Studies on the Development of Data-Driven Intelligent Predictive Maintenance Framework for the Critical Components of Machine Tool Systems in Industry 4.0 Scenario

Submitted in partial fulfilment of the requirements for the award of the degree of

Doctor of Philosophy

by

Nikhil M Thoppil

(Roll No. 701613)

Supervisor (s):

Dr. V Vasu

Prof. C S P Rao

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL

2022

My Humble Effort I Dedicate to My Family and Teachers

This Thesis entitled

Studies on the Development of Data-Driven Intelligent Predictive Maintenance Framework for the Critical Components of Machine Tool Systems in Industry 4.0 Scenario

by **Nikhil M Thoppil** is approved for the

degree of Doctor of Philosophy

Examiners

____

____

____

Supervisor (s)

____

____

Chairman

____

Date: _____________

Abstract

Predictive maintenance of machine tools is gaining wide attention in the manufacturing sector for achieving higher production rates and closer tolerance of machined parts. Due to the continuous operation of machine tools and the nature of work performed on it, wear and tear occur on the sliding and rotating components causing gradual mechanical damages. These mechanical damages on critical machine tool components adversely affect the quality of machined products and overall production efficiency. The present work investigates the development of an intelligent predictive maintenance framework for the critical components of machine tool subsystems.

The investigation first proposes a scientific methodology for the criticality analysis of machine tool systems employing a fuzzy modified failure mode, effects, and criticality analysis (fuzzy FMECA) for the maintenance prioritization of computer numerical control (CNC) lathe subsystems. The subsystems with a higher risk of failure and causing longer downtimes are considered for predictive maintenance. The lathe spindle unit is identified as the most critical subsystem with the highest estimated risk value of 848.2. The information on potential failure modes of components is utilized for the sensor selection in condition monitoring of machines. Following, an accelerated run-to-failure lathe spindle experimental test rig is fabricated to acquire machine tool health degradation data for analyzing the intelligent prognostic regression models for predictive analytics. Vibration signals representing machinery health degradation patterns are collected from critical lathe spindle locations. Vibration signature features in time, frequency, and time-frequency domain revealing superior machine degradation patterns are employed for training the prognostic algorithms. A neighborhood component analysis (NCA) based feature weighting scheme is used for selecting the most relevant features for regression analysis. Data conditioning and data selection can avoid underfitting and overfitting of the prognostic models.

Data-driven prognostics regression algorithms are observed efficient for the machinery health prognostics and Remaining Useful Life (RUL) estimations. Long Short-Term Memory (LSTM)/Bidirectional-LSTM (bi-LSTM) deep neural network models, Support Vector Machine (SVM) machine learning model, and exponential degradation statistical estimator model are utilized to develop intelligent predictive models for the prognostic analysis of CNC lathe spindle unit. However, optimizing the network structure and hyperparameters of the learning algorithm is a major challenge in the implementation of learning algorithms for predictive analytics. Bayesian optimized machine learning and deep neural network algorithms are proposed for the predictive analytics of the CNC lathe spindle unit. LSTM/bi-LSTM deep neural architecture-based prognostic algorithms are promising computational techniques for predictive maintenance and RUL estimation. LSTM/bi-LSTM networks and their combination network architectures are explored to evolve intelligent predictive models for the RUL estimation of lathe subsystems. Prediction accuracy of the evolved predictive models for estimating RUL of the lathe subsystems is evaluated using root mean square error (RMSE) and mean absolute percentage error (MAPE). The LSTM + bi-LSTM network architecture is identified to have the best prediction accuracy on lathe spindle RUL estimation with RMSE equals 31.65 and MAPE equals 4.45%. Further, this LSTM + bi-LSTM intelligent predictive model is chosen to develop a real-time IoT-based cloud analytics paradigm with a remote maintenance decision-making dashboard.

Keywords: *Industry 4.0, Internet of Things (IoT), Predictive Maintenance, Remaining Useful Life (RUL), Vibration Signal, Computer Numeric Control (CNC) Lathe; Failure Mode Effects and Criticality Analysis (FMECA), Deep Learning, Long Short-Term Memory (LSTM), Machine Learning, Hyperparameter Optimization.*

Contents

List of Figures

List of Tables

Abbreviation Notation and Nomenclature

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
bi-LSTM	bidirectional LSTM
CBM	Condition-Based Maintenance
Cm	Failure Mode Criticality
CNC	Computer Numerical Control
CNN	Convolution Neural Network
CPMT	Cyber-Physical Machine Tool
CPS	Cyber-Physical System
Cr	Component Criticality
D	Detection
DAQ	Data Acquisition
DBN	Deep Belief Network
FFT	Fast Fourier Transform
FI	Feature Indices
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
FRF	Frequency Response Function
GPR	Gaussian process regression
HI	Health Indicator
IIoT	Industrial Internet of Things
IoT	Internet of Things
KNN	k-Nearest Neighbor
LSTM	Long Short-Term Memory
M2M	Machine-to-Machine
MAE	Mean Absolute Error

MAPE	Mean Absolute Percentage Error
MQTT	Message Queue Telemetry Transport
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
NCA	Neighborhood Component Analysis
O	Occurrence
PCA	Principal Component Analysis
PHM	Prognostic Health Management
REST	Representational state transfer
RFID	Radio Frequency ID
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RPN	Risk Priority Number
RSSq	Root Sum of Square
RUL	Remaining Useful Life
S	Severity
SVM	Support Vector Machine
SVR	Support Vector Regression
TCM	Tool Condition Monitoring
TTF	Time-To-Failure

NCA_{obj}	NCA objective function
λ	NCA Regularization parameter
P_i	Probability
X_i	Feature vector
x_i	Actual RUL
y_i	Predicted RUL
w	Feature weight
n	Number of time-steps in datasets

N	Number of failure runs/Experimental dataset
$C_{(t)}$	LSTM long-term state
$H_{(t)}$	LSTM short-term state
$X_{(t)}$	LSTM previous output
f_t	LSTM forget gate
i_t	LSTM input gate
o_t	LSTM output gate
g	LSTM gate activation function
σ	LSTM input activation function
h	LSTM output activation function
U_i	Input gate weight associated with $X_{(t)}$
V_i	Input gate weight associated with $H_{(t-1)}$
W_i	Input gate weight associated with $C_{(t-1)}$
b_i	Bias weight vector
U_f	Forget gate weight associated with $X_{(t)}$
V_f	Forget gate weight associated with $H_{(t-1)}$
W_f	Forget gate weight associated with $C_{(t-1)}$
U_o	Output gate weight associated with $X_{(t)}$
V_o	Output gate weight associated with $H_{(t-1)}$
W_o	Output gate weight associated with $C_{(t-1)}$
$o_{(t)}$	LSTM Previous memory cell state
$l_{(t)}$	LSTM current layer input
$\overrightarrow{H_{(t)}}$	Forward LSTM hidden state
$\overleftarrow{H_{(t)}}$	Backward LSTM hidden state
$W_{\vec{h}}$	Forward LSTM layer weights
$W_{\overleftarrow{h}}$	Backward LSTM layer weights
b_y	Output layer bias weight vector
$y_{(t)}$	bi-LSTM layer output
χ	Bounded feature domain
x_{best}/x_{next}	Target best/next Bayesian optimization point

$Acq(x)$	Acquisition function
$\underset{x \epsilon \chi}{argmin}(f)$	Value of x in χ for the minimum value of f(x)
$\underset{x \epsilon \chi}{argmax}(Acq)$	Value of x in χ for the maximum value of Acq(x)
$S_n(t_i)$	Normalized time-series data
$s(t_i)$	Time-series dataset
mean	Mean of training dataset
std	Standard deviation of training dataset
H_n	Number of hidden units
FC_n	Number of fully connected layer
D_r	Dropout rate
Epoch	Maximum Epochs
L_r	Initial learning rate
$L2_{rf}$	L2-regularization factor

Declaration

This is to certify that the work presented in the thesis entitled **Studies on the Development of Data-Driven Intelligent Predictive Maintenance Framework for the Critical Components of Machine Tool Systems in Industry 4.0 Scenario** is a bonafide work done by me under the supervision of **Dr. V Vasu** and **Prof. CSP Rao** and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

[illegible]

Nikhil M Thoppil
Roll No. 701613

Date: 07/03/2022

Chapter 1

Introduction

1.1 Background

Manufacturing enterprises face a constant demand for increased productivity and improved product quality at minimum costs of production and maintenance. Manufacturing systems are now designed as highly sophisticated machine tools with advanced Computer Numerical Control (CNC) support to be in phase with the global competition. The reliability of manufacturing systems is revealed by the operating condition of the functional components and subsystems of machine tools. As the manufacturing industry continues to adopt more digital technologies the shortage of an appropriate machine tool maintenance strategy could prove adverse [1]. Due to the continuous operation of machine tools and the nature of work performed on it, wear and tear occurs on the sliding and rotating components causing gradual mechanical damages. These mechanical damages on critical machine tool components adversely affect the quality of machined products and productivity [2]. An efficient machinery health management strategy for the machine tool systems is necessary to withstand the manufacturing industry at zero unexpected failure rates and minimal machinery downtimes [3, 4].

Machinery health management is the practice of keeping the machines in proper functioning condition to produce quality products with maximum efficiency. Machinery health management primarily involves the maintenance of industrial equipment to maximize asset availability ensuring the quality of products manufactured or services offered and ensuring a safe working environment for their workforce [5]. Industrial maintenance has evolved over time, starting from fundamental reactive maintenance through preventive maintenance, and has come to condition-based maintenance (CBM) [6, 7]. The reactive or corrective maintenance strategy is an unplanned maintenance approach where the machine is allowed to operate until failure and then restoring. This maintenance approach can result in unexpected equipment downtime causing severe production loss. Preventive maintenance is a planned maintenance strategy where time-based or periodic maintenance actions are scheduled in advance to prevent failure. However, this maintenance approach causes redundant maintenance activities causing unnecessary expenditures. Reactive and preventive maintenances approaches consume time and resources which otherwise could be utilized for production [1, 7]. Figure 1.1 illustrates the reactive and preventive maintenance strategies for the machinery performance index against service life. In reactive maintenance, the repair activities are performed only after the machine degrade beyond the prescribed service line where it fails to perform its intended functions. The reactive maintenance strategy also performs rehabilitation activities to restore the machine to its initial working condition once it runs into a catastrophic failure. In preventive maintenance, a periodic target line is defined as reaching which the machine is subjected to periodic maintenance activities to restore it to its initial working condition. This maintenance strategy ensures the machinery to operate always in the prescribed service life state.

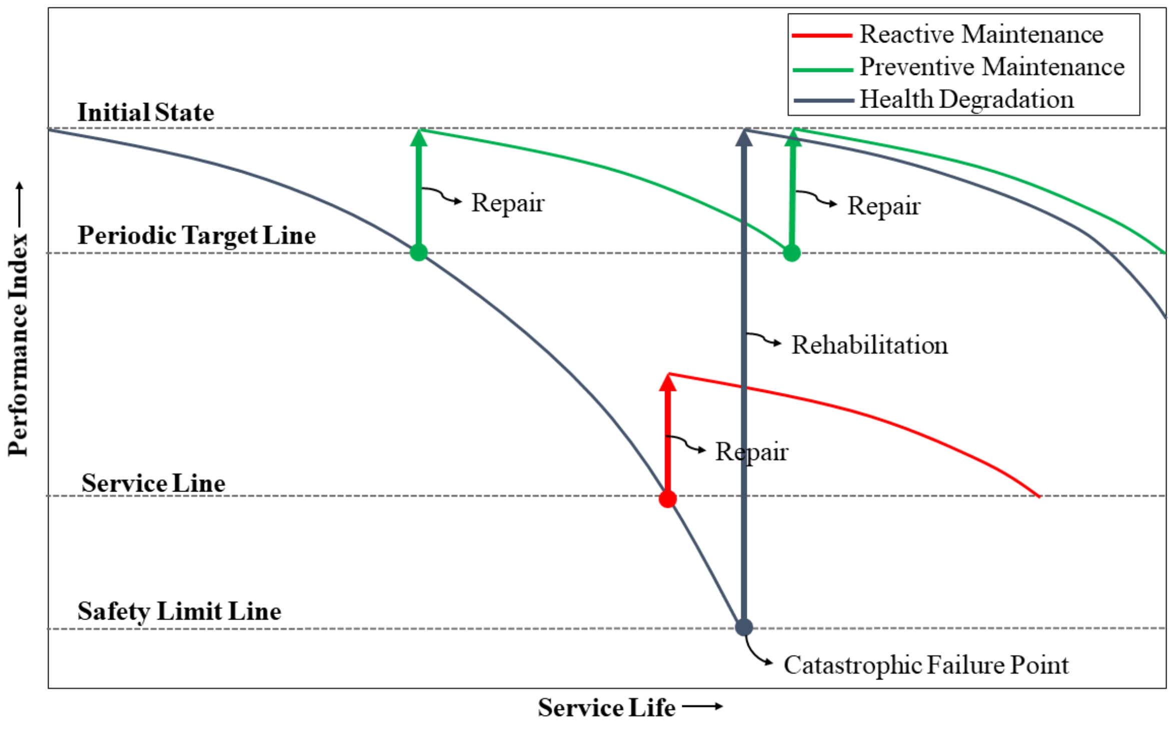


Figure 1.1 Illustration of conventional industrial machinery maintenance strategies on machinery performance index and service life

CBM is a maintenance strategy that monitors the actual condition of an asset for deciding when what maintenance needs to be done. CBM is based on the condition monitoring data collected from operating machinery [8]. Predictive maintenance is an advanced version of CBM, where data analysis tools and intelligent computational techniques are used to predict any upcoming failures far before their actual point of occurrence. Thus, the predictive maintenance approach allows users to well-plan the maintenance activities to perform the right maintenance action at the right point in time with minimal production loss and expenditure [9].

The scope of implementing predictive maintenance of manufacturing systems contributes to enhancing the controls, costs, and quality of production. The predictive maintenance paradigm belongs to Industry 4.0 that it is propped up by several technological enabling factors including a wide range of sensors capable to register any source of information (vibrations, acoustic emissions, temperature, etc.) send from operating machinery, advanced computational resources for analyzing the acquired data, modern Internet of Things (IoT) enabled remote connectivity means, and big data cloud storage and computing technologies that provide real-time update of machinery information for the prognostic analysis [10].

Predictive maintenance generally uses historical and real-time machine health degradation information to recognize the equipment failure patterns and further this information is used to forecast upcoming failures. The predictive maintenance strategy uses condition monitoring tools to track the performance of the machine functional components. In any mechanical system, the machine operating condition is monitored using time-series data such as vibration signal, shock-pulse, acoustic emission, bearing temperature, oil debris, oil pressure, and electric current variations. This involves the application of different sensors, data acquisition systems, data processing, and computation techniques [11]. This causes huge installation costs of sensors, data acquisition, and computational algorithms, and therefore predictive maintenance is limited to the most critical subsystems of the machinery. Hence, the identification of the most critical components of the machine system and associated failure modes are the pre-requisite for employing predictive maintenance [12]. The information on potential failure modes associated with the critical subsystems is necessary for the selection of the most suitable sensors for condition monitoring. The machinery log file information that records either event of machine failure, downtime, restoration, and costs involved are analyzed using failure modes and criticality analysis tool to identify the potential failure modes of machinery functional components and the risk associated [13, 14].

Machine condition monitoring makes the primary component of predictive maintenance. Sensors are used to measure specific machine operating conditions to observe any sign of irregularity that would affect the normal operation of the machine. The machine condition monitoring data provides both diagnostic and prognostic information, like the fault, location of the fault, causes of the fault, and upcoming failure time. The condition monitoring information can also be used for evaluating the quality of products manufactured, mainly its dimensional tolerances and surface quality. Machine vibration monitoring is the most widely employed condition monitoring parameter for mechanical systems. The machine health degradation patterns are extracted for the vibration signals for machinery failure prognostic analysis [6, 15, 16].

Machinery health prognostics primarily perform the estimation of Remaining Useful Life (RUL) or Time-to-Failure (TTF) by analyzing the current operating condition of the machine against the historical machine failure trend pattern. The health prognostics techniques are classified as physics-based approaches, statistical model-based approaches, Artificial Intelligence (AI) approaches, and hybrid approaches [6, 17, 18]. The physics-based approach demands a thorough knowledge of the physics of failure mechanism, which is difficult to execute for complex machinery. Statistical model-based and AI approaches are data-driven approaches, which utilize machinery health degradation data for prognostic analysis. Statistical model-based approaches require only empirical knowledge to establish a relationship between the failure mechanics and statistical model, whereas the AI models use minimal technical aspects of the system. AI approaches use intelligent learning algorithms to learn machinery health degradation patterns, but its implementation was not popular due to the black-box nature of the learning process and the requirement of computers with high computational power. The data-driven approaches use machinery health degradation information to train intelligent predictive models, which are used for RUL estimation of the considered machine [19, 20].

The statistical model-based approaches are the most popularly used prognostic technique for decades. Random coefficient models, autoregressive models, Wiener process models, Gamma process models, inverse Gaussian process models, Markov models, proportional hazards models, exponential degradation model, etc. are the most widely used statistical models for machinery health prognostics [17]. AI approaches have been receiving increasing attention with the advancements in computational capabilities and their superiority in dealing with prognostics problems of complex mechanical systems. Machine learning and deep learning AI models are very popular in machinery health diagnostic and prognostic analysis. The most recognized machine learning and deep learning architectures for failure prediction and RUL estimation include artificial neural network (ANN), neural fuzzy systems, support vector machine (SVM), support vector regression (SVR), k-nearest neighbor (KNN), Gaussian process regression (GPR), recurrent neural network (RNN), long short-term memory (LSTM), deep belief network (DBN), convolution neural network (CNN), etc. [21, 22]. Though the technical advancements encourage the implementation of an AI algorithm for prognostics, hyperparameter optimization and network architecture selection are major challenges before the successful utilization of these techniques. The hyperparameter optimization including both structural and training parameters has a direct influence on the accuracy of the RUL estimation models. The computational search algorithms like grid search, random search, Bayesian search optimization, etc. are employed for the hyperparameter optimization to reduce the computational complexity and improve prediction accuracy [23, 24].

The appealing contribution of the latest technological advancements to the industry is the Industrial Internet of Things (IIoT), which establishes an efficient communication paradigm between the various industrial machinery, systems, and users. IIoT utilizes advanced sensor technologies supporting the IoT, cloud space, and cloud computing facilities for intelligent computational algorithms. These technological advancements can be used to augment the predictive maintenance paradigm for industrial machinery [9, 25]. A maintenance decision support system with a maintenance decision-making dashboard and failure warning system can be offered for the industrial sector. The system is also capable of providing remote access to the industrial machinery health status information and control of industrial activities [26].

Predictive maintenance is not a substitute for traditional maintenance approaches, rather it is considered as a value addition to the total industrial production management. It cannot totally wipe out the need for traditional reactive or preventive maintenance approaches. In any industry, the user should identify if the predictive maintenance suits the machinery concerning the huge installation costs and the fact that only those component faults that can be monitored using sensor technology could be considered [27].

1.2 Motivation of the Thesis

"*Maintenance is a necessary evil*" is a general opinion among industrialists. The maintenance costs constitute a major part of the total operating costs of all manufacturing systems. The loss of production time and product quality that result from an inefficient maintenance approach has serious impacts on industrial manufacturing enterprises. The advent of Industry 4.0 can take advantage of cyber-physical systems techniques to digitalize maintenance strategies and implement IoT-based data-driven automated remote-controlled operations. IoT-based intelligent decision support system for machinery health management can make maintenance scheduling and shop floor work allocations a facile task. Predictive maintenance is a credible solution for machinery health and product quality management. Instead of relying on industrial field failure data or in-plant average life statistics, predictive maintenance uses the machinery condition monitoring data like component vibrations, temperature, acoustic emissions, etc. to obtain insights on the actual operating condition of the manufacturing system. Advancements in sensor technology and IoT connectivity open a wide scope for real-time machinery condition monitoring data generation and data acquisition from anywhere. In the manufacturing sector, predictive maintenance of machine tools is gaining wide attention to bring about higher production rates and closer tolerance of machined parts. However, the additional instrumentation causing huge installation costs and the complexities involved in computational tools pull back the industrialist from implementing predictive maintenance. Data-driven prognostics approaches like AI techniques require minimal technical knowledge on the machinery operations and their failure mechanism. Deep learning is a promising AI computational tool for machinery health prognostics and RUL estimation. However, the implementation of deep learning algorithms for machinery health prognostics involves many challenges.

1.3 Scope of the Work

This thesis work aims to employ intelligent data-driven computational techniques for the prognostic analysis and predictive maintenance of critical machine tool systems. The research aims to investigate the challenges faced during the implementation of a predictive maintenance strategy in the manufacturing industry for their machine tool systems. The study initiates with the criticality analysis of a CNC lathe machine tool for the maintenance prioritization of critical components and potential failure mode identification followed by sensor selection and configures data acquisition system for machine health degradation data recording. Further, data-driven techniques are employed to extract machine health degradation patterns to evolve intelligent predictive models. However, the implementation of intelligent data-driven techniques like deep learning algorithms for machinery health prognostics has enormous challenges, which have a certain scope for discussion in the present era of Industry 4.0. The mandate to have a large size machinery failure data for training deep learning algorithms is regarded as a major limitation. Hyper-parameter optimization, architecture design and data training of deep learning algorithms are still challenging and unpredictable, which can pull back industrialists from implementing intelligent health management of industrial machinery. This research work aims to unveil the black-box nature of deep learning algorithms to make an intelligible prognostic platform with automated hyper-parameter selection to instigate industrialists to set about an autonomous machinery health management system. The primary scope of this thesis is to motivate industrial practitioners to develop an autonomous machinery health management system consisting of the machine health degradation data acquisition system and intelligent prognostic model training algorithms. Such a system improves the overall industrial value and thus fits itself into the present Industry 4.0 era.

1.4 Organization of Thesis

Chapter 1 introduces the research problem. A detailed review of literature is presented in chapter 2 highlighting the gaps areas identified for further research and the aim and objectives of the present study. Chapter 3 discuss the criticality analysis and maintenance prioritization of the CNC lathe machine tool. Chapter 4 details the fabrication of an accelerated run-to-failure experimental setup for time series machinery health degradation data acquisition and data processing to extract useful information on machine failure. Chapter 5 presents the implementation of data-driven prognostic algorithms for machinery health prognostics and RUL estimation. The chapter also discusses a comparison of popular data-driven prognostic models for machinery RUL estimation. Chapter 6 presents the implementation of IoT based remote maintenance decision-making dashboard with cloud data analytics. Chapter 7 presents the summary and conclusions of the entire study and the scope for future work.

Chapter 2

Literature Survey

2.1 Introduction

In recent years, the increasing demand for higher production rates and product quality has led to the induction of more sophisticated machines, which in turn has increased the requirement of more effective maintenance strategies to ensure the overall performance of industrial systems. The evolution of industrial maintenance strategies has reached an intelligent predictive maintenance concept, which detects the upcoming failures in an industrial system far before their actual occurrence. The change has also affected the manufacturing sectors and the machine tool systems, the operating condition of which reflects the production rates and quality of the machined parts. This chapter reviews the predictive maintenance of mechanical systems with respect to machine condition monitoring and data acquisition, and the computational analysis focusing on the data-driven approaches. The machine tool failure mode identification and criticality analysis are discussed for the application of maintenance prioritization. The data-driven computational approaches including statistical estimator models, machine learning approaches, and deep learning approaches are discussed for machinery prognostic regression analysis. Furthermore, the chapter reviews the IoT-based maintenance decision support system for industrial machinery. The chapter ends with discussions on the gap areas in literature, the aim and objectives of the present work, and a flowchart of the present work.

2.2 Machine Tool Maintenance

Machine tool technology is the backbone of the manufacturing industry. The machine tools are often operated in the manufacturing sectors without a planned maintenance strategy which is usually confined to cleaning the equipment and lubricating the moving parts. Such a maintenance approach can result in low-grade products, unplanned downtimes, and catastrophic failures causing huge production loss and a precarious working environment. The prime motive of planned maintenance is to keep the machine tool vibrations within acceptable limits thus ensuring a good operating condition [5, 6]. According to ISO 230 (2012), machine tools should mitigate the types of vibration that produce undesirable effects to avoid *"unacceptable cutting performance with regard to surface finish and accuracy, premature wear or damage of machine components, reduced tool life, unacceptable noise level, physiological harm to operators"* [8].

The CNC machine tools are more complex and sophisticated industrial systems that play a significant role in the modern manufacturing industry. These machine tools perform shaping or machining operations usually, by turning, milling, boring, grinding, shearing, or other forms of deformation. Due to the continuous operation of machine tools and the nature of work performed on it, wear and tear occurs on the sliding and rotating components causing gradual mechanical damages [28]. These mechanical damages on critical machine tool components adversely affect the quality of machined products. The increasing demand for machining quality and closer tolerance of machined parts has raised the need for a systematic maintenance approach for the machine tool systems [4, 29]. A fault-free operating condition of machine tools is of high priority in the manufacturing industry [30, 31]. There has also been a simultaneous growth of industrial technologies that hasten the need for an efficient maintenance approach. The advent of Industry 4.0 has driven the manufacturing industry to achieve a new generation of machine tool systems termed as Cyber-Physical Machine Tool (CPMT) or Machine Tool 4.0, which are intelligent, well connected, extensively accessible, and more adaptive and autonomous in operation, control, and maintenance [4, 32, 33].

The reliable performance of machine tool systems is largely dependent on the inherent reliability of its functional components and maintenance program adopted. The peripheral factors like production planning and work scheduling also influence the machine tool's reliability. On the component level, functional components and subsystems of the CNC machine tool have a considerable influence on its reliability level [3]. The operating condition of each subsystem like CNC unit, spindle unit, linear axis feed drive, hydraulic system, etc. contributes to the overall performance of the machine tool. In a machine tool system, the mechanical failures bring about the major cause of downtime, while the electrical failures are more frequent, though the downtime caused is comparatively very less [29, 34, 35]. The cutting tool wear is another common failure phenomenon in machine tool systems. However, these failures do not cause catastrophic failures of machine tool systems, but highly influence the quality of machined products [36].

The research paper by Harris et.al. [37] is one of the oldest publications on condition monitoring and fault detection of machine tools. Martin [38] has published a review on condition monitoring and fault diagnosis in machine tools and discussed its primary concept and applications. Drake et.al. [39] developed a data acquisition system for machine tool condition monitoring. Saravanan et al. [40] have performed failure data analysis for the condition monitoring of lathe, milling, and grinding machine tools. The author later discussed the condition monitoring of lathe spindle units based on vibration, acoustic emission, surface roughness monitoring, and the Shock Pulse Method (SPM) [34]. Liang et al. [41], Kim et al. [42], Atluru et al. [43] discussed the evolution of process monitoring and control technologies for machine tool systems. Liao et al. [44] and Jay Lee et al. [45] utilized Watchdog Agent® prognostic toolbox for the automatic tool changer and spindle bearing health status prediction integrating both CNC controller data and sensor data. Soft computational data mining techniques [46-48] and finite elements models [49-52] were preferably employed for fault diagnostics and prognostics of machine tool system failures. Statistical data analysis techniques for criticality analysis were used for machine tool failure analysis [2, 3, 53]. Comparatively only a few researchers worked on condition monitoring and maintenance of machine tool functional components. Most research works on advanced machine tool maintenance strategies are focused on cutting tool condition monitoring and cutting tool wear predictions [54, 55].

Recently, a few researchers have studied the development of an intelligent maintenance strategy for machine tool systems [55-57]. The functional components of CNC machine tools like spindle unit [58-60], spindle bearings [61, 62], linear axis feed drive [63, 64], ball screw feed drive [29], transmission system [65], and automatic tool changer [66, 67] are critically investigated for fault detection and diagnostics. The current trend in machine tool maintenance is based on advanced predictive maintenance systems with real-time machine tool monitoring and failure predictions [3, 4, 61, 67-71]. Research in this area is necessary to ensure increased productivity, improved product quality, reduced costs, and to keep the manufacturing technology up-to-date in this fast-growing world.

2.3 Evolution of the Maintenance Strategies

Maintenance can be defined as "*a combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in or restore it to, a state in which it can perform the required function*" (EN 13306-2010) [72]. The primary objective of maintenance is to maximize the asset availability ensuring the quality of products manufactured or services offered. Efficient maintenance of industrial systems also ensures a safe working environment for their workforce.

Industrial maintenance has evolved over time, starting from the fundamental reactive maintenance where any repair or replace actions are initiated only after the occurrence of failure, through the preventive maintenance in which maintenance actions are performed at regular intervals to avoid upcoming failures determined based on statistical reliability analysis, and has come to the CBM where the maintenance is done when a condition monitoring indicator goes over a predefined threshold [30]. The reactive or corrective maintenance strategy is an unplanned maintenance approach where the machine is allowed to operate until failure and then restoring. This can only be considered for less critical systems and only if the consequences of failure are affordable. The preventive maintenance or time-based or routine or periodic maintenance are planned maintenance approaches where the maintenance actions are scheduled in advance to prevent failure. The maintenance is usually determined based on the machine operating manual or operator's experience. The CBM is based on the machine condition monitoring information that indicated machine failure or deterioration. CBM has the primary focus on failure prevention and functionality of its components but also has a secondary focus on the quality of manufactured products or machine operations. A predictive maintenance strategy can be regarded as an advanced version of CBM where the condition monitoring data is used to predict any upcoming failures far before their actual point of occurrence [30, 31]. Table 2.1 summarizes the main characteristics of all the aforementioned maintenance strategies.

Table 2.1. Main characteristics of all maintenance strategies

Maintenance Strategy	**Reactive Maintenance**	**Preventive Maintenance**	**CBM**	**Predictive Maintenance**
Maintenance Frequency	Depends on component failure	Fixed on basis of statistical reliability analysis	Based on condition monitoring data	Determined by prognostic regression analysis
Complexity and Technological Requirements	Low	Medium	High medium	High
Human Intervention Requirements	High	High Medium	Medium	Low

CBM has been the most investigated maintenance strategy by both the research community and industries for machine tool maintenance [31] The discussion mostly includes the selection of the most suitable sensors for machine tool vibration and temperature monitoring with available data analysis techniques, and no consideration of the predictive prognostic analysis was reported. Over a half-decade, researchers had been enthusiastic about the prognostic health management (PHM) concept for industrial machinery, which includes research works on monitoring and analyzing the current health status of the machine and analyzing past machine failure data to predict future machine failures. Implementing the predictive maintenance concept can contribute to reduced maintenance associated costs, increased production time, improved product quality, and reduced risk of catastrophic failures [5, 28].

The predictive maintenance paradigm belongs to Industry 4.0 that it is propped up by several technological enabling factors including a wide range of sensors capable to register any source of information send forth an operating machine (vibrations, acoustic emissions, temperature, etc.), advanced computational resources for analyzing the collected data, modern IoT enabled remote connectivity means, and big data cloud storage and computing technologies that provide real-time update of machine information for prognostic analysis [33, 73].

2.4 Predictive Maintenance

Predictive maintenance strategy allows the industries to have the minimal amount of annual maintenance activities that would be required to keep the machine in its peak operating condition and avoid chances of any unexpected catastrophic failure or downtime. Reactive and preventive maintenances approaches consume time and resources which otherwise could be utilized for production [7]. Predictive maintenance primarily detects early signs of failure in the functional components of machinery and then initiates necessary maintenance actions at the right time. A predictive maintenance strategy is supported by condition monitoring and prognostics algorithms, which perform the analysis of machinery health degradation. The machine condition monitoring data provides both diagnostic and prognostic information, like the fault, location of the fault, causes of the fault, and upcoming failure time. The condition monitoring information can also be used for evaluating the quality of products manufactured, mainly its dimensional tolerances and surface quality [74]. Predictive maintenance generally uses historical and real-time machine health degradation information to estimate the RUL or TTF of the equipment. RUL is the subjective estimate of the length of time a machine can perform its intended functions [9]. The failure of a machine is defined as the point at which it is not able to perform its intended functions in the designed manner. The failure rates throughout the lifetime of a mechanical system are graphically represented on a failure rate vs time plot namely, the bathtub curve as shown in Figure 2.1.

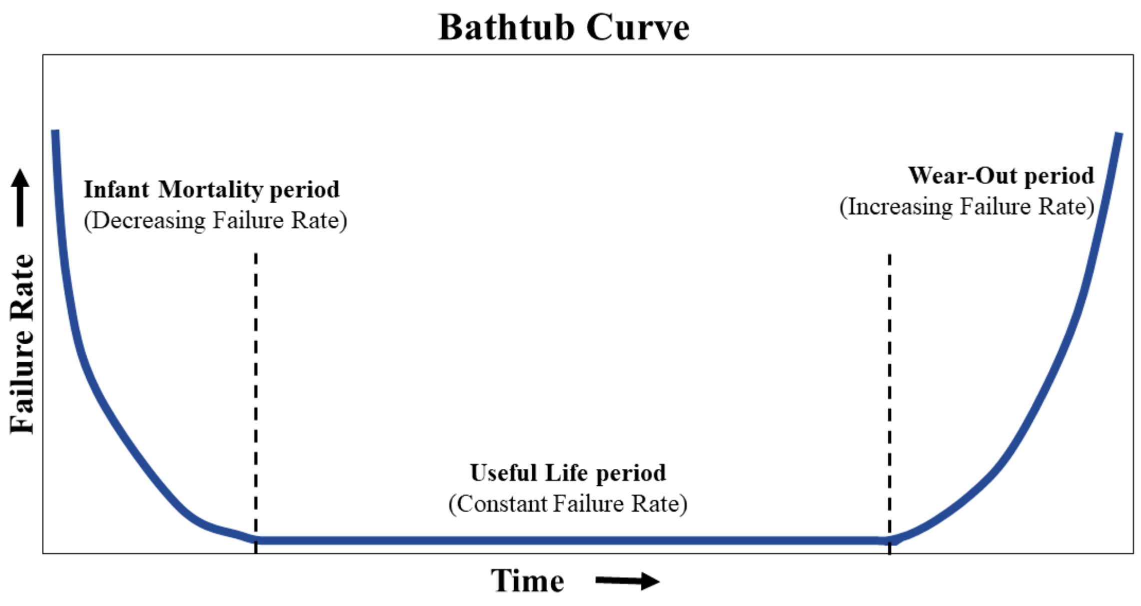


Figure 2.1 Bathtub Curve

The bathtub curve is divided into three regions: early infant mortality period, useful life period, and wear out period. The early infant mortality period is characterized by a high but rapidly decreasing failure rate. The failures in this period are due to minor design and manufacturing flaws. The useful period has a long period of constant failure rate. This period is the best operating lifetime of the machine. In the final wear-out period, the mechanical components begin to wear out approaching the failure. The failure rate rapidly increases at this period of machine lifetime [75].

The predictive maintenance approach consists of three main steps; data acquisition, data processing, and maintenance decision-making. The continuous machine monitoring data gives insight into the changing health status of the machine. The failure trends of a particular machine are deduced from the available historical machine monitoring data. The machine failure trends and real-time machine health status are fed into a prognostic analysis tool to determine the TTF or RUL that helps in maintenance decision making and scheduling [76, 77]. Figure 2.2 shows the flow diagram of the predictive maintenance approach.

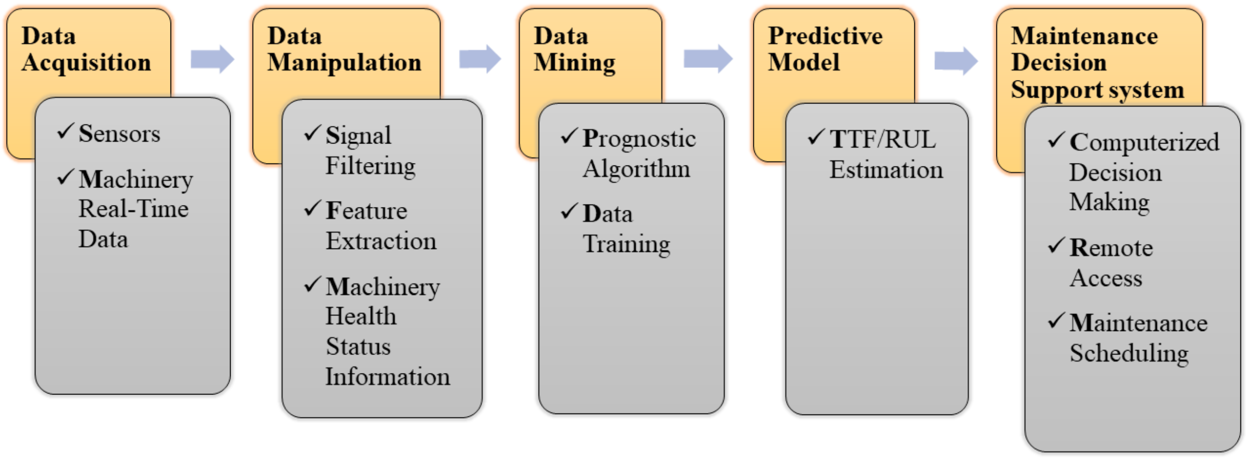


Figure 2.2 Predictive maintenance flow diagram

The concept of condition monitoring and predictive maintenance dates back to decades when experienced maintenance persons use their senses of seeing, hearing, smelling, and touching to detect an early sign of failure and initiate necessary maintenance actions. This expertise-based technique though very old is still being practiced as it is still admired in many situations. The current predictive maintenance approach with the technological substructure got its true application in the industrial world only since the 1990s, which due to the unavailability of suitable sensors and data acquisition systems, and high complexities involved discouraged the industrialist and research community to consider this maintenance approach [7, 78]. The advancements in sensor technology and computational capabilities have made these discouraging factors trivial before the growing demand for efficient maintenance approaches [74]. With the arrival of Industry 4.0, many industries are eager to establish an intelligent predictive maintenance strategy for their machinery [79, 80]. The predictive maintenance approach avoids the situation of over-maintenance or under-maintenance of industrial equipment. It is not a substitute for traditional maintenance approaches, rather it is considered as a value addition to the total industrial production management. It cannot totally wipe out the need for traditional reactive or preventive maintenance approaches. In any industry, the user should identify a suitable maintenance strategy either reactive, preventive, or predictive based on the critical requirement [33, 80].

Implementation of predictive maintenance is a complex proceeding for the industry since it involves meticulous planning of hardware, software, and personnel requirements. Considering the machinery faults, only those machinery faults that can be monitored using sensor technology could be considered. Also, predictive maintenance required a huge installation cost, the user should identify which machinery or machine subsystem should be covered in the predictive maintenance strategy. The general requirements for successful implementation predictive maintenance are summarized as follows:

i. Identify the critical components to be monitored
ii. Identify the parameters that indicate deterioration of component
iii. Selection of suitable sensors and data acquisition system
iv. Selection of suitable condition monitoring techniques and critical thresholds for the monitored parameter
v. Selection of suitable computational algorithm to perform prognostic analysis
vi. Efficient computerized maintenance decision support system [73, 74, 81]

2.5 Machine Criticality Analysis

Although predictive maintenance is getting wide acceptance, its realization in manufacturing sectors requires huge installation costs for sensors technology and intelligent computational algorithms, and therefore, it is limited to the most critical subsystems of the machinery [74, 82]. The information on potential failure modes associated with these critical subsystems is necessary for the selection of the most suitable sensors for condition monitoring.

On account of these factors, the identification of the most critical components of the machine system and associated failure modes makes the prerequisite for employing predictive maintenance. Recently, a few research studies have focused on establishing the necessity of criticality analysis of a mechanical system to support maintenance decision making, pointing out the lack of strong machine criticality analysis methodologies in the industry for maintenance prioritization [83-86]. Gopalakrishnan et al. [12] were critical of the traditional maintenance prioritization practices in the industry, which is operator influenced and thus non-factual. He has also interpreted the connection between machine criticality and maintenance prioritization in an industrial context for productivity improvement.

Recently, many researchers have employed failure mode and effects analysis (FMEA) and failure mode, effects, and criticality analysis (FMECA) techniques for the investigation of potential failure modes and reliability-centered maintenance of machine tools [87-93]. Lo et al. [87] introduced a risk assessment framework for the manufacturing of machine tools using a modified FMEA technique. Gupta et al. [88] presented reliability-centered maintenance with fuzzy FMEA for a milling machine. FMEA was used to identify critical failure modes of components and subsystems of the CNC turning center [89]. Wang et al. [90] used FMECA for CNC lathe with the criticality factor modified for considering the cost required for reducing failure rates. Du et al. [91] presented FMECA of a remanufactured machine tool with a case study of the hobbing machine. Zhou et al. [92] presented a reliability allocation method based on the cubic transformed functions of FMEA. Kim et al. [93] presented a reliability assessment of machine tools using FMEA with a case study of the machining center, which includes web-based main-axis vibration data analysis program and a failure mode estimation algorithm.

FMEA and FMECA are tools designed to identify potential failure modes for a system or process, to determine the risk factor associated with failure modes. These risk factors are further represented on a relative scale for criticality analysis. The information about various failure modes and associated risk factors is used to identify and implement corrective measures for machine components in the order of risk priority. On the application level, FMEA might be termed as process FMEA, design FMEA, system FMEA, etc. [94], but the basic procedure remains the same. FMEA has a wide range of applications from equipment failure analysis to nuclear power product designs for the identification of different failure modes and risks associated with [94, 95]. FMEA is a 70-year-old technique, first introduced by the US Army and modified several times for improved analysis and specific applications [96]. Standards like

MIL-STD-1629A (1980), SAE-J-1739, and SMC REGULATION 800-31 were defined for implementing FMEA/FMECA techniques [88, 95] MIL-STD-1629A [97] is the most widely used standard in failure analysis using FMEA/FMECA. FMEA technique with added criticality analysis and ranking of failure modes or components is termed FMECA [98]. FMECA is a traditional approach adopted to improve the design and reliability of a system.

FMEA proceeds with the failure mode identification and calculation of Risk Priority Number (RPN). RPN is an indicator of the risk associated with the failure mode of the component. RPN is commonly calculated as the product of Severity *(S),* Occurrence *(O),* and Detection *(D)* [96, 97].

$$\text{RPN} = S \times O \times D \tag{2.1}$$

S, *O,* and *D* are indicated by values on a scale from 1 to 10. *S* is the indication of how severe is the cause of failure mode, *O* is the frequency of occurrence of the failure mode, and *D* is the non-detection rating of the failure mode. RPN can range from 1 to 1000, where minimum RPN 1 indicates the least risk priority, and the maximum RPN 1000 indicates the highest risk priority. RPN is used for risk prioritization of failure modes of components [96, 97].

FMEA with an added criticality analysis and risk prioritization of failure modes and components is termed as FMECA [98-101]. The procedures for performing conventional FMECA [100] are as follows:

i. Identification of various failure modes, their potential effects, potential causes, and machine controls for detection at the component level.
ii. Assigning *S*, *O,* and *D* ratings for each of the failure modes.
iii. Calculation of RPN from *S*, *O,* and *D* rating values.
iv. Classification of failure modes based on the criticality ranking.

S, *O,* and *D* ratings for CNC lathe machine failure are defined following the MIL-STD-1629A [97] guidelines and expert elicitation. Furthermore, to determine the RPN for each failure mode the Eq (2.1) is utilized, which takes the product of *S, O,* and *D.* The sum of the RPNs of each failure mode of a component gives the RPN of that particular component. The sum of the RPNs of all individual components under a subsystem gives the RPN of that particular subsystem. RPN is just a number having no units. It is always measured relative to the RPNs of other components of the system [96, 97]. These RPNs indicate the criticality of the CNC lathe components and subsystems, which are further utilized to prepare a criticality ranking for maintenance prioritization.

In certain applications, the criticality is calculated as the product of severity and occurrence [96]. Primarily, there are two approaches to determine the criticality of a failure mode of a component, qualitative analysis, and quantitative analysis. Qualitative analysis is used when the data available is limited or insufficient. Whereas, quantitative analysis is used when enough failure data of the system is available, and this data is used to calculate the criticality number. The failure data required for calculating criticality numbers include failure modes, failure rates, failure ratios, and failure affect probabilities. The method proceeds by calculating the failure mode criticality (Cm) for each failure mode followed by summing up all failure mode criticalities to obtain the component criticality (Cr) [100, 101]. The formulation is adopted from MIL-STD 1629A [97]. The failure mode criticality is calculated as;

$$Cm = \beta\alpha\lambda pt \qquad (2.2)$$

where β is the conditional probability of occurrence of failure mode, α is the Failure mode ratio, λp is the Part failure rate, and t is the total operating time.

Then, the component criticality is calculated as;

$$Cr = \sum (Cm) \qquad (2.3)$$

In spite of its successful implementation in an extensive range of applications, many researchers have criticized conventional FMEA/FMECA methodology pointing out a few drawbacks [96]. The following are the major drawbacks of FMEA/FMECA.

i. The concept of RPN calculation is an extension of the risk matrix defined in MIL-STD-1629A. There is no rationale for considering RPN as a product risk factor [87, 98].
ii. Different sets of *S*, *O*, and *D* give the same RPN. But in real practice, the risk associated may not be identical [99].
iii. There is the erroneous assumption that *S*, *O*, and *D* values have the same significance. This may not be reasonable in practical applications [102, 103].
iv. RPN is not continuous from 1 to 1000. The product of *S*, *O*, and *D* will never make a few values in this range. This creates serious interpretation problems [104].

