing in the testing category. Subsection 5.7 is how the model was assessed and evaluated then the conclusion and remarks come in subsection 5.8.

Table 5.1. *XJTU-SY Bearing Dataset Selected and their Operating Conditions*

#	**Operating Condition (OC)**		**Bearings Identification**		
	Rotating Speed(rmp)	Radial Force(kN)	Training	Validation	Online Testing
1	2100 (35Hz)	12	B11, B14	B13	B12, B14, B15
2	2550 (37.5Hz)	11	B22, B24	B25	B21, B23
3	2400 (40Hz)	10	B31, B34	B32	B33, B35

It should also be noted that the model was developed using TensorFlow 2.7. Models were trained independently based on bearing data from different operating conditions as indicated in table 5.1. For each model to be fully trained, it took close between 3 weeks and to a month or slightly more time to complete. Two multiprocessor computers were used to train models meant for different conditions separately and at the same time. A Windows 10 Enterprise with a processor Intel(R) Xeon(R) CPU E5-2643 v2 @ 3.50GHz 3.50GHz, Installed RAM of 96.0GB and an Ubuntu computer with processors; Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz 2.30GHz Capacity of 4GHz and 62.7GB of RAM. Other parameters and hyperparameters selected for the model are displayed in table 5.2.

5.2. Case Study

The training, testing, and validation of this model were performed using vibration signal data carried out by the Institute of Design Science and Basic Components at Xi'an Jiaotong University (XJTU-SY). The experiment consists of 15 bearings which were used on a testing bed at different operating conditions and resulted in different faults. This study split these bearings into training, validation, and online testing as shown in table 5.1, based on the applied Operation Conditions (OC) and the type of faults that was discovered at the end of the experiment. In this experiment, generated data was sampled in several csv files generated during the operation for each test. Sampling was performed for each file for 1 minute and each file contained 32768 datapoints, which translates to 0.001831 seconds for each data point. There were 2 observations collected, horizontal and vertical vibration data. The former produces a better degradation trend than the latter because of its proximity and position to the bearings and is therefore considered for this study.

Table 5.2. *Hyperparameters and Parameters Used During Training and Validation*

Name	Value/Type
Loss Function	Mean Squared Error
Optimization Algorithm	Adamax
Learning Rate	1e-3
Kernel Regularizers	1e-3
Bias Regularizers	1e-3
Recurrent Regularizers	1e-3
BatchNormalization	After every LSTM layer
Dropout	Ascending and Descending for Encoder and Decoder, respectively
Epochs	200
Early Stopping Patience	2 for val_loss, and the mode was min

5.3. Training the Model

The experimental study that was carried out consisted of 15 bearings, these bearings were split as demonstrated in table 5. 1. The vibration signal data from the selected case study was divided into training, validation, and testing based on the applied conditions. Data were normalized using the minimax normalization algorithm before being fed into the model. The model worked best if trained, validated, and tested with data that was transformed into absolute datapoints (Figure 5.1) and if the data were from a similar condition. The model produced the best training during the training of OC 2 from which validation loss started increasing at the 91st epoch, refer to figure 5.2, while OC 1 and OC 3, 88 and 68 epochs, respectively. It should also be noted that the process of training the model was stopped when the validation loss started to increase.

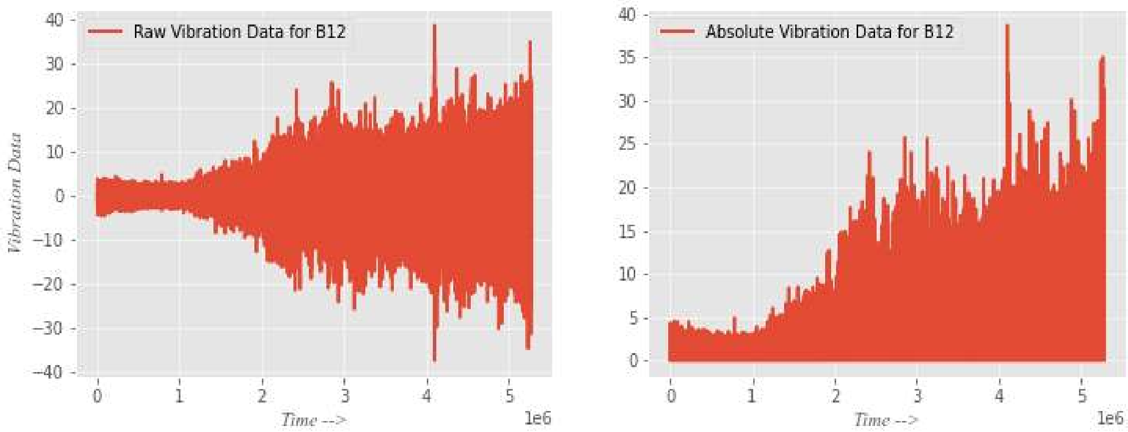


Figure 5.1. *Raw and Absolute Vibration Data for Bearing 12*

The absolute value for all the data was used to remove the fluctuation amplitude between negative and positive. This produced more accuracy and trained the model more compared to the use of the raw data., refer to figure 5.1 which shows the differences between raw and absolute data for bearing 12. Positive or negative vibration data were not considered independently but rather the absolute of the data was considered due to the fact that every single data point in sequence data contributes significantly to the overall results and nevertheless, the mean of all the raw data was not equal to zero. Bearings under OC 2 produced the best training most probably because the normal and fault operations are spread evenly across the lifetime for most bearings under this condition.