Many researchers have presented various modifications to overcome the drawbacks of conventional FMEA [87-90, 98-107]. The fuzzy logic computational technique is extensively applied to improve FMEA/FMECA [98, 102-107]. The fuzzy logic computational technique is used to establish the correlation between *S*, *O,* and *D* with RPN. The fuzzy modified FMEA/FMECA is successfully implemented in various areas of risk assessment like LNG storage facility [98], purchasing process in a hospital [102], etching of an integrated circuit wafer [103], sterilization unit [104], aircraft landing system [105], emergency department in a hospital [106], medical product development [107], etc.

2.6 Machine Condition Monitoring

Machine condition monitoring makes the primary component of predictive maintenance [20, 108]. The advent of machine condition monitoring has positively influenced machine reliability management. The availability of monitored data in digital format opens a wide opportunity for the industries to redefine the limits of the smart manufacturing and maintenance paradigm [73, 109]. It is the procedure of measuring the specific machine parameters while in operation to observe any significant portent that could be indicative of an impending failure. It is a maintenance strategy where appropriate maintenance is done based on the operating condition of the machine. Condition monitoring of mechanical systems involves the continuous measuring of specific equipment parameters, taking note of any irregularities that would affect the normal operation of the equipment and lead to catastrophic failure of the system components. In any mechanical system, the machine operating condition is monitored using sensor data as a vibration signal, shock-pulse, acoustic emission, bearing temperature, oil debris, oil pressure, and electric current variations [110, 111].

Condition monitoring includes specific machine data monitoring and acquisition followed by data analysis to convert the monitored data into useful information indicating machine health status. In order to monitor specific machine data, respective sensors are mounted around the machine critical locations. A data acquisition system converts the raw signal read from the sensors into digital data and is stored for further analysis. Relevant information about the health status of the machine is extracted from this discrete digital data [112].

2.6.1 Vibration-Based Condition Monitoring

Machine vibration is one of the most suitable and trustable condition monitoring signals to know about the machine health condition. The amplitude of the vibration signal indicates the severity of the fault, while the frequency of the vibration signal indicates the source of the fault. In an operating machine, there are many forms of excitations in terms of time-varying forces and torques like forces due to unbalance, forces due to misalignment, dynamic forces at bearing locations, etc. [112]. In rotating machinery, the rotational speed of the machine corresponds to the excitation frequency or forcing frequency. The diagnostic analysis and fault detection of a machine involves the understanding of vibration system characteristics by measuring its transfer function or Frequency Response Function (FRF). FRF is the ratio of the response of the system to its excitation. In a mechanical system, the individual components are considered as a continuous system, whose natural frequencies can be estimated once their stiffness and mass are known. These frequencies mapped against the monitored vibration signal are used to determine the health status machine components [112-114]. The Fast Fourier Transform (FFT) of the monitored time-series vibration signal can detail all fault frequencies associated with the machine. FFT can define individual frequencies for the detection of faults like misalignment, cracked shaft, bowed and bent shaft, unbalanced shaft, looseness, rub, and bearing defects. Separate frequencies are established for bearing inner race, outer race, cage, and balls or rollers [115]. Similarly, fault frequencies are defined for gear faults [52, 116, 117]. The prognostics of a machine involves the understanding of trends portrayed by the condition monitoring signal over a long-term time span [6, 118]. Statistical feature extraction-based data mining techniques are popularly employed for vibration signal trend analysis [116, 119, 120].

The machinery vibration can be measured in terms of displacement, velocity, or acceleration. A simple dial gauge, linearly variable displacement transducer (LVDT), proximity sensor, capacitive probe, position potentiometer, etc. can be used for measuring displacement. A self-generating low-impedance vibration velocity transducer is used for linear vibration velocity measurements. Acceleration measurement is the most widely employed method for vibration monitoring. Acceleration measure is based on the measurement of relative motion of a suspended mass in a casing, where the casing is subjected to a motion. A sensing element is attached to the suspended mass, whose motion is calibrated to the provided measure of acceleration. Piezoelectric accelerometers use piezoelectric crystals as a sensing element, which is placed on the base and top of a mass on the accelerator. Piezoelectric crystals are sensitive to motion in a particular direction. Therefore, it is aligned along the most sensitive axis inside the accelerometer housing. The piezoelectric accelerometer is mounted on the surface of the machine to be monitored utilizing a wax, adhesive, magnet, or stud. Accelerometers are also available with a handheld probe.

Figure 2.3 shows popularly used piezoelectric accelerometers. The accelerometers can be uniaxial measures vibrations only in one direction or triaxial that measures vibrations in all three directions. The sensitivity of an accelerometer is defined after a suitable charge-to-voltage amplifier and is expressed as mV/ms-2. The amplitude of vibration measurement is expressed in g units, where ‘g’ is the acceleration due to gravity (9.8 ms-2) [112].

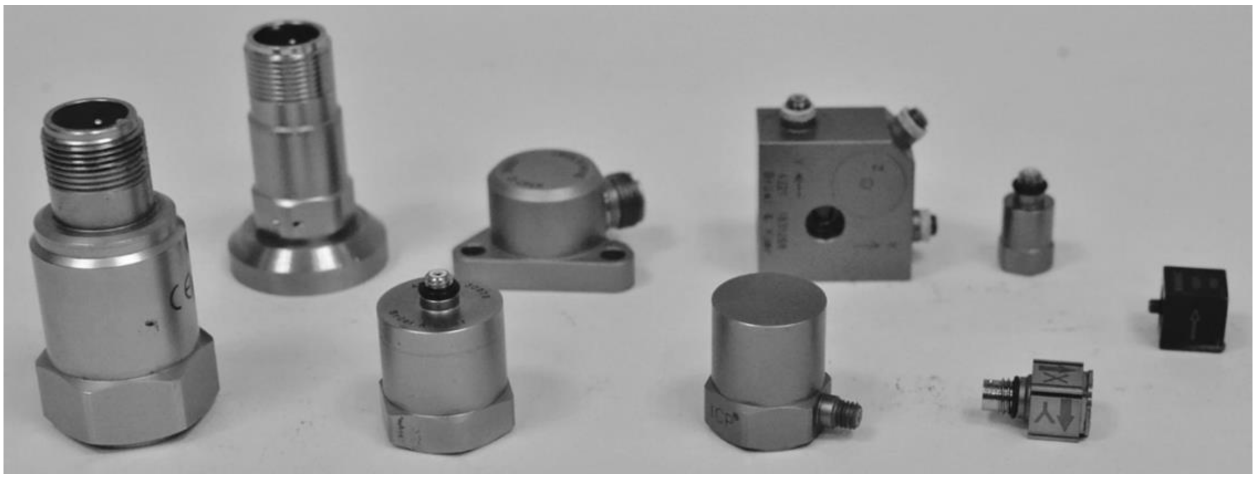


Figure 2.3 Accelerometers used for vibration measurement [112]

2.6.2 Vibration Data Acquisition and Storage

Sensors mounted on the machinery measure analog signals which are collected by an analog-to-digital converter. The purpose of the data acquisition system is to accurately represent the measured analog signal in digital format. A data acquisition system is concerned with two important aspects the sampling frequency and the digital bit size. An inadequate sampling frequency can cause a serious error in data acquisition called aliasing error. In order to prevent aliasing, the signal sampling frequency has to be at least two times higher than the excitation frequency present in the machine, which is known as the Nyquist sampling theorem [66, 121]. Another important aspect of data acquisition is the digital bit size or bits per sample, which determines the measurement resolution. The required measurement resolution is the smallest detectable change in the monitored signal [121]. If the assigned digital bit size is above the required measurement resolution, it is known as digitalization error. To avoid digitalization error, it is preferred to have a higher bit size for the data acquisition system [112].

During the selection of an appropriate transducer to measure the mechanical vibrations, the characteristic parameters to be considered include frequency response, dynamic range, and sensitivity. Frequency response represents the natural frequency of the transducer itself. While performing measurements this frequency response region has to be avoided. The dynamic range relates to the maximum and minimum quantity that can be measured. Sensitivity defines the level of accuracy expected for the signal measurement. It is the smallest fractional change in a device that can be measured [112].

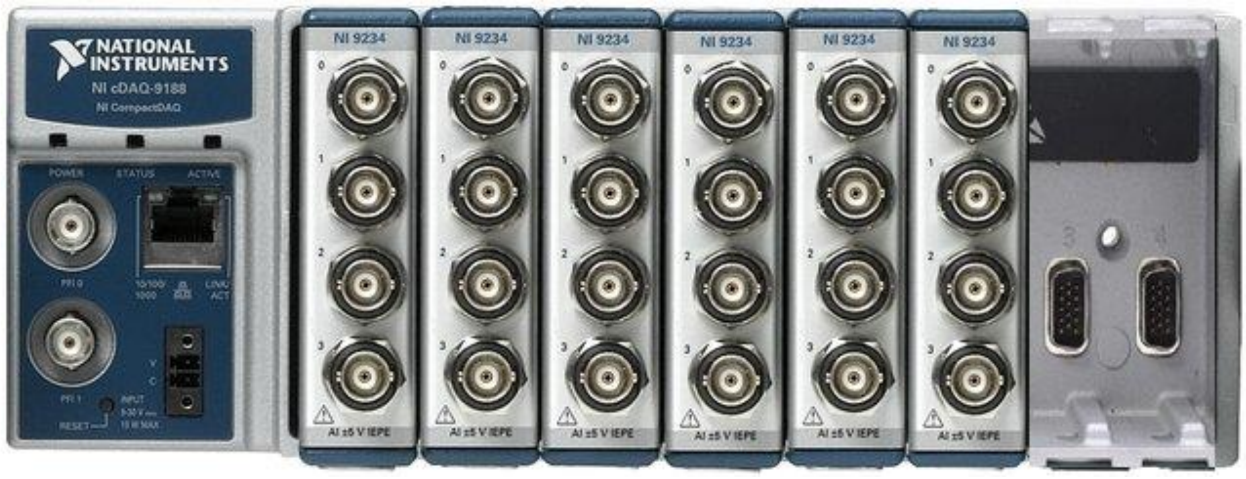


Figure 2.4 A typical multi-channeled data acquisition system

On expanding the condition-based maintenance to predictive maintenance the condition monitoring data has to be stored for later analysis. Digital data recorders like a memory card or a computer hard drive are more prevalent for field data recording. Modern data recorders or data acquisition systems have built-in anti-aliasing low-pass filters that act before analog to digital conversion. These devices are multi-channeled for providing simultaneous access for multiple transducers. A typical multi-channeled data acquisition system is shown in Figure 2.4.

2.6.3 Feature Extraction and Feature Selection

Vibration signal signature features revealing superior machine degradation patterns are extracted from raw time-series vibration data. Signals from rotating machines operating at constant speed are categorized under stationary deterministic signals. The analysis of stationary signals is performed in the time, frequency, and time-frequency domain to extract features representing machine health degradation trends. The time-domain features give an overall sense of the time-series signal and best portray the degradation pattern [20]. The predominant frequencies at which mechanical events occur are analyzed by FFT. As mentioned in 'section 2.6.1', every mechanical component or event has a distinct frequency of occurrence, but it does not provide any information on the degradation pattern. The joint time-frequency domain features give an in-depth portrayal of short mechanical events on the time domain.

The selection of sensitive features from among the extracted features is a significant step in data preparation for prognostic regression analysis. The primary task of feature selection is to discard irrelevant and redundant features, which might cause the overfitting of an evolved prognostic model. Feature selection is usually performed using various feature ranking metrics, which include monotonicity, trendability, linear correlation, etc. Neighborhood Component Analysis (NCA) is a feature learning algorithm that can be effectively implemented to prioritize features for regression analysis through a feature weighting process. NCA is a non-parametric approach, characterized by a feature weighting scheme to select the best subset of features based on the minimization of an objective function that measures the average leave-one-out regression loss over a training data set. The mean absolute deviation of the response values of a randomized regression model from the actual response values is considered as the regression loss function. The algorithm determines the weighting vector w that corresponds to the feature vector X_i. A regularization parameter λ is used to avoid overfitting of the NCA model [122]. The regularized NCA model objective function NCA_{obj} is represented in Eq (2.4).

$$NCA_{obj} = \sum_{i=1}^{n} P_i - \lambda \sum_{m=1}^{r} w_m^2 \tag{2.4}$$

where P_i is the probability of actual RUL being correctly predicted, w_m is the weight assigned to the mth feature, r is the number of features, n is the number of training data. Finally, a relative threshold value of feature weights is set as a cut-off criterion for selecting features for prognostic analysis [122,123]. Table 2.2 presents the outline of regularized NCA algorithm.

Table 2.2 Regularized NCA algorithm outline

Neighborhood Component Analysis algorithm outline	
Step 1	*Recognize training set S={(X_i,x_i), i= 1,2,..N}, number of failure runs N, actual RUL x_i*
Step 2	*Perform 10-fold cross-validation on training set S*
Step 3	*Train the NCA model for each λ value using S in each fold.*
Step 4	*Fit a Gaussian process regression (GPR) model using the selected features.*
Step 5	*Compute regression loss for the corresponding test set in the fold using the GPR model.*
Step 6	*Compute the average loss obtained from the folds for each λ value.*
Step 7	*Tune regularization parameter λ to obtain minimum regression loss*
Step 8	*Fit NCA model with λ_{best} to obtain feature weights*
Step 9	*Assign a relative threshold as the cut-off criterion for feature selection*
Step 10	*Identify the relevant features*

2.7 Prognostics Analysis of Condition Monitoring Data

Prognostics is primarily the forecasting of upcoming failures based on the present and past operating conditions of machines. The computational algorithm for predictive analytics is the most crucial and challenging step in machinery failure prognostics. A prognostic algorithm generally estimates the RUL or TTF by analyzing the historical and current operating condition of the machine. Several RUL prediction approaches are being employed for machinery maintenance decision-making. These approaches can be based on the physics of the failure mechanism or utilizing the machinery failure data (data-driven). Based on the basic techniques and methodologies, the prognostics approaches can be classified into four groups as physics-based approaches, statistical model-based approaches, AI approaches, and any combination of these approaches (hybrid approaches) [6, 18, 54, 124, 125].

Physics-based approach: The physics model-based approach demands a thorough knowledge of the physics of failure mechanism, which is used to develop an analytical model to represent the degradation processes of machinery. For complex mechanical systems, however, it is difficult to understand the physics of the failure mechanism and therefore this approach has a limited application.

Statistical model-based approaches: The statistical model-based approaches require only empirical knowledge about the failure mechanism for establishing the relationship between the failure mechanism and the statistical models. These statistical model-based approaches are also effective in describing the uncertainty associated with the RUL prediction [17].

Artificial Intelligence (AI) approaches: The AI approach uses minimal technical aspects of the system, where intelligent learning algorithms are employed to learn the machinery health degradation patterns [126, 127]. The algorithms learn the machine degradation data to capture its health degradation pattern.

2.7.1 Data-Driven Approaches for RUL Estimation

In a data-driven approach, the historic machine monitoring data representing the health status degradation are used for training computational algorithms to evolve intelligent predictive models. These predictive models can be employed with real-time machine monitoring data for future health status prediction and RUL estimation [18-20, 128]. The data-driven methods are learning-based approaches that discover viable features and prognostics models from the acquired data. These techniques include statistical models and AI models that infer health status information directly from the monitored data. The statistical prediction models are developed by fitting the available machinery failure data into random coefficient models under a probabilistic method without any physics expertise involved. The statistical model-based approaches are the most popularly used prognostic technique for decades. Random coefficient models, autoregressive models, Wiener process models, Gamma process models, inverse Gaussian process models, Markov models, proportional hazards models, exponential degradation model, etc. are the most widely used statistical models for machinery health prognostics [17, 129, 130].

AI is a recently established computational technique that grabbed widespread attention in the area of machinery health diagnostics and prognostics. AI approaches have been receiving increasing attention as it is capable of dealing with prognostics problems of complex mechanical systems because it depends only on the machinery failure data instead of building physics models or statistical models [56]. Unlike the physics-based or statistical model-based approaches, the AI approaches are hard to be explained or lack transparency in operation and are thus called the black box. Machine learning and deep learning AI models are very popular in machinery health diagnostic and prognostic analysis [21, 22]. The most recognized machine learning and deep learning architectures for failure prediction and RUL estimation include neural fuzzy systems, ANN, SVM, SVR, KNN, GPR, RNN, LSTM, DBN, CNN, etc. [21, 22, 27, 127, 131]. Figure 2.5 shows a comparison between the various machinery health management approaches, which is a clear indication of why many recent research works are focused on deep learning techniques.

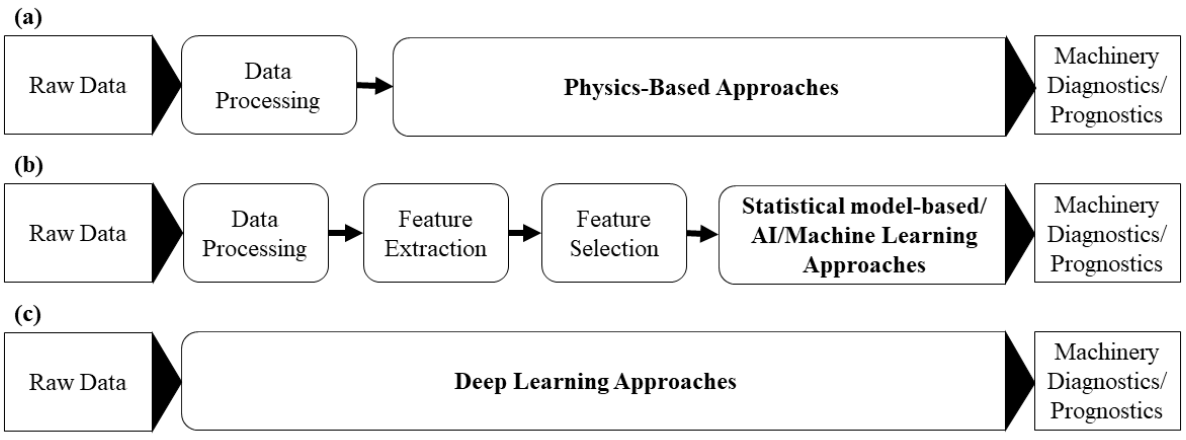

(a)
Raw Data
Data Processing
Physics-Based Approaches
Machinery Diagnostics/ Prognostics
(b)
Raw Data
Data Processing
Feature Extraction
Feature Selection
Statistical model-based/ AI/Machine Learning Approaches
Machinery Diagnostics/ Prognostics
(c)
Raw Data
Deep Learning Approaches
Machinery Diagnostics/ Prognostics

Figure 2.5 Comparison between **(a)** physics model-based approaches, **(b)** statistical model-based/ AI/machine learning approaches, and **(c)** deep learning approaches

2.7.2 RUL Estimation Using Exponential Degradation Estimator Models

An exponential degradation model is utilized for estimating the RUL of the mechanical component. This computational methodology is mostly employed when the component experiences a cumulative degradation, which is the common degradation phenomenon of any mechanical system. The exponential degradation model fits into the machinery health degradation indicator or Health Indicator (HI). This degradation model is extrapolated to find a future time step where the degradation model crosses a predefined threshold value. The difference between this future time step and the present time step gives the required RUL. The failure threshold is usually defined based on previous failure history or chosen as a safe value before the faulty zone on HI. The exponential degradation model is defined in Eq (2.5) as:

$$S(t) = \varphi + \theta(t)e^{\left(\beta(t)t+\varepsilon(t)-\frac{\sigma}{2}\right)} \tag{2.5}$$

where, ϕ is the model intercept, which is constant. $\theta(t)$ is a random variable modeled as a lognormal distribution, $\beta(t)$ is a random variable modeled as a Gaussian distribution, $\varepsilon(t)$ is the model additive noise and is modeled as a normal distribution, σ is the Variance [132]. This degradation model is fit to the constructed HI to predict the RUL of the mechanical component in real-time.

Tseng et al. [130], Wen et al. [133], and Zhang et al. [134] recently employed exponential models for predictive analytics and RUL estimation. Gebraeel et al. [135], first introduced the exponential model for RUL prediction. It is a model-based analytical method that can incorporate both expert knowledge and information from measured data [124, 129]. The exponential models are highly suitable for representing the degradation patterns of a mechanical component, where an exponential-like degradation process can be observed [124, 129, 135]. However, the exponential models are not explored in depth for predictive analytics.

2.7.3 RUL Estimation Using Machine Learning Approaches

Machine learning algorithms are widely employed for the failure classification and prediction of mechanical systems [22]. SVM is the most popularly used machine learning tool for classification problems of machinery failure prediction and RUL estimation [127, 136]. In the past decade, SVM has found its space in regression analysis in real-value function estimation problems like time-series data trend analysis and predictions. In regression analysis SVM is also termed as SVR, the basic operation principle remains the same for both classification and regression analysis. Like NCA, SVM is also a non-parametric technique that relies on kernel functions [137].

Consider a training dataset $T = \{x_n, y_n\}$, where x_n is the feature set of N observations and y_n is the corresponding RUL response vector. The SVM objective is to determine a function $f(x)$ as represented in Eq (2.6), which is a derivative of y_n and the value for each training point x.

$$f(x) = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*)(x_n' x) + b \tag{2.6}$$

where α_n and α_n^* are non-negative multipliers for each observation x_n. and b is the bias term. If either α_n or α^*_n is not zero, then the observation x'_n is called the support vector.

Eq (2.6) represents the linear SVM regression equation. Replacing the dot product $x'_n x$ with a non-linear kernel function k gives the non-linear SVM as represented in Eq (2.7).

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)k(x_n', x) + b \tag{2.7}$$

The function *f(x)* is used to predict new values of the regression problem [137, 138]. The implementation of SVM for degradation data training and RUL estimation requires a good setting for hyperparameters, which is considered highly complicated. The hyperparameter setting largely influences the prediction accuracy of the evolved regression model.

SVM is one of the most popular machine learning techniques proposed by Vapnik in 1999 [139], which has unfurled a wide scope for implementation in machinery health management problems [139-144]. SVR is the popular application form of SVM, which is used for mapping the machine degradation indicators into nonlinear regressions for RUL prediction [145-152].

2.7.4 RUL Estimation Using Deep Learning Approaches

Deep learning approaches have emerged as a promising computational tool for the time-series machinery health degradation analysis for failure prediction and RUL estimation. In deep learning or deep neural network, the data runs through several layers of a neural network algorithm. Deep learning, also known as deep structured learning consists of multiple layers on non-linear processing units. Deep learning is a phrase that leverages a series of nonlinear processing units comprising multiple layers for the flow of information throughout the model. Deep learning adopts a hierarchy of data transformation where the present layer accepts the outcome from the previous layer which is processed and passed to the next layer. Deep learning supports learning from both labeled and unlabeled data [153]. This novel learning approach has been successfully implemented in the fields of computer vision, pattern recognition, image classification, face recognition, facial emotion recognition, natural language processing, speech recognition, health care, time-series data classification, regression, and many more [153, 154].

As shown in Figure 2.6, a deep learning framework for machinery health prognostics is generally composed of the data acquisition stage, deep neural network learning stage, intelligent predictive model construction, and the final machinery health status prediction stage. An intelligent predictive model is evolved from the trained deep learning algorithm, which is further utilized for machinery health prognostics using real-time machine monitoring data [21, 155].

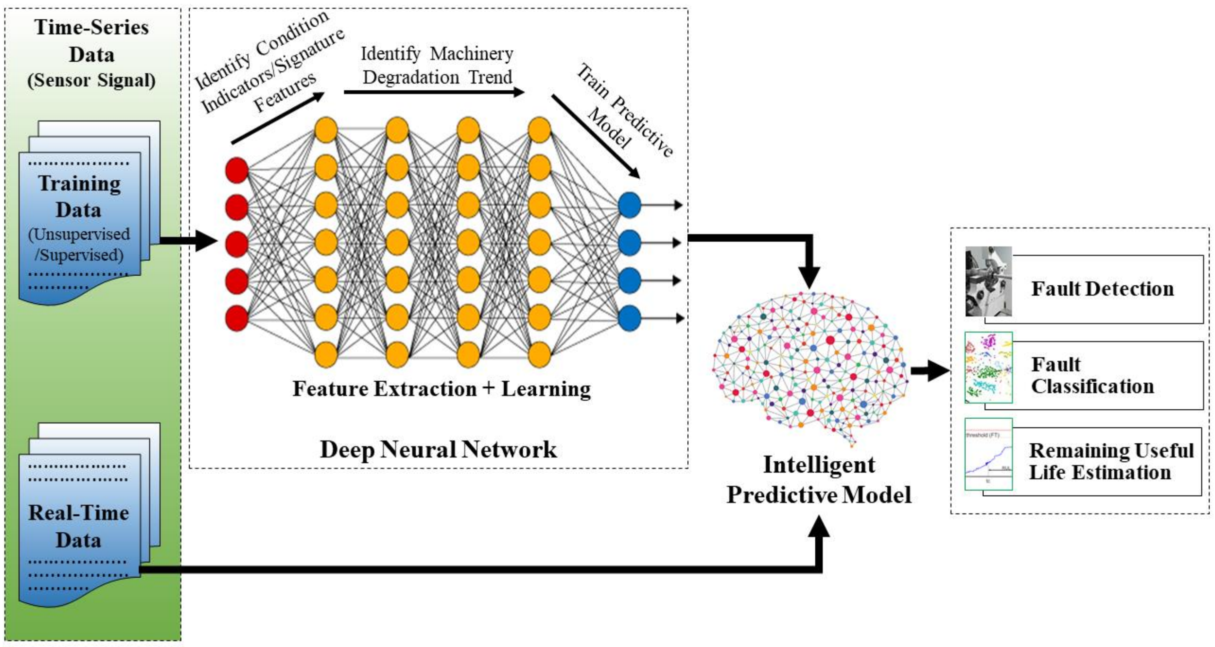
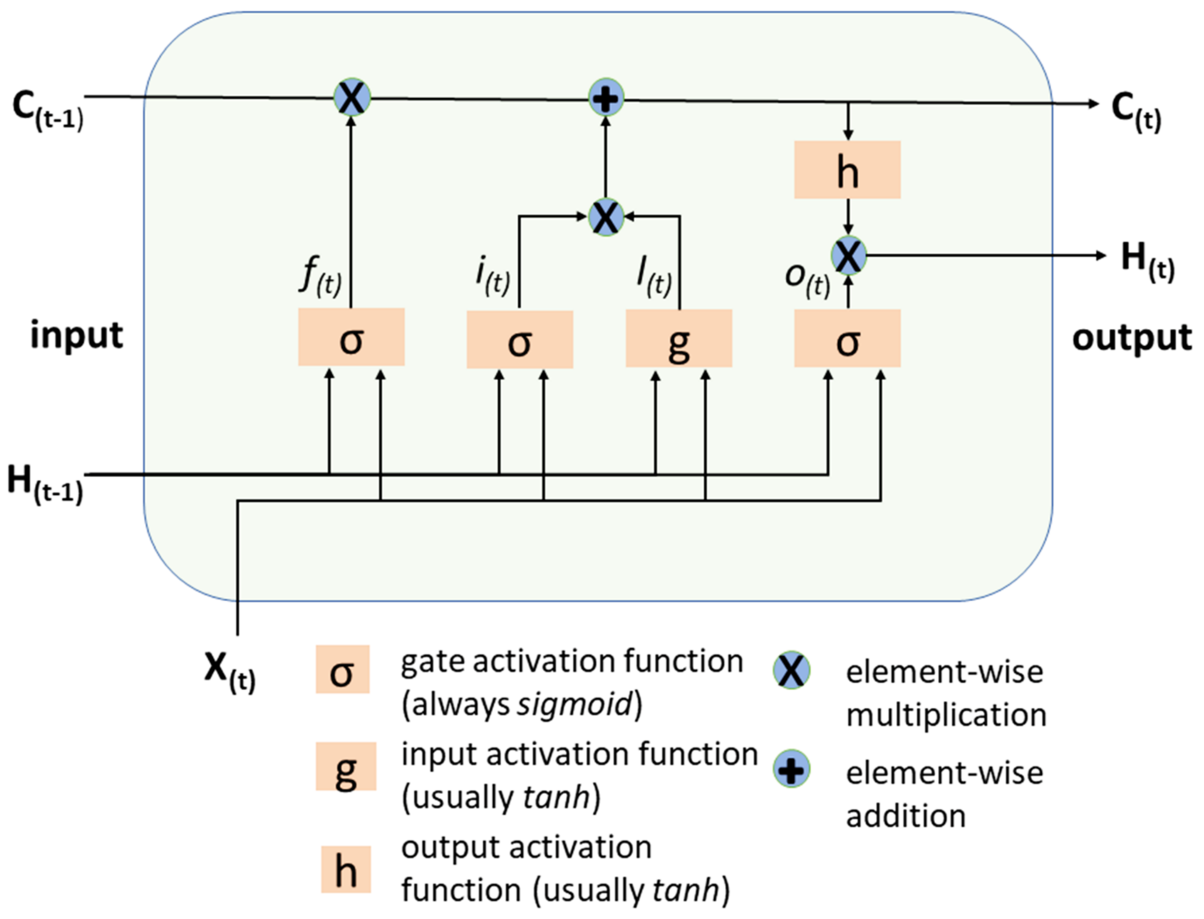

Time-Series Data (Sensor Signal)
Training Data (Unsupervised /Supervised)
Real-Time Data
Identify Condition Indicators/Signature Features
Identify Machinery Degradation Trend
Train Predictive Model
Feature Extraction + Learning
Deep Neural Network
Intelligent Predictive Model
Fault Detection
Fault Classification
Remaining Useful Life Estimation

Figure 2.6. Generalized deep learning framework for machinery health prognostics using time-series data

The LSTM neural network, an advanced variant of RNN, is identified as a powerful computational tool for mining critical information from raw sequential time-series data. In 1997, Hochreiter and Schmidhuber [156] introduced the LSTM neural network to address the vanishing/exploding gradient problem while training using RNN. LSTM has an additional memory unit to remember information for long periods enabling the network to learn long-term dependency, which makes it suitable for RUL estimation from time-series data. A self-recurrent connection node within the memory unit ensures that the long-term information gradient can pass across many time-steps of time-series sequence data without vanishing. The memory unit in LSTM provides storage for information from previous time-steps and passes it to current outputs ensuring the long-term memory in the form of weights that changes during training and the short-term memory in the form of ephemeral activation functions [157].



C(t-1)
C(t)
h
H(t)
f(t)
i(t)
I(t)
o(t)
input
output
σ
σ
g
σ
H(t-1)
X(t)
σ gate activation function (always sigmoid)
g input activation function (usually tanh)
h output activation function (usually tanh)
element-wise multiplication
element-wise addition

Figure 2.7 Generalized LSTM network architecture

Figure 2.7 shows the generalized LSTM network architecture having three control gates; input gate, output gate and forget gate; and two information states; the long-term state $C_{(t)}$ and the short-term state $H_{(t)}$. Initially, the LSTM architecture had only input and output gates, later Gers et al. [158] in 2000 have introduced the forget gate. The LSTM architecture with input gate, output gate and forget gate is entitled as vanilla LSTM. This vanilla version is now popularly referred to as the LSTM architecture. The input gate controls the information stored in the memory cell and output gates control the information extraction from the memory cell. The forget gate controls the discarding of previous information from the memory cell. In an LSTM cell, the input gate combines the current input $X_{(t)}$, previous output $H_{(t-1)}$, and the previous memory cell state $C_{(t-1)}$. f_t, i_t, o_t indicate the three-control gate outputs of forget gate, input gate, and output gate, respectively. The Eqs (2.8), (2.9), and (2.10) represent the gate operation of forget gate, input gate, and output gate, respectively. σ , g, and h are the gate, input, and output activation functions, respectively; $\otimes$ and $\oplus$ denote element-wise multiplication and element-wise addition of vectors, respectively.

$$f_{(t)} = \sigma(U_f . X_{(t)} + V_f . H_{(t-1)} + W_f \otimes C_{(t-1)} + b_f) \tag{2.8}$$

where U_f, V_f, and W_f are the forget gate weights associated with $X_{(t)}$, $H_{(t-1)}$, and $C_{(t-1)}$, respectively, while b_f is the bias weight vector. The output gate combines the current input $X_{(t)}$, previous output $H_{(t-1)}$, and the previous memory cell state $C_{(t-1)}$.

$$i_{(t)} = \sigma(U_i . X_{(t)} + V_i . H_{(t-1)} + W_i \otimes C_{(t-1)} + b_i) \tag{2.9}$$

where U_i, V_i, and W_i are the input gate weights associated with $X_{(t)}$, $H_{(t-1)}$, and $C_{(t-1)}$, respectively, while b_i is the bias weight vector. The activation function values of forget gate $f_{(t)}$ at current time stem t are computed based on the current input $X_{(t)}$, previous output $H_{(t-1)}$, and the previous memory cell state $C_{(t-1)}$.

$$o_{(t)} = \sigma(U_o . X_{(t)} + V_o . H_{(t-1)} + W_o \otimes C_{(t-1)} + b_o) \tag{2.10}$$

where U_o, V_o, and W_o are the output gate weights associated with $X_{(t)}$, $H_{(t-1)}$, and $C_{(t-1)}$, respectively, while b_o is the bias weight vector. The current layer input, $l_{(t)}$ is computed as presented in Eq (2.11).

$$l_{(t)} = g(U_l . X_{(t)} + V_l . H_{(t-1)} + b_l) \tag{2.11}$$

where U_l and V_l are the layer input weights associated with $X_{(t)}$ and $H_{(t-1)}$ respectively, while b_l is the bias weight vector. The current memory cell state $C_{(t)}$ is computed as presented in Eq (2.12).

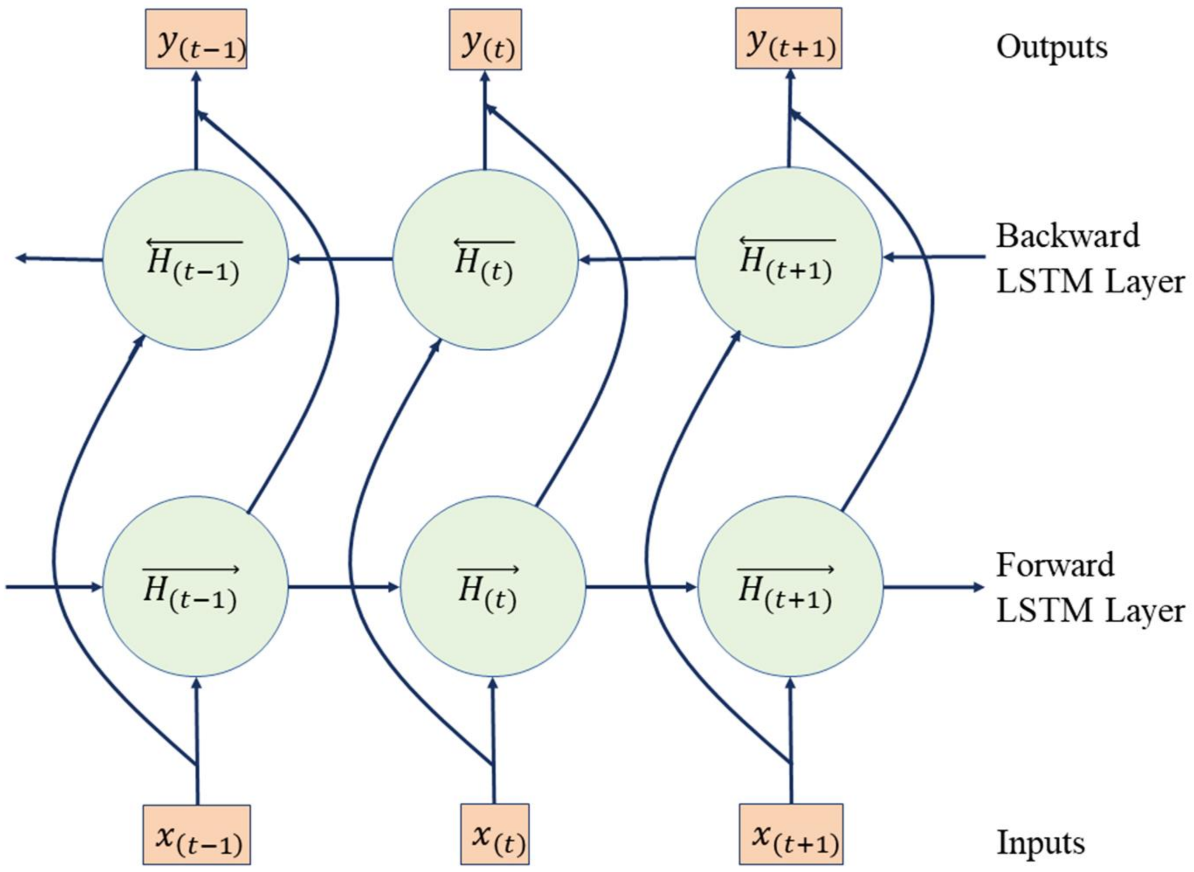
$$C_{(t)} = l_{(t)} \otimes i_{(t)} \oplus C_{(t-1)} \otimes f_{(t)} \tag{2.12}$$

Finally, the current layer output $H_{(t)}$ is computed as presented in Eq (2.13).

$$H_{(t)} = h(C_{(t)}) \otimes o_{(t)} \tag{2.13}$$

LSTM evaluates the temporal relationship between inputs and outputs following a forward learning principle using the Eqs. (2.8) to (2.13). The error values between the input data and the output data of each layer are computed and are reversely seeded to the input gate and forget gate, based on which the weights associated with each gate are updated. This process is repeated for a fixed number of iterations and an optimal value of weights and bias terms are obtained [157, 159].

The Bi-LSTM network learns the bidirectional long-term dependency between time-steps of time-series sequence data, enabling the network to learn from the complete time-series at each time step. The bi-LSTM network comprehensively considers the temporal correlation information between inputs and outputs in both forward and backward time-step directions simultaneously, which makes it perform excellent for RUL estimation problems. The generalized bi-LSTM network architecture is represented in Figure 2.8 where the same input data is fed into forward LSTM cells and backward LSTM cells of respective forward and backward LSTM layers [160,161].



$y_{(t-1)}$
$y_{(t)}$
$y_{(t+1)}$
Outputs
$\overleftarrow{H_{(t-1)}}$
$\overleftarrow{H_{(t)}}$
$\overleftarrow{H_{(t+1)}}$
Backward LSTM Layer
$\overrightarrow{H_{(t-1)}}$
$\overrightarrow{H_{(t)}}$
$\overrightarrow{H_{(t+1)}}$
Forward LSTM Layer
$x_{(t-1)}$
$x_{(t)}$
$x_{(t+1)}$
Inputs

Figure 2.8 Generalized bi-LSTM network architecture

The forward LSTM is computed as:

$$\overrightarrow{H_{(t)}} = LSTM(x_{(t)}, \vec{h}_{(t-1)}) \tag{2.14}$$

The backward LSTM is computed as:

$$\overleftarrow{H_{(t)}} = LSTM(x_{(t)}, \overleftarrow{h}_{(t+1)}) \tag{2.15}$$

The bi-LSTM network computes the forward LSTM hidden state $\overrightarrow{H_{(t)}}$ and the backward LSTM hidden state $\overleftarrow{H_{(t)}}$ simultaneously at each sequence time-step t. LSTM(.) denotes the LSTM evaluation defined in Eqs (2.8) to (2.13). Finally, the two hidden states are concatenated to compute the current layer output $y_{(t)}$ as:

$$y_{(t)} = W_{\vec{h}}\overrightarrow{H_{(t)}} + W_{\overleftarrow{h}}\overleftarrow{H_{(t)}} + b_y \tag{2.16}$$

where $W_{\vec{h}}$ and $W_{\overleftarrow{h}}$ are the forward and backward LSTM layer weights; b_y is the output layer bias weight vector [163].

Recently, the LSTM network and its variants are widely used for learning machine degradation patterns from time-series sensor data for RUL estimations [163-178]. Elsheikh et al. [160], Song et al. [179], Zhang et al. [180], Wang et al. [181] have used bidirectional LSTM (bi-LSTM) and Essien et al. [182] used convolutional LSTM for RUL estimation. Xia et al. [162] and Zhang et al. [183] have used a convolutional LSTM and convolutional bi-LSTM ensemble framework respectively for the RUL estimation.

The machinery prognostics data are time-series sensor signals, which are usually analyzed to develop a regression model for RUL estimation. RNN and its variant LSTM have emerged as the most popular AI approach for handling sequential data like time-series sensor signals using its ability to encode temporal information. The initial implementations of RNN have reported its suitability for RUL prediction [184]. Improved accuracy of RUL prediction was achieved by enhancing the memory capacity of basic RNN architecture which is named Eco-State Network [185, 186]. The LSTM architecture has subsequently proved to be more effective than simple RNN architecture. Fault diagnostics and prognostics of aero-engine units [163] and machine tools [187] mark the initial implementation of LSTM architecture for machinery health prognostics. He et al. [172] have proposed a long short-term memory network with multi-resolution singular value decomposition to accurately locate the fault in vibration signals, and minimize the effect of interfering noise. ElSaid et al. [166] have utilized LSTM for predictive analytics of time-series sensor data with the Ant Colony Optimization approach for optimizing the gates within LSTM. Further, this work was improved by combining CNN with LSTM forming a convolutional bi-directional LSTM [188]. CNN was used to extract robust features from raw sensor signals sequential input and bi-directional LSTM was used to encode temporal information of sequential output of the convolution layer. Zhao et al. [189] have proposed a hybrid approach that combines manual feature extraction and automatic feature learning using an enhanced gated-RNN/Gated Recurrent Unit (GRU) for machinery health monitoring. Elsheikh et al. [160] proposed a bidirectional handshaking LSTM architecture for RUL prediction with a short sequence of monitored data with random starting health conditions and requires a lesser assumption on actual degradation phenomenon. Malhotra et al. [190] proposed an LSTM based encoder-decoder architecture where the encoder transforms a multivariant input sequence into a fixed-length vector which the decoder translates into a target sequence. Xia et al. [162] proposed a hybrid deep learning framework based on convolutional bi-directional long short-term memory with multiple time windows for accurately predicting RUL. The algorithm addresses the inconsistency among the length of condition monitoring data and develops base models with different time window sizes. Zhang B. et al. [171] have formulated bearing health monitoring as a classification problem using LSTM RNN using waveform entropy as the degradation indicator for bearing health degradation. Wu Y. et al. [165], Zhao et al. [169], Wang et al. [170], Zhang et al. [180], and Chen Y. et al. [191] have also developed deep long short-term memory network architectures for RUL estimation based on time-series sensor signals. LSTM and other variants of RNN are reported as the most suitable computational tool for RUL estimation. A majority of recent literature on machinery health prognostics has implemented LSTM architecture as such or with modifiers for prognostics health indicator construction and RUL predictions.

2.7.5 Bayesian Optimization of Hyperparameters for Learning Algorithms

The hyperparameter optimization and network architecture selection are other major challenges before the successful implementation of a deep learning algorithm for prognostics analysis of industrial machinery. The selection of hyperparameters including both training and structural parameters has a direct influence on the performance of the predictive models. Generally, this optimization is performed by trial-and-error methods, which is arduous and time-consuming [162, 168]. Recently, a few literature have discussed the hyperparameter optimization algorithms employing ant colony optimization [166], particle swarm optimization [171], comparative analysis [179], grid search [165, 170, 182], Bayesian search optimization [161, 173], etc. to reduce the computational complexity and improve prediction accuracy.

Bayesian optimization is an effective computational tool for optimizing non-differentiable, discontinuous, and computationally expensive functions. In the Bayesian optimization algorithm framework, the unknown hyperparameters of deep learning architecture are assumed to be random, independent, and have their prior distributions. The algorithm attempts to minimize a scalar objective function *f(x)* for x in a bounded domain χ. The deep learning architecture is represented within the function *f(x)* that returns the deep learning validation error. Bayesian optimization internally maintains a Gaussian process model of the objective function *f(x)* and uses it to train the computational model. This Gaussian process model is updated at each new evaluation of *f(x)* and the next x point is determined using an acquisition function *Acq(x)* performing a local search to find the best apparent feasible point satisfying the constraints. The target of Bayesian optimization is to determine the optimum point x_{best}, which is the value of x for which *f(x)* attains its minimum. x_{best} is computed as:

$$x_{best} = \underset{x \epsilon \chi}{\operatorname{argmin}}(f(x)) \tag{2.17}$$

After each new evaluation of *f(x)*, the next most potential evaluation point x_{next} is the value of x for which *Acq(x)* attains its maximum. x_{next} is computed as:

$$x_{next} = \underset{x \epsilon \chi}{\operatorname{argmax}}(Acq(x)) \tag{2.18}$$

The Bayesian optimization algorithm repeats the Gaussian process modeling and acquisition function at each evaluation of the objective function *f(x)*. The algorithm can terminate at a predefined time limit, maximum iterations, or any other termination criteria reflecting its performance. The Bayesian optimization can be deterministic or stochastic because the optimization results are not reproducible for the same point x [173, 121, 192]. Table 2.3 presents the outline of a Bayesian optimization algorithm for hyperparameter optimization.