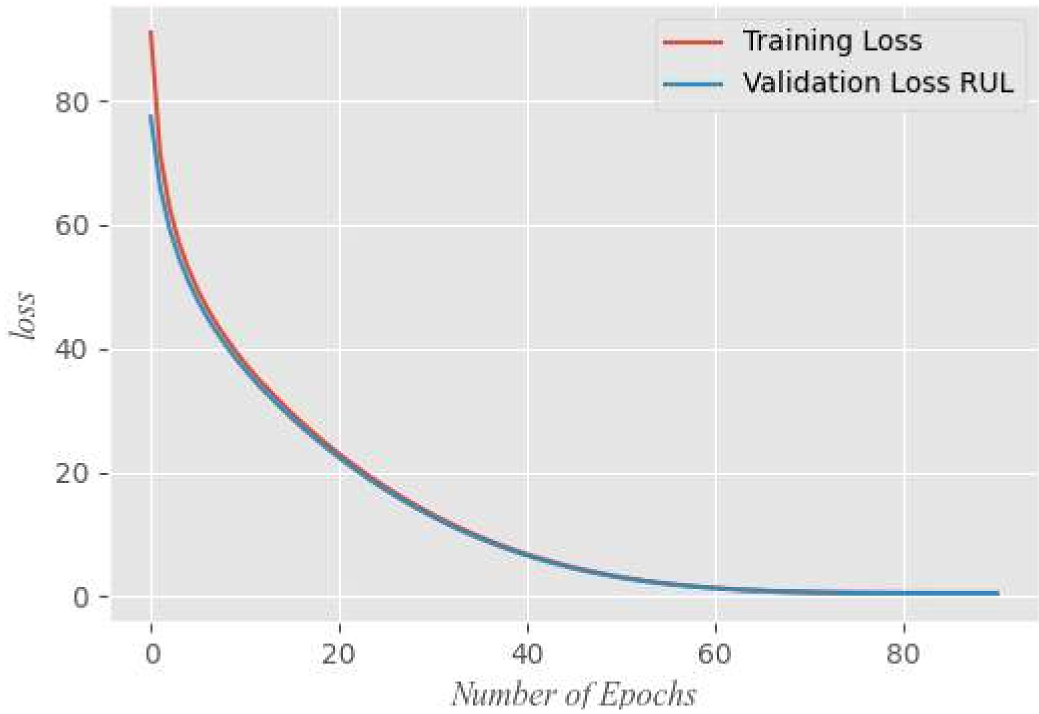


Figure 5.2. *Training Loss Vs Validation Loss for Condition 2*

5.4. TSP Detection and Diagnostics

As discussed in section 3, TSP is the time when the machine experiences the first fault; this is also known as FPT as mentioned earlier. To determine TSP, this study proposed that the absolute value of the Pearson Correlation Coefficient (PCC) for the created dependent and independent values is computed using equation 3.5 in section 3. The result is a scalar value ψ for each pair of batches. Through the investigations that were performed, it was discovered that when PCC values ψ crossed a threshold that is equal to 0.07, TSP is correctly detected, refer to figure 5.3 which shows the detection of TSP in bearings 11, 23, and 24.

The fluctuation of ψ that is displayed in Figure 5.3, does not imply self-healing but rather false self-healing that is caused by the spread of damages. When the damage inflicted on the bearing spreads, the bearing tends to behave as if it is in a normal health condition. However, this rare behavior does not last for long before it again increases beyond the set threshold. This is because the spread damages do not persist for long until they develop other severe damages. Because of these fluctuations, TSP is detected the first time when ψ crosses the threshold and therefore whatever happens after this is not considered even if ψ goes below the threshold. This was true except for bearing B13, B25, and B32 which looked to have outliers that as a result showed earlier detection of

TSP in comparison with published works. When outlier removal techniques were applied, like the moving average of less or equal to 1500 window for this case, they all responded effectively with reliable results compared with other published results as indicated in table 5.3.

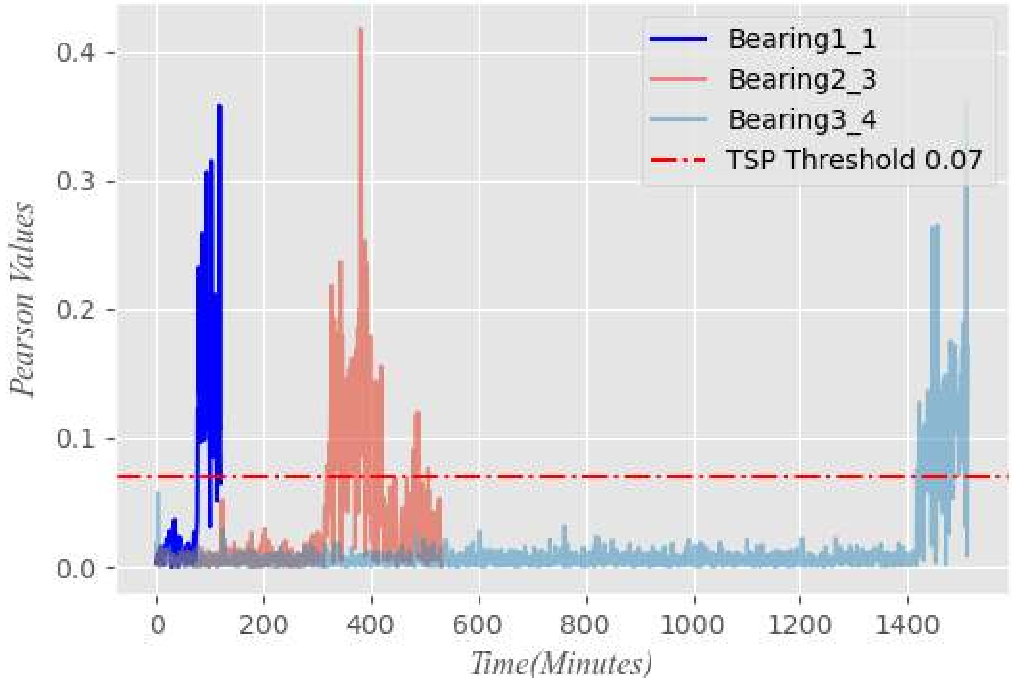


Figure 5.3. *Shows the Increase in PCC Value for TSP Detection*

When TSP is detected, the process of computing RULs begins. The last batch of data from which TSP was detected is fed into the trained proposed model to forecast future data until EOL is reached. When EOL is attained, the model stops forecasting, and thereafter RUL is computed for that first batch received. RUL is the time between TSP and EOL. The RUL computed here is saved as the model received a second batch from the time TSP was detected. When the second batch is received, another RUL is computed. This current RUL is expected to be less than the first RUL because physically the remaining useful life should be decreasing as more batches are received. This process continues iteratively till the EOL of the machine.

Table 5.3. *Proposed TSP Detection Technique in Comparison with Other Methods (All in Minutes)*

Bearing	Bearing Lifetime	Lin's (T. Lin et al., 2022)	Huang's (C.-G. Huang et al., 2021)	Nemani's (Nemani, Lu, Thelen, Hu, & Zimmerman, 2022)	Proposed Method
B11	123	75	76	79	79
B12	161	45	44	55	59
B13	158	61	60	60	89
B14	122	85	--	--	85
B15	52	39	39	26	46
B21	491	455	455	456	455
B22	161	48	48	50	83
B23	533	128	327	316	319
B24	42	31	32	32	31
B25	339	144	114	123	144
B31	2538	2464	2344	2404	2464
B32	2496	2142	--	2450	2142
B33	371	331	340	343	343
B34	1515	1417	1418	1420	1418
B35	114	21	9	8(20)	12

5.5. Model Forecasting Process

The model did not forecast accurately when given data before TSP is detected. When TSPs were detected, the models were fed with the latest batch of data and forecasted future data until EOL is detected. The model successfully estimated the future state of test bearing, some of which are shown in figures 5.4, 5.4, and 5.6, and in table 5.4 which shows the first RUL prediction for bearings selected for online testing mode. The RUL computed for every batch of data received matched with insignificant error with the actual RUL that is computed based on the TSPs of each bearing. Since the data was normalized before being fed into the model, the EOL was set at 1.0, therefore when the model predicted values greater than or equal to EOL, the model was stopped and subsequently RUL was computed for every time the model received a batch of data.

Table 5.4. *Predicted and Actual RUL (mins) of the Case Study After Detecting TSP*

Bearing	No. of Data Points Predicted	Predicted RUL	Actual RUL
B12	3353258	102.330	102
B14	1241766	37. 894	37
B15	251223	7.665	6
B21	1208297	36.914	36
B23	7015580	214.092	214
B33	894077	27.28	28
B34	3151216	96.164	97
B35	3437774	104.909	102

PRUL is determined by equation 3.6 from section 3 which is the time between TSP and EOL. In this study, every datapoint predicted was considered in computing RUL. Data points predicted were counted (dpts) and multiplied by time in seconds for each data point (tdpts) which happens at 0.001831 seconds as mentioned before, this is realized by PRUL = (dpts* tdpts)/60. The division by 60 is meant to convert the PRUL into minutes. This is computed from the dataset that is composed of files collected within 1 minute and each file contains 32768 datapoints, refer to table 5.3 which shows actual and predicted RUL for all the test bearings.

Using the model, RUL was predicted and every time it received a new batch of data as RUL decreased as the bearing approached its EOL. It can be noted from figure 5.7 that the more the data received by the model towards EOL the more accurate it became. The model predicts almost accurately the Actual RUL as the bearing approached its EOL, this gives assurances that EOL will not be reached before the model gives accurate RUL. For safety reasons, it may also be noted that more predictions are less than the Actual RUL as we approach the EOL, this also gives confidence to the model users that it will not allow a catastrophic situation before it is reported. For more results refer to the app. table 1 and app. figure 1 & 2 in the appendix.