Table 2.3 Bayesian optimization algorithm outline

Bayesian optimization algorithm outline	
Step 1	*Randomly initialize a set of hyperparameters from the bounded domain of* χ
Step 2	*Obtain the new set of hyperparameters*
Step 3	*Perform Gaussian process regression to evaluate f(x)*
Step 4	*Determine the next most potential evaluation point* x_{next} *based on Acq(x)*
Step 5	*Add results to solution space*
Step 6	*Repeat Steps 2 to 5 until the termination criterion is reached*
Step 7	*Record the best hyperparameters* x_{best}

2.8 Maintenance Decision Support System

The IoT is having a profound effect on the industrial sector. Utilizing the full benefits from IoT, the industrial predictive maintenance approach can be implemented in its intact form [73, 78]. The advancements in sensor technologies supporting IoT, cloud storage and cloud computing, and intelligent computational approaches have offered an easy to deploy end-to-end communication paradigm which is now popularly known as IIoT [9, 194]. Figure 2.9 shows a typical architecture of IIoT systems portraying different levels of IIoT. The bottom level of IIoT architecture includes the machines or devices labeled as the '*things*', then comes the internet gateway to the cloud space and data pre-processing followed by the data management stage, and finally, the cloud data center where the real-time intelligent computations are occurring.

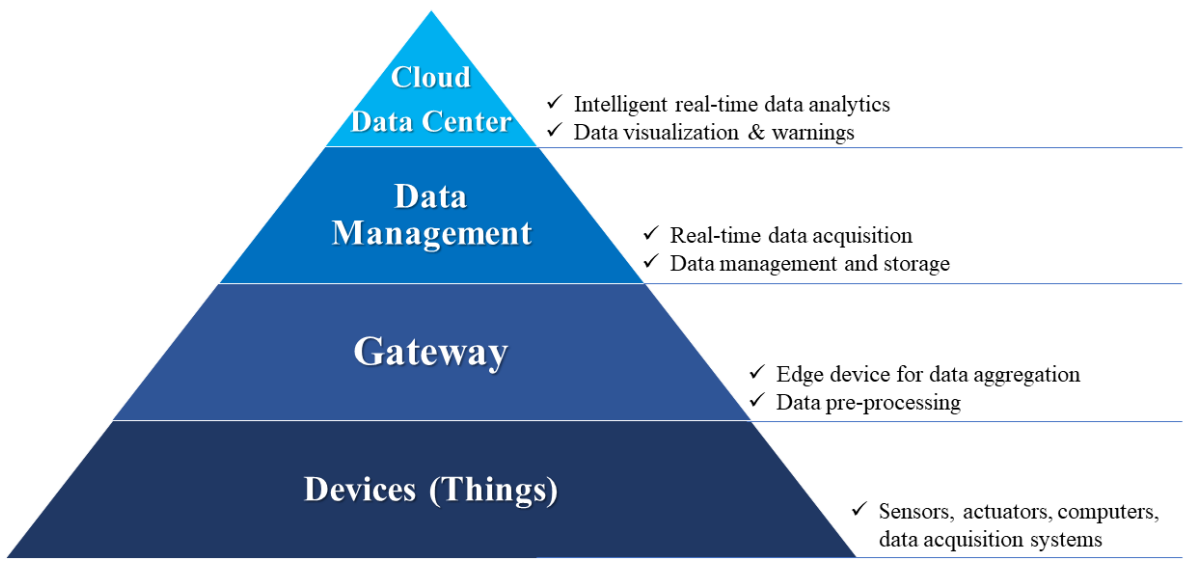

Cloud Data Center
✓ Intelligent real-time data analytics
✓ Data visualization & warnings
Data Management
✓ Real-time data acquisition
✓ Data management and storage
Gateway
✓ Edge device for data aggregation
✓ Data pre-processing
Devices (Things)
✓ Sensors, actuators, computers, data acquisition systems

Figure 2.9 Typical architecture of IIoT systems

Maintenance decision-making is a very crucial task in the industry that has a considerable impact on machine operability and has to be carried out with minimal impact on the production cycle in the industry [26, 194]. Effective maintenance decision-making requires the integration of several factors including the machine health condition, machine work schedule, system configuration, maintenance service, spare parts availability, and maintenance costs [14, 195, 196]. The primary aim of maintenance decision-making is to have minimum machine downtime and improved quality of products [197]. Cloud computing and big data analytics with predictive maintenance decision-making are used to develop an integrated maintenance recommendation system to improve asset lifecycle [26, 198]. A key performance indicator of machinery namely, RUL, TTF, mean time between failure (MTBF), or mean time to repair (MTTR) is determined as the maintenance decision support tool. Intelligent computational models and the computational algorithm to determine the key performance indicator are made available in the cloud space [1, 3]. The decision support system using models is popularly known as a model-driven decision support system. In a model-driven maintenance decision support system, the real-time machine health condition data is evaluated against the computational models to estimate the key performance indicator to predict future failures [199]. This information is made available on a maintenance decision support dashboard to serve as a maintenance decision-making paradigm for industrial machines. This maintenance decision support dashboard provides managers with models and analysis capabilities that can be used during the process of maintenance decision-making [199, 200]. The system is also been used to develop an autonomous warning system on approaching failures [193, 201].

2.9 Gap Areas

Predictive maintenance has become the new trend of PHM for industrial equipment. The ability to predict the need for maintenance of the complex machine tool systems at a certain future point is one of the main challenges in the PHM of the manufacturing industry. An intelligent predictive health management paradigm for manufacturing machinery is inevitable in Industry 4.0. The tool condition monitoring (TCM), problems such as cutting tool wear, cutting tool breakage, and chatter are largely been investigated by the research communities over decades, whereas the research investigations on machine tool functional component condition monitoring and maintenance are limited and inadequate for Industry 4.0 implementations. The manufacturing industries are pulled back from implementing the predictive maintenance approach due to its huge capital investment for the installation of sensors and data acquisition systems, and difficulties to maintain complex computational algorithms. From the research point of view, studying the industrial machinery health degradation and failure analysis has practical complications to implement. Most research works are depended on machine degradation and failure data available in the open repositories, which are used for the development and validation of machinery failure diagnostics and prognostics algorithms. Advanced computational algorithms based on deep learning models are widely employed for solving real-world problems. It is also found effective for sequence data like time-series machine health degradation data. The deep learning architecture design and hyperparameter optimization are the major challenges that need to be addressed for the successful implementation of intelligent predictive analytics. On account of the industry 4.0 era, the advanced sensor technologies with IoT support, cloud storage, and big data analytics could be integrated to develop a model-driven intelligent maintenance decision support dashboard. The gaps identified from the literature are summarized as:

i. Limited literature is available in the public domain on prognostics and predictive maintenance of machine tools. The available literature is largely focused on cutting tool condition monitoring and tool wear analysis.

ii. The possibility of a systematic solution for the high installation costs for sensors and data acquisition systems in the implementation of predictive maintenance is never discussed.

iii. Researchers seldom generate experimental machine health degradation data for the prognostic analysis. The available research works are mostly based on machine health degradation data available in open repositories.

iv. Deep Learning techniques are not explored in depth for Predictive Maintenance applications. On an implementation level, deep learning architecture design, hyper-parameter optimization, and data training are still challenging and unpredictable.

v. IoT-based intelligent maintenance decision support system for the manufacturing sector is a shortfall in the present Industry 4.0 era.

In view of the above, there is a strong need for in-depth scientific understanding and detailed investigations on the development of condition monitoring and predictive maintenance of machine tool systems addressing the huge capital investments and large complexity of its implementation. It is proposed to take up the research on the predictive maintenance of machine too systems, which involves the maintenance prioritization technique to limit predictive maintenance approach to the most critical subsystems are components, generate machine health degradation data, easy employment of intelligent data-driven techniques for prognostic analysis, and IoT-based intelligent maintenance decision support system.

2.10 Aim and Objectives

The thesis work aims to employ a systematic approach for machine tool components maintenance prioritization and data-driven intelligent algorithms for predictive maintenance. The thesis validates the developed algorithms using simulated run-to-failure experimental data. The primary objectives of the work are:

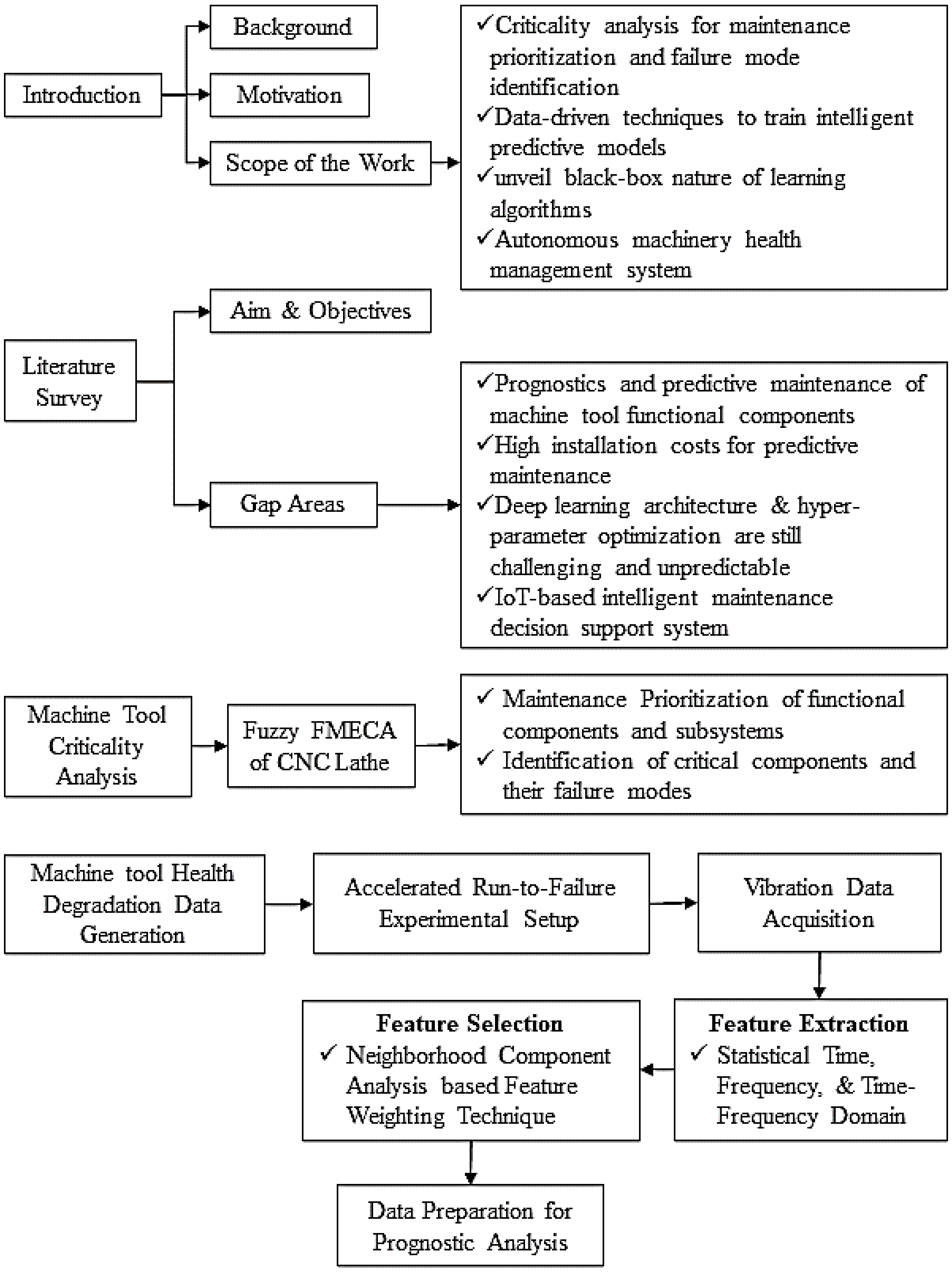
i. To identify the most critical component of a Lathe Machine Tool through failure analysis.

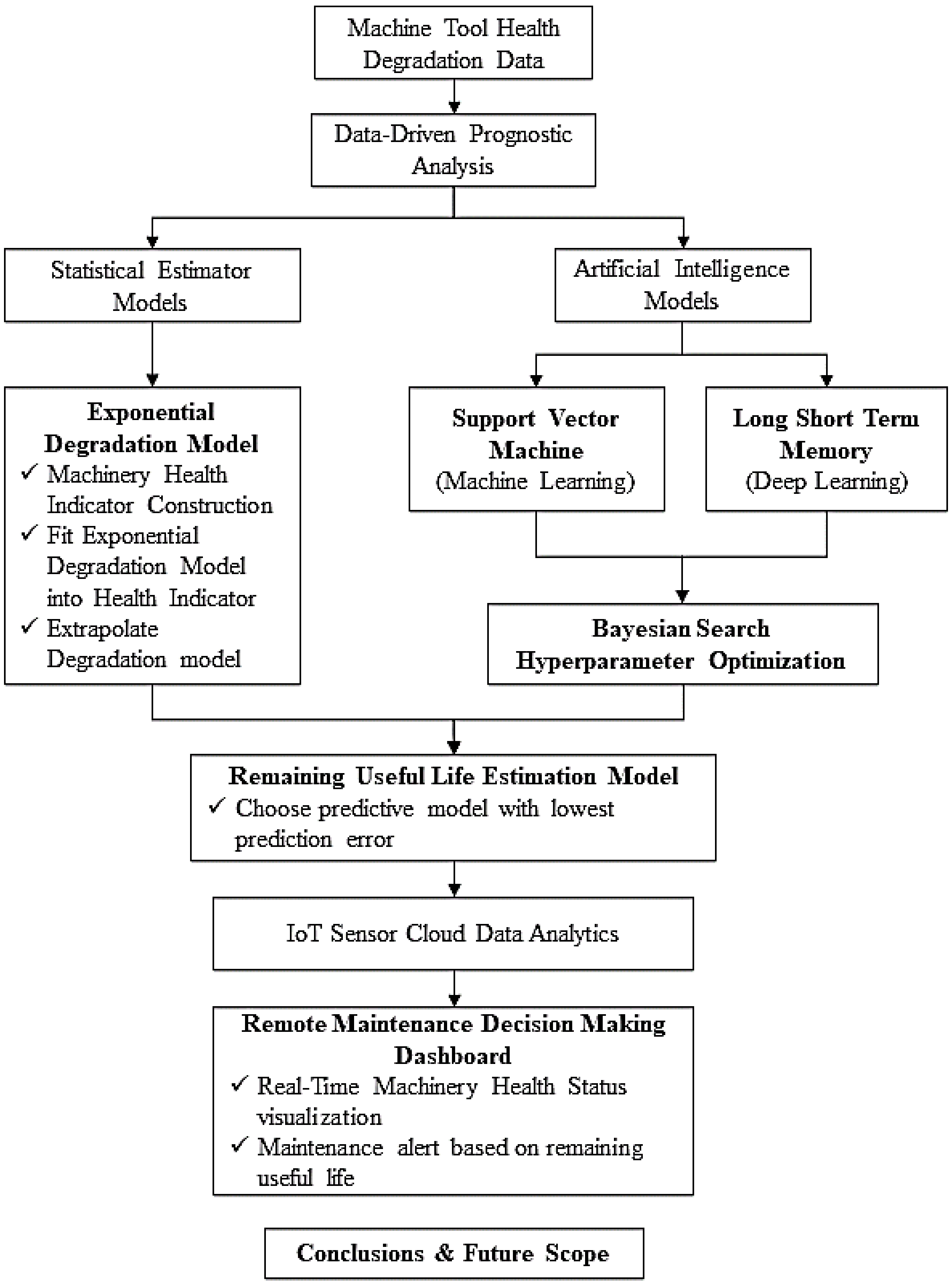
ii. To perform vibration-based Condition Monitoring for the selected features of the identified critical component.

iii. To conduct optimization/simulation-based analysis to predict the RUL of the identified critical component.

iv. To develop a remote maintenance decision-making dashboard over an IoT-based cloud data analytics platform for integrating the evolved prognostic model.

2.11 Flowchart of Thesis


Introduction
Background
Motivation
Scope of the Work
✓Criticality analysis for maintenance prioritization and failure mode identification
✓Data-driven techniques to train intelligent predictive models
✓unveil black-box nature of learning algorithms
✓Autonomous machinery health management system
Literature Survey
Aim & Objectives
Gap Areas
✓Prognostics and predictive maintenance of machine tool functional components
✓High installation costs for predictive maintenance
✓Deep learning architecture & hyper-parameter optimization are still challenging and unpredictable
✓IoT-based intelligent maintenance decision support system
Machine Tool Criticality Analysis
Fuzzy FMECA of CNC Lathe
✓ Maintenance Prioritization of functional components and subsystems
✓ Identification of critical components and their failure modes
Machine tool Health Degradation Data Generation
Accelerated Run-to-Failure Experimental Setup
Vibration Data Acquisition
Feature Extraction
✓ Statistical Time, Frequency, & Time-Frequency Domain
Feature Selection
✓ Neighborhood Component Analysis based Feature Weighting Technique
Data Preparation for Prognostic Analysis


Machine Tool Health Degradation Data
Data-Driven Prognostic Analysis
Statistical Estimator Models
Artificial Intelligence Models
Exponential Degradation Model
✓ Machinery Health Indicator Construction
✓ Fit Exponential Degradation Model into Health Indicator
✓ Extrapolate Degradation model
Support Vector Machine (Machine Learning)
Long Short Term Memory (Deep Learning)
Bayesian Search Hyperparameter Optimization
Remaining Useful Life Estimation Model
✓ Choose predictive model with lowest prediction error
IoT Sensor Cloud Data Analytics
Remote Maintenance Decision Making Dashboard
✓ Real-Time Machinery Health Status visualization
✓ Maintenance alert based on remaining useful life
Conclusions & Future Scope

2.12 Summary

An intelligent predictive health management approach for the manufacturing industry is inevitable in the present generation of smart manufacturing. The manufacturing industry could utilize the recent developments in sensor technologies and computational capabilities to develop a predictive maintenance paradigm to ensure the well-operating condition of the functional components and subsystems of machine tool systems. However, the implementation of predictive maintenance for machinery health management has enormous challenges, which have a certain scope for discussion in the present era of Industry 4.0. Predictive maintenance involves various complex stages of implementation like machinery health degradation data acquisition as time-series sensor signals, which are analyzed for fault detection, prediction, and maintenance decision making. Further, there is a lack of literature available in the public domain addressing the major issue on the practical implementation of predictive maintenance like the huge installation costs for sensors and data acquisition systems, also on optimizing the architectural and learning parameters of the data-mining technique. Thus, the present work aims to investigate the practical implementation of predictive maintenance for CNC lathe machine tool systems with intelligent data-driven computational techniques for machinery health degradation pattern learning. The next chapter presents the pre-requisite analysis for the implementation of predictive maintenance on the criticality analysis of the CNC lathe machine tool for the maintenance prioritization of functional components and subsystems, and to identify its potential failure modes.

Chapter 3

Machine Tool Criticality Analysis and Maintenance Prioritization

3.1 Introduction

As mentioned in the previous chapter ‘section 2.5’, machine criticality analysis is a prerequisite for employing the predictive maintenance strategy for industrial systems. Machine criticality analysis involves the identification of the most critical components and subsystems of the machine system and their associated failure modes. As the predictive maintenance approach is based on machinery condition monitoring data, only those components and subsystems that deteriorate over time emitting signals indicating health degradation are considered for predictive maintenance. Also, the information on potential failure modes associated with these components and subsystems is necessary for the selection of the most suitable sensors for condition monitoring. The machine criticality analysis thus provides a maintenance prioritization of machinery components and subsystems, assisting the user to choose only those components and subsystems which are critical to be considered for predictive maintenance and hence keep the initial investments and implementation costs to a minimum. In this work, the CNC lathe machine tool components and subsystems are subjected to FMECA for the maintenance prioritization and identification of potential failure modes.

3.2 CNC Lathe Criticality Analysis

CNC machine tools are the key production equipment for the manufacturing industry. CNC machine tools, with numerous mechanical moving parts and precise control systems, are prone to malfunctioning and breakdowns. The predictive maintenance of CNC machine tools requires the identification of critical components and subsystems and their potential failure modes [12]. The methodology for criticality analysis and maintenance prioritization involves the following steps:

i. Collection of field failure data of CNC lathe machine tools.
ii. Defining the structure of the CNC lathe system.
iii. Identifying the potential failure modes, failure causes, and machine controls and methods for detecting failures and subsequently framing of S, O, and D rating scales for the failure modes, failure occurrence.
iv. Finally, the computation of RPN for maintenance prioritization.

3.2.1 Field Failure Data of CNC Lathe

Industrial field failure data and expert elicitation constitute the foundation for failure analysis using fuzzy FMECA [202, 203]. The field failure data of 30 CNC lathe machines over 7-years of duration is collected from various industries. The data is in the form of history cards of individual CNC lathe machines, which detail all the maintenance works, repairs, and replacements of the components and subsystems with the date of action and total downtime of the machine tool for each failure. The expert elicitation is an aggregate of opinions of various industry experts. The major causes of failure were due to the structural and material failure of the component. The structural failure includes design and maintenance faults, manufacturing defects, mechanical overload, the presence of debris, and the collision of components. The material failure includes fatigue, wear, corrosion, overheating, insufficient lubrication.

The field failure data and expert elicitation collected from industries are used to identify the potential failure modes of the CNC lathe at the component level. Further, this information is used to assign S, O, and D rating values and in developing the fuzzy FMECA engine. The potential failure modes are identified at the component level and the corresponding risk factor is defined. In FMECA, the aggregate risks allied with failure modes of components in a subsystem represent the risk associated with that particular subsystem [89, 97].

3.2.2 Structure of CNC Lathe

CNC lathes are machine tools with a composite structure having mechanical, hydraulic, and electrical subsystems [34, 202]. In the present study, the prioritization of components for predictive maintenance is limited to only mechanical components and subsystems. Electronics and electrical subsystems, which are very frequent to failures like electronic damages of sensors, relays, blown fuse, etc. are not included [87-93]. Therefore, only seven mechanical subsystems of the CNC lathe machine are investigated using fuzzy FMECA. Individual components of the CNC lathe machine are grouped within different subsystems for failure analysis. Figure 3.1 shows the CNC lathe hierarchy with subsystems and the respective components in each subsystem. The hierarchy structure of a CNC lathe is defined based on expert elicitation and field failure data from industries.

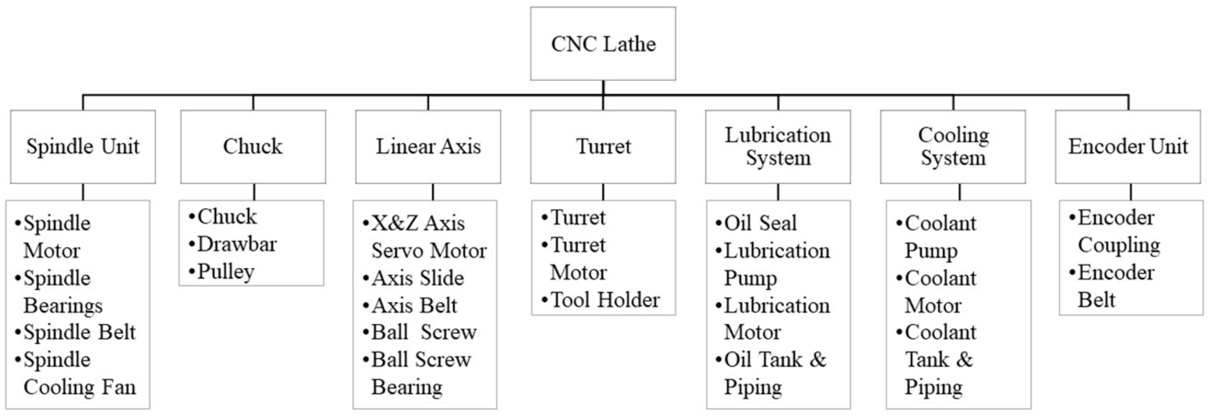

CNC Lathe
Spindle Unit
Chuck
Linear Axis
Turret
Lubrication System
Cooling System
Encoder Unit
•Spindle Motor •Spindle Bearings •Spindle Belt •Spindle Cooling Fan
•Chuck •Drawbar •Pulley
•X&Z Axis Servo Motor •Axis Slide •Axis Belt •Ball Screw •Ball Screw Bearing
•Turret •Turret Motor •Tool Holder
•Oil Seal •Lubrication Pump •Lubrication Motor •Oil Tank & Piping
•Coolant Pump •Coolant Motor •Coolant Tank & Piping
•Encoder Coupling •Encoder Belt

Figure 3.1 Hierarchy of a CNC lathe machine

3.3 Failure Analysis of CNC Lathe

Industrial field failure data and expert elicitation of CNC lathe combined with MIL-STD-1629A [88, 97] guidelines are followed in developing S, O, and D rating scales, which makes the input parameters for FMECA. The industrial field failure data and expert elicitation are analyzed to identify the failure modes associated with each component of the CNC lathe followed by an assessment of potential failure effects, causes of failure, and current controls available to detect the fault. The CNC lathe failure and maintenance information and the MIL-STD-1629A guidelines are concatenated to define the risk classifications of failure modes of CNC lathe machine components in respect of S, O, and D rating scales. The rating scales illustrated in various literature on the application of FMEA/FMECA for CNC lathe machine tool failure analysis are also considered [87-89]. Table 3.1, Table 3.2, and Table 3.3 respectively present the S, O, and D rating scales for the CNC lathe machine. S, O, and D rating varies from 1 to 10. In Table 3.1, the severity rating 1 indicates the least severity, and 10 indicates the most severe case. In Table 3.2, the occurrence rating is framed based on the occurrence probability of component failure, where the occurrence rating of 1 indicates an extremely unlikely occurrence of failure and value 10 indicates the most frequent occurrence of failure, which might be serious. In Table 3.3, the detection rating represents the chance of failure being undetected. The detection rating 1 indicates an almost certain chance for the detection of component failure and value 10 indicates an almost uncertain chance for the detection of component failure. S, O, and D ratings are determined for all the components mentioned in the CNC lathe hierarchy structure (refer to Figure 3.1).

Table 3.1 Severity rating for CNC Lathe

S. No.	**Failure effects severity**	**Ranking Criteria**	***S* scale**
1	Serious	Failure causing harm to the operator without warning	**10**
2	Very Extreme	Failure causing harm to the operator with a warning	**9**
3	Extreme	Failure occurring without any warning and causing no harm to the operator	**8**
4	Major	Failure causing damage to the machine with a warning and causing huge maintenance costs	**7**
5	Significant	Failure has severe effects on the functions of subsystem /component of the machine Significant maintenance costs and production loss	**6**
6	Moderate	Moderately effect on the performance of a subsystem /component Moderate maintenance costs and production loss	**5**
7	Low	Failure has no severe effect on the function or performance of a subsystem /component Low maintenance costs and production loss	**4**
8	Minor	Failure can be solved by minor repair Very low maintenance costs and production loss	**3**
9	Very Minor	Failure has a little effect on machine performance Negligible maintenance costs and production loss	**2**
10	None	Failure has a little or no effect on the performance of the machine	**1**

Table 3.2 Occurrence rating for CNC Lathe

S. No.	**Ranking level**	**Ranking Criteria**	**Occurrence probability**	***O* scale**
1	Frequent	Failure is almost certain	> 0.0600	**10**
2	Very High		0.0500 to 0.0600	**9**
3	High	Failure repetition is expected	0.0400 to 0.0500	**8**
4	High Moderate		0.0300 to 0.0400	**7**
5	Moderate	Failure occurs occasionally	0.0250 to 0.0300	**6**
6	Low		0.0200 to 0.0250	**5**
7	Very Low		0.0175 to 0.0200	**4**
8	Remote	Failure repetition is not expected	0.0100 to 0.0175	**3**
9	Very Remote		0.0080 to 0.0100	**2**
10	Extremely Unlikely	Failure is almost uncertain	< 0.0080	**1**

Table 3.3 Detection rating for CNC Lathe

S. No.	**Likelihood of detection**	**Ranking Criteria**	***D* scale**
1	Almost certain	Design controls will almost certainly detect the potential failure modes	**1**
2	Very High	Very likelihood that the current design controls will detect potential failure modes/task error	**2**
3	High	High chance that the current design controls will detect failure	**3**
4	Moderate-High	Moderately high likelihood that the current design controls detect the potential failure modes before affecting the system performance	**4**
5	Moderate	Moderately likelihood that the current design controls detect the potential failure modes before affecting the system performance	**5**
6	Low	Low likelihood that the current design controls will detect failure modes	**6**
7	Very Low	Very low likelihood that the current design controls will detect failure	**7**
8	Remote	Remote chance that the design controls will detect failure	**8**
9	Very Remote	Defect most likely remains undetected	**9**
10	Almost Uncertain	Failures are not detected	**10**

3.4 Computation of RPN and Maintenance Prioritization

The key part of FMECA is to determine the RPN for failure modes of CNC lathe components, which is computed based on *S*, *O*, and *D* rating values. The conventional method for determining the RPN value of components is by computing the product of *S*, *O*, and *D* rating values of each component (refer to Eq 2.1). As mentioned in ‘section 2.5’, conventional FMEA/FMECA has many drawbacks on practical application. The computations in conventional FMECA are based on the assumption that the input variables are crisp values. Although, due to several uncertainties, these variables are non-crisp in nature, which is the primary cause of disputes about the conventional FMECA technique [87, 107]. These rating values basically represent linguistic variables indicating different risk classifications. Linguistic variables are input/output variables whose values are words or sentences. The qualitative FMECA approach uses linguistic variables to express the risk classification category of severity, occurrence, and detection rating scales. The linguistic terms like certain, uncertain, moderate, low, very low, high, etc. are used to indicate various risk classifications.

In this work, the Fuzzy logic technique is used to assign non-crisp values to these linguist variables. Zadeh [204] in the year 1965, introduced fuzzy sets to assign the linguistic variables to different fuzzy sets. A membership (characteristic) function was defined to correlate the fuzzy sets with linguistic variables. The fuzzy logic was further evolved from this concept. The fuzzy logic computation is used when there are uncertainties in risk factor calculations.

The fuzzy inference engine mostly uses Mamdani's method or Takagi-Sugeno's method [99, 205]. The present work utilizes Mamdani’s method to define a fuzzy FMECA engine, where both the precedent and the succedent are fuzzy propositions. A typical fuzzy logic algorithm proceeds as follow [203, 206]:

i. Fuzzification of quantities

ii. Establishment of fuzzy sets

iii. Establishment of fuzzy rules

iv. Defuzzification of quantities

Fuzzy FMECA is basically a fuzzy decision support system, which offers a more realistic framework for qualitative risk rating scales than traditional crisp values. The methodology for performing fuzzy FMECA are as follows:

i. Create all input and output variables of FMECA in the fuzzy logic platform.

ii. Develop the input membership functions to represent *S*, *O,* and *D*.

iii. Develop the output membership functions to represent RPN

iv. Establish rules to correlate the Fuzzy RPN with the fuzzified *S*, *O,* and *D* linguistic variables.

The membership function for a fuzzy set is a generalization of the characteristic function of crisp sets [204]. Membership functions are used to solve practical problems by experience rather than knowledge. It represents the degree of truth of a valuation. The membership function associates each element with a value in the interval *[0, 1]*. In fuzzy sets, each element is mapped to the interval *[0, 1]* using a membership function. This makes the degree of the truthiness of a statement in fuzzy logic not constrained to either 0 or 1, but to have any values in the range *[0,1]*. Consequently, the fuzzy set with a vague boundary is used to represent crisp values. The establishment of this correlation is known as fuzzification [99]. This fuzziness is best characterized by its membership function. The membership function allows the graphical representation of the fuzzy set.

Simple functions are used to build the membership function for a fuzzy set. The most commonly used base functions include the triangular function, trapezoidal function, Gaussian function, generalized bell function, sigmoid function, etc. [206]. In this work, the Gaussian membership function is used to represent each linguistic variable. The Gaussian function is a smooth, concise notation, and non-zero at all points, which makes it a popular method for specifying fuzzy sets [204-206]. Moreover, polynomial-based curves are commonly used to represent fuzzy membership functions. The Gaussian function is defined using two parameters, mean (μ) and standard deviation (σ), which indicate the center and width of the membership function respectively. Figure 3.2 illustrates a typical Gaussian function representation. The smaller the standard deviation, the narrower will be the bell curve. The standard deviation value is tuned so that the membership functions in a fuzzy set have suitable overlapping to avoid any chance of a gap in the linguistic variable domain. The Gaussian membership function can be represented by Eq (3.1), where *x* can be any of the crisp values of the FMECA input and output variables [206].

$$f(x;\mu,\sigma) = e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{3.1}$$

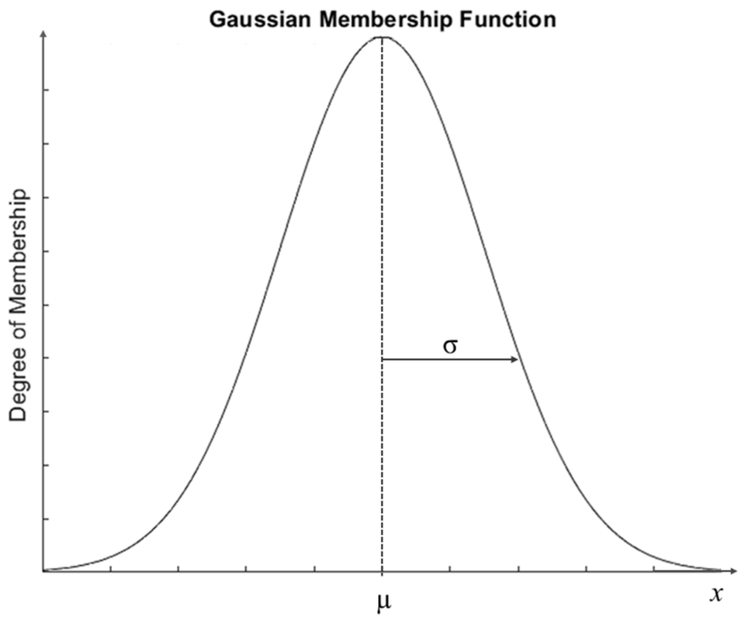


Figure 3.2 Typical Gaussian membership function

In fuzzy FMECA, the variables are not defined by sharp boundaries. The crisp input values of *S*, *O*, and *D* rating scales are transformed into non-crisp fuzzy values from the linguistic terms using membership functions. These linguistic input variables are fed into the fuzzy engine. In order to represent RPN using fuzzy membership functions, a risk classification must be made based on RPNs. RPNs are also represented using linguistic variables. The fuzzy engine returns a linguistic output variable representing RPN, which is defuzzified to obtain a crisp value for RPN.

The input to the defuzzification stage is a fuzzy linguistic variable and the output is a crisp value, which is denoted as the fuzzy RPN. This is a reverse mapping of crisp values from membership functions. The centroid method is the most commonly used defuzzification technique that returns the central point of the area under the fuzzy set, which is a crisp value [100, 203] The overall structure for the fuzzy modeling of FMECA is illustrated in Figure 3.3.

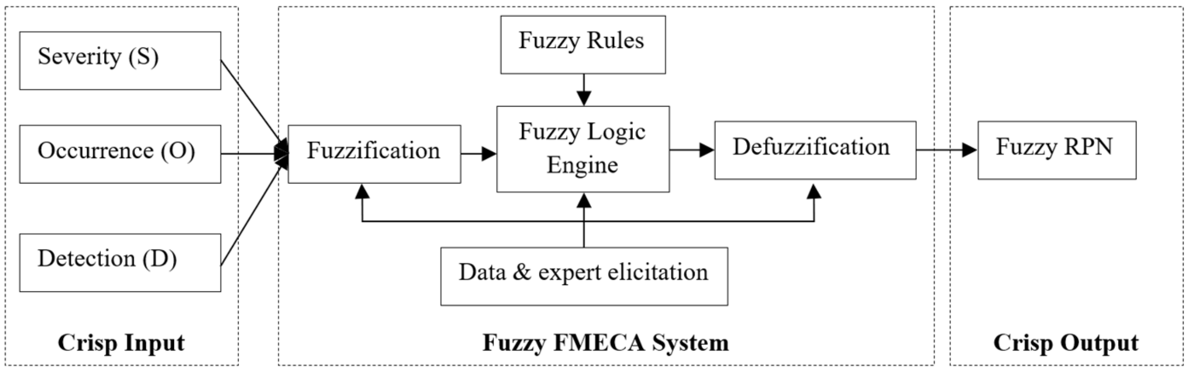


Figure 3.3 Flow diagram for fuzzy modeling of FMECA

The collated field failure data of the CNC lathe is subjected to criticality analysis using FMECA and fuzzy FMECA methods. The major objective of the study is to identify the most critical components and subsystems of a CNC lathe and prepare a priority list for implementing a predictive maintenance strategy. The potential failure modes, potential effects, potential causes, and design controls for detection are identified for every component of the CNC lathe using the field failure data, and further, Tables 3.1, 3.2, and 3.3 are used to assign *S*, *O,* and *D* ratings for each component. The fuzzy FMECA proceeds are shown in Figure 3.3. In order to determine the fuzzy RPNs for failure modes of components, the input and output linguistic variables are fuzzified.

Figure 3.4 shows the fuzzy representation of the severity, occurrence, and detection rating scales. This makes the basic non-crisp inputs for a fuzzy FMECA. Similarly, the output RPNs are fuzzified, but it requires an RPN rating scale for the CNC lathe. Table 3.4 presents the 10-scale risk classification of RPN, which defines a set of RPNs to a particular risk category. Risk classification and ranking criteria for RPN are developed by integrating the conventional RPNs with industrial expert elicitation. Each class is assigned the values from 1 to 10, the value 10 indicates the category with the highest risk and the value 1 indicates the category with the least risk. Like the input membership functions, the Gaussian membership function is used to transform RPN into a fuzzy RPN. Figure 3.5 shows the output membership function for fuzzy RPN. This fuzzy representation of RPNs is used for the defuzzification of the linguistic output variables to give crisp fuzzy RPNs.

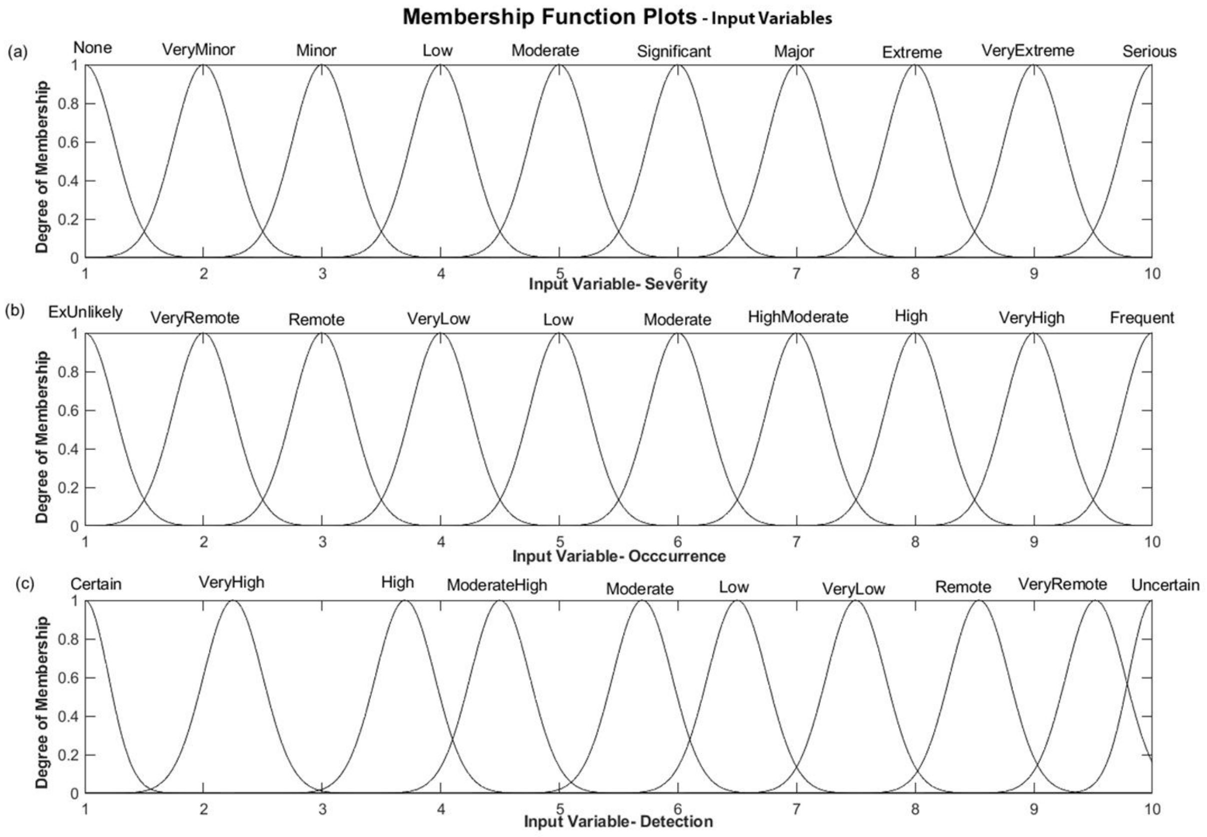


Figure 3.4 Membership function for input variables **(a)** Severity, **(b)** Occurrence, **(c)** Detection

Table 3.4 Failure Classification based on RPN rating scale

S. No.	Linguistic Variable	Ranking Criteria	Rank
1	Very High	$450 \leq RPN \leq 1000$	10
2	High	$300 \leq RPN \leq 449$	9
3	Low High	$217 \leq RPN \leq 299$	8
4	High Medium	$141 \leq RPN \leq 216$	7
5	Medium	$81 \leq RPN \leq 140$	6
6	Low Medium	$50 \leq RPN \leq 80$	5
7	High Low	$30 \leq RPN \leq 49$	4
8	Low	$17 \leq RPN \leq 29$	3
9	Very Low	$9 \leq RPN \leq 16$	2
10	None	$1 \leq RPN \leq 8$	1

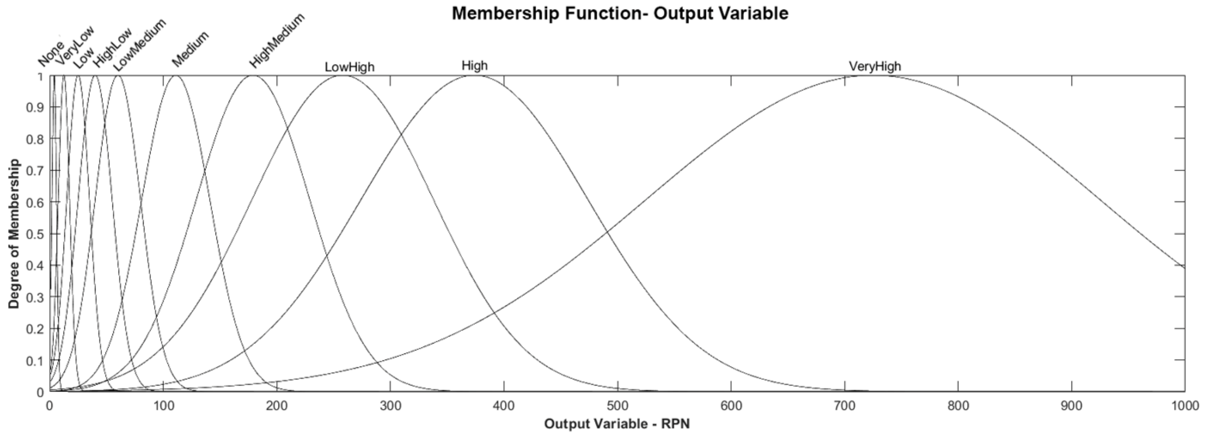


Figure 3.5 Membership function for output variable-RPN

The fuzzy rules are defined to correlate the input and output membership function. The *if-then* rule is used to establish the fuzzy relation between the inputs *S*, *O*, *D*, and the output RPN. Following the 10-scale ratings for *S*, *O*, and *D*, 1000 *if-then* rules are developed using the information extracted from expert elicitation. These rules are intended to portray every possible combination of *S*, *O*, and *D* rating scales. *If-then* rules are defined as follows:

*"**If** Severity is Serious **and** Occurrence is Moderate **and** Detection is Uncertain **then** RPN is Very High"*

*"**If** Severity is Minor **and** Occurrence is Frequent **and** Detection is Uncertain **then** RPN is High"*

*"**If** Severity is Major **and** Occurrence is Remote **and** Detection is Certain **then** RPN is Low"*

Similar 1000 rules are defined to represent the fuzzy FMECA of the CNC lathe. A typical worksheet for qualitative FMECA is used to display the complete FMECA and fuzzy FMECA of a CNC lathe machine as presented in Table 3.5. Columns of the FMECA worksheet include part name, potential failure modes, the potential effects and causes of the respective failure mode, the current machine controls detection, severity, occurrence, and detection values for the failure mode of the component, and RPNs. An additional column is included in the conventional FMECA worksheet to display fuzzy RPNs. This could provide a clear comparison between the conventional RPN and the fuzzy RPN. The FMECA relates the potential failure modes to potential effects and root causes, which give a clear knowledge about the failure of a component.

Table 3.5 FMECA sheet for CNC Lathe with RPN and Fuzzy RPN

Part No.	**Part Name**	**Potential failure mode(s)**	**Potential effect(s) of failure**	**Potential cause(s) of failure**	**Current machine controls Detection**	**Risk assessment**				**Fuzzy RPN**
						S	***O***	***D***	**RPN**	
1	Turret	Indexing error	Affects engaging of the cutting tool, Bearing damage	Trapping of chips	Visual Inspection	5	6	7	**210**	**255**
		Lock problem	Improper Clamping, Chance of accident	Wear, Trapping of chips	Visual Inspection	6	10	3	**180**	**178**
2	Turret Motor	Noisy Operation, Overheating	Winding or bearing damaged	Wrong position of the tool	Visual Inspection	4	1	3	**12**	**13.5**
		Electric short circuit	Faulty Indexing	Winding burns out	Electric Failure Alarm	6	1	7	**42**	**40.9**
3	Tool Holder	Improper clamping, Play	Tool slip, increased tool charter	Fatigue	Visual Inspection	3	5	2	**30**	**40.1**

4	Chuck	Worn out, Sudden loss of grip	Workpiece runout, Improper clamping	Overload, material fatigue	Inspection using instruments and gauges	6	10	5	**300**	**280**
5	Drawbar	Worn Thread	Improper workpiece clamping	Thread failure	No job clamping	5	10	3	**150**	**178**
6	Pulley	Loosen Key, Worn Sheave	Pulley play, Belt slip, Belt break	Wear, Spindle bearing fault	Noise and jerking movement of the belt	4	1	5	**20**	**17.7**
7	Spindle Motor	Noisy Operation, Overheating	Bearing damaged, Over Heating	Motor bearing or bush failure	Noise and overheating, Increased Vibrations	7	3	3	**63**	**66**
		Electric short circuit	Winding Burn out, Stopped Working	Winding burns out	Overheating, Electric failure Alarm	6	2	6	**72**	**65**
8	Spindle Belt	Improper tensioning, Belt fatigue, Worn Belt,	Broken Belt, improper power transmission, noisy operation, overheating	Fatigue, Wear, pulley misalignment	Increased noise of slipping and chatter	3	8	4	**96**	**83.2**

9	Spindle Bearing	Bearing deformation	Noisy Operation, Overheating, Spindle runout, Damage bearing elements	Wear and Deformation of bearing elements, Overload, Misalignment	Increased noise and vibration, Chatter mark on Workpiece	7	10	7	**490**	**568**
10	Spindle Cooling Fan Motor	Over Heating	Noisy operation, Overheating	Short circuit, Contamination	Fan stopped working	5	4	3	**60**	**66**
11	X & Z Axis Servomotor	Worn bearing, Electric short-circuit	Noisy operation, overheating	Overload due to improper slide movement	Overheating, Noise	8	4	6	**192**	**179**
		Worn coupling	Vibrations	Misalignment	Jerking Noise	4	3	5	**60**	**45.8**
12	Axis Belt	Worn Belt, Belt fatigue, Improper tensioning	Overheating, Noisy operation	Fatigue, Wear, pulley misalignment	Visual Inspection	4	1	6	**24**	**22.7**
13	Axis Slide	Not Smooth, Play	Lack of precision	Misalignment, wear	no smooth movement, increased load on motor	5	3	3	**45**	**42.6**

14	Ball Screw	Bent Screw, Play	Misalignment	Chip trapping, wear, Fatigue	Noisy Unsmooth operation	6	1	5	**30**	**27.8**
15	Ball Screw Bearing	Unsmooth operation	Noisy operation, Overheating	Deformation, chip trapping	Noise, Vibration	6	5	5	**150**	**129**
16	Oil Seal	Leakage	Pressure loss, Wastage of oil	Contaminants	Visual Inspection	3	3	5	**45**	**40.4**
17	Lubrication pump	Vane Blocked	Improper lubrication	Entry of chips, deformation	Drop in oil pressure	2	1	6	**12**	**13.3**
18	Lubrication Motor	Burnt Winding	Stop working	Oil leakage into the winding	Oil Spilling	5	1	5	**25**	**23.7**
		Noisy Operation, Overheating	Noisy Operation	Wear of bearing	Noise, Overheating	4	2	6	**48**	**40.2**
19	Oil Tank and Piping	Leakage, Blocked or improper flow	Inadequate Lubrication, Lubricant wastage	Blockage due to Contaminants	Visual Inspection	3	4	4	**48**	**40**
20	Coolant Pump	Worn or blocked vanes	Inadequate pressure,	Blocked filter, contaminants	Inadequate flow of coolant	2	1	3	**6**	**4.79**

			Improper Coolant flow							
21	Coolant Motor	Burnt Winding	Stop working	Leakage of coolant into the winding	stop working	6	1	3	**18**	**26.8**
		Noisy Operation, Overheating	Noisy Operation	Wear of bearing	Noise, Overheating	4	2	4	**32**	**30**
22	Coolant Tank and Piping	Leakage, Blocked or improper flow	Inadequate Coolant	Blockage due to Contaminants	Visual Inspection	2	1	5	**10**	**5.41**
23	Encoder Coupling	Fail to encode	Wrong axis and turret position	Fatigue, aging	Wrong positioning	5	1	5	**25**	**23.7**
24	Encoder Belt	Rubbing and Overheating	Wrong axis and turret position	Fatigue, wear	Noise, vibration, no movement	3	1	6	**18**	**19.3**

In Table 3.5, it is observed that the same RPN is produced for different combinations of *S*, *O*, and *D* values with the conventional FMECA, which is one of its major drawbacks. On the other hand, all the fuzzy RPNs are unique. For example, tool holder and ball screw have (*S*, *O*, *D*) combinations as (3, 5, 2) and (6, 1, 5) respectively, which give the same RPN 30 with conventional FMECA. The same set of input variables for tool holder and ball screw in fuzzy FMECA gives fuzzy RPNs 40.1 and 27.8 respectively. Thus, fuzzy RPN overcomes the major drawback of conventional FMECA in calculating RPN and makes it better for risk prioritization. This is due to the result of the fuzzification of the linguistic variables of input and output parameters.

RPN is a relative quantity indicating the risk associated with different failure modes of the components of machinery. The RPN of a component is calculated as the aggregate of RPNs of all failure modes of that component as represented by Eq (3.2). The RPN of a subsystem is calculated as the sum of RPNs of all components within that subsystem as represented by Eq (3.3) [89, 96].

$$\text{RPN}_{\text{Component}} = \text{RPN}_{\text{Failure Mode 1}} + \text{RPN}_{\text{Failure Mode 2}} + \ldots + \text{RPN}_{\text{Failure Mode n}} \qquad (3.2)$$

where n is the total number of failure modes of that particular component.

$$\text{RPN}_{\text{Subsystem}} = \text{RPN}_{\text{Component 1}} + \text{RPN}_{\text{Component 2}} + \ldots + \text{RPN}_{\text{Component m}} \qquad (3.3)$$

where m is the total number of components under the considered subsystem.

The RPN of components is calculated using Eq (3.2). A priority rank is given to the CNC lathe components based on conventional RPN and fuzzy RPNs from larger to smallest as presented in Table 3.6. Due to the repetition of RPNs in conventional FMECA, there is a chance of a tie between the priority ranks of components. In such cases, the product of *S* and *O* is considered and the component having a higher product value is given a higher rank. However, this problem does not appear when the ranking is based on a fuzzy RPN, where all values are unique. Therefore, the risk priority ranking based on a fuzzy RPN is considered for the predictive maintenance of the CNC lathe. It is observed that spindle bearing has the highest RPN and the fuzzy RPN 490 and 568, respectively. Hence, the spindle bearing is reported as the most critical component of a CNC lathe machine, followed by turret and chuck.

Table 3.6 Criticality Ranking of CNC Lathe Components based on RPN and Fuzzy RPN

Part No.	**Part Name**	**RPN**	**RPN Priority Rank**	**Fuzzy RPN**	**Fuzzy RPN Priority Rank**
1	**Spindle Bearing**	**490**	**1**	**568**	**1**
2	Turret	390	2	443	2
3	Chuck	300	3	280	3
4	X & Z Axis Servomotor	252	4	224.8	4
5	Drawbar	150	5	178	5
6	Spindle Motor	135	7	131	6
7	Ball Screw Bearing	150	6	129	7
8	Spindle Belt	96	8	83.2	8
9	Spindle Cooling Fan Motor	60	10	66	9
10	Lubrication Motor	73	9	63.9	10
11	Coolant Motor	50	12	56.8	11
12	Turret Motor	54	11	54.5	12
13	Axis Slide	45	14	42.6	13
14	Oil Seal	45	15	40.4	14
15	Tool Holder	30	16	40.1	15
16	Oil Tank and Piping	48	13	40	16
17	Ball Screw	30	17	27.8	17
18	Encoder Coupling	25	18	23.7	18
19	Axis Belt	24	19	22.7	19
20	Encoder Belt	18	21	19.3	20
21	Pulley	20	20	17.7	21
22	Lubrication pump	12	22	13.3	22
23	Coolant Tank and Piping	10	23	5.41	23
24	Coolant Pump	6	24	4.79	24

Further, the RPNs of subsystems are calculated as the sum of the RPNs of each component belonging to that subsystem using Eq (3.3). Similarly, the fuzzy RPNs are also calculated for the subsystems. These fuzzy RPNs are utilized to prepare a maintenance priority ranking for CNC lathe subsystems as presented in Table 3.7. A comparison of the conventional and fuzzy RPN is provided and the subsystems are arranged according to the fuzzy RPN priority ranking. It is observed that the ranking based on fuzzy RPN is more in agreement with industrial expert elicitation. The spindle unit of the CNC lathe machine is identified as the most critical subsystem with the conventional and fuzzy RPNs 781 and 848.2, respectively. The criticality analysis based on both conventional and fuzzy RPNs has established the spindle unit as the most critical subsystem. The spindle unit of a CNC Lathe Machine tool includes subsystems such as spindle motor, spindle bearings, spindle belt, and spindle cooling fan. The subsystem fuzzy RPN values of the spindle motor, spindle bearings, spindle belt, and spindle cooling fan are obtained as 131, 568, 83.2, and 66 respectively. The highest fuzzy RPN value is obtained for the spindle bearings whose potential cause of failure is reported as wear and deformation of beaning components. The fuzzy RPN values of the spindle motor, spindle belt, and spindle cooling fan are far-off for comparison with that of spindle bearings. Hence, all other causes of failure such as overheating, short-circuit, contamination, etc. have negligible contributions toward the failure spindle unit.

Hence, it can be stated that the predictive maintenance strategy must be implemented for the spindle unit of a CNC lathe machine tool. The turret, chuck, and linear axis subsystems also have high fuzzy RPNs. These subsystems can also be considered for predictive maintenance. The other subsystems might be considered for preventive or reactive maintenance.

Table 3.7 Criticality Ranking of CNC Lathe Subsystem based on RPN and Fuzzy RPN

Subsystem	RPN	RPN Priority Rank	Fuzzy RPN	Fuzzy RPN Priority Rank
Spindle	**781**	**1**	**848.2**	**1**
Turret	474	3	527.5	2
Chuck	470	4	475.7	3
Linear Axis	501	2	446.9	4
Lubrication	178	5	157.6	5
Cooling	67	6	67	6
Encoder	43	7	43	7

3.5 Summary

This chapter presents a systematic methodology for the criticality analysis and maintenance prioritization of the CNC lathe machine tool for the application of picking out critical lathe components and subsystems for predictive maintenance. Criticality analysis of CNC lathe is performed to identify the most critical subsystems and their potential failure modes from a maintenance perspective and hence limit the implementation of predictive maintenance to the identified critical subsystems. One of the most widely used criticality analysis techniques FMECA, which is improved with fuzzy logic computation is utilized for risk prioritization of the CNC lathe machine tool. The failure modes of components and subsystems of the CNC lathe are identified and the risk associated with each component and subsystem is determined. Furthermore, a maintenance priority rank is generated based on the risk factor associated with the components with respect to the failure modes. Industrial field failure data and expert elicitation constitute major input for performing failure and criticality analysis. This data is used to calculate the RPNs following the conventional and fuzzy improved FMECA.

FMECA relates the potential failure modes to potential effects and root causes. This knowledge can be utilized in the phenomenon of sensors selection and installation for the condition monitoring of critical components. The spindle unit of a CNC lathe is identified as the most critical subsystem with the highest RPN, followed by the turret, chuck, and linear axis. The wear and deformation of bearings causing increased noise and vibration are identified as the potential failure modes and failure effects for the lathe spindle unit. The comparison of the results of conventional and fuzzy FMECA highlights the benefits of fuzzy FMECA over conventional methodology. The fuzzy FMECA results seem to be more reasonable and in agreement with the industrial data and expert elicitation. The study also proves that the primary drawbacks of conventional FMECA are eliminated with the implementation of fuzzy logic computational techniques.

Chapter 4

Condition Monitoring Data Acquisition and Data Preparation

4.1 Introduction

In the present chapter, the experimental setup for condition monitoring data acquisition and data preparation for prognostic regression analysis is described. This chapter includes the fabrication of an accelerated run-to-failure experimental test rig with sensors and a data acquisition system for condition monitoring data acquisition and the conversion of raw condition monitoring data into useful machinery health degradation information for prognostic regression analysis. The chapter also discusses the standardization of condition monitoring data and response variables.

4.2 Experimental Test Rig

The predictive maintenance analysis of the lathe spindle unit requires machinery health degradation data from the initial healthy state to the final faulty state. The available literature largely has utilized the open-source aircraft turbofan engine degradation data [19-24, 31-37, 40] or bearing degradation data [25- 27, 39] for validating their RUL estimation algorithms. A few researchers have generated their machinery degradation data like reciprocating compressor degradation data [28], cutting tool wear monitoring data [29], gear failure data [30], can-making machine degradation data [38], etc.

A spindle test-bed built by TechSolve [207] used a frequency drive, electric motor, Poly-V belt transmission, and simplified spindle using two bearings identical to the ones used in the horizontal machining center. Figure 4.1 (a) shows the spindle testbed including the motor, the belt transmission, and the actual spindle. A loading mechanism pulling on the nose of the spindle was added to accelerate the degradation. The force pushing down on the spindle nose was kept approximately constant throughout all tests. The section of the loading mechanism is located under the supporting stand (see Figure 4.1 (b)). A uniaxial accelerometer was placed on the spindle housing, on top of the back bearing. A thermocouple was inserted in a hole drilled into the spindle housing, close to the back bearings outer surface.

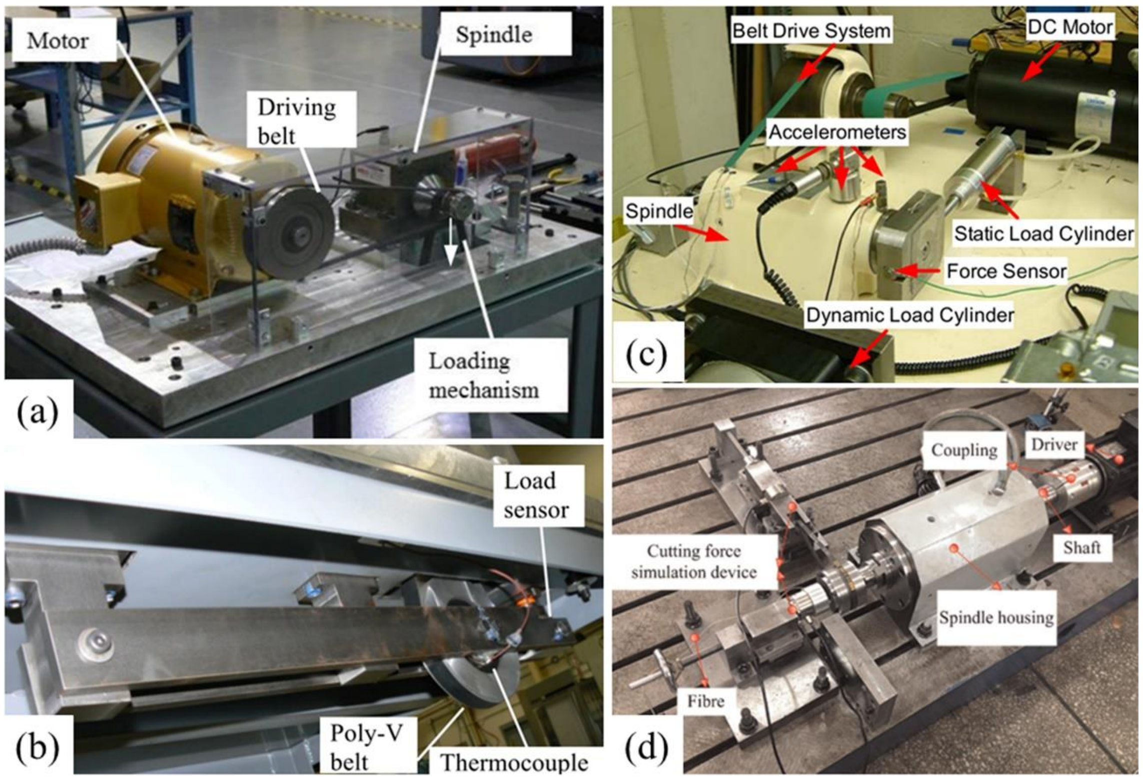


Figure 4.1 Machine tool spindle test rigs in literature (a) and (b) [207], (c) [208], and (d) [209]

Similarly, Figure 4.1 (c) [208] shows a spindle is supported by four angular contact ball bearings of 42 mm outer diameter on its front and rear ends. Two air cylinders (static and dynamic load cylinder) apply constant and impulsive loads to the spindle, simulating the spindle operation conditions under stable preload (e.g., when machining a workpiece under constant speed and feed) or shock input (e.g., when impacted due to tool-workpiece interaction). Four accelerometers were placed at the front and rear ends of the spindle, within the loading and unloading zones of the bearings, to measure their vibrations. Literature proposing a methodology for obtaining an optimal bearing preload to improve the spindle work accuracy of machine tools used a similar machine tool spindle test rig that includes the driver, coupling, shaft, housing, and cutting force simulation device. The experimental test rig is shown in Figure 4.1 (d) [209]. Fiber Bragg grating (FBG) temperature sensors are utilized to measure the temperature rise at the bearing outer ring.

In the present work, an accelerated run-to-failure experimental test setup is fabricated to acquire spindle health degradation data. The experimental test rig primarily comprises a lathe spindle assembly, a drive motor with the belt-pulley arrangement, and a loading arrangement. The speed (rpm) of the spindle unit is varied for different runs by varying the driver pulley ratio. The loading arrangement is placed replacing the chuck of the lathe spindle. The arrangement applies a constant radial load on the spindle and it also ensures that there is no shock load acting on the spindle. As in any mechanical system, the vibration signal is identified as the most suitable parameter for condition monitoring of the spindle unit test rig. 'PCB-603C01' accelerometer sensors are employed for vibration monitoring of the spindle unit. The accelerometer sensors are mounted to the spindle housing by the adhesive mounting technique. A mounting pad is first fixed firmly to the spindle housing using superior metal adhesives. Further, the accelerometer is screw fitted to the mounting pad. The mounting pads offer a very high-frequency response which is comparable to that of adhesives and stud mounting, and this could avoid the damage of a sensor during the removal process. The accelerometer model has a sensitivity of 100 mV/g, a frequency range of 0.5 to 10000 Hz, and a measurement range of ±50 g (g = 9.8m/s2). The development of a fault or deterioration in the condition of machines is indicated by an increase in overall vibration levels. 'NI-9234' sound and vibration module with 'cRIO 9171' data acquisition (DAQ) system are used to record monitored vibration signals. Table 4.1 presents the detailed specifications of the equipment the vibration sensor, the data acquisition module, and the data acquisition interface system used in the experimental setup. The accelerometer sensors are mounted directly on the top of spindle bearings and are connected to the data acquisition device. The data acquisition system acquires vibration data from the accelerometers and transfers it to the connected computer storage. The data flow across various components in the experimental setup is illustrated in Figure 4.2.

Table 4.1 Detailed specification of equipment used in the experimental setup

Equipment Type	**Equipment Name**	**Equipment Specification**
Vibration Sensor	PCB 603C01 Accelerometer	▪ Sensitivity: (±10%)100 mV/g (10.2 mV/(m/s^2)) ▪ Frequency Range: 0.5 to 10000 Hz ▪ Sensing Element: Ceramic ▪ Measurement Range: ±50 g (±490 m/s^2) ▪ Weight: 1.8 oz (51 gm)
Data Acquisition Module	NI 9234 Sound & Vibration Module	▪ 4 Channel ▪ 51.2 kS/s/channel Simultaneous Sampling ▪ Signal Range: ±5 V, 24 Bit ▪ AC Coupling & AC/DC Coupling ▪ IEPE type
Data Acquisition System (Computer Interface)	cRIO 9171 USB DAQ System	▪ 1-Slot, USB CompactDAQ Chassis ▪ Plug-and-play simplicity of USB to sensor ▪ Supported on DAQmx Driver Software

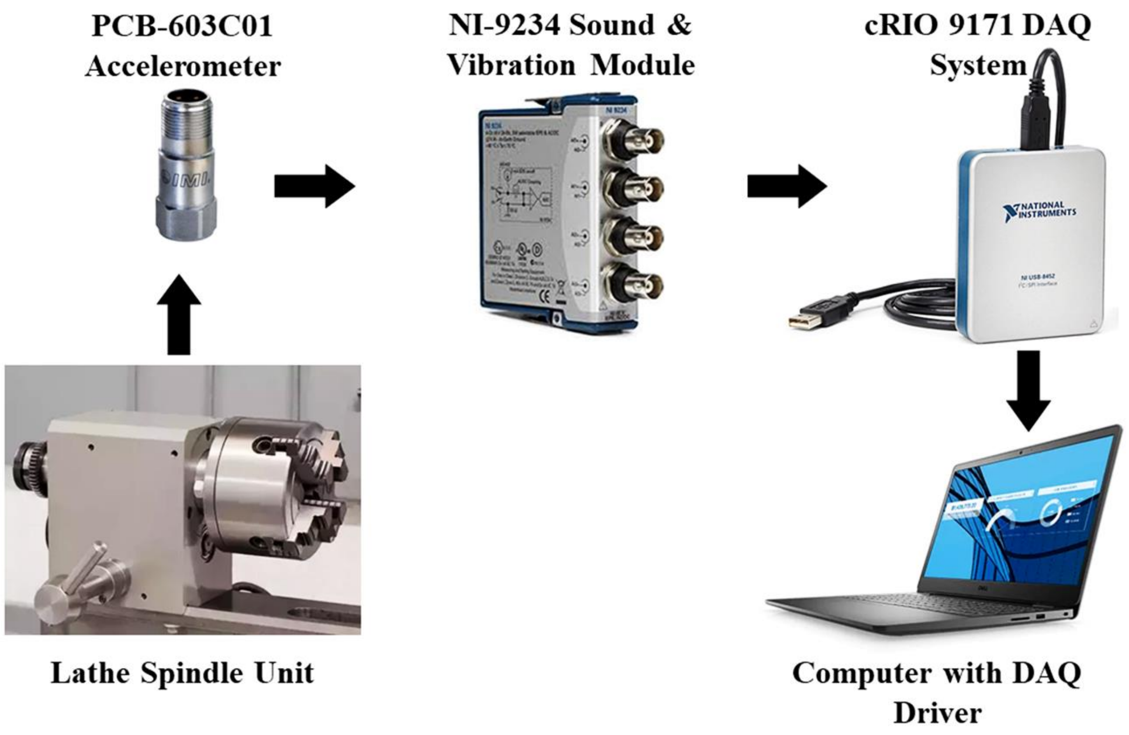


Figure 4.2 Data flow across various components in the experimental setup

The entire experimental setup is assembled on a heavy machine worktable using anti-vibration gaskets to minimize the presence of noise in the condition monitoring data. The accelerated run-to-failure experimental test rig with component labels is shown in Figure 4.3. The accelerometer data is acquired at a sampling rate of 25.6 kHz following a time-step of once every 60 seconds. Each of these acquired data packets represents the health state of the spindle unit at that time step. The machine health degradation data is represented by these packets in their respective time-steps. Each degradation data represents the health degradation of the spindle unit from a healthy state to a faulty state. The faulty state of any machine component is when it is not able to perform its operation in a predefined desired manner. The faulty state of a lathe spindle unit can be defined as the point at which it is unable to produce machined parts within the tolerance limits of surface finish and dimensional accuracy. Accelerometer reading (vibration amplitude) above 40 g is defined as the failure state of the experimental test rig considered.

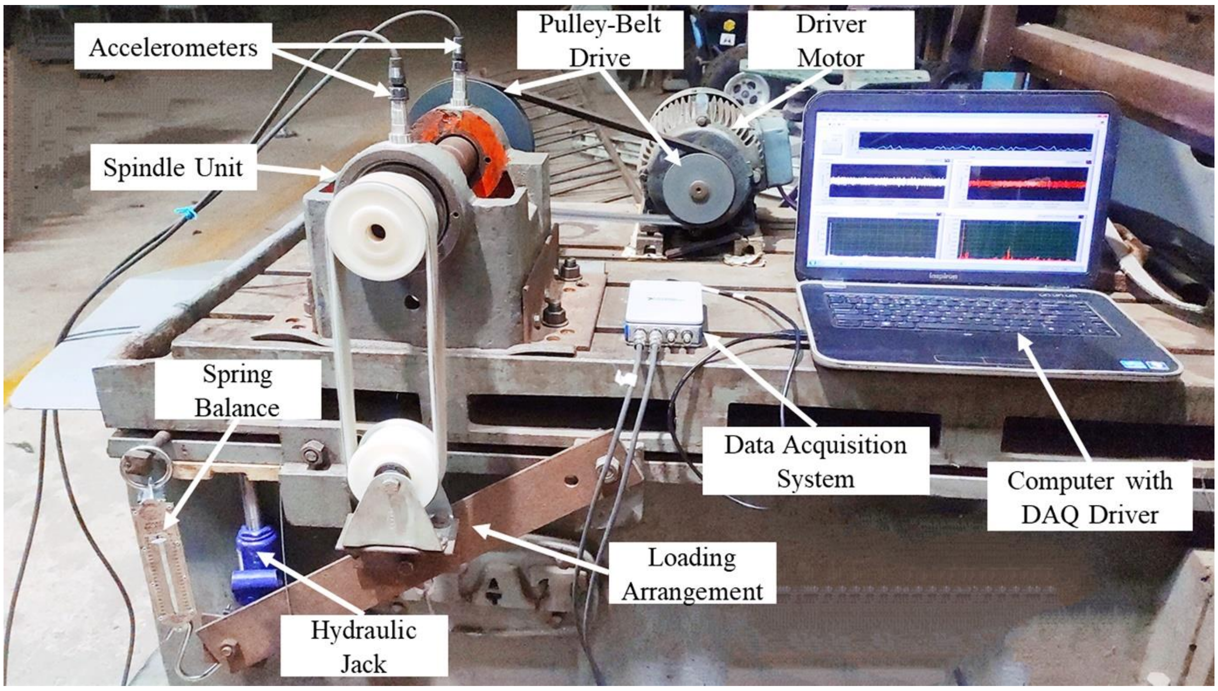


Figure 4.3 Accelerated run-to-failure experimental test rig

The lathe spindle health degradation data includes 14 accelerated run-to-failure vibration signal datasets performed at a constant load for five different rotational speeds 823 rpm, 900 rpm, 1400 rpm, 1800 rpm, and 2520 rpm. The chosen rotational speeds can simulate the maximum possible variation in the functioning of the lathe spindle for the observed vibration signals. The rotational speeds above 2520 rpm produced very high and rapidly varying vibration signals in a short run-time and rotational speed below 823 rpm failed to produce any significant variation (increase) in the vibration signal even after long run-time. Thus, the five rotational speeds are chosen on practical grounds. Figure 4.4 shows all 14 acquired raw vibration data presented on time series plots with their respective rotational speeds.

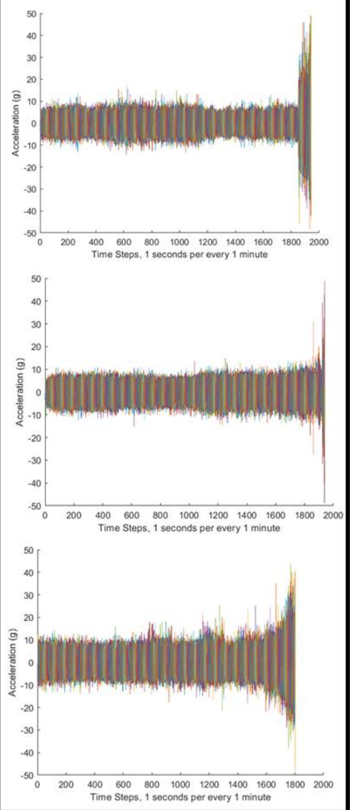
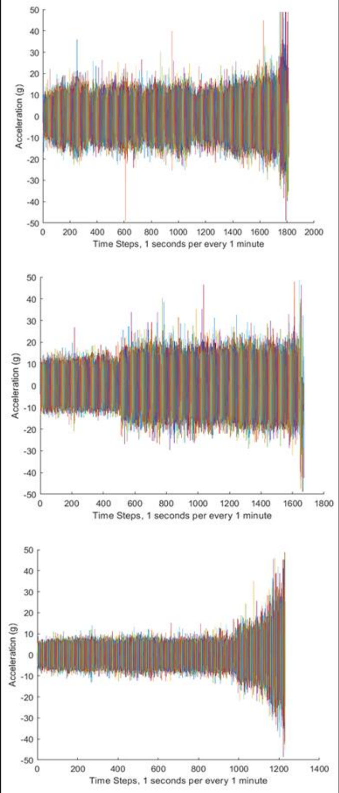
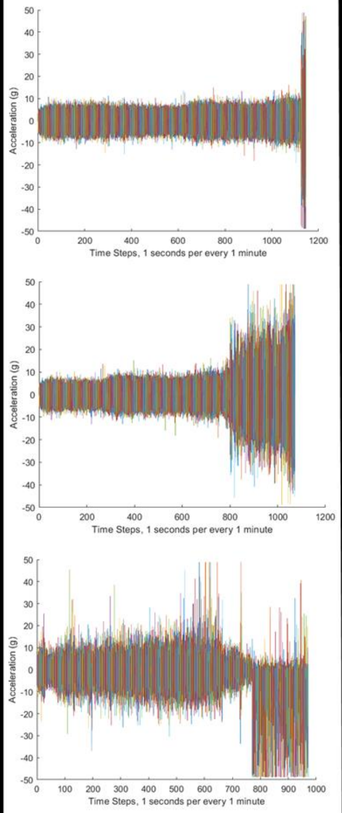
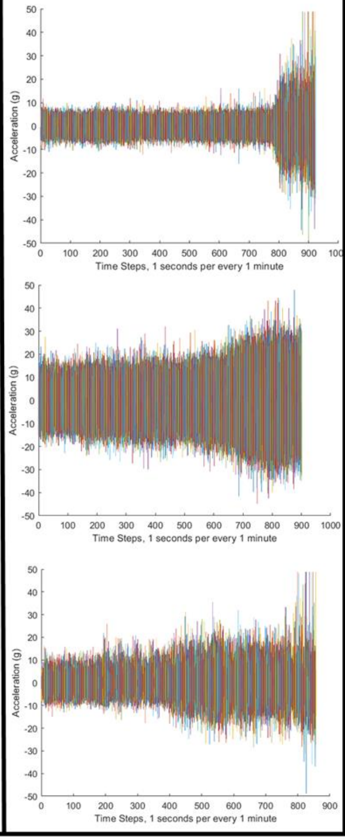
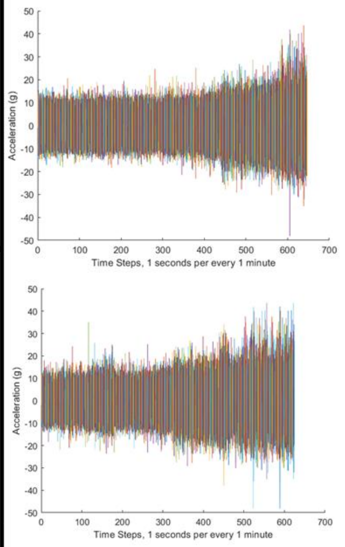
Spindle Speed (rmp)	823	900	1440	1800	2520
Raw Vibration Data in Time Series Plot					

Figure 4.4 Acquired 14 raw vibration data presented on time series plot

Table 4.2 lists the lathe spindle health degradation data with maximum life obtained for each run denominated in time-steps (one time-step equals 60 seconds). The spindle bearings (32211, 32212 tapered roller bearings) are replaced with new bearings and the spindle unit is completely reconditioned before each failure runs.

Table 4.2 Lathe spindle health degradation data with maximum life

S. No.	Speed (rpm)	Maximum Life (Time-Steps)
1	823	1942
2		1926
3		1816
4	900	1802
5		1674
6		1230
7	1440	1146
8		1072
9		972
10	1800	923
11		901
12		857
13	2520	647
14		624

4.3 Vibration Signal Processing

4.3.1 Feature Extraction

As mentioned in the introduction section of the thesis, the machinery degradation data comprising at most a single vibration sensor data might cause underfitting of the deep learning model. Vibration signal signature features revealing superior machinery degradation patterns are considered for deep network training. The raw vibration signals are first subjected to wavelet denoising with ‘db5’ mother wavelet at level-4 to filter out high-frequency noise signals [50]. Then, statistical features are extracted from the denoised signal using time, frequency, and time-frequency domain analysis. The features extracted from the denoised vibration signal include; mean, standard deviation, root mean square (RMS), root sum of square (RSSq), peak-to-peak, crest factor, impulse factor, margin factor, skewness, kurtosis, shape factor, mean frequency, time domain energy, wavelet energy, spectral entropy, spectral kurtosis, Shannon entropy, Log entropy, normal entropy, approximate entropy, joint moment, and mean peak frequency. [51, 52] (See Appendix I for MATLAB code for vibration signal signature feature extraction). Table 4.3 summarizes the statistical features with their computation formulae.

Table 4.3 Statistical features in time, frequency, and time-frequency domain

FI	Feature	Formula	FI	Feature Index	Formula
1	***Mean***	$M = \frac{1}{N}\sum_{i=1}^{N} x_i$	12	***Peak-to-peak***	$P2P = \max\lvert x_i\rvert - \min\lvert x_i\rvert$
2	***Standard deviation***	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})}$	13	***RSSq***	$RSSq = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert^2}$
3	***RMS***	$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	14	***Crest factor***	$CF = \frac{max\lvert x_i\rvert}{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
4	***Impulse factor***	$IF = \frac{max\lvert x_i\rvert}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}}$	15	***Margin factor***	$MF = \frac{max\lvert x_i\rvert}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert^2}}$
5	***Skewness***	$Sk = \frac{\sum_{i=1}^{N}(x_i - m)^3}{(N-1)\sigma^3}$	16	***Kurtosis***	$Ku = \frac{\sum_{i=1}^{N}(x_i - m)^4}{(N-1)\sigma^4}$
6	***Shape factor***	$SF = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$	17	***Mean frequency***	$meanfreq = \frac{1}{N}\sum_{i=1}^{N} f_i$
7	***Energy (Time domain)***	$E = \sum_{i=1}^{N} x_i^2$	18	***Wavelet energy***	$WE = \sum_{i=1}^{N}\frac{\omega t_{\emptyset}^2(i)}{N}$
8	***Spectral entropy***	$e(p) = -\sum_{i=1}^{n} p(x_i)\, log_2(p(x_i))$ *P(.) = probability function*	19	***Spectral kurtosis***	$K(f) = \frac{\lvert S(t,f)\rvert^4}{(\lvert S(t,f)\rvert^2)^2} - 2, f \neq 0$ *S(.)=STFT of* x_i*, t=time, f=frequency*
9	***Shannon entropy***	$sEntropy = -\sum_{i=1}^{N} x_i^2 \log x_i^2$	20	***Log entropy***	$lEntropy = \sum_{i=1}^{N} \log x_i^2$
10	***Normal entropy***	$nEntropy = \sum_{i=1}^{N}\lvert x_i\rvert^p$ *p = signal threshold*	21	***Approx. entropy***	$AE = \emptyset_m - \emptyset_{m+1}$ $\emptyset_m = (N - m + 1)^{-1}\sum_{i=1}^{N-m+1}\log(N_i)$
11	***Joint moment***	$momentJ = \iint t^n w^m P(t,\omega)\,dt\,d\omega$ m= *order, P(.) = marginal distribution, t=first temporal moment, ω=spectral time-freq. moment*	22	***Mean peak frequency***	$meanPeakFreq = \frac{1}{T}\int_0^T argmax_w(Sp(t,\omega))dt$ *Sp(.) = Spectrogram, ω=individual frequency*

***FI = Feature Index**

4.3.2 Feature Selection

The vibration features from 14 failure runs are concatenated to form a compound dataset, which is then subjected to an NCA-based feature selection criterion. A regularized NCA algorithm is executed using the compound feature dataset representing all 14 failure runs to determine the weighting vector corresponding to each feature. The NCA result is illustrated in Figure 4.5 showing the weighting vectors and corresponding feature indices (FI). The features with their indices are presented in Table 4.3. A relative threshold value for feature weights equals to 15.0 is assigned from random trials as the cut-off criterion to select the most relevant feature subsets for all 14 failure datasets. The features standard deviation, RMS, energy, normal entropy, peak-to-peak, RSSq, log entropy clears the set threshold and are selected for the RUL estimation of the lathe spindle unit using the data-driven prognostic algorithms.

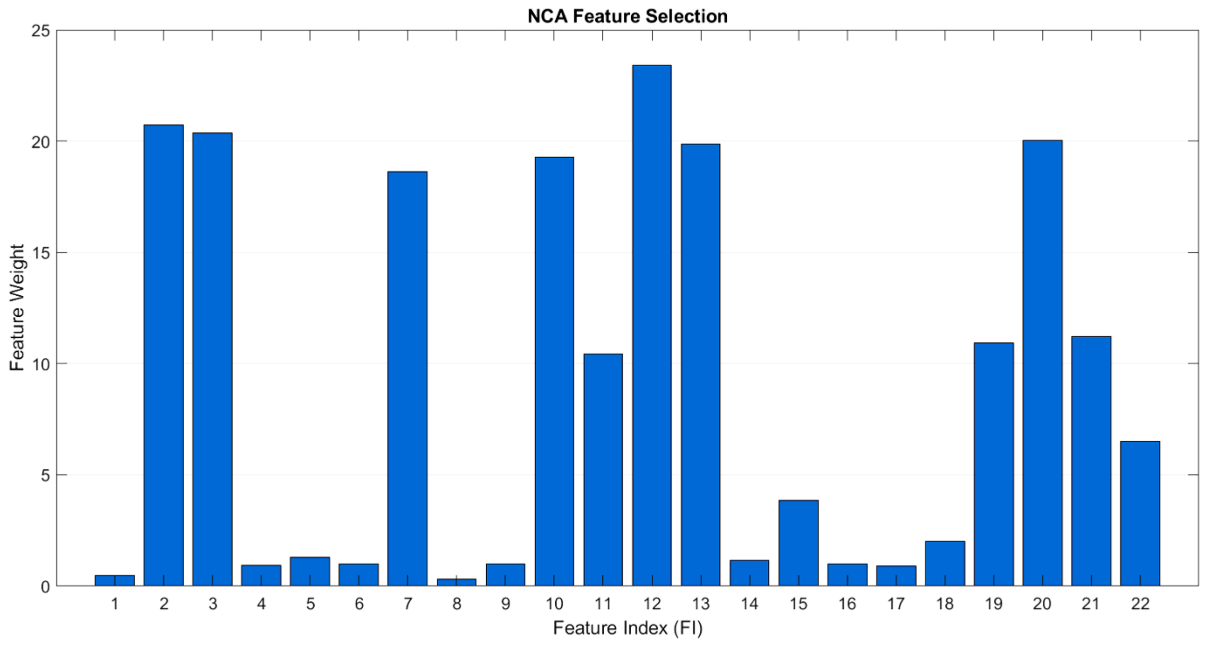


Figure 4.5 NCA feature selection - Feature weights and corresponding feature index

All the remaining features are discarded at this stage of analysis. Training the prognostic algorithms using these selected features could maximize the prediction accuracy of evolved predictive models. (See Appendix II for MATLAB code for NCA regression-based feature selection)

4.4 Preparing Data

All the selected features representing a single machine (lathe spindle) failure run constitutes a machine health degradation dataset. In preparing data for predictive analytics, the available dataset is first divided into the training set, validation set, and testing set. The training dataset is used to train the prognostic model by minimizing the error between the predicted and actual values. The validation dataset is used to select an evolved predictive model before possible overfitting. The testing dataset is used to evaluate the generalization capability of the trained model. The training, testing, and validation dataset are randomly selected following an 80:10:10 split ratio. The total of 14 machine health degradation datasets is divided into 10 training datasets, 2 validation datasets, and 2 testing datasets.

This time-series training dataset is then normalized to have zero mean and unit variance. The validation and testing datasets are also normalized for the same mean and standard deviation. The normalized time-series data $S_n(t_i)$ for any time-series dataset $S(t_i)$ can be expressed as:

$$S_n(t_i) = \frac{S(t_i) - mean}{std} \tag{4.1}$$

where, *mean* is the mean of the training dataset, and *std* is the standard deviation of the training dataset [53]. The time-steps from the initial healthy state to the final faulty state represents the response variable, which is the RUL at each instance from healthy to faulty state. It is assumed that the lathe spindle does not start to degrade at the beginning healthy state of each failure runs instead, a response clip of around 20% of total life is applied until when there is no health degradation or drop in RUL [32, 40]. Beyond the response clip region, the RUL drops in each time step to reach zero. Figure 4.6 portrays a typical perspective of response clipping for time series data.

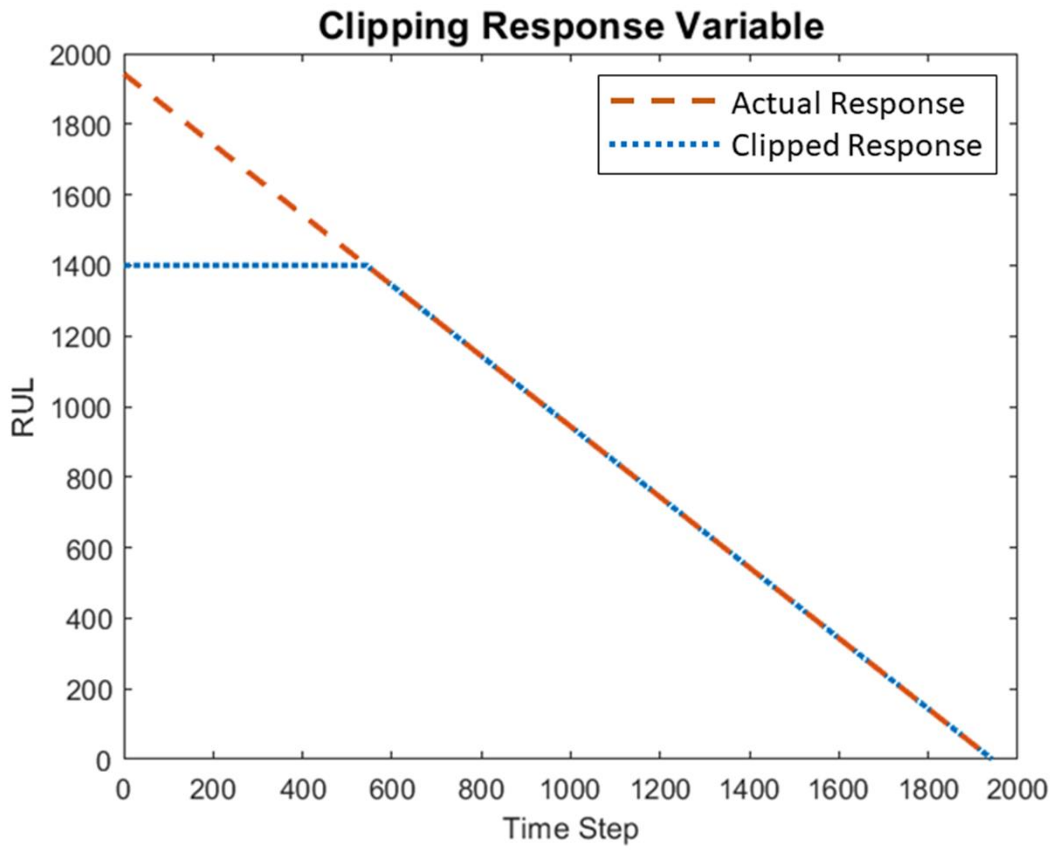


Figure 4.6 Typical clipped response variable

4.5 Summary

The vibration signals representing lathe spindle health degradation are acquired from an accelerated run-to-failure experimental test rig are first analyzed to extract time, frequency, and time-frequency features. These extracted vibration signature features are then subjected to NCA-based feature selection criteria to identify the relevant features for prognostic analysis. All the selected features representing a lathe spindle failure run constitute a machine health degradation dataset. Such 14 datasets are weeded out from the entire extracted feature datasets. This feature processing methodology is expected to nullify the concerns due to the smaller data size. In preparing data for predictive analytics, the available dataset is first divided into the training set, validation set, and testing set. The time-series dataset is then normalized to have zero mean and unit variance. It is also assumed that the lathe spindle does not start to degrade at the beginning healthy state of each failure run. Hence, a response clip is applied until where there is no health degradation or drop in RUL.