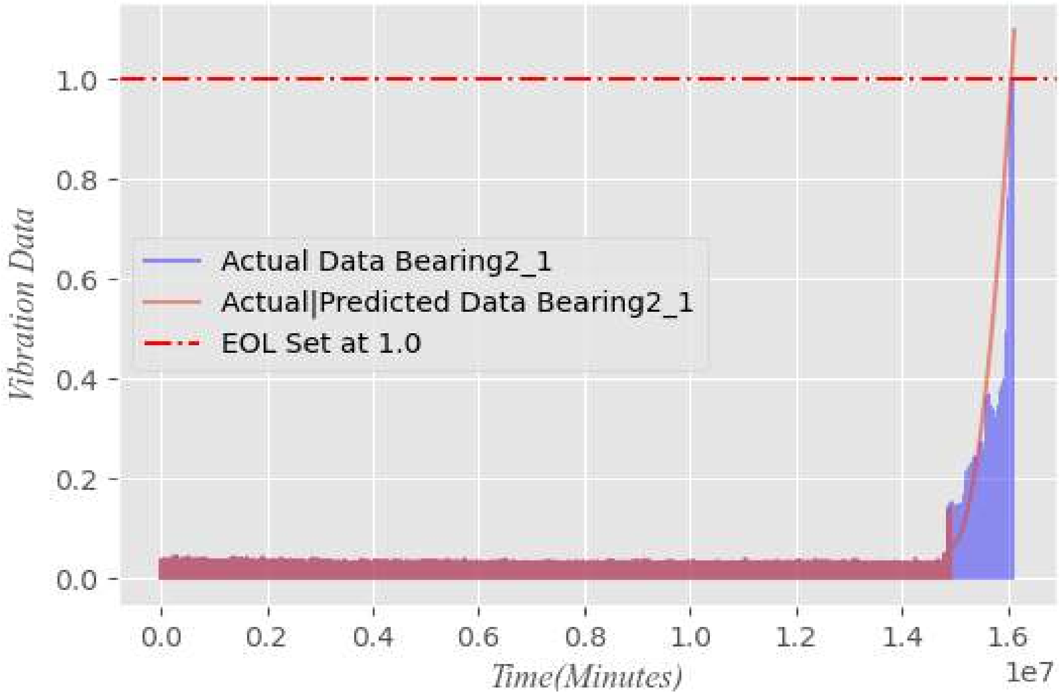


Figure 5.4. *Showing Forecasting Made After Detecting TSP for Bearing21*

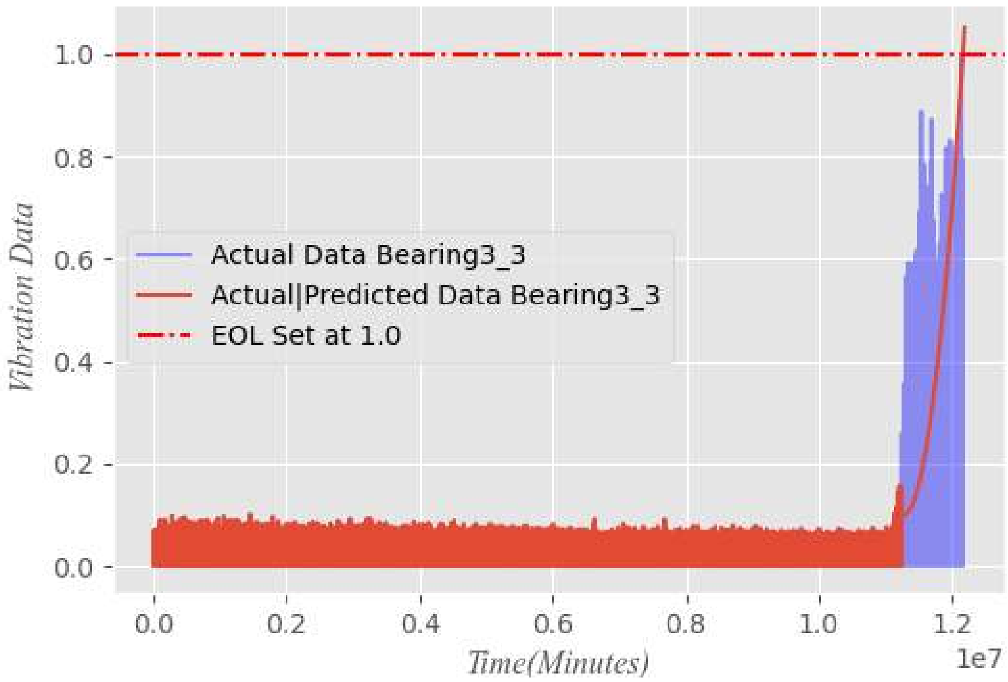


Figure 5.5. *Showing Forecasting Made After Detecting TSP for Bearing33*

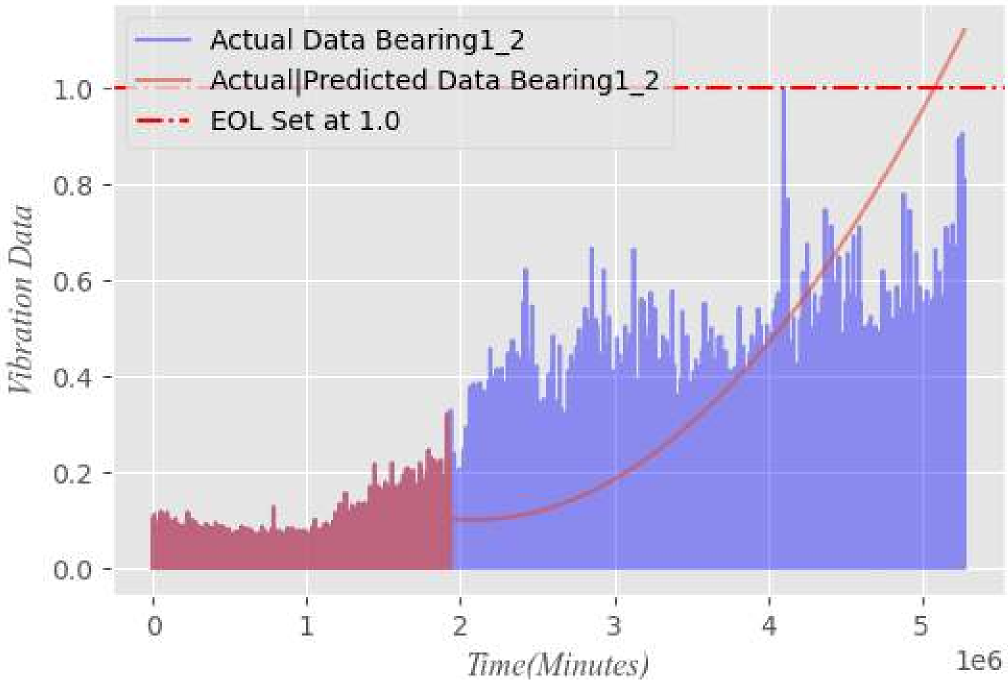


Figure 5.6. *Showing Forecasting Made After Detecting TSP for Bearing12*

5.6. Model Evaluation

To evaluate the proposed model in this research results from Biao Wang (Du et al., 2022; B. Wang et al., 2018) were used for comparison. In Biao's article, results were obtained from four models that were evaluated in relation to the model that was proposed in the article. The model was named Hybrid Prognostics Approach (HPA). The compared models were Relevance Vector Machine (RVM), Deep Belief Network (DBN), Particle Filtering (PF), and Extended Kalman Filtering (EKF) with HPA. The results demonstrated the superiority of HPA over the other 4 models.

In this research, the results that were produced by Reseq2seq model were compared with the results from the five models, RVM, DBN, PF, EKF, and HPA from the article that is mentioned in the previous paragraph, refer to table 5.5 and figure 5.8. It can be deduced from table 5.4 and figure 5.8 that the Reseq2seq model generally outperformed all the other models proposed as all best results are bolded in the table.

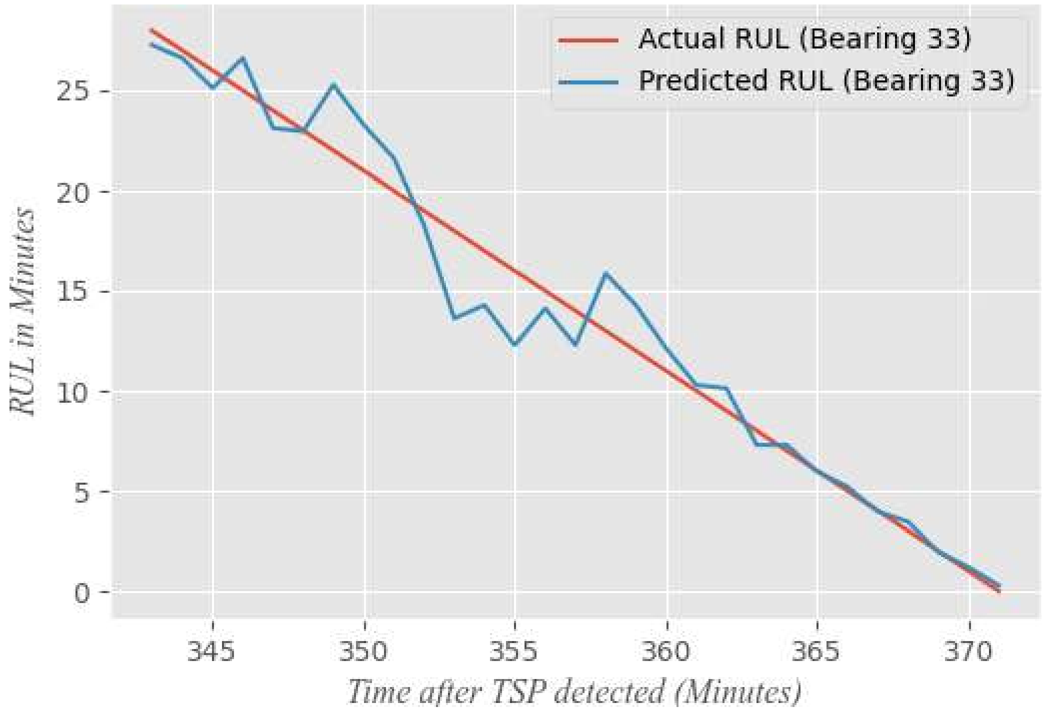


Figure 5.7. *Showing Actual RUL vs Predicted RUL for Bearing33*

The bearings that were put under the category of online test from the dataset, whose results were compared with the results of the other model, indicate that for bearing 12, HPA outperformed the proposed model with just *5.53%* superiority.

Table 5.5. *Comparison of Evaluations Metrics with the Proposed Model (CRA)*

Bearing	RVM	DBN	PF	EKF	HPA	Proposed Method	Percentage Comparison
B12	0.1815	0.6248	0.7256	0.3500	**0.8546**	0.8003	-5.53%
B14	0.3722	-0.947	0.2305	0.6839	0.7240	**0.78879**	**6.49%**
B15	0.6122	0.6636	0.4311	0.5042	**0.7878**	0.5345	-25.33%
B21	0.5718	0.5518	0.3963	0.5750	0.8621	**0.8910**	**2.89%**
B23	0.6172	0.9013	0.7364	0.8800	0.9612	**0.9617**	**0.5%**
B33	-0.0883	0.5167	0.1283	-0.9643	0.8887	**0.8900**	**0.13%**
B34	0.6570	0.6050	0.7830	0.6670	0.8133	**0.91711**	**9.78%**
B35	0.4881	0.5618	0.3857	0.5040	0.6512	**0.9406**	**28.64%**

In the case of bearing 14, the Reseq2seq model outperformed all the other models by 6.46%, however, it was again outperformed by 25.33% for bearing 15. Reseq2seq model gained moment with bearings 21, 23, 33, 34, and 35 as it was superior to the other models by 2.89%, 0.5%, 0.13%, 9.78%, and 28.94%, respectively. In general, based on the comparative study, Reseq2seq model outperformed all the other models by *17.87%*.