Chapter 5

Data-Driven Prognostic Analysis for Remaining Useful Life Estimation

5.1 Introduction

This chapter presents a detailed discussion on the implementation of the data-driven intelligent computational algorithms for the prognostic analysis of lathe spindle unit RUL estimations. The computational algorithms for predictive modeling are the most crucial and challenging step in machine failure prognostics. These algorithms can be physics-based, model-based, data-driven, or a hybrid combination of any of these. The physics-based approach demands a thorough knowledge of the physics of the system and the model-based approach requires mathematical descriptions for the health status degradation of the mechanical system [19, 20]. Whereas, the data-driven approach uses minimal technical aspects of the system. In a data-driven approach, the historic machine monitoring data representing the health status degradation are used for training computational algorithms to evolve intelligent predictive models [18-20, 128]. Data-driven approaches include statistical approaches and AI approaches. In this work, three data-driven prognostic algorithms based on the deep learning model, machine learning model, and statistical estimator model are developed for the RUL estimation of the lathe spindle unit.

5.2 RUL Estimation Employing Deep Learning Model

The data-driven methods are learning-based approaches that discover viable features and prognostics models from the acquired data. These techniques include statistical models and AI models that infer health status information directly from the monitored data. AI models learn machinery health degradation patterns from the available machine health degradation data. Machine learning and deep learning AI models are very popular in machinery health diagnostic and prognostic analysis. The most recognized deep learning architectures for failure prediction and RUL estimation include RNN, LSTM, DBN, CNN, etc. [21, 22].

Over the past decade, deep learning approaches have emerged as a promising computational tool for predictive analytics in engineering and industrial applications. The LSTM-RNN deep learning architectures are also gaining wide reorganization in analyzing the time series machinery health degradation analysis for failure prediction and RUL estimations [163-178]. LSTM network is an advanced variant of RNN which has been recognized as a powerful computational tool for mining critical information from raw sequential time series data. Song et al. [179], Zhang et al. [180], Elsheikh et al. [160], Wang et al. [181] have used bidirectional LSTM (bi-LSTM) and Essien et al. [182] used convolutional LSTM for RUL estimation. Zhang et al. [183] and Xia et al. [162] have used a convolutional LSTM and convolutional bi-LSTM ensemble framework respectively for the RUL estimation. Generally, the deep learning network has a natural structure to learn machinery degradation patterns directly from raw machinery monitoring data. However, it demands a large size machine degradation data with multiple sensor observations to avoid the problem of underfitting during RUL estimations [210]. Wang et al. [170], Zhang et al. [171], He et al. [172] have processed the machine degradation data to extract signature features representing the machine degradation trend, which is then utilized for training the LSTM deep network for accurate RUL estimation. The potential of the LSTM network to learn from raw sensor data minimizes the computational complexity, at the same time the mandate to have large size machine degradation data is regarded as the major limitation on the implementation of LSTM deep neural networks in industries.

The hyperparameter optimization and network architecture selection is another major challenge before the successful implementation of deep learning algorithms for prognostics analysis of industrial machinery. The selection of hyperparameters including both training and structural parameters has a direct influence on the performance of the predictive models. Generally, this optimization is performed by trial-and-error methods, which is arduous and time-consuming [162, 168]. Recently, a few literature have discussed the LSTM hyperparameter optimization algorithms employing ant colony optimization [166], particle swarm optimization [171], comparative analysis [179], grid search [165, 170, 182], Bayesian search optimization [161, 173], etc. to reduce the computational complexity and improve prediction accuracy. Bayesian search optimization provides a refined approach and has been shown to outperform other algorithms [211]. Bayesian optimization techniques are popularly employed to optimize non-differentiable, discontinuous, and expensive functions. The algorithm uses an acquisition function that estimates the next point to evaluate, which makes the algorithm optimize the hyperparameters in a minimum number of iterations [173, 211]. Its capability to converge at optimized values in the minimum number of iterations makes it appropriate for expensive and computationally complicated algorithms like deep learning algorithms. The mandate to have large-sized data for prognostic analysis and the black-box nature of learning algorithms remains a major challenge before the practical implementation of deep learning architectures for predictive maintenance. There exists an insistent need for further research works owing to the vast applicability of the LSTM deep learning network in prognostic predictive maintenance of industrial machinery.

The thesis proposes a novel Bayesian optimization LSTM/bi-LSTM deep learning approach with an automated hyperparameter optimization paradigm for the RUL estimation of the lathe spindle unit. The prepared vibration signature features for prognostic regression analysis using time, frequency, and time-frequency domain analysis and NCA-based feature selection criteria (refer to section 4.3) are employed for training the deep learning models. This feature processing methodology is expected to nullify the concerns due to the smaller data size. The prepared lathe spindle health degradation dataset comprising the training, testing, and validation datasets are employed for training the Bayesian optimized LSTM/bi-LSTM and their combination deep network architectures for the prognostic analysis. The Bayesian optimization of LSTM/bi-LSTM deep network hyperparameters is expected to knock down the black-box nature of these learning algorithms.

The training and validation datasets are fed into the Bayesian optimization LSTM/bi-LSTM network algorithm to evolve accurate predictive models for RUL estimation. In the proposed learning architecture, the LSTM/bi-LSTM network algorithm is executed within the

Bayesian optimization algorithm. The flow diagram for the Bayesian optimization LSTM/bi-LSTM network algorithm is shown in Figure 5.1. The LSTM/bi-LSTM deep learning architectures with predefined fixed ranges of hyperparameters are assigned as the objective function to a Bayesian optimization algorithm. The objective function trains the LSTM/bi-LSTM network and returns the validation error to the Bayesian optimization algorithm. After each iteration, a new set of values are assigned to the hyperparameters and the process continues until a termination criterion is reached [173, 211]. The termination criterion for the algorithm is set to a predefined condition of completing 30 iterations. The condition is arbitrarily chosen based on random trials considering a parity between the computational expense and prediction accuracy. Finally, the chosen predictive models are tested on the independent test sets. (See Appendix III for MATLAB code for Bayesian optimization deep learning model based prognostic algorithm)

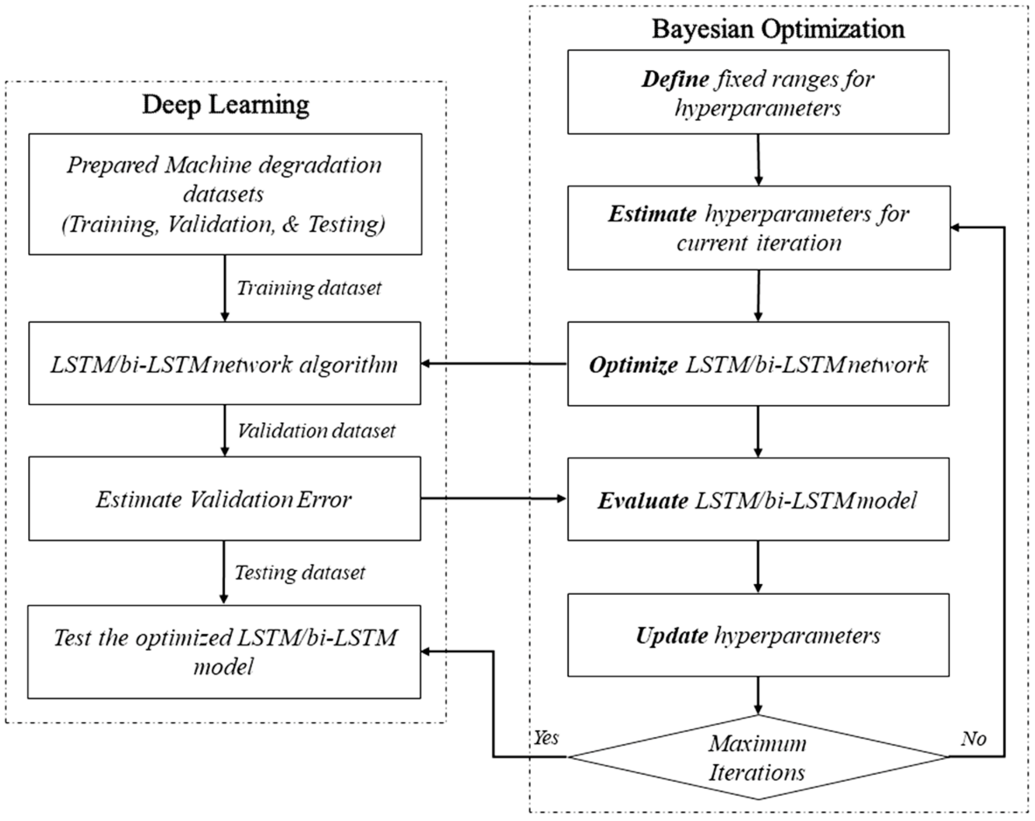

Bayesian Optimization
Deep Learning
Define fixed ranges for hyperparameters
Prepared Machine degradation datasets (Training, Validation, & Testing)
Estimate hyperparameters for current iteration
Training dataset
LSTM/bi-LSTM network algorithm
Optimize LSTM/bi-LSTM network
Validation dataset
Estimate Validation Error
Evaluate LSTM/bi-LSTM model
Testing dataset
Test the optimized LSTM/bi-LSTM model
Update hyperparameters
Yes
Maximum Iterations
No

Figure 5.1 Flow diagram for Bayesian optimization LSTM/bi-LSTM network algorithm

The algorithm is proposed to optimize the deep learning hyperparameters towards the achievement of a minimum validation error. The validation error is defined as the mean absolute error (MAE) between the predicted machine health degradation trend for the validation dataset and the actual machine health degradation trend as shown in Eq. 5.1

$$Validation\ Error = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{5.1}$$

where y_i is the predicted machine health degradation trend, x_i is the actual machine degradation trend, and n is the total time-steps of the validation dataset. The degradation trend is the RUL of machine components at different time-steps starting from healthy to a failure state.

In order establish a better understanding of the LSTM/bi-LSTM network algorithms, the Bayesian optimization is separately performed for the network structure and hypermeters optimization of LSTM, bi-LSTM, LSTM + bi-LSTM, bi-LSTM + LSTM, LSTM + LSTM, and bi-LSTM + bi-LSTM network architectures. The deep network architectures with single LSTM/bi-LSTM network models include only six layers namely the sequence input layer, the LSTM layer, the fully connected layer, the dropout layer, the fully connected output layer, and the regression layer. On the other hand, those deep network architectures with LSTM/bi-LSTM combination network models have nine layers including the second LSTM/bi-LSTM network model and an additional set of the fully connected layer and the dropout layer for the second LSTM/bi-LSTM network models. Table 5.1 and Table 5.2 presents the layer framework of these deep learning architectures. A sequence input layer inputs sequence data to a network. The fully connected layer establishes the connection between every neuron in one layer of the LSTM/bi-LSTM network model to every neuron on preceding layers. The dropout layers regulate the learning process to avoid overfitting of LSTM/bi-LSTM network models. A regression layer computes the error loss for regression tasks. In the combination network models, the second LSTM/bi-LSTM network layer requires an additional set of the fully connected layer and the dropout layer. The fully connected layer, the dropout layer, and the second LSTM/bi-LSTM layer make the addition of extra three layers.

Table 5.1 Layer framework of LSTM and bi-LSTM network architectures

Layer No.	LSTM	bi-LSTM
1	*Sequence input layer*	*Sequence input layer*
2	*LSTM layer*	*bi-LSTM layer*
3	*Fully connected layer*	*Fully connected layer*
4	*Dropout layer*	*Dropout layer*
5	*Fully connected output layer*	*Fully connected output layer*
6	*Regression layer*	*Regression layer*

Table 5.2 Layer framework of LSTM + bi-LSTM, bi-LSTM + LSTM, LSTM + LSTM, and bi-LSTM + bi-LSTM network architectures

Layer No.	LSTM + bi-LSTM	bi-LSTM + LSTM	LSTM + LSTM	bi-LSTM + bi-LSTM
1	*Sequence input layer*	*Sequence input layer*	*Sequence input layer*	*Sequence input layer*
2	*LSTM layer*	*bi-LSTM layer*	*LSTM layer*	*bi-LSTM layer*
3	*Fully connected layer*	*Fully connected layer*	*Fully connected layer*	*Fully connected layer*
4	*Dropout layer*	*Dropout layer*	*Dropout layer*	*Dropout layer*
5	*bi-LSTM layer*	*LSTM layer*	*LSTM layer*	*bi-LSTM layer*
6	*Fully connected layer*	*Fully connected layer*	*Fully connected layer*	*Fully connected layer*
7	*Dropout layer*	*Dropout layer*	*Dropout layer*	*Dropout layer*
8	*Fully connected output layer*	*Fully connected output layer*	*Fully connected output layer*	*Fully connected output layer*
9	*Regression layer*	*Regression layer*	*Regression layer*	*Regression layer*

The hyperparameters control the behavior of deep learning architecture. The most relevant hyperparameters involved in the training of LSTM/bi-LSTM deep learning architecture are as follows: [192, 212]

(a) Number of hidden units (H_n): It corresponds to the amount of information remembered between time-steps. Too large H_n value might cause overfitting of training data. It can vary from a few dozen to a few thousand.

(b) Number of fully connected layer (FC_n): All neurons in the fully connected layer are connected to all neurons in the previous layer. This combines all the features learned across the layers to identify hidden patterns.

(c) Dropout rate (D_r): It randomly sets input elements to zero thus changing the underlying network structure between iterations. Higher D_r causes more elements being dropped during training. It influences the performance of the evolved model by enhancing model generalization.

(d) Maximum epochs (Epoch): A epoch corresponds to the full pass of the training algorithm over the entire dataset.

(e) Initial learning rate (L_r): It is used for training the algorithm. Too low L_r takes a long training time and too high L_r might conclude at a suboptimal result or diverge.

(f) L2-regularization factor ($L2_{rf}$): It corresponds to the weight decay factor that is responsible to reduce training data overfitting.

In the Bayesian optimization algorithm, the hyperparameters are initiated with predefined fixed ranges based on random trials. The algorithm is executed with random hyperparameters combinations varying one at a time. Only those ranges of hyperparameter showing a significant variation in the final result are chosen for the study. Table 5.3 shows the hyperparameters with their respective predefined fixed ranges. The performance of the LSTM/bi-LSTM network architectures for RUL estimation of lathe spindle is evaluated using the root mean square error (RMSE) between the actual and predicted RUL. RMSE is computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}} \quad (5.2)$$

where n is total time-steps, y_i is predicted values, and x_i is actual values.

Table 5.3 Hyperparameters of deep learning architectures with respective fixed ranges

Hyperparameters					
Network Structure			***Training Parameters***		
H_n	FC_n	D_r	***Epoch***	L_r	$L2_{rf}$
[100 300]	[25 250]	[0.2 0.8]	[100 300]	[1e-3 0.1]	[1e-5 1e-3]

The algorithms employed in this work are developed and executed on a computer having an Intel Core-i7 (3.60 GHz) processor with 8 GB RAM using 64-bit MATLAB software. The prepared lathe spindle health degradation dataset comprising 10 training datasets, 2 validation datasets, and 2 testing datasets are employed for prognostic analysis using the Bayesian optimization LSTM/bi-LSTM learning algorithm. Bayesian optimization of LSTM, bi-LSTM, LSTM + bi-LSTM, bi-LSTM + LSTM, LSTM + LSTM, and bi-LSTM + bi-LSTM network architectures are performed separately using the training and validation datasets to identify their best hyperparameters. The Bayesian optimization algorithm tune the hyperparameters towards the attainment of a minimum validation error. The algorithm terminates after completing 30 iterations to produce the optimized hyperparameters for the minimum validation error. Table 5.4 presents the Bayesian optimized hyperparameter sets for all the considered deep network architectures. The LSTM/bi-LSTM deep network architectures trained using these optimized hyperparameters are compared to identify the most accurate predictive model.

Table 5.4 Bayesian optimized hyperparameters of LSTM/bi-LSTM network architectures

Network Architecture	H_n-1	FC_n-1	D_r-1	H_n-2	FC_n-2	D_r-2	***Epoch***	L_r	$L2_{rf}$
LSTM	242	116	0.63	---	---	---	296	0.009	4.0e-4
bi-LSTM	104	199	0.36	---	---	---	239	0.011	2.5e-4
LSTM + bi-LSTM	114	129	0.38	270	84	0.21	207	0.0034	5.1e-5
bi-LSTM + LSTM	125	127	0.63	107	152	0.30	216	0.0012	1.9e-4
LSTM + LSTM	221	180	0.21	179	39	0.22	247	0.0098	1.8e-5
bi-LSTM + bi-LSTM	110	184	0.55	119	75	0.20	223	0.0019	1.8e-4

The evolved predictive models are examined to determine the prediction accuracy on an independent testing dataset. The RMSE between the actual and predicted RUL is estimated for all predictive models. Test observation plots between actual RUL and predicted RUL for LSTM/bi-LSTM network architectures with their respective prediction accuracy (RMSE) are illustrated in Figure 5.2. The predicted RUL well compliments the actual lathe spindle life degradation pattern including the response clip pinned on the actual life pattern. It is observed that not all the deeper LSTM and bi-LSTM network combination architectures give better accuracy than the single LSTM or bi-LSTM network architectures for spindle lathe RUL estimation. The LSTM + bi-LSTM network architecture is identified to have the best prediction accuracy on lathe spindle RUL estimation with RMSE equals 31.65 followed by the single LSTM architecture with RMSE equals 40.01. The LSTM architecture is well efficient in digging up hidden patterns from time-series data. Further employing a bi-LSTM network can refine the learned degradation patterns yielding accurate estimations. It is also observed that the order of placing the LSTM/bi-LSTM layer in the layer framework has a very high impact on the training process and thus the prediction accuracy. The deep learning architectures with a bi-LSTM network at the beginning are observed to produce comparatively lesser accurate predictions. This might be because the bi-LSTM network learns the time series data from both directions leads to overfitting of the evolved predictive model. The Bayesian optimization of LSTM/bi-LSTM and their combination architectures identifies the best individual hyperparameter sets for maximum prediction accuracy on lathe spindle data. The proposed Bayesian optimization deep learning algorithm provides an integrated self-optimization of hyperparameters and an RUL estimation platform for the predictive analytics of the lathe spindle unit.

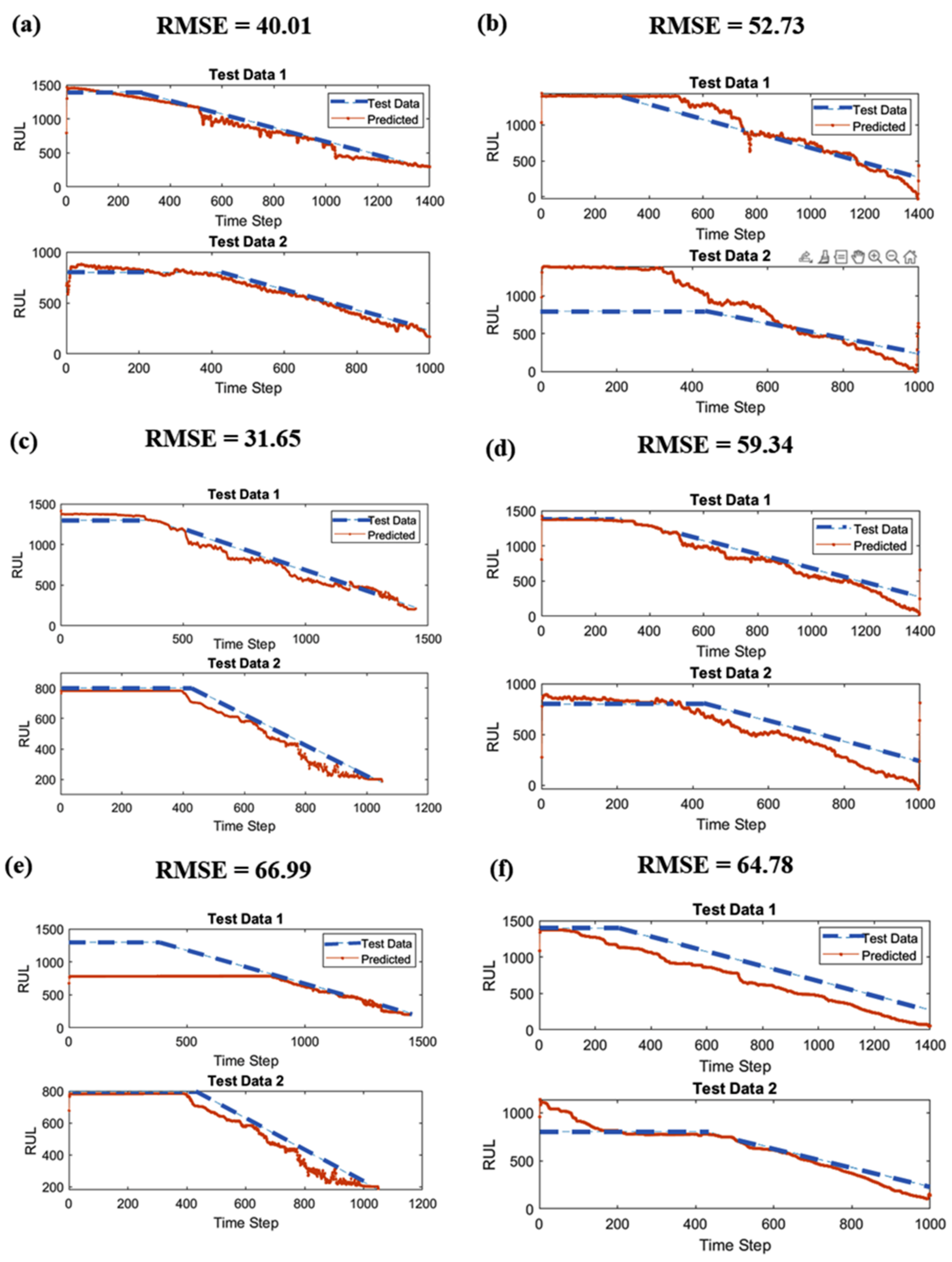
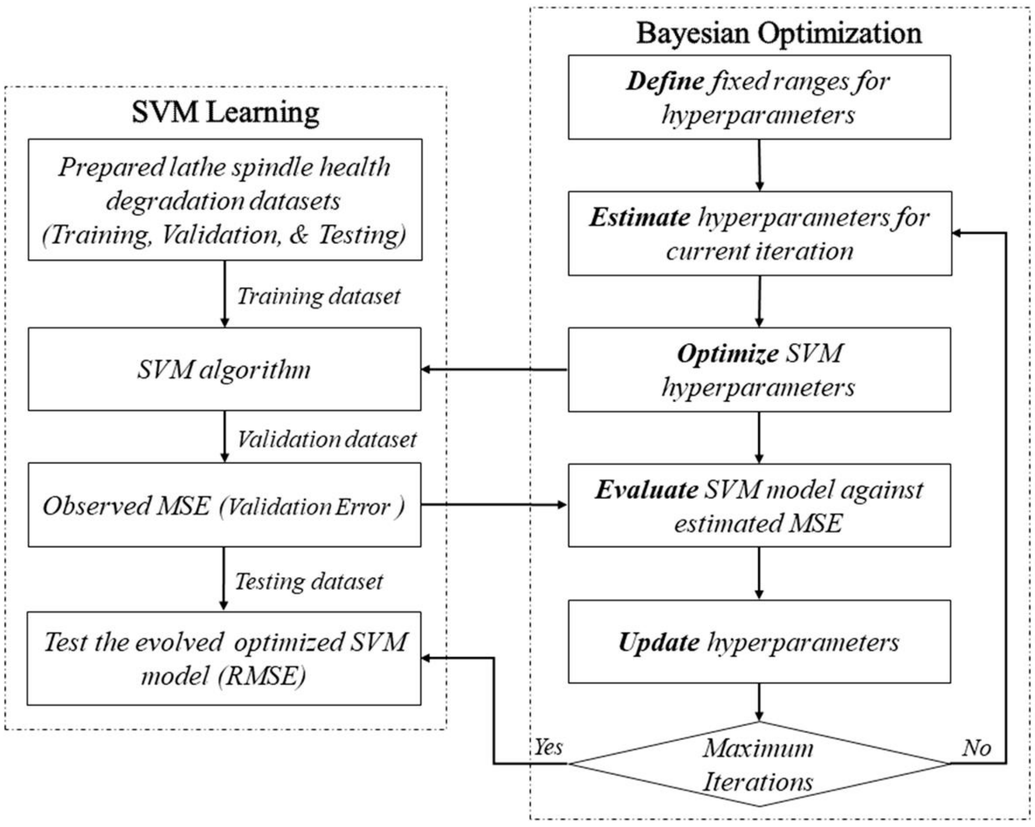


Figure 5.2 Actual RUL vs predicted RUL for (a) LSTM, (b) bi-LSTM, (c) LSTM + bi-LSTM, (d) bi-LSTM + LSTM (e) LSTM + LSTM, (f) bi-LSTM + bi-LSTM models

5.3 RUL Estimation Employing Machine Learning Model

Machine learning algorithms are widely employed for the failure classification and prediction of mechanical systems [22]. SVM is the most popularly employed learning approach for machine failure prediction and RUL estimation [127, 136]. Yan et al. [138] employed an SVM classifier to assess the degradation stage of bearing, which is further utilized to exploit the optimal RUL prediction. Chen et al. [213] proposed a framework for RUL estimation of an aircraft engine using the lifecycle data and performance deteriorated parameter data based on the theory of similarity index and SVM. Louen et al. [214] proposed a new health feature creation approach using a binary SVM classifier, which is also used to obtain fault detection for the RUL estimation. Benkedjouh et al. [145] employed the isometric feature mapping reduction technique for nonlinear feature reduction and SVR for the RUL estimation of mechanical bearings. As in the case of deep learning algorithms, the options for the internal parameters/hyperparameters of SVM can strongly influence the prediction accuracy. Manually assigning these hyperparameters might not serve the purpose as it is considered transcendent [23, 215, 216]. Similar to the Bayesian optimization of LSTM/bi-LSTM deep network hyperparameter optimization algorithm, the SVM machine learning algorithm is executed within the Bayesian optimization algorithm against the minimization of prediction error to obtain the best hyperparameter set for a given training data.

The SVM machine learning architectures with predefined fixed ranges of hyperparameters are assigned as the objective function to a Bayesian optimization algorithm. The objective function trains the SVM model and returns the validation error to the Bayesian optimization algorithm. After each iteration, a new set of values are assigned to the hyperparameters and the process continues until a termination criterion is reached [23]. The termination criterion for the algorithm is set to a predefined condition of completing 30 iterations as similar to LSTM/bi-LSTM deep learning architecture optimization. Finally, the evolved SVM regression model for the best-optimized hyperparameter set is tested towards the independent testing dataset to obtain the RUL and prediction error. The flow diagram of the Bayesian optimization SVM learning algorithm is shown in Figure 5.3. (See Appendix IV for MATLAB code for Bayesian optimization machine learning model based prognostic algorithm)



SVM Learning
Prepared lathe spindle health degradation datasets (Training, Validation, & Testing)
Training dataset
SVM algorithm
Validation dataset
Observed MSE (Validation Error)
Testing dataset
Test the evolved optimized SVM model (RMSE)
Bayesian Optimization
Define fixed ranges for hyperparameters
Estimate hyperparameters for current iteration
Optimize SVM hyperparameters
Evaluate SVM model against estimated MSE
Update hyperparameters
Maximum Iterations
Yes
No

Figure 5.3 Bayesian Optimization SVM algorithm flow diagram

The overall algorithm is aiming to minimize the Bayesian optimization validation error, which is computed using the MSE as presented in Eq. 5.1 [217]. Bayesian optimization algorithm estimates the MSE between the predicted and actual RUL for every iteration for the assigned set of hyperparameters. The hyperparameter set for every iteration is expected to minimize the prediction error to obtain an estimated MSE. This estimated MSE is compared against the actual observed MSE to determine the hyperparameter set for the next iteration. In each iteration, the SVM model fitness is evaluated using RMSE between the actual and the predicted RUL as presented in Eq. 5.2 [218].

The SVM algorithm is repeatedly executed within the Bayesian optimization algorithm to converge at a set of hyperparameters that causes minimum prediction error. The major hyperparameters considered for optimization includes

(a) *Box constraint:* It constrains the α coefficients that the α coefficient values cannot exceed the box constrain value. The Positive values log-scaled in the range [0.001,1000] are assigned for box constrain

(b) *Kernel function:* Kernel functions are mathematical functions used to map the training data to the required form. A kernel function among Linear, Gaussian, Quadratic, or Cubic, which best represents the training data is selected.

(c) *Kernel scale:* The SVM algorithm divides the elements of the predictor vector by kernel scale. Positive values log-scaled in the range [0.001,1000] are assigned for kernel function.

(d) *Epsilon:* Epsilon is a margin of tolerance in the support vector selection error. It is half the width of epsilon-insensitive band specified as positive values log-scaled in the range [0.001,100] * *iqr(y) /* 1.349, where *iqr(y)* is the interquartile range of response variable y [219, 220].

Table 5.5 shows the hyperparameters of SVM machine learning architecture with their respective predefined fixed ranges. The algorithm iterates repeatedly for a different set of SVM hyperparameters within these fixed ranges.

Table 5.5 Hyperparameters of SVM machine learning architecture with respective fixed ranges

Hyperparameters Fixed Ranges			
Box Constraint	***Kernel Function***	***Kernel Scale***	***Epsilon***
0.001-1000	Linear, Gaussian, Quadratic, or Cubic	0.001-1000	0.47739 - 47739.066

The algorithms employed in this work are developed and executed on a computer having an Intel Core-i7 (3.60 GHz) processor with 8 GB RAM using 64-bit MATLAB software. The prepared lathe spindle health degradation dataset comprising 10 training datasets, 2 validation datasets, and 2 testing datasets are employed for prognostic analysis using the Bayesian optimization SVM learning algorithm. First, the Bayesian optimization SVM learning algorithm is executed using the training and validation dataset. The SVM model is trained within the Bayesian optimization algorithm, which automatically optimizes the hyperparameters to evolve the best accurate RUL estimation model. The algorithm iterates repeatedly to tune the SVM hyperparameters for a minimum estimated MSE between the predicted and actual RUL on the validation dataset. The Bayesian optimization algorithm terminates after 30 iterations. The progress of Bayesian optimization to identify a minimum observed MSE against an estimated minimum MSE for each iteration is represented in an MSE plot as shown in Figure 5.4.

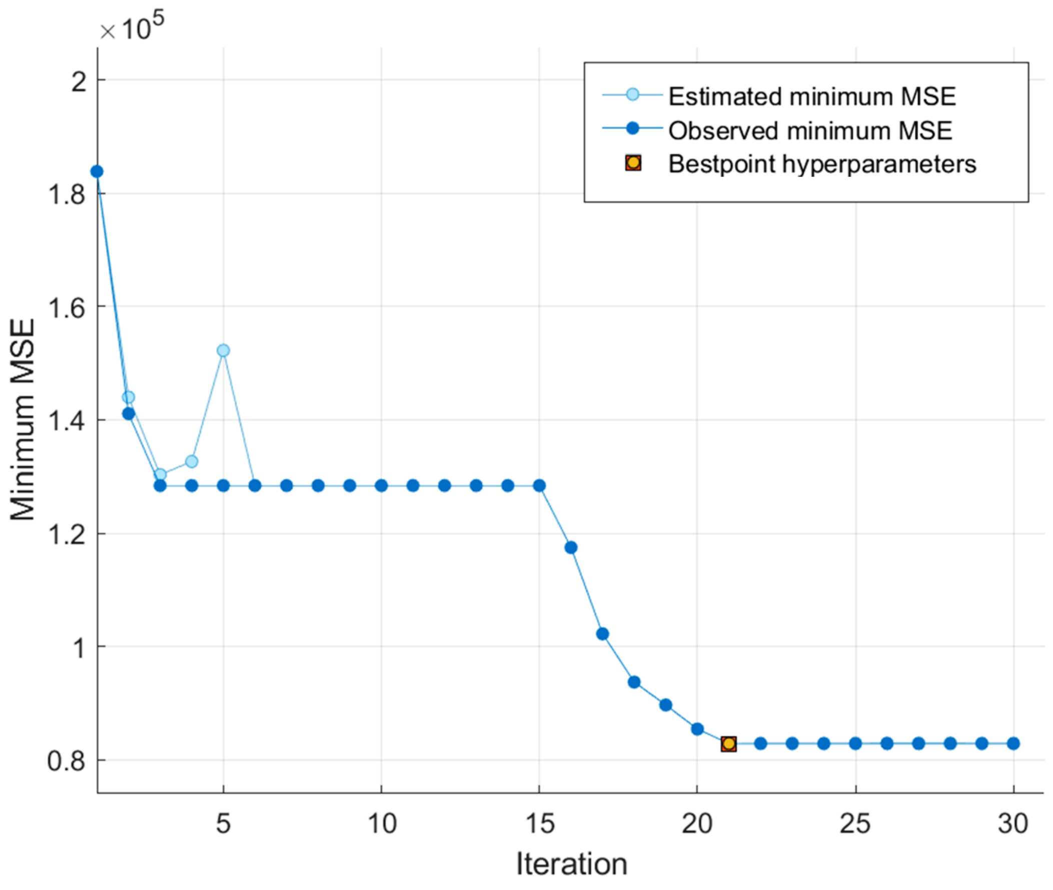


Figure 5.4 MSE plot for Bayesian optimization

The estimated minimum MSE at each iteration is an estimate of the MSE for all the hyperparameters sets considered till each respective iteration. The hyperparameter set assigned to each iteration is expected to achieve this minimum estimated MSE. The observed minimum MSE is the minimum of actual MSE obtained up to the current iteration. In Figure 5.4, it can be observed that the best hyperparameter set is obtained at iteration 21, at which the hyperparameter values correspond to the most accurate SVM model for the given datasets. The optimized SVM hyperparameters are shown in Table 5.6.

Table 5.6 Bayesian optimized SVM hyperparameters

Hyperparameters	**Optimized Result**
Box Constrain	536.89
Kernal Function	Gaussian
Kernal Scale	0.058
Epsilon	76.52

The SVM model with the Bayesian optimized hyperparameters set is tested using the two independent test datasets. The variation of actual RUL vs predicted RUL for the two test datasets is shown in Figure 5.5. The actual RUL values of test datasets are response-clipped to an RUL value equal to 800. It can be observed that the predicted RUL attempts to trace the actual RUL pattern from the initial to the final time-steps. The pattern tracing is observed to improve towards the final time-steps for both test data 1 and test data 2. The overall predicted RUL is having an acceptable agreement with the actual RUL. An RMSE equal to 206.23 is obtained as a quantitative measure of prediction accuracy for the Bayesian optimized SVM model for the given dataset.

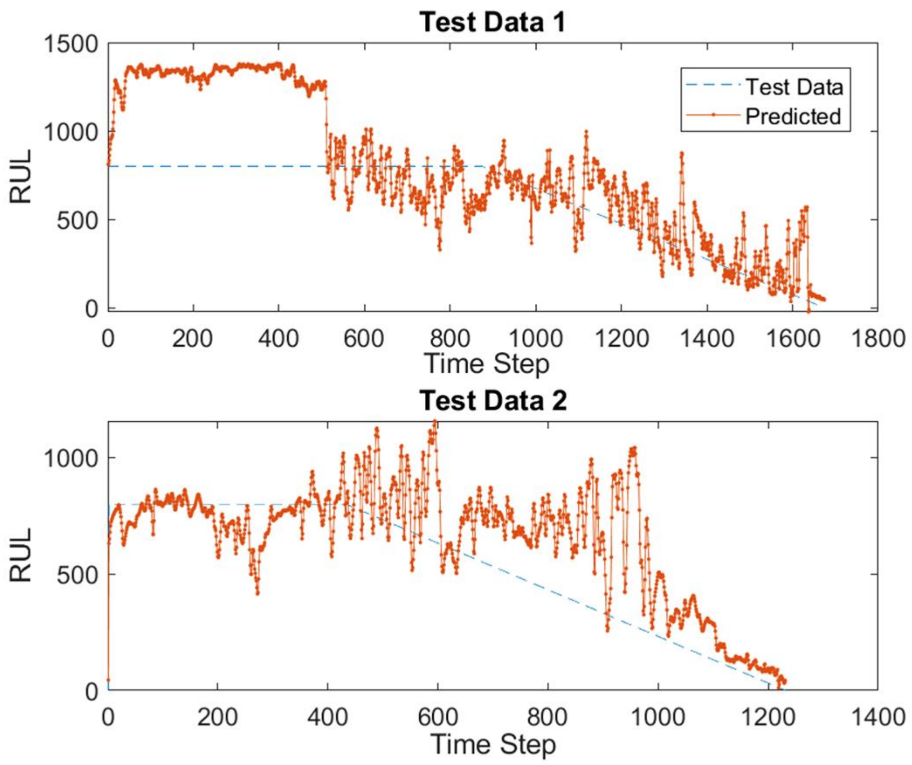


Figure 5.5 SVM model -the variation of actual RUL vs predicted RUL

The prediction results obtained fairly follows the actual experimental test observations. SVM prognostic regression analysis is an effective tool for RUL estimation from time-series machine degradation data. As well, the Bayesian optimization approach for the best hyperparameters setting is proved effective for machine learning algorithms.

5.4 RUL Estimation Employing Statistical Estimator Model

The statistical data-driven regression approaches are reported as an easy-to-implement approach for RUL estimation using time-series sequence data [17]. It is effective in addressing the uncertainty of the degradation phenomenon and its influence on RUL estimations. Auto-Regressive (AR) models, Random coefficient models, Wiener process models, Gamma process models, Inverse Gaussian process models, Markov models, Proportional hazards models, and their different variants are the most popularly used statistical data-driven techniques for machine prognostics analysis (6, 221).

Recently, the statistical data-driven approaches have made a progression from the basic state estimator models to more specific RUL estimation models like the similarity-based models, exponential degradation models, and survival-based models. The similarity-based RUL prediction approach is widely used for industrial data analytics (222, 223). Liu et al. [224] developed a distance similarity and spatial direction similarity-based health index for RUL prediction. Wen et al. [133] have presented an exponential degradation model and Mahalanobis distance approach for the RUL estimation of ball screw systems. Zhang et al. [134] proposed an exponential degradation model and particle filter-based RUL estimation approach Lithium-ion battery. Tseng et al. [130] proposed an exponential-depression degradation model for the optimization of accelerated degradation test allocation problems. Li et al. [129] proposed an improved exponential degradation model for the RUL estimation of bearings. Gebraeel et al. [135], first introduced the exponential model for RUL prediction. It is a model-based analytical method that can incorporate both expert knowledge and information from measured data [124, 129]. The exponential models are highly suitable for representing the degradation patterns of a mechanical component, where an exponential-like degradation process can be observed [225, 226]. It is also useful when the component experiences cumulative degradation where the cause of degradation from multiple sources are acting together [124, 129, 135, 226]. However, the exponential models are not explored in depth for predictive analytics.

In this section, a prognostic approach combining the Principal Component Analysis (PCA) and exponential degradation model is proposed for the RUL estimation of the lathe spindle unit. The proposed algorithm put forward a less complicated and economic computational platform by integrating the abilities of PCA to reduce dimensionality while preserving variance of the prepared vibration monitoring data and the exponential degradation model to represent the complex degradation patterns of mechanical components. The methodology includes lathe spindle HI construction and RUL estimation. The PCA is employed for the HI construction and the exponential degradation model is employed for RUL estimation.

5.4.1 Feature Dimensionality Reduction and Degradation HI Construction

Dimensionality reduction is the process of converting a set of data having large dimensions to data with smaller dimensions, ensuring that there is no loss of useful information during the conversion process. This technique can also fuse the data generating a single data feature that conveys similar information concisely. One of the most widely used dimensionality reduction-based fusion approaches, namely the PCA - Principal Component Analysis, also known as the Karhunen–Loeve transformation is utilized in this work [227, 228].

PCA is the most commonly used statistical tool for HI construction by dimension reduction and feature fusion. The method generates a new set of variables, called principal components, which are linear combinations of all selected features [227, 229, 230]. The HI construction by PCA is described in the following steps.

i. *Standardization:* - This step aims to standardize the range of vibration signal features to ensure an equal contribution of all features. Once standardized, all features are transformed to a common scale.

$$Standardized\ value = \frac{value - mean}{standard\ deviation} \tag{5.3}$$

ii. *Covariance Matrix Computation:* -This step determines how the features are varying from the mean with respect to each other, which is to find any correlation between them. The positive sign of covariance indicates a correlation and the negative sign indicates an inverse correlation.

iii. *Compute Eigenvectors and Eigenvalues of Covariance Matrix:* -These are used to estimate the principal components. The computed principal components are uncorrelated and store maximum information about the original features. The first principal component has the maximum possible information, the next maximum information in the second, and so on.

iv. *Feature Vector:* -In this step, the lesser significant (low eigenvalues) components are discarded and the remaining ones are selected to form a feature vector. This makes dimensionality reduction. Finally, cast the data along the principal component axes. This is done by multiplying the transpose of standardized original data set by the transpose of the feature vector.

The prognostic analysis employing the exponential degradation model utilizes a single lathe spindle health degradation dataset. The dataset with a maximum life of 1072 time-steps is chosen for validation of the proposed algorithm. The features selected after NCA based feature selection criterion (refer to Chapter 4, section 4.3.2) are analyzed with PCA to construct the HI. The constructed HI is a linear combination of these selected features. Figure 5.6 shows the analysis of the data space of the first two principal components.

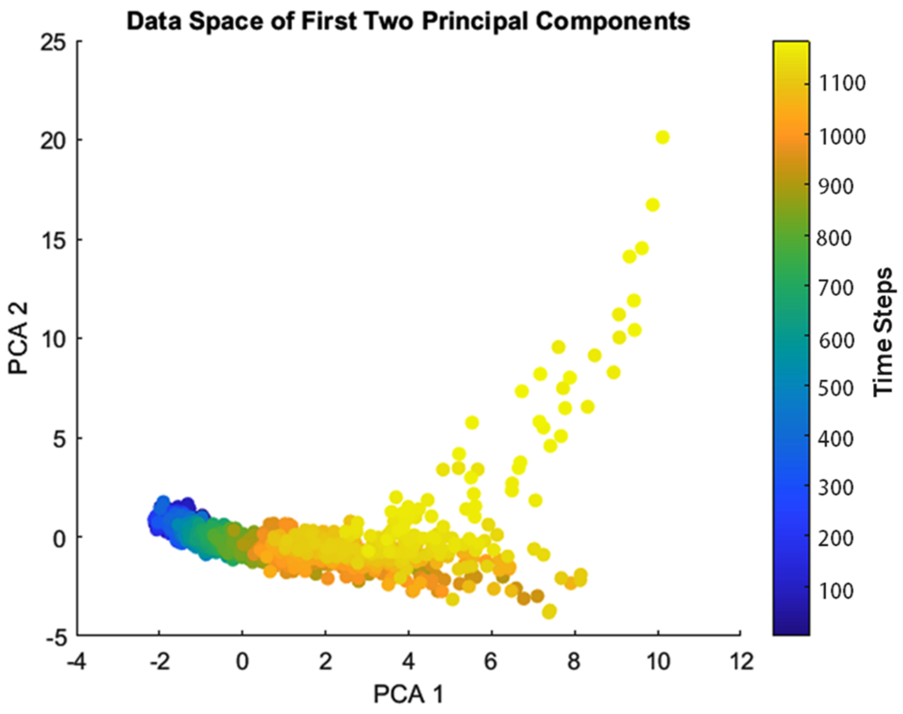


Figure 5.6 Data space for the first two principal components

The first principal component increases as the bearing approach to failure and therefore, it is recognized as the best to represent bearing degradation [231]. The first principal component is further smoothened using a moving mean filter to produce a reliable degradation trend of the bearing from normal to the faulty stage. Figure 5.7 shows the smoothened HI for bearing degradation. The HI shows a steady trend at the beginning of component life, which indicates the normal healthy condition of the component. This stage is followed by a gradually increasing trend of HI, which marks the sprouting of faults in the mechanical unit. A sudden increase in trend indicates the component approaching failure and finally the failure. This smoothened HI is used to train the exponential degradation model algorithm for RUL estimation. (See Appendix V for MATLAB code for exponential degradation model-based prognostic algorithm)

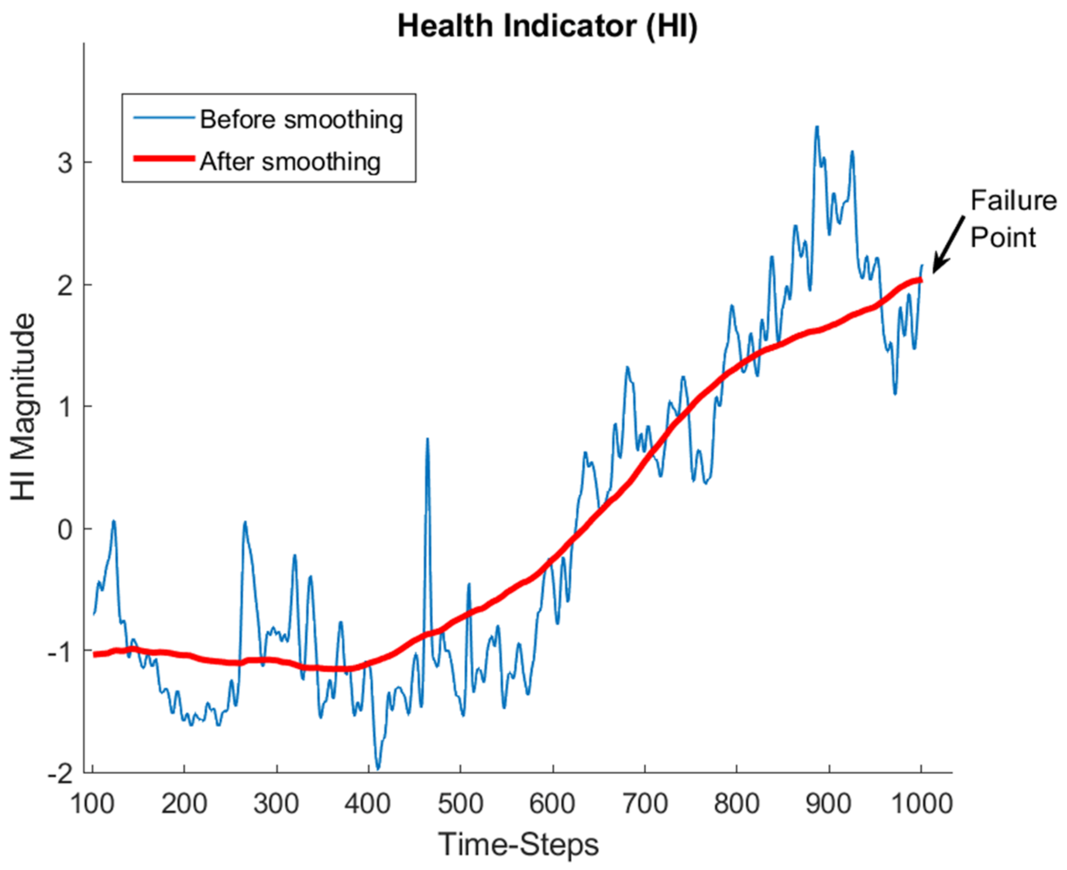


Figure 5.7 Constructed Health Indicator (HI)

5.4.2 Lathe Spindle RUL Estimation Using Exponential Degradation Model

The exponential degradation model is utilized for estimating the RUL of the lathe spindle unit. This computational methodology is mostly employed when the component experiences a cumulative degradation, which is the common degradation phenomenon of any mechanical system. The basic concept of the approach is a simple curve fitting and extrapolation process. The exponential degradation model fits into the constructed HI. This degradation model is extrapolated to find a future time step where the degradation model crosses a predefined threshold value. The difference between this future time step and the present time step gives the required RUL. This failure threshold is usually defined based on previous failure history or chosen as a safe value before the faulty zone on HI. Assuming that no historical data is available for the dataset, the last value of the HI is chosen as the failure threshold.

The constructed HI is provided as the input for the RUL computation algorithm and the HI value corresponding to the time step-1072 (the last HI value) is assigned as the threshold for failure detection. To validate the prediction algorithm, only a part of the HI from the beginning to time step-900 is provided as the input to train the exponential degradation model. The evolved exponential degradation model truly fits into the actual degradation trend represented by the HI. Figure 5.8 shows the exponential degradation model fitted into the constructed HI for the RUL estimation with a 95% confidence interval. The algorithm estimates the prediction model to cross the failure threshold at time step-958, which computes the RUL as 58-time steps. The error of predicted lifetime from the actual experimental value is 114-time steps.

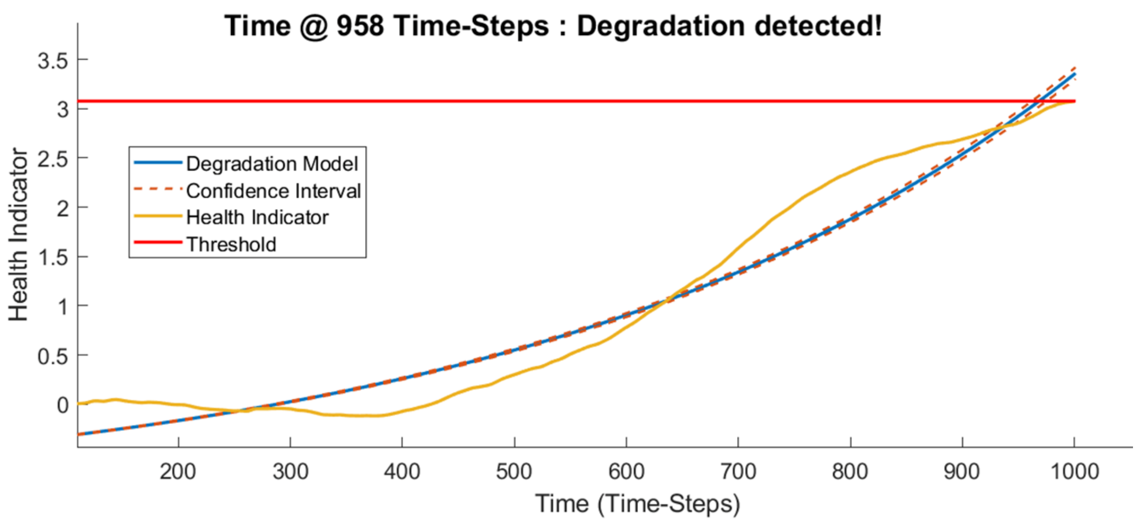


Figure 5.8 Exponential degradation model for RUL Prediction

5.5 Comparison of Predictive Accuracy of Considered Data-Driven Prognostic Models

For further evaluation of the proposed predictive analytics approaches, a comparison of the prominent data-driven approaches the deep learning model, machine learning model, and statistical estimator model is included in the context of discussions. To make an easy intelligible comparison of the prediction accuracy of considered data-driven prognostic models the mean absolute percentage error (MAPE) between the predicted and actual RUL are estimated [232].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{(x_i - y_i)}{x_i}\right| \tag{5.4}$$

where x_i is actual RUL, y_i is predicted RUL, and n is sample length. A comparison is made for the RUL prediction error determined as RMSE and MAPE for the considered data-driven prognostics approaches. A comparison has also been made on the computational complexity of three data-driven approaches in terms of the time for computation.

A comparison metrics of the considered data-driven prognostics approach LSTM/bi-LSTM deep learning models, exponential degradation estimator model, and SVM machine learning model are presented in Table 5.7. The comparison metrics include the RUL prediction errors (determined as RMSE and MAPE) and computational complexity (determined as computational time). It is observed that the deep learning models outperform the machine learning and statistical estimator model in the RUL estimation of the lathe spindle unit.

Table 5.7 Comparison of prediction accuracy and computational complexity of data-driven models

Data-Driven Approach	**RMSE**	**MAPE (%)**	**Computational Time (Hours)**
LSTM	40.01	6.09	4.49
bi-LSTM	52.73	7.01	13.01
LSTM + bi-LSTM	31.65	4.45	18.60
bi-LSTM + LSTM	59.34	7.14	20.34
LSTM + LSTM	66.99	10.40	9.51
bi-LSTM + bi-LSTM	64.18	9.52	23.82
Exponential Degradation model	114.00	10.63	0.28
SVM model	206.23	23.18	1.83

The deeper LSTM + bi-LSTM leaning architecture is trained best on the lathe spindle health degradation data with an RMSE of 31.65 and MAPE of 4.45%. The RMSE and MAPE of the exponential degradation model are 114 and 10.63% respectively, and those for the SVM model are 206.23 and 23.18% respectively. The prediction errors observed for the exponential degradation model and SVM model are high compared to the prediction error of LSTM/bi-LSTM deep learning models. Considering the computational complexity, a comparison is also made based on the computational time. It is observed that the computational time for deep learning or machine learning models is not even in the range of comparison with the statistical estimator model. The observed computation time for the exponential degradation estimator model is 0.28 hours (approx. 17 minutes), whereas the computation time for learning algorithms is from around 2 hours to 24 hours. This is because the prognostic analysis employing an exponential degradation model does not involve any hyperparameter optimization and is basically a curve fitting technique. Machine learning models having less complicated network architectures spend 1.83 hours for their Bayesian optimized hyperparameter tuning and RUL estimation using the SVM model. The deep learning models on the other hand consumed greater time for Bayesian optimized hyperparameter tuning and RUL estimations ranging from 4.49 hours to 23.82 hours. It is also observed that the simple deep learning architectures involving a single LSTM or bi-LSTM layer consume lesser time compared to more complicated deeper LSTM/bi-LSTM combination model architectures. The bi-LSTM model that learns the time-series data from both directions is observed to consume greater time compared to LSTM models.

5.6 Summary

In this work, three data-driven prognostic algorithms based on the deep learning model, machine learning model, and statistical estimator model are developed for the RUL estimation of the lathe spindle unit. Bayesian optimization-based self-optimizing hyperparameter algorithms are developed for deep learning and machine architectures. The Bayesian optimization LSTM/bi-LSTM network algorithm is executed separately for LSTM/bi-LSTM and their combination network architectures to identify the best hyperparameters. The optimized LSTM/bi-LSTM deep learning architectures were tested on an independent dataset to determine their prediction accuracy. Similarly, a Bayesian optimization SVM machine learning algorithm is executed on the lathe spindle dataset for RUL estimation with the best-optimized hyperparameter sets. An exponential degradation statistical estimator model is also used to fit into the lathe spindle health degradation model for RUL estimation analysis. The LSTM/bi-LSTM deep networks are observed effective for prognostic regression analysis using time-series sequence data. The long-term and short-term memory gradients in the LSTM networks can better unravel the hidden trend patterns from time-series data. Training a deep learning algorithm using the extracted vibration signature features avoids the possibility of underfitting the predictive models. The useful machinery degradation information extracted from the raw vibration signals well trains the data-driven prognostic models for RUL estimations. Feature extraction and feature selection approaches have nullified the mandate to have larger-sized data for training intelligent learning models. Bayesian optimization-based self-tuning of hyperparameters partially knocks down the black-box nature deep learning algorithm as the proposed methodology completely avoids the hectic task of manually setting the hyperparameters for training learning algorithms.

Chapter 6

IoT-Based Real-Time Remote Maintenance Decision-Making Dashboard

6.1 Introduction

The present chapter put forward an intelligent system architecture for Internet-of-Things/IoT-based real-time machinery monitoring and intelligent predictive maintenance of the lathe spindle unit. IoT, as a critical Industry 4.0 enabler unfolds the concepts of smart industries and smart machines for efficient machinery health management for higher productivity. The architecture requires the development of a failure alert system and a remote maintenance decision-making dashboard incorporating cloud space storage and cloud computing technologies. A real-time IoT data acquisition and data analytics framework is developed for real-time machinery health status monitoring and analysis to aid in the maintenance decision-making process. The project uses the ThingSpeak™ IoT analytics platform service offered by MathWorks® to establish a real-time lathe spindle health monitoring and health status analysis studies. ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Real-time condition monitoring and upcoming failure prediction information on the ThingSpeak IoT platform provide enhanced effective decision-making in terms of machinery maintenance scheduling and shopfloor allocations.

6.2 Potential of IoT Technology

The term ‘Internet-of-Things / IoT’, coined by Kevin Ashton in 1999 [233], was in use for the past few years and will continue to be a major area to explore in the industrial sector. The concept of IoT dates back to the early 1990s, as utilized in ‘Radio Frequency ID (RFID)’ chips, where the information is transmitted over radio waves to radio waves to communicate its identity and other information [234]. Essentially, IoT refers to providing devices/things representation in the digital realm using unique identities and establishing connectivity among them in a network space. Devices/things connected in such a network space can communicate among themselves, transfer data and information over the internet without any human interventions, hence realizing machine-to-machine (M2M) communication. M2M communication enables networked devices/things to exchange data and execute actions based on preprogrammed algorithms without any human assistance [78]. An IoT software primarily involves the features of data collection, data integration, real-time analytics, and application. The data collection is achieved through a wide variety of sensors systems and associated data acquisition systems. It also establishes M2M communication within the networked devices. The data integration system ensures the required cooperation and networking between the networked devices for smooth communication. It manages various communication protocols to establish connectivity. Real-time analytics involves algorithms that use data from various devices to make viable decisions or execute actions. Finally, the application extends the reach of the existing system and software to a real-world scenario for effective improvement in practical implementations [78, 235].

Building an IoT system requires the devices/things, the internet gateway, the cloud space for storage and analytics, the analytics platform, and finally the user interface. Figure 6.1 shows the major components in a generalized IoT architecture. The devices/things include a sensor that continuously collects data about themselves and the environment and transmits it to the next layer. These devices/things might be connected over a wire, wi-fi, Bluetooth, etc. The IoT gateway act as a middle layer between the devices/things and cloud space which manages the bidirectional data traffic between the networks and protocols. The gateway is to translate different networks and protocols to ensure interoperability of the connected devices/things. Gateway offers a certain level of security for the network and transmitted data. IoT cloud offers tools to collect, process, manage, and store a huge amount of different data in real-time. The

IoT services can easily access these data remotely and make critical analyses when required. IoT analytics is the process of converting meaningful insights from the collected sensor data. Intelligent algorithms are employed to analyze the sensor data over a cloud computing platform for the improvement of products and services. The user interfaces are the visible, tangible part of the IoT system which are accessible to users [5, 236].

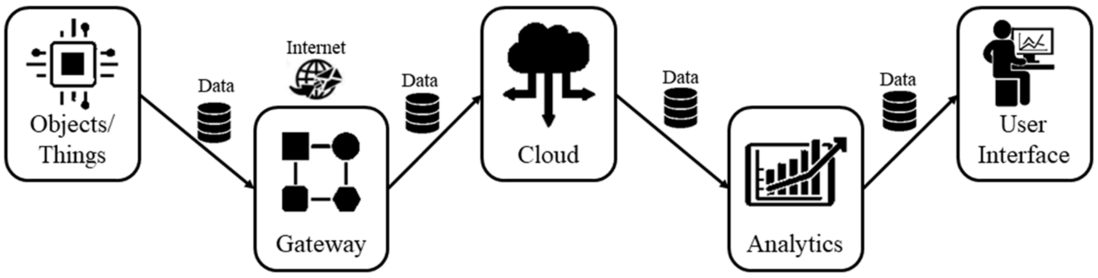


Figure 6.1 Major components of a generalized IoT architecture

The conceptual IoT technologies were successfully implemented in a wide range of real-world applications. These include smart home architecture with connected home appliances [73, 236-238], healthcare framework [239-241], smart cities [242-244], smart transportation [245, 246], smart agriculture with intelligent climate and soil condition monitoring [247-252], manufacturing monitoring with fault diagnosis and prognosis systems [194, 253-255], and many more. IoT-based weather monitoring and forecast systems can make early warnings on natural disasters like floods, earthquakes, storms, hurricanes, etc. [256, 257].

In this Industry 4.0 era, the IoT technologies and artificial intelligence algorithms based on big data analytics are explored in developing innovative solutions for industrial limitations in achieving improved production efficiency. Recent studies have also proved the potential of IoT technologies and big data analytics for improving efficiency, quality and realizing data-oriented predictive maintenance of industrial systems at reduced costs. Ayvaz et al. [201] proposed a data-driven predictive maintenance system for production lines in the manufacturing sector. The real-time IoT sensor data are analyzed employing machine learning algorithms to detect potential failures before their occurrence. Rymaszewska et al. [258] addressed how organizations offering product services can reap the benefits of IoT technologies through the analysis of IoT implementation case studies in manufacturing sectors. Lee et al. [196] introduced an effective cyber-physical system (CPS) architecture for supporting multi-site and multi-products manufacturing focusing on a case in the manufacturing of vehicles' highintensity discharge headlights and cable modules, which are usually manufactured with several multi-manufactured sites. Aheleroff et al. [73] discussed smart appliances under industry 4.0 on transforming conventional home appliances to IoT-enabled smart systems. Khademi et al. [259] proposed a method to convert rotating machinery as IoT-enabled devices using a new generation accelerometer for vibration monitoring. Compare et al. [78] has made a deeper understanding on the relevance of IoT in achieving a maturity stage for the real-world predictive maintenance applications and discussed major research limitation on deployment of IoT-enabled predictive maintenance in the industry. Mekid et al. [260] proposed an IoT-based condition monitoring solution for cutting tool wear monitoring and failure prognostics on the cloud server. Chandra et al. [261] proposed an IoT-based reliable remote monitoring method over the on-site method for the monitoring and control of diesel generators installed for electricity production. Karthik et al. [262] proposed a model for IoT-based preventive maintenance of the alternators and motors inside the aircraft to ensure more safety before every take-off. Tan et a. [263] addressed the implementation of a digital twin-oriented simulation in an IoT-enabled manufacturing environment that synchronizes real-world information in real-time into the digital twin cyberspace. Moens et al. [264] presented the 'Smart Maintenance Living Lab', an open test and research platform that consists of a fleet of drivetrain systems for accelerated lifetime tests of rolling-element bearings, a scalable IoT middleware cloud platform for reliable data ingestion and persistence, and a dynamic dashboard application for fleet monitoring and visualization. Noyjeen et al. [265] developed an IoT-based three-phase induction motor monitoring and diagnosis system. Santiago et al. [266] proposed an efficient IoT-based predictive maintenance system able to identify, predict and notify the occurrence of failure events in Heating, Ventilation, and Air-Conditioning (HVAC) systems. Tao et al. [267] investigated the application of IoT technologies in cloud manufacturing to achieve intelligent perception and access to various manufacturing resources.

6.3 Outline of the Proposed IoT Framework

In this work, an effort is made to seamlessly integrate the relevant information extracted from real-time condition monitoring data, the predictive analytics model trained using historical machine degradation data, machinery health status visualization dashboard, and a real-time upcoming failure warning initiation system to provide meaningful insights into the acquired machinery degradation information. The machinery health degradation information is acquired from the lathe spindle unit using the sensors and data acquisition system. This data is further analyzed to extract useful information for prognostic analysis (refer to Chapter 4). Intelligent prognostic algorithms are trained using the extracted machinery health degradation information to evolve intelligent predictive models for the RUL estimation of lathe spindle unit (refer to Chapter 5).

The best-chosen RUL estimation model is uploaded to the cloud space to make it available for real-time predictive analytics. The real-time machinery health degradation data representing the current operating condition of the lathe spindle is acquired and is made available in the cloud space. An intelligent cloud computing algorithm is developed to estimate the RUL of the current operating lathe spindle unit. This information on the cloud analytics platform is considered for the real-time assessment of upcoming failures and trigger email alerts warnings. The real-time machinery health degradation information of the lathe spindle unit is also made available on a webpage visualization interface. The real-time machinery health status information and failure warnings based on the estimated RUL can be put into use for the predictive maintenance of the lathe machine tool. Figure 6.2 shows the outline of the proposed IoT cloud analytics framework.

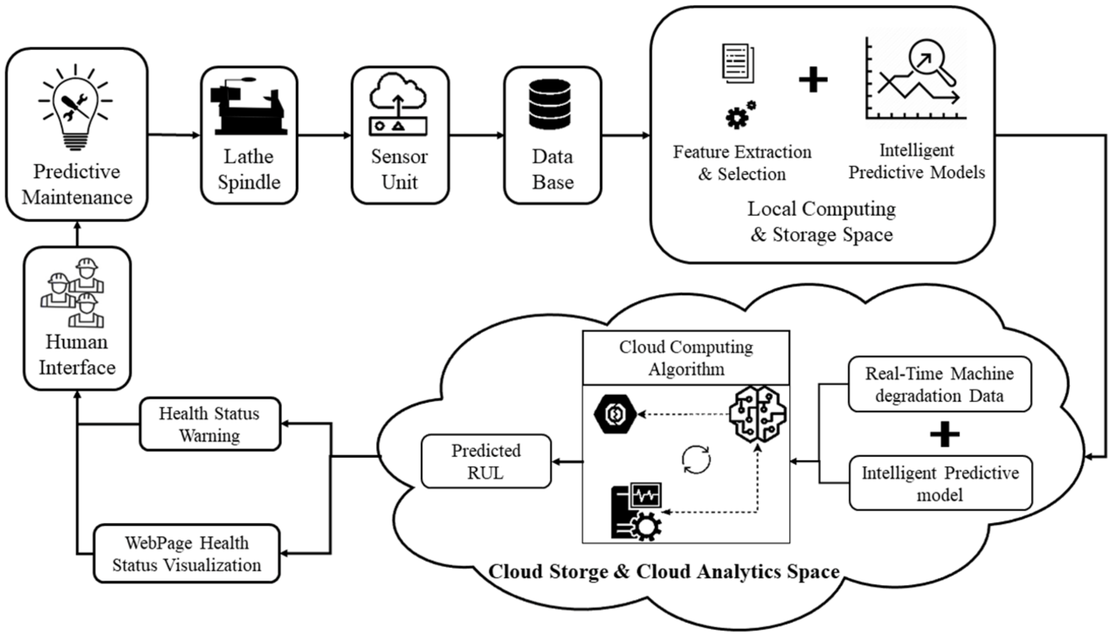

Predictive Maintenance
Lathe Spindle
Sensor Unit
Data Base
Feature Extraction & Selection
Intelligent Predictive Models
Local Computing & Storage Space
Human Interface
Cloud Computing Algorithm
Real-Time Machine degradation Data
Health Status Warning
Predicted RUL
Intelligent Predictive model
WebPage Health Status Visualization
Cloud Storge & Cloud Analytics Space

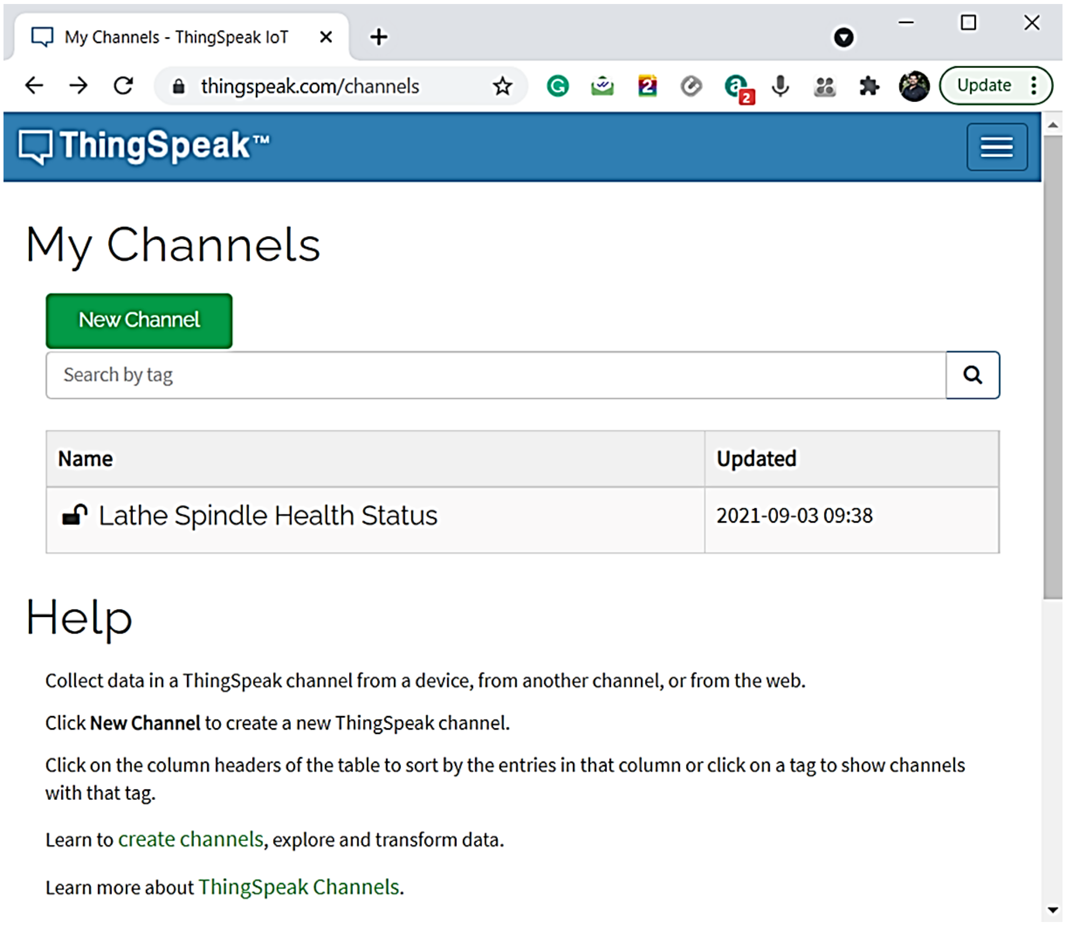
Figure 6.2 Outline of IoT cloud analytics framework

6.4 Cloud-Based Data Management and Predictive Analytics Framework

The cloud-based data management and predictive analytics framework comprises an IoT platform to enable communication with the lathe spindle unit, monitor live data from the local computing and storage system, and manage historical data and evolved predictive models for real-time prognostic analysis. In the present work, the ThingSpeak™ IoT analytics platform service offered by MathWorks® is employed to realize the cloud-based data management and predictive analytics framework. The local computing and storage system is connected to the ThingSpeak IoT platform and the lathe spindle health status visualization and failure alert warning systems were integrated on the ThingSpeak IoT platform.

6.4.1 Overview of ThingSpeak™ IoT Web Service

ThingSpeak is a web-based IoT analytics platform service that allows the user to aggregate, visualize, and analyze real-time data streams in the cloud space [268, 269]. ThingSpeak uses channels to store data transmitted from local storage systems or ground devices over the internet. Figure 6.3 shows a screenshot of the ThingSpeak IoT analytics platform with a channel created to store lathe spindle health status information. Data can be sent to or retrieved from ThingSpeak channels using a REST API or MQTT API (REST- Representational state transfer, MQTT- Message Queue Telemetry Transport, API- Application Programming Interface). The REST API calls are used to create and update the ThingSpeak channels and the MQTT API is used to update the ThingSpeak channels. MQTT is an OASIS and ISO standard messaging protocol for the IoT. In addition, ThingSpeak also encourages cloud-to-cloud interactions, which is utilized in this work to obtain the evolved predictive models for RUL estimation [270].


My Channels - ThingSpeak IoT

thingspeak.com/channels

ThingSpeak™

My Channels

New Channel

Search by tag

Name	Updated
Lathe Spindle Health Status	2021-09-03 09:38

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

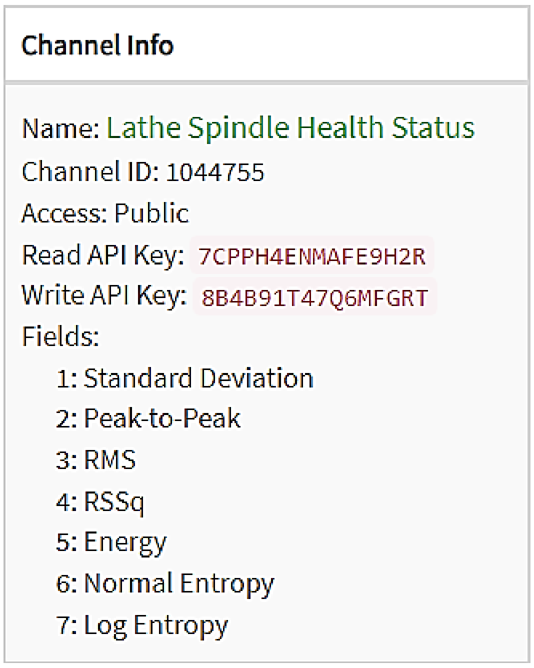
Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Figure 6.3 Screenshot of ThingSpeak^{TM} IoT analytics platform with a live channel created to store lathe spindle health status information

6.4.2 Communication with the IoT Platform

The core element for communication in the ThingSpeak IoT web service is a ThingSpeak channel that establishes connectivity among the devices/things over the internet. ThingSpeak allows the user to store and retrieve data to and from the channel in real-time. The ThingSpeak channel has fields for data, fields for device station location, and fields for status for varied sensed data. REST API calls 'GET' and 'POST' can be used to send and retrieve data to and from the channel respectively. The MQTT 'Publish' method can also be used to update the channel fields and MQTT 'Subscribe' can be used to retrieve channel data. MATLAB functional codes are also employed to write or read data to and from the ThingSpeak channel. The ThingSpeak channel uses separate API keys namely the 'Write API key' and 'Read API key' to write and read data from channel respectively. Figure 6.4 shows a screenshot of the 'lathe spindle health status' channel information with the respective channel ID, the write and read APIs, and the available fields of machinery health degradation data. The data are stored in the ThingSpeak channel in either JSON, XML, or CSV data formats.


Channel Info

Name: Lathe Spindle Health Status
Channel ID: 1044755
Access: Public
Read API Key: 7CPPH4ENMAFE9H2R
Write API Key: 8B4B91T47Q6MFGRT
Fields:
1: Standard Deviation
2: Peak-to-Peak
3: RMS
4: RSSq
5: Energy
6: Normal Entropy
7: Log Entropy

Figure 6.4 Lathe spindle health status channel information

In the present work a MATLAB function code using the channel 'Write API key' is defined to upload the real-time lathe spindle health degradation data from the local storage space to the IoT ThingSpeak channel. The selected vibration signature features mentioned in 'section 4.3.2' are sent to the 'lathe spindle health status' monitoring channel employing the 'Write API key' function. The live data acquired from the lathe spindle test setup is first analyzed to extract the meaningful vibration signature feature (section 4.3) and then selected features uploaded to the ThingSpeak data collection channel at a rate of once in every 60 seconds (See Appendix VI for MATLAB code to upload data to ThingSpeak cloud space). The different vibration signature features are uploaded to separate fields of data defined in the ThingSpeak channel namely standard deviation, peak-to-peak, RMS, RSSq, energy, normal entropy, and log entropy. Similarly, a MATLAB function code using the channel 'Read API key' is defined to retrieve the data from the channel when required for analysis. The data read from the channel fed in a cloud computing space where the live lathe spindle health degradation data is analyzed against the evolved predictive model for the RUL estimation of lathe spindle unit and further failure alert warning through email notifications. The evolved predictive model required for the RUL estimation is made available in the cloud space through the 'Drop Box' cloud storage service provider. The identified best predictive model, the LSTM + bi-LSTM predictive model is uploaded to the DropBox server and is called in the ThingSpeak cloud analytics space when required. Figure 6.5 shows the flow of data in the proposed IoT system.

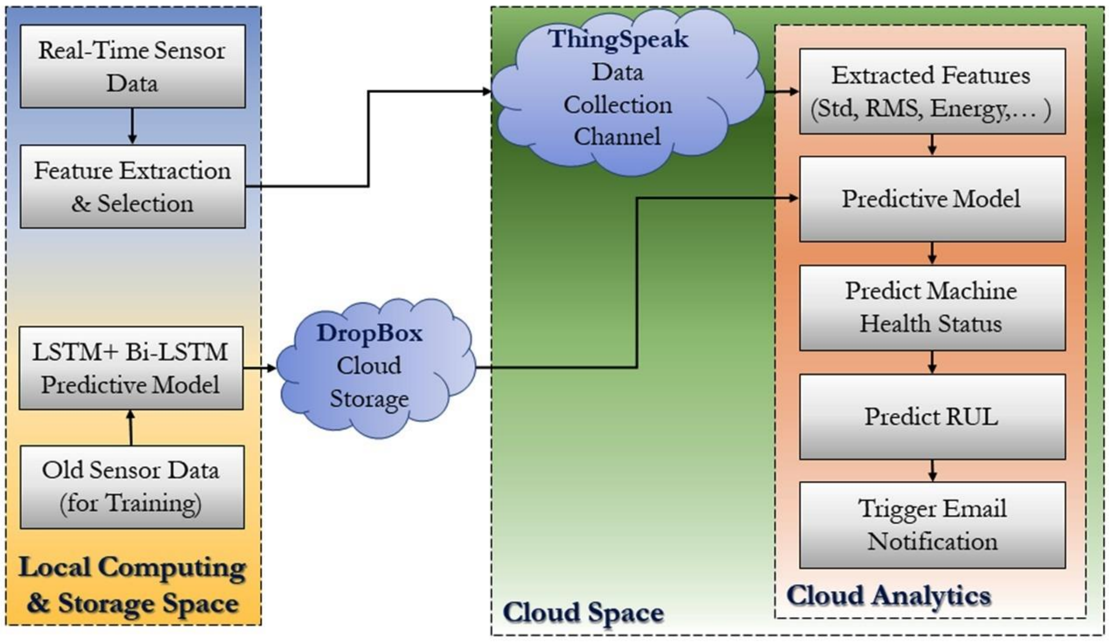

Real-Time Sensor Data
Feature Extraction & Selection
LSTM+ Bi-LSTM Predictive Model
Old Sensor Data (for Training)
Local Computing & Storage Space
DropBox Cloud Storage
ThingSpeak Data Collection Channel
Extracted Features (Std, RMS, Energy,...)
Predictive Model
Predict Machine Health Status
Predict RUL
Trigger Email Notification
Cloud Analytics
Cloud Space

Figure 6.5 Flow of data in IoT Platform

6.5 Cloud Computing Architecture and User Interface for Predictive Analytics

ThingSpeak IoT web service provides a MATLAB analysis cloud computing paradigm for analyzing the channel data. The architecture comprises the cloud storage, cloud computing, and user interface stages. Figure 6.6 shows the flow diagram of IoT cloud computing architecture and user interface for the predictive analytics of the lathe spindle unit. The IoT cloud computing architecture and the user interface are discussed in detail in the following sections.

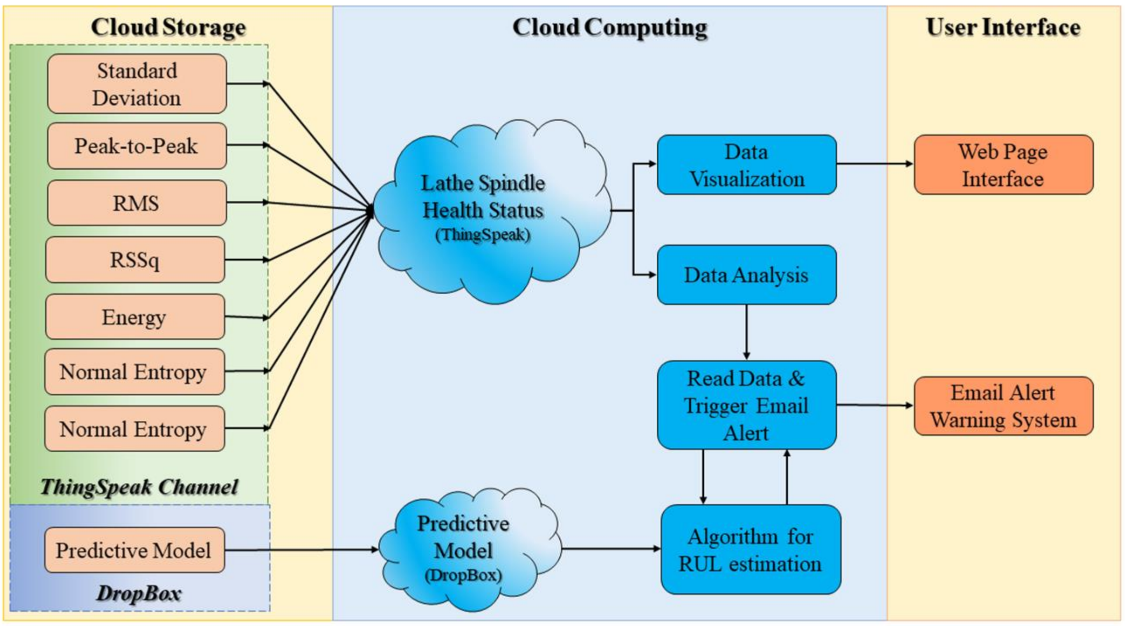


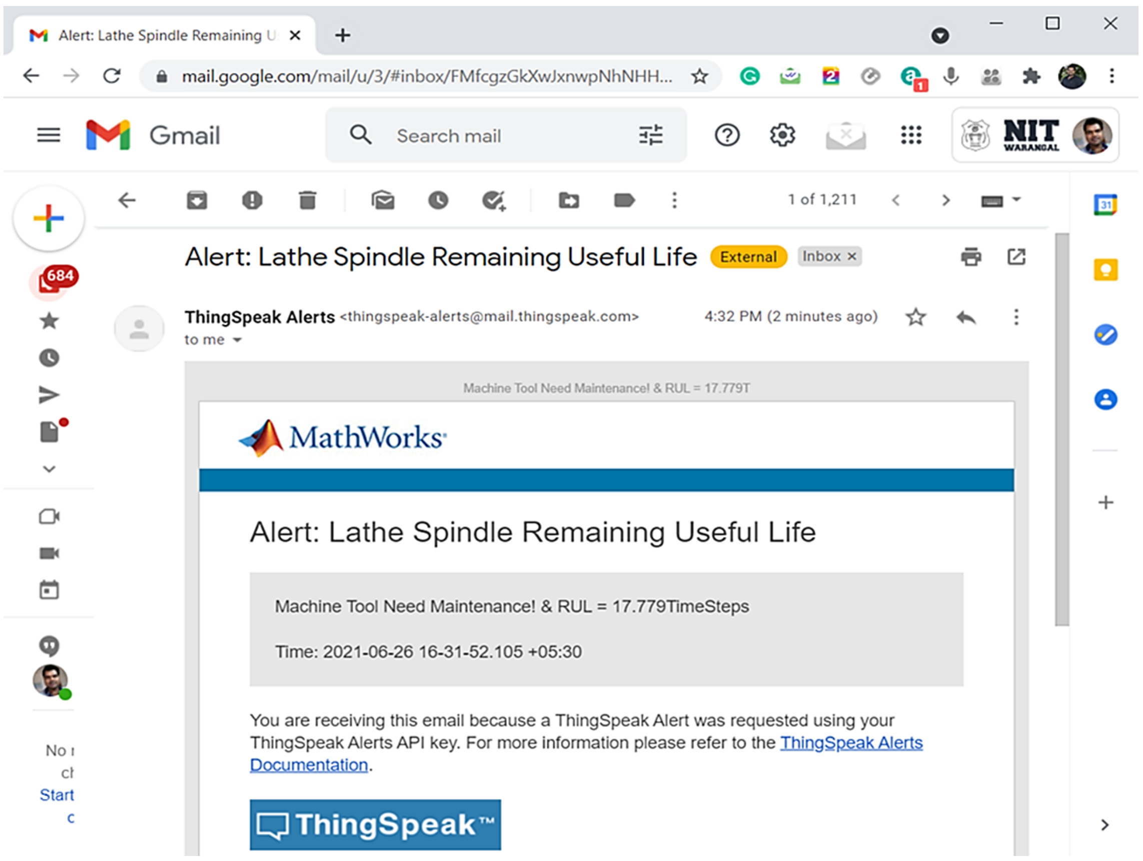
Figure 6.6 IoT cloud computing architecture and user interface for the predictive analytics of lathe spindle unit

6.5.1 Lathe Spindle RUL Estimation and Failure Warning System

The RUL estimation required the lathe spindle real-time health status information and the evolved intelligent predictive model. The real-time lathe spindle health degradation data stored in the 'lathe spindle health status' channel is read using the 'Read API key' and the intelligent predictive model stored in a 'Drop Box' cloud storage is called using the 'DropBox Access Token', which the ''Drop Box API key'. The RUL life estimation algorithm first retrieves the intelligent predictive model and then the features representing the real-time health status of the lathe spindle. (See Appendix VII for the MATLAB code for cloud computing algorithm for the real-time RUL estimation and email alert warnings)

The algorithm predicts the RUL of the lathe spindle by analyzing the current health status information and historical health degradation patterns. The current health status of the lathe spindle is revealed through the extracted vibration signature features standard deviation, peak-to-peak, RMS, RSSq, energy, normal entropy, and log entropy, which are made available in the cloud space. The historical health degradation patterns of the lathe spindle are represented through the evolved LSTM + Bi-LSTM prognostic regression model. The degradation pattern of the features is analyzed against the evolved prognostic regression model to estimate the RUL of the lathe spindle unit. The algorithm automatically executes at regular intervals of time and also can be executed at any time in the interest of the user. The evaluated RUL values are saved for further analysis.

The determined RUL values are then used as a facet to trigger the failure warning system through email alerts. The system is designed to evaluate the RUL at a regular interval of time and if the obtained RUL value crosses a predefined safe threshold, the system automatically triggers the email alerts. The email alert warning includes a message on if the machine is running well or if the machine requires maintenance. The email also provides the available RUL of the spindle unit, which also provides the user to make a suitable decision regarding machinery maintenance and planning. Figure 6.7 provides a screenshot of the generated email alert warning.


Alert: Lathe Spindle Remaining Useful Life

ThingSpeak Alerts <thingspeak-alerts@mail.thingspeak.com> 4:32 PM (2 minutes ago)

to me

Machine Tool Need Maintenance! & RUL = 17.779T

Alert: Lathe Spindle Remaining Useful Life

Machine Tool Need Maintenance! & RUL = 17.779TimeSteps

Time: 2021-06-26 16-31-52.105 +05:30

You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the ThingSpeak Alerts Documentation.

Figure 6.7 Screenshot of generated email alert warning

6.5.2 Webpage Visualization of Lathe Spindle Health Status

The outputs of the proposed IoT architecture are integrated into a web page user interface that allows the user to have live monitoring of lathe spindle health status from anywhere in the world. The live lathe spindle health degradation data sent to the ThingSpeak channel is reflected as such in the webpage interface on a real-time scale. Figure 6.8 shows screenshots on the developed webpage user interface.

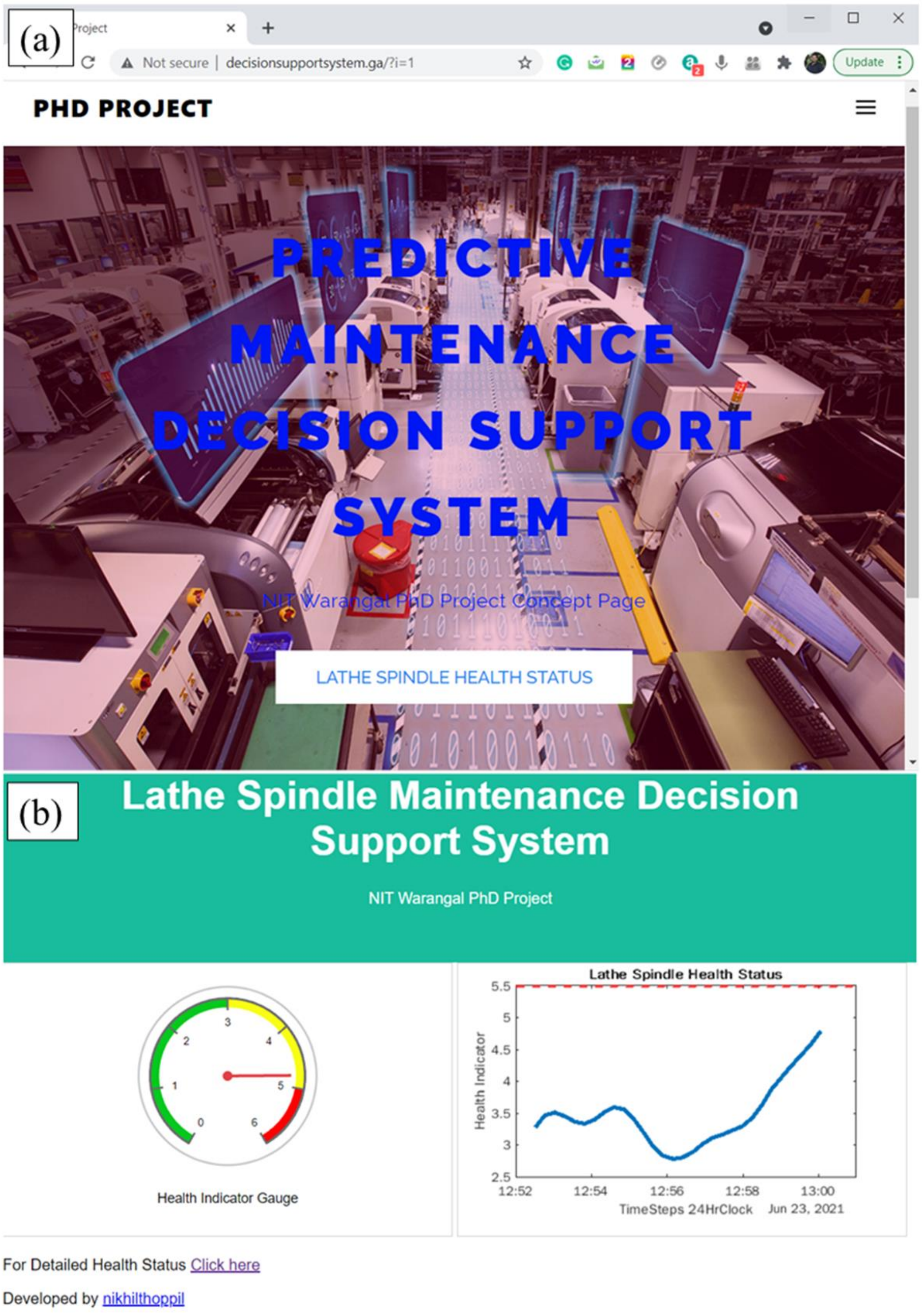

(a)
Project
Not secure | decisionsupportsystem.ga/?i=1
Update
PHD PROJECT
PREDICTIVE MAINTENANCE DECISION SUPPORT SYSTEM
NIT Warangal PhD Project Concept Page
LATHE SPINDLE HEALTH STATUS
(b)
Lathe Spindle Maintenance Decision Support System
NIT Warangal PhD Project
Health Indicator Gauge
Lathe Spindle Health Status
Health Indicator
TimeSteps 24HrClock
Jun 23, 2021
For Detailed Health Status Click here
Developed by nikhilthoppil

Figure 6.8 Screenshot of developed web page interface (a) main introduction page, (b) Health Indicator dial gauge and degradation trend

Figure 6.8 (a) displays the introduction page detailing the identity and purpose of the webpage. A live health status indicator dial gauge is also designed and integrated into the webpage for easy monitoring of lathe spindle health status. Figure 6.8 (b) displays the live health status indicator dial gauge that has three stages of dial gauge indicating the safe (Green), alert (Yellow), and faulty (Red) zones of lathe spindle operational lifetime. The page also displays the health degradation pattern of the current operating lathe spindle. As shown in Figure 6.9, the webpage can also redirect to a detailed graphical representation of the lathe spindle health degradation patterns that are represented through the selected vibration signature features. The real-time lathe spindle health status information available on the webpage interface enables the user to make appropriate decisions regarding the maintenance planning of the lathe machine tool system. The interface can be utilized as a real-time maintenance decision-making dashboard for the lathe machine tool system.