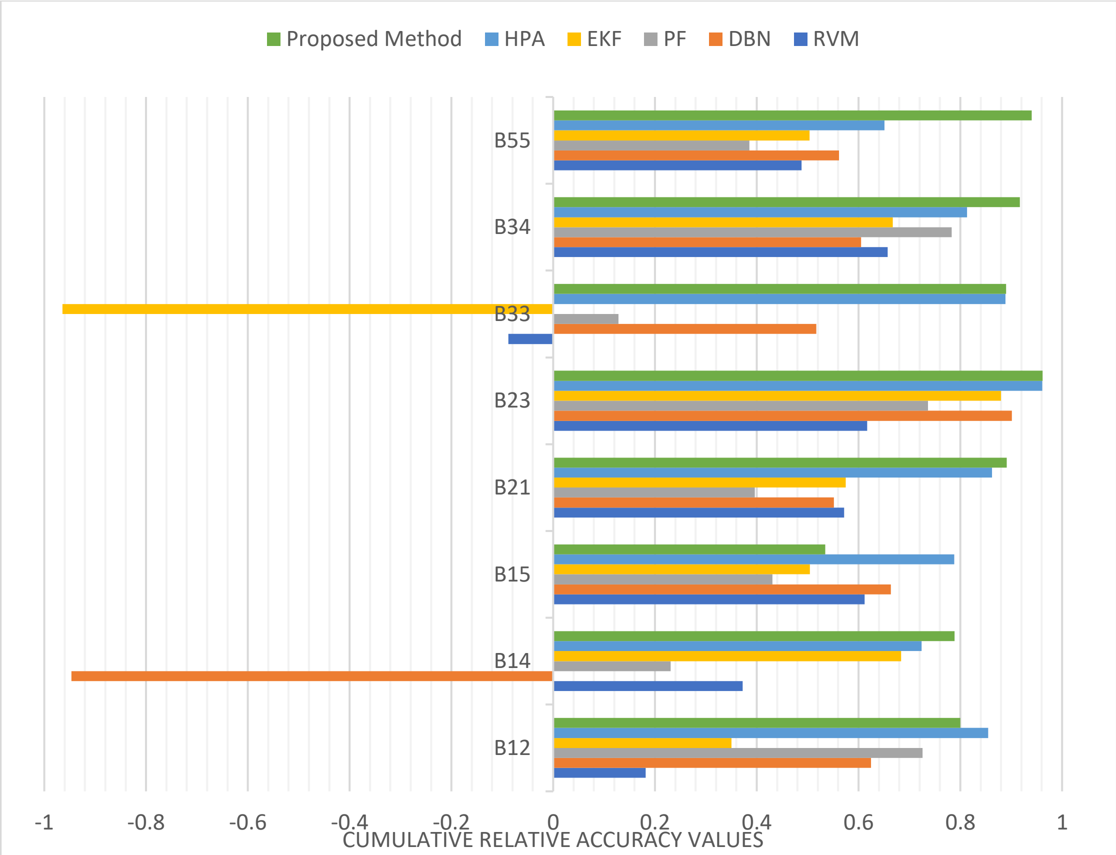


Figure 5.8. *A Clustered Bar Chart Demonstrating the Superiority of the Proposed Model*

5.7. Remarks and Conclusion

A deep Residual Sequence Autoencoder network model is proposed in this study. The model is comprised of two independent models, the residual Encoder, and the residual Decoder. The formation of the proposed model is by matching the output produced from the Encoder to the input required for the Decoder. Bearing vibration data was preprocessed in batches of 60 seconds where each batch was converted into a 3D shape in conformity with the proposed model inputs. Using this data, the model was trained until the validation loss started to increase. At this stage, the model was stopped from training because it is presumed that it had been trained enough to predict and forecast data expected in the next 60 seconds. During the testing phase, the model was able to forecast future data with minimal errors. Generally, the model proposed aided reliably prognostics and diagnostics of rolling elements in machines. In all the cases the model was tested, it demonstrated predicting PRUL that is almost equal to ARUL as the mechanical rolling elements approaches its EOL. When the model was evaluated, it demonstrated superiority of *17.87%* over the models that it compared with.

Even though the model produced reliable results, it should be noted that it underperformed in forecasting the future state of the bearings, and estimation of RUL before detection of TSP, which can be an area that requires further investigation. The possibility of a model to predict run-to-failure data of the machine by just receiving the first batch of the data is another open area for research. During data preprocessing, if more accuracy is needed, the lookback value should be considered. The less the lookback value the more the accuracy. However, it should be noted that the lesser the timestamp/lookback value the more the data created, and this consequently translates into more computer resources required. Therefore, a large amount of computer resources is required to manifest into this research to attain more accuracy. Another factor that calls for further research would be training the model once and apply it on all the bearings for testing; however, this may also require high computational power. Finally, it can be concluded that the Reseq2seq model outperformed models evaluated alongside it and demonstrated that it can be reliable if applied in real life in related PHM projects.

REFERENCES

Ahmad, W., Khan, S. A., & Kim, J.-M. (2017). A hybrid prognostics technique for rolling element bearings using adaptive predictive models. *IEEE Transactions on Industrial Electronics, 65*(2), 1577-1584.

Aivaliotis, P., Georgoulias, K., & Chryssolouris, G. (2019). The use of Digital Twin for predictive maintenance in manufacturing. *International Journal of Computer Integrated Manufacturing, 32*(11), 1067-1080.

Aydemir, G., & Acar, B. (2020). Anomaly monitoring improves remaining useful life estimation of industrial machinery. *Journal of Manufacturing Systems, 56*, 463-469.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks, 5*(2), 157-166.

Benkedjouh, T., Medjaher, K., Zerhouni, N., & Rechak, S. (2013). Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence, 26*(7), 1751-1760.

Cakir, M., Guvenc, M. A., & Mistikoglu, S. (2021). The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system. *Computers & Industrial Engineering, 151*, 106948.

Cao, P., Zhang, S., & Tang, J. (2018). Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning. *Ieee Access, 6*, 26241-26253.

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.

Chen, D., Qin, Y., Wang, Y., & Zhou, J. (2021). Health indicator construction by quadratic function-based deep convolutional auto-encoder and its application into bearing RUL prediction. *ISA transactions, 114*, 44-56.

Cheng, J. C., Chen, W., Chen, K., & Wang, Q. (2020). Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Automation in Construction, 112*, 103087.

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*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ding, G., Wang, W., & Zhao, J. (2022). Prediction of remaining useful life of rolling bearing based on fractal dimension and convolutional neural network. *Measurement and Control*, 00202940211065674.

Ding, Y., Ding, P., Zhao, X., Cao, Y., & Jia, M. (2022). Transfer Learning for Remaining Useful Life Prediction Across Operating Conditions Based on Multisource Domain Adaptation. *IEEE/ASME Transactions on Mechatronics*.

Dong, S., Wen, G., Lei, Z., & Zhang, Z. (2021). Transfer learning for bearing performance degradation assessment based on deep hierarchical features. *ISA transactions, 108*, 343-355.

Du, W., Hou, X., & Wang, H. (2022). Time-Varying Degradation Model for Remaining Useful Life Prediction of Rolling Bearings under Variable Rotational Speed. *Applied Sciences, 12*(8), 4044.

Elsheikh, A., Yacout, S., & Ouali, M.-S. (2019). Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing, 323*, 148-156.

Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems, 91*(2), 179-189.

Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence, 92*, 103678.

Gallicchio, C., & Micheli, A. (2017). Deep echo state network (deepesn): A brief survey. *arXiv preprint arXiv:1712.04323*.

García, E., Montés, N., & Alacreu, M. (2018). *Towards a Knowledge-driven Maintenance Support System for Manufacturing Lines.* Paper presented at the ICINCO (1).

Guo, J., Xu, L., & Chen, E. (2020). *Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation.* Paper presented at the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Hanif, M. S., & Bilal, M. (2020). Competitive residual neural network for image classification. *ICT Express, 6*(1), 28-37.

He, J., Li, X., Chen, Y., Chen, D., Guo, J., & Zhou, Y. (2021). Deep transfer learning method based on 1D-CNN for bearing fault diagnosis. *Shock and Vibration, 2021*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780.

Huang, C.-G., Huang, H.-Z., Li, Y.-F., & Peng, W. (2021). A novel deep convolutional neural network-bootstrap integrated method for RUL prediction of rolling bearing. *Journal of Manufacturing Systems, 61*, 757-772.

Huang, W.-C., Hayashi, T., Wu, Y.-C., Kameoka, H., & Toda, T. (2021). Pretraining techniques for sequence-to-sequence voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*, 745-755.

Javed, K., Gouriveau, R., & Zerhouni, N. (2017). State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mechanical systems and signal processing, 94*, 214-236.

Javed, K., Gouriveau, R., Zerhouni, N., & Nectoux, P. (2014). Enabling health monitoring approach based on vibration data for accurate prognostics. *IEEE Transactions on Industrial Electronics, 62*(1), 647-656.

Jia, F., Lei, Y., Lin, J., Zhou, X., & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical systems and signal processing, 72*, 303-315.

Jimenez, J. J. M., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems, 56*, 539-557.

Khamoudj, C. E., Benbouzid-Si Tayeb, F., Benatchba, K., Benbouzid, M., & Djaafri, A. (2020). A Learning Variable Neighborhood Search Approach for Induction Machines Bearing Failures Detection and Diagnosis. *Energies, 13*(11), 2953.

Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical systems and signal processing, 107*, 241-265.

Khorram, A., Khalooei, M., & Rezghi, M. (2021). End-to-end CNN+ LSTM deep learning approach for bearing fault diagnosis. *Applied Intelligence, 51*(2), 736-751.

Kumar, A., Parkash, C., Vashishtha, G., Tang, H., Kundu, P., & Xiang, J. (2022). State-space modeling and novel entropy-based health indicator for dynamic degradation monitoring of rolling element bearing. *Reliability Engineering & System Safety*, 108356.

Lee, S., Yu, H., Yang, H., Song, I., Choi, J., Yang, J., . . . Kwon, J. (2021). A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment. *Applied Sciences, 11*(4), 1564.

Li, C., Zhang, W., Peng, G., & Liu, S. (2017). Bearing fault diagnosis using fully-connected winner-take-all autoencoder. *Ieee Access, 6*, 6103-6115.

Li, J., Zi, Y., Wang, Y., & Yang, Y. (2022). Health Indicator Construction Method of Bearings Based on Wasserstein Dual-Domain Adversarial Networks Under Normal Data Only. *IEEE Transactions on Industrial Electronics, 69*(10), 10615-10624.

Liang, Z., & Parlikad, A. K. (2020). Predictive group maintenance for multi-system multi-component networks. *Reliability Engineering & System Safety, 195*, 106704.

Lin, R., Yu, Y., Wang, H., Che, C., & Ni, X. (2022). Remaining useful life prediction in prognostics using multi-scale sequence and Long Short-Term Memory network⋆. *Journal of Computational Science, 57*, 101508.

Lin, T., Wang, H., Guo, X., Wang, P., & Song, L. (2022). A novel prediction network for remaining useful life of rotating machinery. *The International Journal of Advanced Manufacturing Technology*, 1-10.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., . . . Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics, 8*, 726-742.

Long, J., Zhang, S., & Li, C. (2019). Evolving deep echo state networks for intelligent fault diagnosis. *IEEE Transactions on Industrial Informatics, 16*(7), 4928-4937.

Luo, J., & Zhang, X. (2022). Convolutional neural network based on attention mechanism and Bi-LSTM for bearing remaining life prediction. *Applied Intelligence, 52*(1), 1076-1091.

Mao, W., Chen, J., Liu, J., & Liang, X. (2022). Self-supervised Deep Domain-Adversarial Regression Adaptation for Online Remaining Useful Life Prediction of Rolling Bearing under Unknown Working Condition. *IEEE Transactions on Industrial Informatics*.

Mao, W., Feng, W., Liu, Y., Zhang, D., & Liang, X. (2021). A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis. *Mechanical systems and signal processing, 150*, 107233.

Mao, W., Tian, S., Fan, J., Liang, X., & Safian, A. (2020). Online detection of bearing incipient fault with semi-supervised architecture and deep feature representation. *Journal of Manufacturing Systems, 55*, 179-198.

Miao, M., Yu, J., & Zhao, Z. (2022). A sparse domain adaption network for remaining useful life prediction of rolling bearings under different working conditions. *Reliability Engineering & System Safety, 219*, 108259.

Mobley, R. K. (2002). *An introduction to predictive maintenance*: Elsevier.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). *PRONOSTIA: An experimental platform for bearings accelerated degradation tests.* Paper presented at the IEEE International Conference on Prognostics and Health Management, PHM'12.

Nemani, V. P., Lu, H., Thelen, A., Hu, C., & Zimmerman, A. T. (2021). Ensembles of Probabilistic LSTM Predictors and Correctors for Bearing Prognostics Using Industrial Standards. *Neurocomputing*.