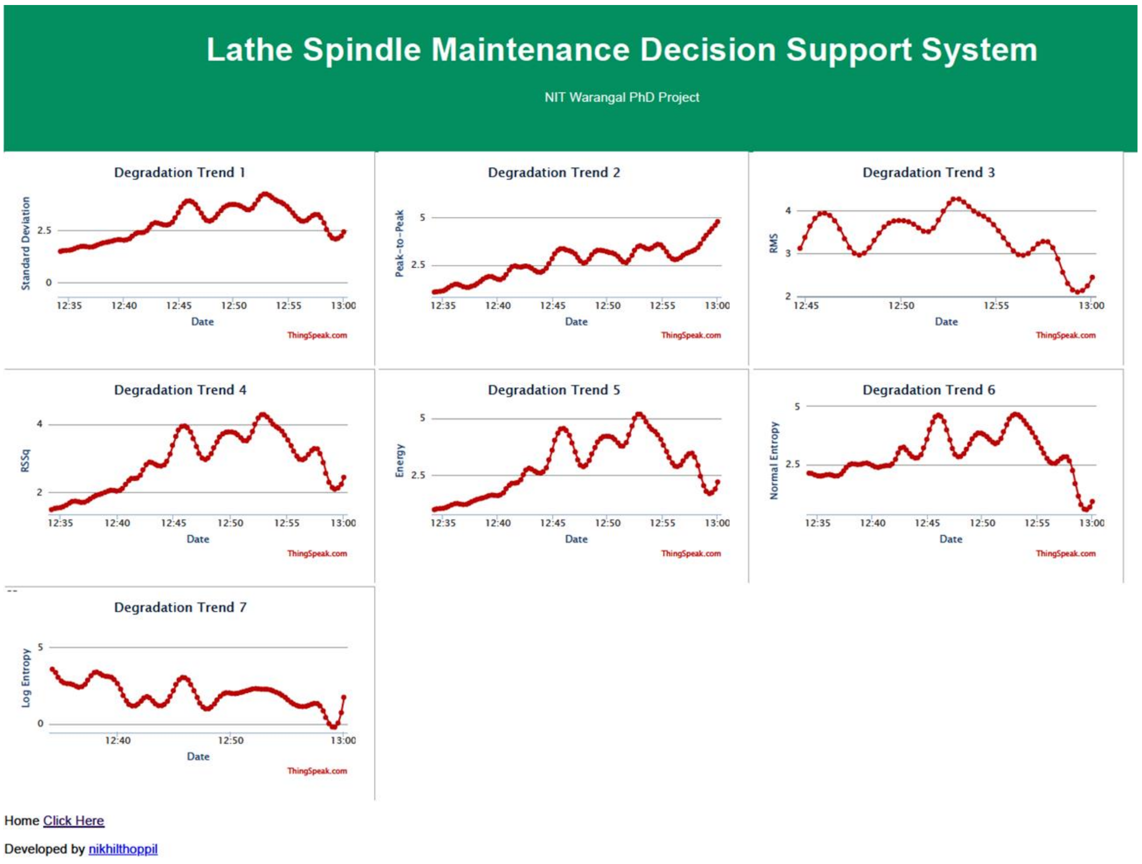


Figure 6.9 Screenshot of Detailed webpage view of lathe spindle degradation trend

6.6 Summary

The chapter presents a real-time maintenance decision-making dashboard collaborating the modern digital technological paradigms IoT, Big Data Analytics, Cloud Computing, and sophisticated predictive analytics algorithms. In this work, a simple and efficient framework is proposed for real-time IoT-based machinery health status monitoring and maintenance decision-making. The ThingSpeak IoT web service platform and DropBox cloud storage services are utilized for developing the IoT-based real-time maintenance decision-making dashboard. The real-time lathe spindle monitoring vibration data is analyzed to extract signature features and then the selected features are simultaneously sent to a cloud space storage on a real-time basis. The best evolved data-driven prognostic model is also stored in a cloud space to make it available for predictive analytics. The features representing the current health status of the lathe spindle are analyzed against the trained predictive model to estimate the real-time RUL of the lathe spindle unit. This RUL value crossing a predefined threshold automatically triggers email warnings to alert the user regarding the maintenance of the lathe machine tool. The features in the cloud space are integrated into a webpage user interface to provide a real-time health status visualization of the lathe spindle unit. The webpage interface displays a health indicator dial gauge calibrated to the real-time health status of the lathe spindle.

Chapter 7

Summary and Conclusions

7.1 Summary

The present research work comprehends and contributes to the existing understanding of the condition monitoring and intelligent predictive maintenance of industrial machinery. The studies provide insights into the potential of intelligent data-driven learning algorithms, advanced big data technologies, and IoT-based cloud computational techniques in the application of machinery health degradation data monitoring, data exploration, and prognostic regression analysis. The thesis portrays the design and development of an intelligent predictive maintenance framework for the critical components of the CNC lathe machine tool in an industry 4.0 scenario. The study utilizes the computational techniques fuzzy modified FMECA for the maintenance prioritization of CNC machine tool components and the data-driven prognostic algorithms such as LSTM deep learning models, SVM machine learning models, and exponential degradation statistical estimator models to evolve intelligent RUL estimation models. The generated machinery health degradation data is employed to train the intelligent data-driven prognostic models. The study is expected to motivate industrial practitioners for developing a remote machinery health monitoring and data acquisition system with an intelligent predictive maintenance module, thus making them a worthful candidate to compete in the current Industry 4.0 era.

7.2 Conclusions

This research work is focused on studies on the development of a data-driven intelligent predictive maintenance framework for the critical components of the CNC lathe machine tool in an industry 4.0 scenario. Based on the research work, the followings are the conclusions:

(a) Criticality analysis of the CNC lathe machine tool subsystems is performed to identify the most critical subsystems and their potential failure modes from a maintenance perspective and hence limit the implementation of predictive maintenance to the identified critical subsystems. The Fuzzy modified FMECA technique is effectively employed for the maintenance prioritization of lathe machine tool subsystems. The spindle unit of a CNC lathe is identified as the most critical subsystem with the highest fuzzy RPN value of 848.2. Furthermore, FMECA relates the potential failure modes to potential effects and root causes. The wear and deformation of spindle bearings causing increased noise and vibration are identified as the potential failure causes and effects in the lathe spindle unit. This knowledge can be utilized in the phenomenon of sensors selection and installation of the condition monitoring system for the critical components.

(b) An accelerated run-to-failure experimental test rig with sensors and a data acquisition system is fabricated to obtain the operational health degradation patterns of a CNC lathe spindle unit. The acquired vibration sensor signals do not provide any meaningful information regarding the health degradation patterns of the lathe spindle unit. These sensor signals need to be preprocessed and explored to extract useful information for effectual learning using computational algorithms. The Vibration signature feature in the time, frequency, and time-frequency domain extracts the health degradation patterns from the raw vibration signals. Further, an NCA-based feature selection algorithm identified the features' standard deviation, peak-to-peak, RMS, RSSq, energy, normal entropy, and log entropy as significant for prognostic regression analysis. Lathe spindle accelerated run-to-failure health degradation data acquired from 14 individual experimental runs are prepared for the prognostic regression analysis employing intelligent data-driven algorithms. As well, the feature extraction and feature selection approach employed for the preparation of health degradation data have nullified the mandate to have larger-sized data for training intelligent learning models.

(c) Three data-driven algorithms, the deep learning LSTM/bi-LSTM network models, machine learning SVM model, and statistical estimator exponential degradation model algorithms are successfully employed for the prognostic regression analysis. The selected vibration signature features representing the lathe spindle health degradation are used to train the data-driven algorithms to evolve intelligent RUL estimation models. The LSTM + bi-LSTM network architecture is identified to have the best prediction accuracy on lathe spindle RUL estimation with RMSE equals 31.65 followed by the single LSTM architecture with RMSE equals 40.01. The LSTM architecture is well efficient in digging up hidden patterns from time-series data. Further employing a bi-LSTM network can refine the learned degradation patterns yielding accurate estimations. Furthermore, an RMSE equal to 206.23 is obtained as the prediction accuracy for the SVM model and an RMSE of 114.00 is obtained for the exponential degradation model.

(d) One of the major challenges in training deep learning and machine learning algorithms for prognostic regression analysis is the selection of suitable hyperparameter values to obtain maximum prediction accuracy. A Bayesian optimized deep learning and machine learning architecture are proposed to have an automated hypermeter tuning to evolve the best intelligent predictive models. Intricacy in manually setting the best hyperparameters is overcome through the Bayesian Optimization approach. Bayesian optimization-based self-tuning of hyperparameters partially knocks down the black-box nature deep learning algorithm as the proposed methodology completely avoids the hectic task of manually setting the hyperparameters for training learning algorithms.

(e) In order to make a further evaluation on the prediction accuracy and computational complexity of the proposed data-driven approaches the deep learning model, machine learning model, and statistical estimator model, the MAPE and maximum computational time for each algorithm are compared. The deeper LSTM + bi-LSTM leaning architecture is trained best on the lathe spindle health degradation data with a MAPE of 4.45%. The MAPE of the exponential degradation model is 10.63% respectively, and that for the SVM model is 23.18% respectively. The observed computation time for the exponential degradation estimator model is approximately 17 minutes, whereas the computation time for learning algorithms ranges from 2 hours to 24 hours. This is because the prognostic analysis employing an exponential degradation model does not involve any hyperparameter optimization and is basically a curve fitting technique.

(f) An IoT cloud-based simple and efficient predictive maintenance framework are proposed for real-time machinery health status monitoring and RUL estimation. The vibration signature features representing the lathe spindle health degradation patterns and the intelligent predictive models for RUL estimation are seamlessly integrated to realize a real-time remote maintenance decision-making dashboard. The acquired live lathe spindle health degradation data is analyzed against the trained intelligent predicted model to estimate the RUL. The estimated RUL values crossing a predefined threshold automatically trigger email warnings intimating the user regarding the maintenance of the lathe spindle. The real-time lathe spindle health status information is also made available in a webpage user interface, which allows the user to access the lathe spindle live health status information from anywhere in the world.

With the advancements in technology, industrial practitioners can integrate new technologies, including AI and machine learning, IoT, and cloud computing and analytics into their production facilities. The thesis provides insight into the scope and challenges of the Industry 4.0 era and future machine tool technologies. Advanced sensor technology and computational algorithms are suitably integrated to develop a remote machinery health monitoring and data acquisition system with an intelligent predictive maintenance module. The machine criticality analysis can identify the most critical machine system and its subsystems that need to be considered for predictive maintenance. The information on the potential failure of causes component provides insights into the selection of suitable sensors for machinery condition monitoring. The sensors and associated data acquisition systems are programmed to monitor and acquire machine operating data on a real-time basis. The extracted data is further analyzed to obtain meaningful information on machinery health degradation to better train intelligent predictive algorithms. Subsequently, the acquired machinery condition monitoring information is made available in the IoT cloud computing and analytics platform for further predictive analytics from anywhere in the world.

7.3 Scope for Future Work

(a) Multi-sensor data fusion with CNC controller data can be explored to improve the quality and information content of machinery degradation data, which consequently improves the performance of prognostic regression models.

(b) A large-size multi-sensor dataset for training deep learning algorithms could avoid the requirement for feature extraction and selection.

(c) The application of hybrid computational algorithms can be studied to better train the machinery health degradation data for RUL estimation.

(d) IoT-based data analytics can be elaborated to control the operation of industrial machinery concerning the present machinery condition monitoring data.

(e) Multi-sensor data and CNC controller data can be mapped to surface roughness and dimensional tolerances to develop an advanced predictive paradigm of machined product quality.

Appendix I

Vibration Signal Signature Feature Extraction

Raw Vibration Signal Processing & Feature Extraction

```
N = 14;
featureTableArray = cell(N,1);
SfeatureArray = cell(N,1);
SfeatureTableArray = cell(N,1);

for i=1:N
    file=sprintf('SDA%d.mat',i);
    feature = sprintf('allFeatures%d',i);

    load(file);
    len = length(singleDataArray);

    fs = 25600;
    tstart = 0;

    %Plot raw vibration signals
    figure
    hold on
    for k = 1:len
        v= singleDataArray{k,1}(:,1);
        t = tstart + (1:length(v))/fs;
        plot(t, v)
        tstart = t(end);
    end
    hold off

       xlabel('Time (minutes), 1 seconds per every 60 seconds')
       ylabel('Acceleration (g)')

tstart = 0;
SE_array = cell(len, 1);
denoisedata_array = cell(len,1);
for k = 1:len
```

```
% Denoise using Wavelet analysis
    v = wdenoise(singleDataArray{k,1}(:,1), 8,...
                 'Wavelet', 'sym4', ...
                 'DenoisingMethod', 'Bayes', ...
                 'ThresholdRule', 'Median', ...
                 'NoiseEstimate', 'LevelIndependent');

     denoisedata_array{k,1} = v;

% spectral entropy
    t = tstart + (1:length(v))/fs;
    [se,te] = pentropy(v,t');
    SE_array{k,1} = se;

end
    features = matfile(feature,'Writable', true);

for k = 1:len
    v = denoisedata_array{k,1}(:,1);

    SE = SE_array{k,1};
    SPW = spectralPowerFeatures(v,fs);
    SPK = spectralPeaksFeatures(v,fs);
    order = [2,2];

 % Time Domain Features
    features.mean(k,1) = mean(v);
    features.Std(k,1) = std(v);
    features.Skewness(k,1) = skewness(v);
    features.Kurtosis(k,1) = kurtosis(v);
    features.Peak2Peak(k,1) = peak2peak(v);
    features.RMS(k,1) = rms(v);
    features.Peak2RMS(k,1) = peak2rms(v);
    features.RSSq(k,1) = rssq(v);
    features.CrestFactor(k,1) = max(v)/features.RMS(k,1);
    features.ShapeFactor(k,1) = features.RMS(k,1)/mean(abs(v));
    features.ImpulseFactor(k,1) = max(v)/mean(abs(v));
    features.MarginFactor(k,1) = max(v)/mean(abs(v))^2;
    features.Energy(k,1) = sum(v.^2);
    features.Entropy(k,1) = wentropy(v,'norm',1.1);
    features.ShEntropy(k,1) = wentropy(v,'shannon');
    features.LogEntropy(k,1) = wentropy(v,'log energy');
    features.SpEntropy(k,1) = pentropy(v,fs,'Instantaneous',false);
    %Wavelet packet node energy
    [c,l] = wavedec(v,3,'sym4');%1D wavelet decomposition
    [Ea,Ed] = wenergy(c,l);
    features.WPDEnergy(k,1) = Ea;
    features.momentJ(k,1) = tfmoment(v,time,order);%seconds(1:1:N)

    features.corDim(k,1) = correlationDimension(v);
    features.approxEnt(k,1) = approximateEntropy(v);
    features.lyapExp(k,1) = lyapunovExponent(v,fs);
    %P0 is the spectrogram, fvec is the frequencyvector and
    % tvec is the time vector.
    [~,fvec,tvec,P_k] = spectrogram(v,500,450,512,fs);
    [~,I] = max(P_k);
    features.meanPeakFreq(k,1) = mean(fvec(I));
    features.RMSF = sqrt(sum((abs(S)/length(S)).^2));
    features.meanfreq(k,1) = meanfreq(v,fs);
    features.medfreq(k,1) = medfreq(v,fs);
    features.spuriousfreedr(k,1) = sfdr(v,fs);
    features.distrotionRatio(k,1) = sinad(v,fs);
    features.intercept(k,1) = toi(v,fs);
    features.bw(k,1) = obw(v,fs);
    features.bp(k,1) = bandpower(v);
    features.pbw(k,1) = powerbw(v,fs);

    % Spectral Kurtosis related features
    features.SKMean(k,1) = mean(SK);
    features.SKStd(k,1) = std(SK);
    features.SKSkewness(k,1) = skewness(SK);
    features.SKKurtosis(k,1) = kurtosis(SK);

    % Spectral Entropy related features
    features.SEMean(k,1) = mean(SE);
    features.SEStd(k,1) = std(SE);
    features.SESkewness(k,1) = skewness(SE);
    features.SEKurtosis(k,1) = kurtosis(SE);
    features.SPower(k,1) = SPW(1);
    features.SPeakpos(k,1) = SPK(1);
    features.SPeakpow(k,1) = SPK(7);
end
featureTableArray{i,1} = table(features.mean,features.Std,...
features.Skewness,features.Kurtosis,features.Peak2Peak,features.RMS,
features.Peak2RMS,...
feature.RSSq,features.CrestFactor,features.ShapeFactor,features.Impu
lseFactor,...
features.MarginFactor,features.Energy,features.Entropy,features.ShEn
tropy,...
features.LogEntropy,features.SpEntropy,features.momentJ,features.cor
Dim,...
features.approxEnt,features.lyapExp,features.WPDEnergy,features.mean
freq,...
features.medfreq,features.spuriousfreedr,features.distrotionRatio,..
.
features.intercept,features.bw,features.bp,features.pbw,features.SKM
ean,...
features.SKStd,features.SKSkewness,features.SKKurtosis,features.SEMe
an,...
features.SEStd,features.SESkewness,features.SEKurtosis,features.SPow
er,...
features.SPeakpos,features.SPeakpow,...
'VariableNames',{'mean','Std','Skewness','Kurtosis','Peak2Peak',...
'RMS','Peak2RMS','RSSq','CrestFactor','ShapeFactor','ImpluseFactor',
...
'MarginFactor','Energy','Entropy','ShEntropy','LogEntropy','SpEntrop
y',...
'WPDEnergy','momentJ','corDim','approxEntropy','lyapExp','MeanFreq',
...
'MedianFreq','SFDR','DistrotionRatio',...
'3rdIntercept','BandWidth','BandPower','PowerBW',...
'SKMean','SKStd','SKSkewness','SKKurtosis','SEMean',...
'SEStd','SESkewness','SEKurtosis','SPower','SPeakPos','SPeakPow'});
%smooth features
Sfeaturetable = smoothdata( featureTableArray{i,1});
SfeatureTableArray{i,1} = Sfeaturetable;
SfeatureArray{i,1} = table2array(Sfeaturetable);
end
N = 14;
SfeatureTimeTableArray =cell(N,1);
for i = 1:N
    lenTable = size(SfeatureTableArray{i,1},1);
    SfeatureTimeTableArray{i,1} =
table2timetable(SfeatureTableArray{i,1},'RowTimes',seconds(1:1:lenTa
ble));
end
```

Array of feature matrix

```
selectedFeatureArray = cell(N,1);
for o = 1:N
    selectedFeatureArray{o,1} =
table2array(selectedFeatureTableArray{o,1});
end
SortedfeatureSelectedArray = cell(N,1);
%Plot Extracted Features
for j=1:N
    featureSelected = selectedFeatureTableArray{j,1};
    featureSelectedNorm = normalize(featureSelected)
    featuretable = timetable2table(featureSelectedNorm);
    featuretable = removevars(featuretable,{'Time'});
    featurearray = table2array(featuretable);
    SortedfeatureSelectedArray{j,1} = featurearray;
    s = size(featurearray,2);
    names = featureSelectedNorm.Properties.VariableNames;
    nS   = sqrt(s);
    nCol = ceil(nS);
    nRow = nCol - (nCol * nCol - s > nCol - 1);
    figure
    hold on
    for i = 1:s
      subplot(nRow,nCol,i);
      plot(featureSelectedNorm.Time,featurearray(:,i));
      title(names{i});
    end
end
```

Appendix II

NCA Regression-Based Feature Selection

Load the sample data.

```
N = size(featureTableArray, 1);
tn = 2; %number of test data
TrainDataArray = cell(N-tn,1);

for i = 1:N-tn
    TrainDataArray{i,1} = table2array(featureTableArray{i,1});
 %RUL for each TrainData
    s = size(TrainDataArray{i,1} ,1);
    timeSteps = 1:1:s; %time column of featuretable
    TrainDataArray{i,1}(:,end+1) = fliplr(timeSteps)';
end

%make single feature matrix
TrainData = cell2mat(TrainDataArray);
XTrain = TrainData(:,1:end-1);%end-1%Predictors
YTrain = TrainData(:,end);

mu = mean(XTrain,2);
sigma = std(XTrain,0,2);
XTrain = (XTrain-mu)./sigma;
%Fit a neighborhood component analysis (NCA)
% model for regression to detect the relevant features.
nca = fsrnca(XTrain,YTrain,'Standardize',1,'Lambda',0.0666);
%Plot the feature weights.

figure()
plot(nca.FeatureWeights,'ro')
xlabel('Feature Index')
ylabel('Feature Weight')
grid on
TestDataArray = cell(2,1);
i = 1;
for k = N-(tn-1):N
    TestDataArray{i,1} = table2array(featureTableArray{k,1});
 %RUL for each TrainData
    s = size(TestDataArray{i,1} ,1);
    timeSteps = 1:1:s; %time column of featuretable
    TestDataArray{i,1}(:,end+1) = fliplr(timeSteps)';
    i = i+1;
end
```

```
%make single feature matrix
TestData = cell2mat(TestDataArray);
XTest = TestData(:,1:end-1);%end-1%Predictors
YTest = TestData(:,end);

mu = mean(XTest,2);
sigma = std(XTest,0,2);
XTest = (XTest-mu)./sigma;
```

Compute the regression loss.

```
L = loss(nca,XTest,YTest,'LossFunction','mad')
```

Compute the predicted response values for the test set and plot them versus the actual response.

```
YPred = predict(nca,XTest);
figure()
plot(YPred,YTest,'bo')
xlabel('Predicted response')
ylabel('Actual response')
xlim([0 1300])
ylim([0 1300])
SelectedFeatureArray = cell(N,1);
for m = 1:N
    SelectedFeatureArray{m,1} = featureTableArray{m,1}...
                                (:,nca.FeatureWeights(:,:)>1);

end
for j=1:N
    lenTable = size(SelectedFeatureArray{j,1},1);
    TimeStep = 1:1:lenTable;
    featurearray = table2array(SelectedFeatureArray{j,1});
    s = size(featurearray,2);
    names = SelectedFeatureArray{j,1}.Properties.VariableNames;
    nS   = sqrt(s);
    nCol = ceil(nS);
    nRow = nCol - (nCol * nCol - s > nCol - 1);
    figure
    hold on
    for i = 1:s
      subplot(nRow,nCol,i);
      plot(TimeStep,featurearray(:,i));
      title(names{i});
    end
end
```

Appendix III

Bayesian Optimization Deep Learning Model-Based Prognostic Algorithm

Load Data

```
load ('SDATrainC.mat');
numofFailData = size(TrainDataArray, 1);
numofFeatures = size(TrainDataArray{1,1}, 2);
XTrain = cell(numofFailData,1);
YTrain = cell(numofFailData,1);
for i = 1:numofFailData
    fieldData = table2array(TrainDataArray{i,1});
    X = fieldData;
    XTrain{i} = X';
    s = size(X,1);
    timeSteps = 1:1:s;
%   timeSteps = timeSteps/max(timeSteps);
    Y = fliplr(timeSteps);
    YTrain{i} = Y;
end
%Normalize Data
mu = mean([XTrain{:}],2);
sigma = std([XTrain{:}],0,2);
XTrain = cellfun(@(X) (X-mu)./sigma,XTrain,'UniformOutput',false);
%Clip Data
for i = 1:numel(YTrain)
    s = size(TrainDataArray{i,1},1);
    if (s<900)
      thr = 550;
    elseif(s<1300)
      thr = 800;
    else
      thr = 1400;
    end
    YTrain{i}(YTrain{i} > thr) = thr;
end
%Sort Data
for i=1:numel(XTrain)
    sequence = XTrain{i};
    sequenceLengths(i) = size(sequence,2);
end
[sequenceLengths,idx] = sort(sequenceLengths,'descend');
XTrain = XTrain(idx);
YTrain = YTrain(idx);
```

```
%Test Validation Data
load('SDATestC.mat');
load('SDArulC.mat');
numofTestData = size(TestDataArray,1);
numofTestFeatures = size(TestDataArray{1,1},2);
XTest = cell(numofTestData,1);
YTest = cell(numofTestData,1);
% YTestSteps = cell(numofTestData,1);
for i = 1:numofTestData
    fieldData = table2array(TestDataArray{i,1});
    X = fieldData;
    XTest{i} = X';
end
for j = 1:numofTestData
    X = XTest{j};
    sequenceLength = size(X,2);
    rul = rulData(j);
    YTest{j} = rul+sequenceLength-1:-1:rul;
end
XTest = cellfun(@(X) (X-mu)./sigma,XTest,'UniformOutput',false);
for i = 1:numel(XTest)
    s = size(XTest{i,1},2);
    if (s<900)
      thr = 550;
    elseif(s<1300)
      thr = 800;
    else
      thr = 1400;
    end
    YTest{i}(YTest{i} > thr) = thr;
end

% XValidation = XTest;
% YValidation = YTest;
XTest = XTest{1,1};
YTest = YTest{1,1};
%Validation Data
load ('SDAValC1.mat');
numofValData = size(ValDataArray, 1);
XValidation = cell(numofValData,1);
YValidation = cell(numofValData,1);
for i = 1:numofValData
    fieldData = table2array(ValDataArray{i,1});
    X = fieldData;
    XValidation{i} = X';
    s = size(X,1);
    timeSteps = 1:1:s; %time column of featuretable

    Y = fliplr(timeSteps);
    YValidation{i} = Y;
end
```

Choose Variables to Optimize

```
optimVars = [optimizableVariable('InitialLearnRate',[0.0001
0.04],'Transform','log')
            optimizableVariable('numHiddenUnits1',[100 500],'Type','integer')
%              optimizableVariable('numHiddenUnits2',[100
300],'Type','integer')
            optimizableVariable('ConnectedLayer1',[25 250],'Type','integer')
%              optimizableVariable('ConnectedLayer2',[25 250],'Type','integer')
            optimizableVariable('dropout1',[0.2 0.8])
%              optimizableVariable('dropout2',[0.2 0.8])
            optimizableVariable('maxEpochs',[100 300],'Type','integer')
            optimizableVariable('L2Regularization',[1e-5 1e-
2],'Transform','log')];
        %          optimizableVariable('ConnectedLayer1',[25
200],'Type','integer')
        %          optimizableVariable('dropout2',[0.2 0.8])
        %          optimizableVariable('ConnectedLayer2',[25
200],'Type','integer')
        %          optimizableVariable('miniBatchSize',[1 2],'Type','integer')
```

Perform Bayesian Optimization

```
ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation);
```

Perform Bayesian optimization by minimizing the classification error on the validation set.

```
BayesObject = bayesopt(ObjFcn,optimVars, ...
    'MaxTime',36*60*60, ...
    'IsObjectiveDeterministic',false, ...
    'UseParallel',false);
```

Evaluate Final Network

Load the best network found in the optimization and its validation accuracy.

```
bestIdx = BayesObject.IndexOfMinimumTrace(end);
fileName = BayesObject.UserDataTrace{bestIdx};
savedStruct = load(fileName);
valError = savedStruct.valError
```

Predict the labels of the test set and calculate the test error. Treat the classification of each image in the test set as independent events with a certain probability of success, which means that the number of incorrectly classified images follows a binomial distribution. Use this to calculate the standard error (`testErrorSE`) and an approximate 95% confidence interval (`testError95CI`) of the generalization error rate. This method is often called the *Wald method*. `bayesopt` determines the best network using the validation set without exposing the network to the test set. It is then possible that the test error is higher than the validation error.

```
[YPredicted,probs] = predict(savedStruct.trainedNet, XTest);
testError = abs(mean(YPredicted - YTest))
NTest = numel(YTest);
testErrorSE = sqrt(testError*(1-testError)/NTest);
testError95CI = [testError - 1.96*testErrorSE, testError + 1.96*testErrorSE]
```

Objective Function for Optimization

Define the objective function for optimization.

```
function ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation)
ObjFcn = @valErrorFun;
    function [valError,cons,fileName] = valErrorFun(optVars)
```

Define the LSTM network architecture.

```
numResponses = size(YTrain{1},1);
featureDimension = size(XTrain{1},1);
% numHiddenUnits = 210;

layers = [ ...
    sequenceInputLayer(featureDimension)
    lstmLayer(optVars.numHiddenUnits1,'OutputMode','sequence')
    fullyConnectedLayer(optVars.ConnectedLayer1)%50
    dropoutLayer(optVars.dropout1)%0.2
%     fullyConnectedLayer(optVars.ConnectedLayer2)%50
%     dropoutLayer(optVars.dropout2)%0.4
    fullyConnectedLayer(numResponses)
    regressionLayer];
%     bilstmLayer(numHiddenUnits2,'OutputMode','sequence')
%     fullyConnectedLayer(optVars.ConnectedLayer2)%50
%     dropoutLayer(optVars.dropout2)%0.4
%     fullyConnectedLayer(optVars.ConnectedLayer2)
%     dropoutLayer(optVars.dropout2)
```

```
miniBatchSize = 2;
validationFrequency = floor(numel(YTrain)/miniBatchSize);
% maxEpochs = 150;

options = trainingOptions('adam', ...
    'MaxEpochs',optVars.maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'InitialLearnRate',optVars.InitialLearnRate, ...
    'GradientThreshold',1, ...
    'L2Regularization',optVars.L2Regularization,...%0.0001
    'GradientDecayFactor',0.95,...%0.95
    'Shuffle','never', ...
    'Plots','training-progress',...
    'Verbose',0,...
    'ValidationData',{XValidation,YValidation}, ...
    'ValidationFrequency',validationFrequency);

%         options = trainingOptions('sgdm', ...
%             'InitialLearnRate',optVars.InitialLearnRate, ...
%             'Momentum',optVars.Momentum, ...
%             'MaxEpochs',60, ...
%             'LearnRateSchedule','piecewise', ...
%             'LearnRateDropPeriod',40, ...
%             'LearnRateDropFactor',0.1, ...
%             'MiniBatchSize',miniBatchSize, ...
%             'L2Regularization',optVars.L2Regularization, ...
%             'Shuffle','every-epoch', ...
%             'Verbose',false, ...
%             'Plots','training-progress', ...
%             'ValidationData',{XValidation,YValidation}, ...
%             'ValidationFrequency',validationFrequency);
```

Train the network and plot the training progress during training. Close all training plots after training finishes.

```
        trainedNet = trainNetwork(XTrain,YTrain,layers,options);
        close(findall(groot,'Tag','NNET_LSTM_TRAININGPLOT_FIGURE'))
```

Evaluate the trained network on the validation set, calculate the predicted image labels, and calculate the error rate on the validation data.

```
        YPredicted = predict(trainedNet,XValidation);
        valError = abs(mean([YPredicted{:}] - [YValidation{:}]));
```

```
        fileName = num2str(valError) + ".mat";
        save(fileName,'trainedNet','valError','options')
        cons = [];

    end
end
```

Appendix IV
Bayesian Optimization Machine Learning Model-Based Prognostic Algorithm

```
%clear
load ('SDATrainC.mat');
load ('SDAValC1.mat');
numofFailData = size(TrainDataArray, 1);
numofFeatures = size(TrainDataArray{1,1}, 2);
XTrain = cell(numofFailData,1);
YTrain = cell(numofFailData,1);
for i = 1:numofFailData
    fieldData = table2array(TrainDataArray{i,1});
    X = fieldData;
    XTrain{i} = X';
    s = size(X,1);
    timeSteps = 1:1:s; %time column of featuretable
    Y = fliplr(timeSteps);
    YTrain{i} = Y;
end
numofValData = size(ValDataArray, 1);
XVal = cell(numofValData,1);
YVal = cell(numofValData,1);
for i = 1:numofValData
    fieldData = table2array(ValDataArray{i,1});
    X = fieldData;
    XVal{i} = X';
    s = size(X,1);
    timeSteps = 1:1:s; %time column of featuretable

    Y = fliplr(timeSteps);
    YVal{i} = Y;
end
```

Normalize Training Data X

Normalize the training set to have zero mean and unit variance.

```
mu = mean([XTrain{:}],2);
sigma = std([XTrain{:}],0,2);
XTrain = cellfun(@(X) (X-mu)./sigma,XTrain,'UniformOutput',false);
XVal = cellfun(@(X) (X-mu)./sigma,XVal,'UniformOutput',false);
```

Clip Responses

```
% thr = 700;
for i = 1:numel(YTrain)
%     YTrain{i}(YTrain{i} > thr) = thr;
    s = size(TrainDataArray{i,1},1);
    if (s<900)
      thr = 550;  %550
    elseif(s<1300)
      thr = 800;  %800
    else
      thr = 1400;  %1300
    end
    YTrain{i}(YTrain{i} > thr) = thr;
end
for i = 1:numel(YVal)
        if(s<900)
            thr = 550;
        elseif(s<1300)
            thr = 800;
        else
            thr = 1400;
        end
    YVal{i}(YVal{i} > thr) = thr;
end
```

Normalize Training Data Y

```
% ymu = mean([YTrain{:}],2);
% ysigma = std([YTrain{:}],0,2);
% YTrain = cellfun(@(Y) ((Y-ymu)./ysigma),YTrain,'UniformOutput',false);
% YVal = cellfun(@(Y) ((Y-ymu)./ysigma),YVal,'UniformOutput',false);
% for i = 1:numofFailData
%     XTrain{i} = XTrain{i}';
%     YTrain{i} = YTrain{i}';
% end
%
% for i = 1:numofValData
%     XVal{i} = XVal{i}';
%     YVal{i} = YVal{i}';
% % end
% XTrainArray = vertcat(XTrain{:});
% YTrainArray = vertcat(YTrain{:});
XTrainArray = horzcat(XTrain{:});
YTrainArray = horzcat(YTrain{:});
XValArray = horzcat(XVal{:});
YValArray = horzcat(YVal{:});
load('FGSVMModelXnorm.mat')
yfit = FGSVMModel.predictFcn(XValArray);
```

Plot

```
    yfit = smoothdata(yfit, 3);
    yfit = yfit';
    subplot(2,1,1)
    plot(YValArray(:,1:1674),'--')
    hold on
    plot(yfit(:,1:1674),'.-')
    hold off
    title("Test Data " + 1)
    xlabel("Time Step")
    ylabel("RUL")

     subplot(2,1,2)
    plot(YValArray(:,1674:2904),'--')
    hold on
    plot(yfit(:,1674:2904),'.-')
    hold off
    title("Test Data " + 2)
    xlabel("Time Step")
    ylabel("RUL")
legend(["Test Data" "Predicted"])
ypred = yfit;
yval = YValArray;
RMSE = sqrt(mean(abs(ypred-yval).^2))
MAPE = mean(abs(ypred-yval)./yval)
ypred1 =yfit(:,1:1674);
yval1 = YValArray(:,1:1674);
RMSE1 = sqrt(mean(abs(ypred1-yval1).^2))
MAPE1 = mean(abs(ypred1-yval1)./yval1)
ypred2 =yfit(:,1674:2904);
yval2 = YValArray(:,1674:2904);
RMSE2 = sqrt(mean(abs(ypred2-yval2).^2))
MAPE2 = mean(abs(ypred2-yval2)./yval2)
function [trainedModel, validationRMSE] = trainRegressionModel(trainingData,
responseData)

% Convert input to table
inputTable = array2table(trainingData', 'VariableNames', {'row_1', 'row_2',
'row_3', 'row_4', 'row_5', 'row_6', 'row_7'});

predictorNames = {'row_1', 'row_2', 'row_3', 'row_4', 'row_5', 'row_6',
'row_7'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false, false, false];

% Train a regression model
% This code specifies all the model options and trains the model.
```

```
regressionSVM = fitrsvm(...
    predictors, ...
    response, ...
    'KernelFunction', 'polynomial', ...
    'PolynomialOrder', 3, ...
    'KernelScale', 1, ...
    'BoxConstraint', 0.6207549522638148, ...
    'Epsilon', 0.4858736702340352, ...
    'Standardize', true);

% Create the result struct with predict function
predictorExtractionFcn = @(x) array2table(x', 'VariableNames',
predictorNames);
svmPredictFcn = @(x) predict(regressionSVM, x);
trainedModel.predictFcn = @(x) svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedModel.RegressionSVM = regressionSVM;
trainedModel.About = 'This struct is a trained model exported from Regression
Learner R2021a.';
trainedModel.HowToPredict = sprintf('To make predictions on a new predictor
row matrix, X, use: \n  yfit = c.predictFcn(X) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nX must
contain exactly 7 rows because this model was trained using 7 predictors. \nX
must contain only predictor rows in exactly the same order and format as your
training \ndata. Do not include the response row or any rows you did not
import into the app. \n \nFor more information, see <a
href="matlab:helpview(fullfile(docroot, ''stats'', ''stats.map''),
''appregression_exportmodeltoworkspace'')">How to predict using an exported
model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
% Convert input to table
inputTable = array2table(trainingData', 'VariableNames', {'row_1', 'row_2',
'row_3', 'row_4', 'row_5', 'row_6', 'row_7'});

predictorNames = {'row_1', 'row_2', 'row_3', 'row_4', 'row_5', 'row_6',
'row_7'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false, false, false];

% Set up holdout validation
cvp = cvpartition(size(response, 1), 'Holdout', 0.25);
trainingPredictors = predictors(cvp.training, :);
trainingResponse = response(cvp.training, :);
trainingIsCategoricalPredictor = isCategoricalPredictor;
```

```
% Train a regression model
% This code specifies all the model options and trains the model.
regressionSVM = fitrsvm(...
    trainingPredictors, ...
    trainingResponse, ...
    'KernelFunction', 'polynomial', ...
    'PolynomialOrder', 3, ...
    'KernelScale', 1, ...
    'BoxConstraint', 0.6207549522638148, ...
    'Epsilon', 0.4858736702340352, ...
    'Standardize', true);

% Create the result struct with predict function
svmPredictFcn = @(x) predict(regressionSVM, x);
validationPredictFcn = @(x) svmPredictFcn(x);

% Add additional fields to the result struct


% Compute validation predictions
validationPredictors = predictors(cvp.test, :);
validationResponse = response(cvp.test, :);
validationPredictions = validationPredictFcn(validationPredictors);

% Compute validation RMSE
isNotMissing = ~isnan(validationPredictions) & ~isnan(validationResponse);
validationRMSE = sqrt(nansum(( validationPredictions - validationResponse
).^2) / numel(validationResponse(isNotMissing) ));
```

Appendix V
Exponential Degradation Model-Based Prognostic Algorithm

Training Data

```
breaktime = 23000;%23750;
breakpoint = find(SfeatureTimeTable.Time <= breaktime, 1, 'last');
trainData = SfeatureTimeTable(1:breakpoint, :);
trainDataSelected = trainData(:,featureImportance{:,:}>0.015)
featureSelected = SfeatureTimeTable(:,featureImportance{:,:}>0.015)
%trainDataSelected = trainData(:, featureImportance{:,:}>0.01);
%feature_array{1,1}(:, featureImportance(:,:)>0.035);
meanTrain = mean(trainDataSelected{:,:});
sdTrain = std(trainDataSelected{:,:});
trainDataNormalized = (trainDataSelected{:,:} - meanTrain)./sdTrain;
coef = pca(trainDataNormalized);
PCA1 = (featureSelected{:,:} - meanTrain) ./ sdTrain * coef(:, 1);
PCA2 = (featureSelected{:,:} - meanTrain) ./ sdTrain * coef(:, 2);
figure
numData = size(featuretable, 1);
scatter(PCA1, PCA2,[], 1:numData,'filled');
xlabel('PCA 1')
ylabel('PCA 2')
cbar = colorbar;
ylabel(cbar, 'Time (x10 seconds)')

healthIndicator = PCA1;
healthIndicator = medfilt1(healthIndicator, 6);
healthIndicator = smoothdata(healthIndicator);
figure
hold on
plot(SfeatureTimeTable.Time/10, PCA1, '-')
plot(SfeatureTimeTable.Time/10, healthIndicator,...
    '-r','LineWidth',1.5)
legend('Before smoothing', 'After smoothing')
xlabel('Time (x10 seconds)')
title('Health Indicator')
hold off
healthIndicator = healthIndicator - healthIndicator(1);
threshold = healthIndicator(end);
```

```
mdl = exponentialDegradationModel(...
    'Theta', 5, ...
    'ThetaVariance', 1e6, ...
    'Beta', 5, ...
    'BetaVariance', 1e6, ...
    'Phi', -1, ...
    'NoiseVariance', (0.1*threshold/(threshold + 1))^2, ...
    'SlopeDetectionLevel', 1);
% Keep records at each iteration
totalTime = (length(healthIndicator) - 1);
estRULs = zeros(totalTime, 1);
trueRULs = zeros(totalTime, 1);
CIRULs = zeros(totalTime, 2);
pdfRULs = cell(totalTime, 1);
% Create figures and axes for plot updating
figure
ax1 = subplot(2, 1, 1);
ax2 = subplot(2, 1, 2);
for currentTime = 1:totalTime
    % Update model parameter posterior distribution
    update(mdl, [currentTime healthIndicator(currentTime)])
    % Predict Remaining Useful Life
    [estRUL, CIRUL, pdfRUL] = predictRUL(mdl, ...
                                         [currentTime
healthIndicator(currentTime)], ...
                                         threshold);
    trueRUL = totalTime - currentTime + 1;
    % Updating RUL distribution plot
    helperPlotTrend(ax1, currentTime, healthIndicator, mdl, threshold, 'x10
seconds');
    helperPlotRUL(ax2, trueRUL, estRUL, CIRUL, pdfRUL, 'x10 seconds')
    % Keep prediction results
    estRULs(currentTime) = estRUL;
    trueRULs(currentTime) = trueRUL;
    CIRULs(currentTime, :) = CIRUL;
    pdfRULs{currentTime} = pdfRUL;

    % Pause 0.1 seconds to make the animation visible
    pause(0.1)
end
```

Predict the RUL for the bearing.

```
estRULoverall = predictRUL(mdl,threshold)
```

Appendix VI

Upload Data to ThingSpeak Cloud Space

Write Data to ThingSpeak Channel

```
load("normTestFeatureData.mat")
s = size(normTestFeatureData,1);
for i=1:s

    data = normTestFeatureData(i,:);
    % Generate timestamps for the data
    tStamps = datetime('now');

    channelID = 1044755; % Change to your Channel ID
    writeKey = '8B4B91T47Q6MFGRT'; % Change to your Write API Key

    % Write 10 values to each field of your channel along with timestamps
    thingSpeakWrite(channelID,data,'TimeStamp',tStamps,'WriteKey',writeKey)
    pause(15)
    r = s-i
end
```