Nemani, V. P., Lu, H., Thelen, A., Hu, C., & Zimmerman, A. T. (2022). Ensembles of probabilistic LSTM predictors and correctors for bearing prognostics using industrial standards. *Neurocomputing, 491*, 575-596.

Ni, Q., Ji, J., & Feng, K. (2022). Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network. *IEEE Transactions on Industrial Informatics*.

Nieves Avendano, D., Vandermoortele, N., Soete, C., Moens, P., Ompusunggu, A. P., Deschrijver, D., & Van Hoecke, S. (2022). A Semi-Supervised Approach with Monotonic Constraints for Improved Remaining Useful Life Estimation. *Sensors, 22*(4), 1590.

Ochella, S., Shafiee, M., & Dinmohammadi, F. (2022). Artificial intelligence in prognostics and health management of engineering systems. *Engineering Applications of Artificial Intelligence, 108*, 104552.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). *On the difficulty of training recurrent neural networks.* Paper presented at the International conference on machine learning.

Pech, M., Vrchota, J., & Bednář, J. (2021). Predictive Maintenance and Intelligent Sensors in Smart Factory. *Sensors, 21*(4), 1470.

Pirhooshyaran, M., & Snyder, L. V. (2020). Forecasting, hindcasting and feature selection of ocean waves via recurrent and sequence-to-sequence networks. *Ocean Engineering, 207*, 107424.

Porotsky, S., & Bluvband, Z. (2012). *Remaining useful life estimation for systems with non-trendability behaviour.* Paper presented at the 2012 IEEE Conference on Prognostics and Health Management.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., . . . Zhou, M. (2020). Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Rao, Y., He, L., & Zhu, J. (2017). *A residual convolutional neural network for pan-shaprening.* Paper presented at the 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP).

Rigamonti, M., Baraldi, P., Zio, E., Roychoudhury, I., Goebel, K., & Poll, S. (2016). *Echo state network for the remaining useful life prediction of a turbofan engine.* Paper presented at the PHM Society European Conference.

Schwendemann, S., Amjad, Z., & Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry, 125*, 103380.

Sehovac, L., & Grolinger, K. (2020). Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention. *Ieee Access, 8*, 36411-36426.

Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics, 15*(4), 2446-2455.

Skomski, E., Lee, J.-Y., Kim, W., Chandan, V., Katipamula, S., & Hutchinson, B. (2020). Sequence-to-sequence neural networks for short-term electrical load forecasting in commercial office buildings. *Energy and Buildings, 226*, 110350.

Song, W., Liu, H., & Zio, E. (2022). Long-range dependence and heavy tail characteristics for remaining useful life prediction in rolling bearing degradation. *Applied Mathematical Modelling, 102*, 268-284.

Suh, S., Lukowicz, P., & Lee, Y. O. (2022). Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks. *Knowledge-Based Systems, 237*, 107866.

Sutrisno, E., Oh, H., Vasan, A. S. S., & Pecht, M. (2012). *Estimation of remaining useful life of ball bearings using data driven methodologies.* Paper presented at the 2012 ieee conference on prognostics and health management.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to sequence learning with neural networks.* Paper presented at the Advances in neural information processing systems.

Tang, Y.-J., Pang, Y.-H., & Liu, B. (2020). IDP-Seq2Seq: identification of intrinsically disordered regions based on sequence to sequence learning. *Bioinformatics, 36*(21), 5177-5186.

Tobon-Mejia, D. A., Medjaher, K., & Zerhouni, N. (2010). *The ISO 13381-1 standard's failure prognostics process through an example.* Paper presented at the 2010 Prognostics and System Health Management Conference.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). *Show and tell: A neural image caption generator.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Wan, S., Li, X., Zhang, Y., Liu, S., Hong, J., & Wang, D. (2022). Bearing remaining useful life prediction with convolutional long short-term memory fusion networks. *Reliability Engineering & System Safety, 224*, 108528.

Wang, B., Lei, Y., Li, N., & Li, N. (2018). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability, 69*(1), 401-412.

Wang, D., Wang, H.-W., Li, C., Lu, K.-F., Peng, Z.-R., Zhao, J., . . . Pan, J. (2020). Roadside air quality forecasting in shanghai with a novel sequence-to-sequence model. *International Journal of Environmental Research and Public Health, 17*(24), 9471.

Wang, J., Wang, D., Wang, S., Li, W., & Song, K. (2021). Fault Diagnosis of Bearings Based on Multi-Sensor Information Fusion and 2D Convolutional Neural Network. *Ieee Access, 9*, 23717-23725.

Wang, J., Wen, G., Yang, S., & Liu, Y. (2018). *Remaining useful life estimation in prognostics using deep bidirectional lstm neural network.* Paper presented at the 2018 Prognostics and System Health Management Conference (PHM-Chongqing).

Wang, W., Lei, Y., Yan, T., Li, N., & Nandi, A. (2022). Residual convolution long short-term memory network for machines remaining useful life prediction and uncertainty quantification. *Journal of Dynamics, Monitoring and Diagnostics, 1*(1), 2-8.

Wang, X., Shen, C., Xia, M., Wang, D., Zhu, J., & Zhu, Z. (2020). Multi-scale deep intra-class transfer learning for bearing fault diagnosis. *Reliability Engineering & System Safety, 202*, 107050.

Wang, Y., Wu, J., Cheng, Y., Wang, J., & Hu, K. (2022). Memory-enhanced hybrid deep learning networks for remaining useful life prognostics of mechanical equipment. *Measurement, 187*, 110354.

Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing, 275*, 167-179.

Xiang, S., Qin, Y., Zhu, C., Wang, Y., & Chen, H. (2020). Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. *Engineering Applications of Artificial Intelligence, 91*, 103587.

Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water resources research, 56*(1), e2019WR025326.

Xiao, L., Liu, Z., Zhang, Y., Zheng, Y., & Cheng, C. (2020). Degradation assessment of bearings with trend-reconstruct-based features selection and gated recurrent unit network. *Measurement, 165*, 108064.

Xu, J., Duan, S., Chen, W., Wang, D., & Fan, Y. (2022). SACGNet: A Remaining Useful Life Prediction of Bearing with Self-Attention Augmented Convolution GRU Network. *Lubricants, 10*(2), 21.

Xu, M., Baraldi, P., Al-Dahidi, S., & Zio, E. (2020). Fault prognostics by an ensemble of Echo State Networks in presence of event based measurements. *Engineering Applications of Artificial Intelligence, 87*, 103346.

Yan, M., Wang, X., Wang, B., Chang, M., & Muhammad, I. (2020). Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. *ISA transactions, 98*, 471-482.

Yu, W., Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y. (2019). A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics, 16*(1), 183-192.

Yu, W., Kim, I. Y., & Mechefske, C. (2021). Analysis of different RNN autoencoder variants for time series classification and machine prognostics. *Mechanical systems and signal processing, 149*, 107322.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zhang, B., Zhang, S., & Li, W. (2019). Bearing performance degradation assessment using long short-term memory recurrent network. *Computers in Industry, 106*, 14-29.

Zhang, S., Sun, Z., Wang, M., Long, J., Bai, Y., & Li, C. (2019). Deep fuzzy echo state networks for machinery fault diagnosis. *IEEE Transactions on Fuzzy Systems, 28*(7), 1205-1218.

Zhang, S., Ye, F., Wang, B., & Habetler, T. G. (2020). Semi-supervised bearing fault diagnosis and classification using variational autoencoder-based deep generative models. *IEEE Sensors Journal, 21*(5), 6476-6486.

Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal, 13*(3), 2213-2227.

Zhao, D., Feng, L., & Wang, X. (2022). Bearing Remaining Useful Life Estimation Based on Encoder and Gated Recurrent Units. *IAENG International Journal of Computer Science, 49*(1).

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., . . . Torr, P. H. (2021). *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Zhu, J., Jiang, Q., Shen, Y., Qian, C., Xu, F., & Zhu, Q. (2022). Application of recurrent neural network to mechanical fault diagnosis: a review. *Journal of Mechanical Science and Technology*, 1-16.

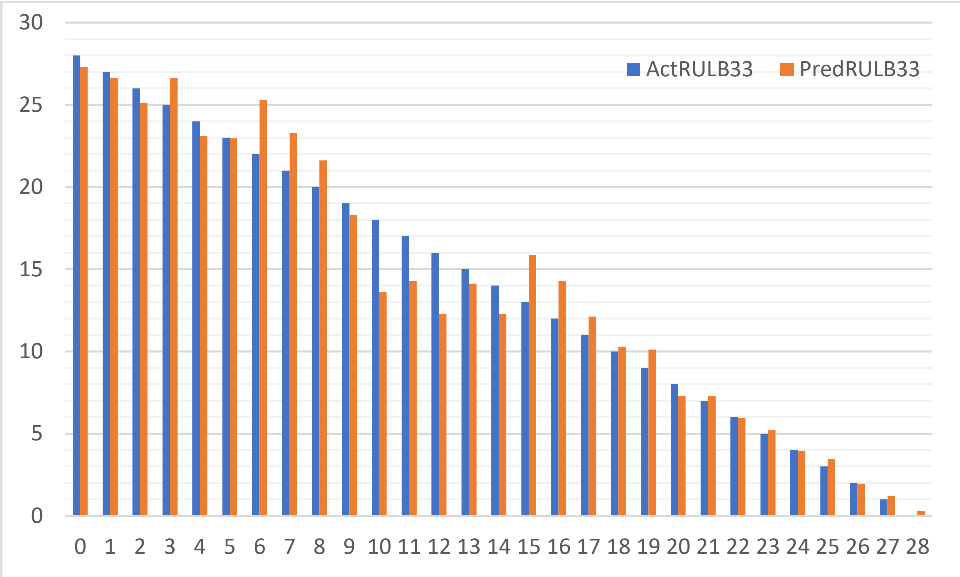
Zhuang, J., Jia, M., & Zhao, X. (2022). An adversarial transfer network with supervised metric for remaining useful life prediction of rolling bearing under multiple working conditions. *Reliability Engineering & System Safety*, 108599.

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 106889.

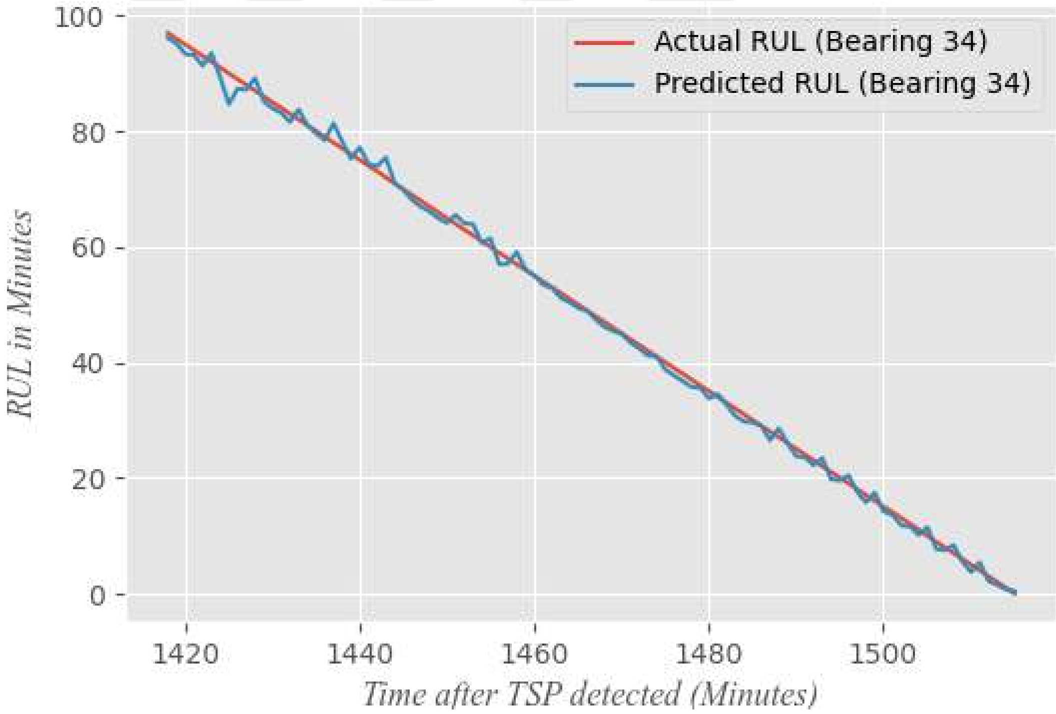
APPENDIX

App. Table 1. *Demonstration of Complete RUL for Bearing 33*

Actual RUL	Datapoints predicted	Predicted RUL
28	894077	27.28425
27	872232	26.61761
26	823079	25.11763
25	872233	26.61764
24	757543	23.11769
23	752083	22.95107
22	828544	25.2844
21	763007	23.28443
20	708393	21.61779
19	599165	18.28452
18	446244	13.61788
17	468091	14.28458
16	402554	12.28461
15	462631	14.11796
14	402556	12.28467
13	519979	15.86803
12	468095	14.2847
11	397097	12.11808
10	337022	10.28479
9	331561	10.11814
8	238717	7.284847
7	238718	7.284878
6	195027	5.951574
5	170451	5.201596
4	129491	3.951634
3	113107	3.451649
2	63955	1.951693
1	39379	1.201716
0	9342	0.285087



App. Figure 1. Actual and Predicted *RUL Comparison for Bearing 34*



App. Figure 2: Actual and Predicted *RUL Comparison for Bearing 34*