Appendix VII

Cloud Computing Algorithm for the Real-Time RUL Estimation and Email Alert Warnings

```
% ThingSpea & DropBox credentials
firstChID = 1044755; %FILL IN first channel's ID
firstReadAPIKey = '7CPPH4ENMAFE9H2R'; %FILL IN first channel's Read API key
secondChID = 1044755;  %FILL IN second channel's ID
secondWriteAPIKey = 'JCQ414YH411POYRA'; %FILL IN second channel's Write API
key
secondReadAPIKey = '02MXH1N0PYB9H77H'; %FILL IN second channel's Read API key
dropBoxAccessToken
='gezLS6x0vLQAAAAAAAAAAdX0AYPfwWSO7z0gHfleIO0UxqbF1IIamgo6siuVOINS'; %FILL IN
dropbox access token

thresholdRUL = 60; %email will be sent if the fan's TTS is less than this
value (in minutes)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Provide the ThingSpeak alerts API key.  All alerts API keys start with TAK.
alertApiKey = 'TAKA0XIG6PYEXOQXE';

% Set the address for the HTTTP call
alertUrl="https://api.thingspeak.com/alerts/send";

% webwrite uses weboptions to add required headers.  Alerts needs a
ThingSpeak-Alerts-API-Key header.
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey
]);

% Set the email subject.
alertSubject = sprintf("Lathe Spindle Remaining Useful Life");

% Read the recent data.
featureData =
thingSpeakRead(firstChID,'ReadKey',firstReadAPIKey,'NumDays',10); %past 10
Days data
testData = featureData';




% Check to make sure the data was read correctly from the channel.
if isempty(featureData)
```

```
    alertBody = ' No data read from machine subsystem. ';
else


    %[mdl, labels] = getmodel(dropBoxAccessToken); %Get the predictive model
from DropBox

    rawdata = downloadFromDropbox(dropBoxAccessToken,'PredictiveModel.mat');
    f = fopen('PredictiveModel.mat','w');
        fwrite(f,rawdata);
        fclose(f);
        T = load('PredictiveModel.mat');


    %#function network
    YPred = predict(T.netmdl,testData,'MiniBatchSize',1);
    YPredLast = YPred(end);

    % Set the outgoing message
    if (YPredLast>thresholdRUL)
        alertBody = 'Machine Tool operating Good!';
    elseif (YPredLast<=thresholdRUL)
        RUL = YPredLast;
        alertBody ="Machine Tool Need Maintenance! and RUL = " +
num2str(RUL,'%0.3f');
    end
end

 % Catch errors so the MATLAB code does not disable a TimeControl if it fails
try
    webwrite(alertUrl , "body", alertBody, "subject", alertSubject, options);
catch someException
    fprintf("Failed to send alert: %s\n", someException.message);
end

%functions

function [mdl,labels] = getmodel(dropBoxAccessToken)
    rawdata = downloadFromDropbox(dropBoxAccessToken,'PredictiveModel.mat');
    f = fopen('PredictiveModel.mat','w');
    fwrite(f,rawdata);
    fclose(f);
    thefile = matfile('PredictiveModel.mat');
    mdl = thefile.trainedModel;
    labels = thefile.labels;
end
```

```
function output = downloadFromDropbox(dropboxAccessToken,varargin)
    narginchk(1,2);

    FName = varargin{1};

    % Generate the custom header
    headerFields = {'Authorization', ['Bearer ', dropboxAccessToken]};
    headerFields{2,1} = 'Dropbox-API-Arg';
    headerFields{2,2} = sprintf('{"path": "/%s"}',FName);
    headerFields{3,1} = 'Content-Type';
    headerFields{3,2} = 'application/octet-stream';
    headerFields = string(headerFields);

    % Set the options for WEBREAD
    opt = weboptions;
    opt.MediaType = 'application/octet-stream';
    opt.CharacterEncoding = 'ISO-8859-1';
    opt.RequestMethod = 'post';
    opt.HeaderFields = headerFields;

    % Upload the file
    try
        tempOutput =
webread('https://content.dropboxapi.com/2/files/download',
opt);%https://content.dropboxapi.com/2/files/download'
    catch someException

throw(addCause(MException('downloadFromDropbox:unableToDownloadFile','Unable
to download file.'),someException));
    end

    % If user requested output, pass along WEBWRITE output
    if isequal(nargout,1)
        output = tempOutput;
    end
end

function output = uploadToDropbox(dropboxAccessToken,dataFile)

    % Check if input file exists
    if ~exist(dataFile,'file')
        throw(MException('uploadToDropbox:fileNotFound','Input file was not
found.'));
    end

    % Read file contents
    try
        fid = fopen(dataFile, 'r');
        data = char(fread(fid)');
        fclose(fid);
    catch someException
        throw(addCause(MException('uploadToDropbox:unableToReadFile','Unable
to read input file.'),someException));
    end

    % Generate the custom header
    [~,remoteFName, remoteExt] = fileparts(dataFile);
    headerFields = {'Authorization', ['Bearer ', dropboxAccessToken]};
    headerFields{2,1} = 'Dropbox-API-Arg';
    headerFields{2,2} = sprintf('{"path": "/%s%s", "mode": "overwrite",
"autorename": false, "mute": false}',remoteFName, remoteExt);
    headerFields{3,1} = 'Content-Type';
    headerFields{3,2} = 'application/octet-stream';
    headerFields = string(headerFields);

    % Set the options for WEBWRITE
    opt = weboptions;
    opt.MediaType = 'application/octet-stream';
    opt.CharacterEncoding = 'ISO-8859-1';
    opt.RequestMethod = 'post';
    opt.HeaderFields = headerFields;

    % Upload the file
    try
        tempOutput = webwrite('https://content.dropboxapi.com/2/files/upload',
data, opt);
    catch someException
        throw(addCause(MException('uploadToDropbox:unableToUploadFile','Unable
to upload file.'),someException));
    end

    % If user requested output, pass along WEBWRITE output
    if isequal(nargout,1)
        output = tempOutput;
    end
end
```

References

1. Yao, X., Zhou, J., Lin, Y., Li, Y., Yu, H., and Liu, Y., 2019, “Smart manufacturing based on cyber-physical systems and beyond,” J. Intell. Manuf., 30(8), pp. 2805–2817. https://doi.org/10.1007/s10845-017-1384-5
2. Janak, L., Stetina, J., Fiala, Z., and Hadas, Z., 2016, “Quantities and Sensors for Machine Tool Spindle Condition Monitoring”, MM Sci J pp. 1648–1653 doi: 10.17973/MMSJ.2016_12_2016204
3. Laloix, T., Iung, B., Voisin, A., and Romagne, E., 2019, “Parameter identification of health indicator aggregation for decision-making in predictive maintenance: Application to machine tool,” CIRP Ann., (68)1, pp. 483–486. https://doi.org/10.1016/j.cirp.2019.03.020
4. Liu, C., Vengayil, H., Zhong, R. Y., and Xu, X., 2018, “A systematic development method for cyber-physical machine tools,” J. Manuf. Syst., 48, pp. 13–24. https://doi.org/10.1016/j.jmsy.2018.02.001
5. Lee, G. Y. et al., 2018, “Machine health management in smart factory: A review,” J. Mech. Sci. Technol., 32(3), pp. 987–1009. https://doi.org/10.1007/s12206-018-0201-1
6. Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, “Machinery health prognostics: A systematic review from data acquisition to RUL prediction,” Mech. Syst. Signal Process., vol. 104, pp. 799–834, 2018. https://doi.org/10.1016/j.ymssp.2017.11.016
7. Ciocoiu, L., Siemieniuch, C. E., & Hubbard, E. M., 2017, “From preventative to predictive maintenance: The organisational challenge”. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 231(10), 1174–1185. https://doi.org/10.1177%2F0954409717701785
8. Rastegari, A., 2017, “Condition Based Maintenance in the Manufacturing Industry: From Strategy to Implementation”, Mälardalen University Press Dissertations.
9. Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., and Li, G.P., 2020, “Predictive maintenance in the Industry 4.0: A systematic literature review,” Comput. Ind. Eng., vol. 150, no. April 2019, p. 106889. https://doi.org/10.1016/j.cie.2020.106889
10. Feng, S. C., Bernstein, W. Z., Hedberg, T. Jr., and Feeney, A. B., 2017, "Toward Knowledge Management for Smart Manufacturing." ASME. J. Comput. Inf. Sci. Eng., 17(3), pp. 031016-1–9. https://doi.org/10.1115/1.4037178

11. Pech, M., Vrchota, J., and Bednář, J., 2021, "Predictive maintenance and intelligent sensors in smart factory: Review," Sensors, vol. 21, no. 4, pp. 1–39. https://doi.org/10.3390/s21041470
12. Gopalakrishnan, M., and Skoogh, A., 2018, "Machine criticality based maintenance prioritization Identifying productivity improvement potential," Int. J. Product. Perform. Manag., 67(4), pp. 654–672. https://doi.org/10.1108/IJPPM-07-2017-0168
13. Wu, Z., Liu, W. & Nie, W. Literature review and prospect of the development and application of FMEA in manufacturing industry. Int J Adv Manuf Technol 112, 1409–1436 (2021). https://doi.org/10.1007/s00170-020-06425-0
14. Gopalakrishnan, M., Subramaniyan, M., and Skoogh, A., 2020, "Data-driven machine criticality assessment–maintenance decision support for increased productivity," Prod. Plan. Control, vol. 0, no. 0, pp. 1–19. https://doi.org/10.1080/09537287.2020.1817601
15. Malla, C. & Panigrahi, I., 2019, "Review of Condition Monitoring of Rolling Element Bearing Using Vibration Analysis and Other Techniques". Journal of Vibration Engineering & Technologies, 7, 407–414. https://doi.org/10.1007/s42417-019-00119-y
16. Caesarendra, W. & Tjahjowidodo, T., 2017, "A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing". Machines, 5, 21. https://doi.org/10.3390/machines5040021
17. Si, X., Wang, W., Hu, C., and Zhou, D., 2011, "Remaining useful life estimation – A review on the statistical data driven approaches," Eur. J. Oper. Res., 213(1), pp. 1–14. http://dx.doi.org/10.1016/j.ejor.2010.11.018
18. Liao, L., and Köttig, F., 2016, "A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction," Appl. Soft Comput. J., 44, pp. 191–199. http://dx.doi.org/10.1016/j.asoc.2016.03.013
19. Xu, G., et al., 2019, "Data-driven fault diagnostics and prognostics for predictive maintenance: A brief overview," IEEE Int. Conf. Autom. Sci. Eng., 1, pp. 103–108. https://doi.org/10.1109/COASE.2019.8843068
20. Zhang, W., Yang, D., and Wang, H., 2019, "Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey," IEEE Syst. J., 13(3), pp. 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565

21. Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D., 2018, “Deep learning for smart manufacturing: Methods and applications,” J. Manuf. Syst., 48, pp. 144–156. https://doi.org/10.1016/j.jmsy.2018.01.003
22. Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., and Alcalá, S. G. S., 2019, “A systematic literature review of machine learning methods applied to predictive maintenance,” Comput. Ind. Eng., 137, p. 106024. https://doi.org/10.1016/j.cie.2019.106024
23. Yang, L., and Shami, A., 2020, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” Neurocomputing, vol. 415, pp. 295–316, https://doi.org/10.1016/j.neucom.2020.07.061
24. Yu T., and Zhu, H., 2020, “Hyper-parameter optimization: A review of algorithms and applications,” arXiv, pp. 1–56. arXiv:2003.05689
25. Liu, C., Vengayil, H., Lu, Y., and Xu, X., 2019, “A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect,” J. Manuf. Syst., 51, pp. 61–74. https://doi.org/10.1016/j.jmsy.2019.04.006
26. Bumblauskas, D., Gemmill, D., Igou, A., and Anzengruber, J., 2017, “Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics,” Expert Syst. Appl., vol. 90, pp. 303–317. http://dx.doi.org/10.1016/j.eswa.2017.08.025
27. Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W., and Ducoffe, M., 2020, “Engineering Applications of Artificial Intelligence Potential, challenges and future directions for deep learning in prognostics and health management applications,” Eng. Appl. Artif. Intell., vol. 92, no. January, p. 103678. https://doi.org/10.1016/j.engappai.2020.103678
28. Adams, S., Malinowski, M., Heddy, G., Choo, B., and Beling, P. A., 2017, “The WEAR methodology for prognostics and health management implementation in manufacturing.” Journal of Manufacturing Systems, The Society of Manufacturing Engineers, 45, 82–96. https://doi.org/10.1016/j.jmsy.2017.07.002
29. Luo, B., Wang, H., Liu, H., Li, B., and Peng, F., 2018, “Early Fault Detection of Machine Tools Based on Deep Learning and Dynamic Identification.” IEEE Transactions on Industrial Electronics, 66(1), 509–518. https://doi.org/10.1109/TIE.2018.2807414
30. Baur, M., Albertelli, P., and Monno, M., 2020, “A review of prognostics and health management of machine tools.” International Journal of Advanced Manufacturing Technology, The International Journal of Advanced Manufacturing Technology, 2843–2863. https://doi.org/10.1007/s00170-020-05202-3

31. Goyal, D., and Pabla, B. S., 2015, “Condition based maintenance of machine tools-A review.” CIRP Journal of Manufacturing Science and Technology, CIRP, 10, 24–35. http://dx.doi.org/10.1016/j.cirpj.2015.05.004
32. Shin, I., Lee, J., Lee, J. Y., Jung, K., Kwon, D., Youn, B. D., Jang, H. S., and Choi, J. H., 2018, “A Framework for Prognostics and Health Management Applications toward Smart Manufacturing Systems.” International Journal of Precision Engineering and Manufacturing - Green Technology, 5(4), 535–554. https://doi.org/10.1007/s40684-018-0055-0
33. Li, Z., Wang, Y., and Wang, K. S., 2017, “Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario.” Advances in Manufacturing, Shanghai University, 5(4), 377–387. https://doi.org/10.1007/s40436-017-0203-8
34. Saravanan, S., Yadava, G. S., and Rao, P. V., 2006, “Condition monitoring studies on spindle bearing of a lathe.” International Journal of Advanced Manufacturing Technology, 28, 993–1005. https://doi.org/10.1007/s00170-004-2449-0
35. Teti, R., Jemielniak, K., O’Donnell, G., and Dornfeld, D. (2010). “Advanced monitoring of machining operations.” CIRP Annals - Manufacturing Technology, 59(2), 717–739. https://doi.org/10.1016/j.cirp.2010.05.010
36. Serin, G., Sener, B., Ozbayoglu, A.M., and Unver, H.O., 2020, “Review of tool condition monitoring in machining and opportunities for deep learning.” International Journal of Advanced Manufacturing Technology, The International Journal of Advanced Manufacturing Technology, 109(3–4), 953–974. https://doi.org/10.1007/s00170-020-05449-w
37. Harris, C. G., Williams, J. H., and Davies, A., 1989, “Condition monitoring of machine tools.” International Journal of Production Research, 27(9), 1445–1464. http://dx.doi.org/10.1080/00207548908942633
38. Martint, K.F., 1994, “A Review by Discussion of Condition Monitoring and Fault Diagnosis in Machine Tools.” Int. J. Mach. Tools Manufact, 34(4), 527–551. https://doi.org/10.1016/0890-6955(94)90083-3
39. Drake, P.R., Jennings, A.D., Grosvenor, R.I., and Whittleton, D., 1995, “A data acquisition system for machine tool condition monitoring.” Quality and Reliability Engineering International, 11(1), 15–26. DOI: 10.1002/qre.4680110104

40. Saravanan, S., Yadava, G.S., and Rao. P.V., 2003, “Machine tool failure data analysis for condition monitoring application.” Proceedings of the 11th National Conference on Machines and Mechanism (NaCoMM-2003), IIT Delhi

41. Liang, S. Y., Hecker, R. L., and Landers, R. G., 2004, “Machining Process Monitoring and Control: The State-of-the-Art.” Journal of Manufacturing Science and Engineering, 126(2), 297. https://doi.org/10.1115/1.1707035

42. Kim, D.H., Kim, S.H., and Song, J.Y., 2005, “Diagnosing the cause of operational faults in machine tools with an open architecture CNC.” Journal of Mechanical Science and Technology, 19(8), 1597–1610. https://doi.org/10.1007/BF03023937

43. Atluru, S., Huang, S.H., and Snyder, J.P., 2012, “A smart machine supervisory system framework.” International Journal of Advanced Manufacturing Technology, 58(5–8), 563–572. https://doi.org/10.1007/s00170-011-3405-4

44. Liao, L., and Lee, J. (2010). “Design of a reconfigurable prognostics platform for machine tools.” Expert Systems with Applications, Elsevier Ltd, 37(1), 240–252. http://dx.doi.org/10.1016/j.eswa.2009.05.004

45. Lee, J., Yang, S., Lapira, E., Kao, H., and Yen, N., 2013, “Methodology and Framework of a Cloud-Based Prognostics and Health Management System for Manufacturing Industry.” Chemical Engineering Transactions, 33, 205–210. https://doi.org/10.3303/CET1333035

46. Liao, L., 2014, “Discovering prognostic features using genetic programming in remaining useful life prediction.” IEEE Transactions on Industrial Electronics, 61(5), 2464–2472. https://doi.org/10.1109/TIE.2013.2270212

47. Kumar, N.B.K, & Kumar, D.C.D., Rachappa, C.T., 2013, “Condition Monitoring of CNC Machine Tool Accuracy with Renishaw Equipment.” 2(12), 7861–7866.

48. Wang, K. S., Sharma, V. S., and Yu, Q., 2014, “A Review of Data Mining Technologies for Condition Based Monitoring for Machine Tools.” Advanced Materials Research, 1039, 155–162. https://doi.org/10.4028/www.scientific.net/AMR.1039.155

49. Li, Y., Cao, H., and Chen, X., 2015 “Modelling and vibration analysis of machine tool spindle system with bearing defects.” International Journal of Mechatronics and Manufacturing Systems, 8(1–2) pp. 33-48. http://dx.doi.org/10.1504/IJMMS.2015.071686

50. Hashemi, S. M., Sambandamurthy, H., and Ghaemi, H., 2015, “Vibration Analysis of Machine Tool Spindle Systems: A Calibrated Finite Element Model.” V04BT04A077 ASME, Montreal, Quebec, Canada. https://doi.org/10.1115/IMECE2014-39547

51. Armendia, M., Peysson, F., & Euhus, D., 2016, “Twin-Control: A New Concept Towards Machine Tool Health Management.” PHM Society European Conference, 3(1). https://doi.org/10.36001/phme.2016.v3i1.1584
52. Xi, S., Cao, H., and Chen, X. (2019). “Dynamic modeling of spindle bearing system and vibration response investigation.” Mechanical Systems and Signal Processing, Elsevier Ltd, 114, 486–511. https://doi.org/10.1016/j.ymssp.2018.05.028
53. Turygin, A., Mosyurchak, A., Zhalo, M., & Hammer, M., 2016, “Maintenance and technical diagnostics of machine tools”. pp. 1491-1496.
54. Tobon-Mejia, D. A., Medjaher, K., and Zerhouni, N., 2012, “CNC machine tools wear diagnostic and prognostic by using dynamic Bayesian networks.” Mechanical Systems and Signal Processing, Elsevier, 28, 167–182. http://dx.doi.org/10.1016/j.ymssp.2011.10.018
55. Luo, W., Hu, T., Ye, Y., Zhang, C., and Wei, Y., 2020, “A hybrid predictive maintenance approach for CNC machine tool driven by Digital Twin.” Robotics and Computer-Integrated Manufacturing, Elsevier Ltd, 65(March), 101974. https://doi.org/10.1016/j.rcim.2020.101974
56. Chen, J., Hu, P., Zhou, H., Yang, J., Xie, J., Jiang, Y., Gao, Z., and Zhang, C., 2019, “Toward Intelligent Machine Tool.” Engineering, Chinese Academy of Engineering, 5(4), 679–690. https://doi.org/10.1016/j.eng.2019.07.018
57. Ghosh, A. K., Ullah, A. S., Teti, R., and Kubo, A., 2021, “Developing sensor signal-based digital twins for intelligent machine tools.” Journal of Industrial Information Integration, Elsevier Inc., 24, 100242. https://doi.org/10.1016/j.jii.2021.100242
58. Vogl, G. W., and Donmez, M. A., 2015, “A defect-driven diagnostic method for machine tool spindles.” CIRP Annals - Manufacturing Technology, CIRP, 64, 377–380. http://dx.doi.org/10.1016/j.cirp.2015.04.103
59. Cao, H., Zhang, X., and Chen, X., 2017, “The concept and progress of intelligent spindles: A review.” International Journal of Machine Tools & Manufacture, 112(October 2016), 21–52. http://dx.doi.org/10.1016/j.ijmachtools.2016.10.005
60. Rastegari A., 2019, “Vibration Analysis of Machine Tool Spindle Units”. In: Mathew J., Lim C., Ma L., Sands D., Cholette M., Borghesani P. (eds) Asset Intelligence through Integration and Interoperability and Contemporary Vibration Engineering Technologies. Lecture Notes in Mechanical Engineering. Springer, Cham, pp. 511-522. https://doi.org/10.1007/978-3-319-95711-1_51

61. Goyal, D., and Pabla, B. S., 2016, "Development of non-contact structural health monitoring system for machine tools." Journal of Applied Research and Technology, 14(4), 245–258. http://dx.doi.org/10.1016/j.jart.2016.06.003
62. Lee, W. J., Wu, H., Yun, H., Kim, H., Jun, M. B. G., and Sutherland, J. W., 2019, "Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data." Procedia CIRP, Elsevier B.V., 80, 506–511. https://doi.org/10.1016/j.procir.2018.12.019
63. Vogl, G. W., Donmez, M. A., and Archenti, A., 2016, "Diagnostics for geometric performance of machine tool linear axes (IMU)." CIRP Annals - Manufacturing Technology, CIRP, 65(1), 377–380. http://dx.doi.org/10.1016/j.cirp.2016.04.117
64. Schmidt, B., and Wang, L., 2018, "Predictive Maintenance of Machine Tool Linear Axes: A Case from Manufacturing Industry." Procedia Manufacturing, Elsevier B.V., 17, 118–125. https://doi.org/10.1016/j.promfg.2018.10.022
65. Senthilkumar, M., Vikram, M., and Pradeep, B., 2015, "Vibration monitoring for defect diagnosis on a machine tool: A Comprehensive case study." International Journal of Acoustics and Vibrations, 20(1), 4–9.
66. Chen, S. L., Cheng, Y. T., and Su, C. F., 2015, "Analysis on machine tool systems using spindle vibration monitoring for automatic tool changer." Advances in Mechanical Engineering, 7(12), 1–12. https://doi.org/10.1177/1687814015620331
67. Chen, S. L., Su, C. F., and Cheng, Y. T., 2016, "A novel framework for diagnosing automatic tool changer and tool life based on cloud computing." Advances in Mechanical Engineering, 8(3), 1–12. https://doi.org/10.1177/1687814016637319
68. Traini, E., Bruno, G., D'Antonio, G., and Lombardi, F., 2019, "Machine learning framework for predictive maintenance in milling." IFAC-PapersOnLine, Elsevier Ltd, 52(13), 177–182. https://doi.org/10.1016/j.ifacol.2019.11.172
69. Qiao, Q., Wang, J., Ye, L., and Gao, R. X., 2019, "Digital twin for machining tool condition prediction." Procedia CIRP, Elsevier B.V., 81, 1388–1393. https://doi.org/10.1016/j.procir.2019.04.049
70. Jimenez-Cortadi, A., Irigoien, I., Boto, F., Sierra, B., and Rodriguez, G., 2020, "Predictive maintenance on the machining process and machine tool." Applied Sciences (Switzerland), 10(1). https://doi.org/10.3390/app10010224
71. Guo, J., Yang, Z., Chen, C., Luo, W., and Hu, W., 2021, "Real-Time Prediction of Remaining Useful Life and Preventive Maintenance Strategy Based on Digital Twin."

Journal of Computing and Information Science in Engineering, 21(3). https://doi.org/10.1115/1.4049153

72. The European Standard EN 13306:2010 Maintenance terminology.
73. Aheleroff, S., Xu, X., Lu, Y., Aristizabal, M., Pablo Velásquez, J., Joa, B., and Valencia, Y., 2020, “IoT-enabled smart appliances under industry 4.0: A case study.” Advanced Engineering Informatics, Elsevier, 43(December 2019), 101043. https://doi.org/10.1016/j.aei.2020.101043
74. Selcuk, S., 2017, “Predictive maintenance, its implementation and latest trends.” Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231(9), 1670–1679. https://doi.org/10.1177/0954405415601640
75. Pandey, P., Mukhopadhyay, A. K., and Chattopadhyaya, S., 2018, “Reliability analysis and failure rate evaluation for critical subsystems of the dragline.” Journal of the Brazilian Society of Mechanical Sciences and Engineering, Springer Berlin Heidelberg, 40(2), 1–11. https://doi.org/10.1007/s40430-018-1016-9
76. Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D., 2014, “Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications.” Mechanical Systems and Signal Processing, Elsevier, 42, 314–334. http://dx.doi.org/10.1016/j.ymssp.2013.06.004
77. Bagul, YG, Zeid, I, & Kamarthi, SV., 2008, "Overview of Remaining Useful Life Methodologies." Proceedings of the ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3: 28th Computers and Information in Engineering Conference, Parts A and B. Brooklyn, New York, USA. August 3–6, pp. 1391-1400. ASME. https://doi.org/10.1115/DETC2008-49938
78. Compare, M., Baraldi, P., and Zio, E., 2019, “Challenges to IoT-enabled Predictive Maintenance for Industry 4.0.” IEEE Internet of Things Journal, IEEE, vol. 7, no. 5, pp. 4585-4597. https://doi.org/10.1109/JIOT.2019.2957029
79. Lee, J., Bagheri, B., and Kao, H. A., 2015, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems.” Manufacturing Letters, Society of Manufacturing Engineers (SME), 3(2015), 18–23. http://dx.doi.org/10.1016/j.mfglet.2014.12.001
80. Ruiz-Sarmiento, J. R., Monroy, J., Moreno, F. A., Galindo, C., Bonelo, J. M., and Gonzalez-Jimenez, J., 2020, “A predictive model for the maintenance of industrial machinery in the

context of industry 4.0.” Engineering Applications of Artificial Intelligence, Elsevier Ltd, 87(January 2019), 103289. https://doi.org/10.1016/j.engappai.2019.103289

81. Vogl, G. W., Weiss, B. a, and Donmez, M. A., 2014, “Standards for Prognostics and Health Management (PHM) Techniques within Manufacturing Operations.” Annual Conference of the Prognostics and Health Management Society, 2013, 1–13. https://doi.org/10.36001/phmconf.2014.v6i1.2503
82. Waterbury, A. C., and Wright, P. K., 2013, “Vibration energy harvesting to power condition monitoring sensors for industrial and manufacturing equipment.” Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 227(6), 1187–1202. https://doi.org/10.1177/0954406212457895
83. Singh, J., Singh, S., and Singh, A., 2019, “Distribution transformer failure modes, effects and criticality analysis (FMECA).” Engineering Failure Analysis, Elsevier, 99, 180–191. https://doi.org/10.1016/j.engfailanal.2019.02.014
84. Medjaher, K., Tobon-Mejia, D. a, and Zerhouni, N., 2012, “Remaining Useful Life Estimation of Critical Components with Application to Bearings.” IEEE Transactions on Reliability, 61(2), 292–302. https://doi.org/10.1109/TR.2012.2194175
85. Elbadawi, I. A. Q., Yusmawiza, W. A., Ali, N. Ben, and Ahmad, A., 2018, “Application of Failure Mode Effect and Criticality Analysis (FMECA) to a Computer Integrated Manufacturing (CIM) Conveyor Belt.” Engineering, Technology & Applied Science Research, 8(3), 3023–3027. https://doi.org/10.48084/etasr.2043
86. Crespo Márquez, A., Moreu De Leõn, P., Sola Rosique, A., and Gõmez Fernández, J. F., 2016, “Criticality Analysis for Maintenance Purposes: A Study for Complex In-service Engineering Assets.” Quality and Reliability Engineering International, 32(2), 519–533. https://doi.org/10.1002/qre.1769
87. Lo, H. W., Liou, J. J. H., Huang, C. N., and Chuang, Y. C., 2019, “A novel failure mode and effect analysis model for machine tool risk analysis.” Reliability Engineering and System Safety, Elsevier Ltd, 183(November 2018), 173–183. https://doi.org/10.1016/j.ress.2018.11.018
88. Gupta, G., and Mishra, R. P., 2017, “A Failure Mode Effect and Criticality Analysis of Conventional Milling Machine Using Fuzzy Logic: Case Study of RCM.” Quality and Reliability Engineering International, 33(2), 347–356. https://doi.org/10.1002/qre.2011

89. Patil, R. B., and Kothavale, B. S., 2018, “Failure modes and effects analysis (FMEA) of computerized numerical control (CNC) turning center.” International Review of Mechanical Engineering, 12(1), 78–87. https://doi.org/10.15866/ireme.v12i1.14156
90. Wang, H., Zhang, Y. M., Yang, Z., and Wang, W., 2018, “Investigation on the multifactor reliability allocation method for CNC lathes based on modified criticality and objective information.” Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 232(9), 1647–1656. https://doi.org/10.1177/0954406217706094
91. Du, Y., Liao, L., and Wang, L., 2017, “Failure Mode, Effects and Criticality Analysis of Remanufactured Machine Tools in Service.” International Journal of Precision Engineering And Manufacturing, 18(3), 425–434. https://doi.org/10.1007/s12541-017-0051-2
92. Zhou, Y., Yunpeng, Z. H. U., Hongrui, R. E. N., and Yimin, Z., 2015, “Comprehensive Reliability Allocation Method for CNC Lathes Based on Cubic Transformed Functions of Failure Mode and Effects Analysis.” Chinese Journal of Mechanical Engineering, 28, pp. 315-324. https://doi.org/10.3901/CJME.2015.0105.004
93. Kim, B., Lee, S., Kim, J., and Song, J., 2006, “Reliability Assessment Approach Using Failure Mode Analysis in Machining Center.” Key Engineering Materials, 321, 1535–1538. https://doi.org/10.4028/www.scientific.net/KEM.321-323.1535
94. Spreafico, C., Russo, D., and Rizzi, C., 2017, “A state-of-the-art review of FMEA / FMECA including patents.” Computer Science Review, Elsevier Inc., 25, 19–28. http://dx.doi.org/10.1016/j.cosrev.2017.05.002
95. Wu, Z., Ming, X. G., Song, W., Zhu, B., and Xu, Z., 2012, “Nuclear product design knowledge system based on FMEA method in new product development.” Arabian Journal for Science and Engineering, 39(3), 2191–2203. https://doi.org/10.1007/s13369-013-0726-7
96. Borgovini, R., Pemberton, S., Rossi, M., 1993, “Failure Mode. Effects and Criticality Analysis”, Reliability Analysis Center. Rome
97. United States Department of Defense MIL-STD-1629A – Military Standard Procedures for Performing a Failure Mode, Effects and Criticality Analysis. System Reliability Center (1980)
98. Renjith, V. R., Jose kalathil, M., Kumar, P. H., and Madhavan, D., 2018, “Fuzzy FMECA (failure mode effect and criticality analysis) of LNG storage facility.” Journal of Loss

Prevention in the Process Industries, Elsevier, 56(November 2017), 537–547. https://doi.org/10.1016/j.jlp.2018.01.002

99. Liu, H., Liu, L., and Liu, N., 2013, “Risk evaluation approaches in failure mode and effects analysis: A literature review.” Expert Systems with Applications, Elsevier Ltd, 40(2), 828–838. http://dx.doi.org/10.1016/j.eswa.2012.08.010
100. Stamatis, D.H., 2015, “The ASQ Pocket Guide to Failure Mode and Effect Analysis (FMEA)”. American Society for Quality (ASQ). Milwaukee.
101. Bowles, J. B., and Peláez, C. E., 1995, “Fuzzy logic prioritization of failures in a system failure mode, effects and criticality analysis.” Reliability Engineering and System Safety, 50(2), 203–213. https://doi.org/10.1016/0951-8320(95)00068-D
102. Kumru, M., and Yıldız, P., 2013, “Fuzzy FMEA application to improve purchasing process in a public hospital.” Applied Soft Computing Journal, Elsevier B.V., 13(1), 721–733. http://dx.doi.org/10.1016/j.asoc.2012.08.007
103. Chang, K. H., and Cheng, C. H., 2010, “A risk assessment methodology using intuitionistic fuzzy set in FMEA.” International Journal of Systems Science, 41(12), 1457–1471. https://doi.org/10.1080/00207720903353633
104. Dağsuyu, C., Göçmen, E., Narlı, M., and Kokangül, A., 2016, “Classical and fuzzy FMEA risk analysis in a sterilization unit.” Computers and Industrial Engineering, 101, 286–294. http://dx.doi.org/10.1016/j.cie.2016.09.015
105. Wang, W., Liu, X., Qin, Y., and Fu, Y., 2018, “A risk evaluation and prioritization method for FMEA with prospect theory and Choquet integral.” Safety Science, Elsevier, 110(June), 152–163. https://doi.org/10.1016/j.ssci.2018.08.009
106. Chanamool, N., and Naenna, T., 2016, “Fuzzy FMEA application to improve decision-making process in an emergency department.” Applied Soft Computing Journal, Elsevier B.V., 43, 441–453. http://dx.doi.org/10.1016/j.asoc.2016.01.007
107. Wang, L., Hu, Y. P., Liu, H. C., and Shi, H., 2019, “A linguistic risk prioritization approach for failure mode and effects analysis: A case study of medical product development.” Quality and Reliability Engineering International, (January), 1–18. https://doi.org/10.1002/qre.2472
108. Cernuda C., 2019, “On the Relevance of Preprocessing in Predictive Maintenance for Dynamic Systems”. In: Lughofer E., Sayed-Mouchaweh M. (eds) Predictive Maintenance in Dynamic Systems. Springer, Cham. https://doi.org/10.1007/978-3-030-05645-2_3

109. Xu, L. Da, and Duan, L., 2019, “Big data for cyber physical systems in industry 4.0: a survey.” Enterprise Information Systems, 13(2), 148–169. https://doi.org/10.1080/17517575.2018.1442934

110. Li, N., Gebraeel, N., Lei, Y., Bian, L., and Si, X., 2019, “Remaining useful life prediction of machinery under time-varying operating conditions based on a two-factor state-space model.” Reliability Engineering and System Safety, 186(February), 88–100. https://doi.org/10.1016/j.ress.2019.02.017

111. Krishnakumar, P., Rameshkumar, K., and Ramachandran, K. I., 2018, “Feature level fusion of vibration and acoustic emission signals in tool condition monitoring using machine learning classifiers.” International Journal of Prognostics and Health Management, 9(1), 1–15. https://doi.org/10.36001/ijphm.2018.v9i1.2694

112. Mohanty, A. R., 2015, “Machinery condition monitoring principles and practices”. Taylor & Francis Group, LLC.

113. Ritou, M., Rabréau, C., Le, S., Furet, B., and Dumur, D., 2018, “Influence of spindle condition on the dynamic behavior.” CIRP Annals - Manufacturing Technology, 67, 419–422. https://doi.org/10.1016/j.cirp.2018.03.007

114. Tian, Y., Liu, Z., and Dong, X., 2019, “Bearing deformation of heavy-duty machine tool-foundation systems and deformation detection methods.” Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233(9), 3232–3245. https://doi.org/10.1177/0954406218813396

115. Bhuiyan, M. S. H., Choudhury, I. A., and Dahari, M., 2014, “Monitoring the tool wear, surface roughness and chip formation occurrences using multiple sensors in turning.” Journal of Manufacturing Systems, The Society of Manufacturing Engineers, 33(4), 476–487. http://dx.doi.org/10.1016/j.jmsy.2014.04.005

116. Li, W., Zhu, Z., Jiang, F., Zhou, G., and Chen, G., 2015, “Fault diagnosis of rotating machinery with a novel statistical feature extraction and evaluation method.” Mechanical Systems and Signal Processing, Elsevier, 50–51, 414–426. http://dx.doi.org/10.1016/j.ymssp.2014.05.034

117. Cerrada, M., Sánchez, R. V., Li, C., Pacheco, F., Cabrera, D., Valente de Oliveira, J., and Vásquez, R. E., 2018, “A review on data-driven fault severity assessment in rolling bearings.” Mechanical Systems and Signal Processing, Elsevier Ltd, 99, 169–196. http://dx.doi.org/10.1016/j.ymssp.2017.06.012

118. Sankararaman, S., 2015, “Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction.” Mechanical Systems and Signal Processing, Elsevier, 52–53, 228–247. http://dx.doi.org/10.1016/j.ymssp.2014.05.029
119. Zhang, Y., Duan, L., and Duan, M., 2019, “A new feature extraction approach using improved symbolic aggregate approximation for machinery intelligent diagnosis.” Measurement: Journal of the International Measurement Confederation, Elsevier Ltd, 133, 468–478. https://doi.org/10.1016/j.measurement.2018.10.045
120. Feng, H., Chen, R., and Wang, Y., 2018, “Feature extraction for fault diagnosis based on wavelet packet decomposition: An application on linear rolling guide.” 10(8), 1–12. https://doi.org/10.1177/1687814018796367
121. Lauro, C. H., Brandão, L. C., Baldo, D., Reis, R. A., and Davim, J. P., 2014, “Monitoring and processing signal applied in machining processes - A review.” Measurement: Journal of the International Measurement Confederation, Elsevier Ltd, 58, 73–86. http://dx.doi.org/10.1016/j.measurement.2014.08.035
122. Malan N. S., and Sharma, S., 2019, “Feature selection using regularized neighbourhood component analysis to enhance the classification performance of motor imagery signals,” Comput. Biol. Med., 107, pp. 118–126. https://doi.org/10.1016/j.compbiomed.2019.02.009
123. Yang, W., Wang, K., and Zuo, W., 2012, “Neighborhood component feature selection for high-dimensional data,” J. Comput., 7(1), pp. 162–168. https://doi.org/10.4304/jcp.7.1.161-168
124. Ahmadzadeh, F., and Lundberg, J., 2014, “Remaining useful life estimation: review.” International Journal of System Assurance Engineering and Management, 5(4), 461–474. https://doi.org/10.1007/s13198-013-0195-0
125. He, M., and He, D., 2019, “A new hybrid deep signal processing approach for bearing fault diagnosis using vibration signals.” Neurocomputing, Elsevier B.V., 396, 542–555. https://doi.org/10.1016/j.neucom.2018.12.088
126. Hamadache, M., Ha, J., Jungho, J., and Byeng, P., 2019, “A comprehensive review of artificial intelligence - based approaches for rolling element bearing PHM: shallow and deep learning.” JMST Advances, Korean Society of Mechanical Engineers, 1(1), 125–151. https://doi.org/10.1007/s42791-019-0016-y

127. Liu, R., Yang, B., Zio, E., and Chen, X., 2018, “Artificial intelligence for fault diagnosis of rotating machinery: A review.” Mechanical Systems and Signal Processing, Elsevier Ltd, 108, 33–47. https://doi.org/10.1016/j.ymssp.2018.02.016
128. Wu, J., Hu, K., Cheng, Y., Zhu, H., Shao, X., and Wang, Y., 2020, “Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network,” ISA Trans., 97, pp. 241–250. https://doi.org/10.1016/j.isatra.2019.07.004
129. Li, N., Lei, Y., Lin, J., and Ding, S. X., 2015, “An Improved Exponential Model for Predicting Remaining Useful Life of Rolling Element Bearings.” IEEE Transactions on Industrial Electronics, IEEE, 62(12), 7762–7773. https://doi.org/10.1109/TIE.2015.2455055
130. Tseng, S., and Lee, I., 2015, “Optimum Allocation Rule for Accelerated Degradation Tests with a Class of Exponential-Dispersion Degradation Models Optimum Allocation Rule for Accelerated Degradation Tests.” 58:2, 244-254. https://doi.org/10.1080/00401706.2015.1033109
131. Rao, B. K. N., 2021, “The Role of Artificial Intelligence (AI) in Condition Monitoring and Diagnostic Engineering Management (COMADEM): A Literature Survey. American Journal of Artificial Intelligence, 5(1), pp. 17-37. DOI: 10.11648/j.ajai.20210501.12
132. Gebraeel, N., 2006, “Sensory-Updated Residual Life Distributions for Components with Exponential Degradation Patterns”. IEEE Transactions on Automation Science and Engineering, 3(4), 382–393. https://doi.org/10.1109/TASE.2006.876609
133. Wen, J. and Gao, H., 2018, “Remaining useful life prediction of the ball screw system based on weighted Mahalanobis distance and an exponential model”. Journal of Vibroengineering, 20(4):1691-1707. https://doi.org/10.21595/jve.2018.19099
134. Zhang, L., Mu, Z., & Sun, C., 2018, “Remaining Useful Life Prediction for Lithium- ion Batteries Based on Exponential Model and Particle Filter”. IEEE Access, 6, 17729-17740. https://doi.org/10.1109/ACCESS.2018.2816684
135. Gebraeel, N. Z., Lawley, M. A., Li, R., & Jennifer, K. R., 2005, “Residual-life distributions from component degradation signals: A Bayesian approach”. IIE Transactions, 37(6):543-557. https://doi.org/10.1080/07408170590929018
136. Huang, H.Z., Wang, H.K., Li, Y.F., Zhang, L., and Liu, Z., 2015, “Support vector machine based estimation of remaining useful life: current research status and future trends,” J. Mech. Sci. Technol., vol. 29, pp. 151–163, https://doi.org/10.1007/s12206-014-1222-z

137. Xue, Z., Zhang, Y., Cheng, C., and Ma, G., 2020, “Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression,” Neurocomputing, vol. 376, pp. 95–102, https://doi.org/10.1016/j.neucom.2019.09.074

138. Yan, M., Wang, X., Wang, B., Chang, M., and Muhammad, I., “Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model,” ISA Trans., vol. 98, pp. 471–482, 2020. https://doi.org/10.1016/j.isatra.2019.08.058

139. Vapnik, V.N., 1999, “An overview of statistical learning theory”. IEEE Trans.\ Neural Networks 10:988–999. doi: 10.1109/72.788640

140. Dong S., and Luo T., 2013, “Bearing degradation process prediction based on the PCA and optimized LS-SVM model”. Measurement 46:3143–3152. https://doi.org/10.1016/j.measurement.2013.06.038

141. Omoregbee, H.O., and Heyns, P.S., 2019, “Fault Classification of Low-Speed Bearings Based on Support Vector Machine for Regression and Genetic Algorithms Using Acoustic Emission”. J Vib Eng Technol 7:455–464. https://doi.org/10.1007/s42417-019-00143-y

142. Kimotho, J.K., Sondermann-Woelke, C., Meyer, T., and Sextro, W., 2013, “Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution–particle swarm optimization”. Chem Eng Trans 33:619–624. DOI: 10.3303/CET1333104

143. Sloukia. F., Aroussi. M.E., Medromi. H., and Wahbi. M., 2013, “Bearings prognostic using mixture of Gaussians hidden Markov model and support vector machine”. ACS International Conference on Computer Systems and Applications, IEEE, Ifrane. 1–4. doi: 10.1109/AICCSA.2013.6616438

144. Lobato, T.H.G., da Silva, R.R., da Costa, E.S., et al., 2020, “An Integrated Approach to Rotating Machinery Fault Diagnosis Using, EEMD, SVM, and Augmented Data”. J Vib Eng Technol 8:403–408. https://doi.org/10.1007/s42417-019-00167-4

145. Benkedjouh, T., Medjaher, K., Zerhouni, N., and Rechak, S., 2013, “Remaining useful life estimation based on nonlinear feature reduction and support vector regression”. Eng Appl Artif Intell 26:1751–1760. https://doi.org/10.1016/j.engappai.2013.02.006

146. Benkedjouh, T., Medjaher, K., Zerhouni, N., and Rechak, S., 2015, “Health assessment and life prediction of cutting tools based on support vector regression”. J Intell Manuf 26:213–223. https://doi.org/10.1007/s10845-013-0774-6

147. Liu, J., Vitelli, V., and Zio, E., Seraoui, R., 2015, “A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data”, IEEE Trans Reliab 64:1203–1213. doi: 10.1109/TR.2015.2427156

148. Liu, J., and Zio, E., 2016, “An adaptive online learning approach for support vector regression: Online-SVR-FID”. Mech Syst Signal Process 76–77:796–809. https://doi.org/10.1016/j.ymssp.2016.02.056

149. Fumeo, E., Oneto, L., and Anguita, D., 2015, “Condition based maintenance in railway transportation systems based on big data streaming analysis”. Procedia Comput Sci 53:437–446. https://doi.org/10.1016/j.procs.2015.07.321

150. Loutas, T.H., Roulias, D., and Georgoulas, G., 2013, “Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic E-support vectors regression”. IEEE Trans Reliab 62:821–832. doi: 10.1109/TR.2013.2285318

151. Khelif, R., Chebel-Morello, B., Malinowski, S., Laajili, E., Fnaiech, F., and Zerhouni, N., 2017, “Direct remaining useful life estimation based on support vector regression”. IEEE Trans Industr Electron 64:2276–2285. doi: 10.1109/TIE.2016.2623260

152. Soualhi, A., Medjaher, K., and Zerhouni, N., 2015, “Bearing health monitoring based on Hilbert-Huang transform, support vector machine, and regression”. IEEE Trans Instrum Meas 64:52–62. doi: 10.1109/TIM.2014.2330494

153. Lecun, Y., Bengio, Y., and Hinton, G., 2015, “Deep learning”. Nature 521:436–444. doi:10.1038/nature14539

154. Emmert-streib, F., Yang, Z., Feng, H., Tripathi, S., and Emmert-streib, F., 2020, “An Introductory Review of Deep Learning for Prediction Models With Big Data.” 3(February), 1–23. https://doi.org/10.3389/frai.2020.00004

155. Khan, S., and Yairi, T., 2018, “A review on the application of deep learning in system health management”. Mech Syst Signal Process 107:241–265 https://doi.org/10.1016/j.ymssp.2017.11.024

156. Hochreiter S., and Schmidhuber, J., 1997, “Long Short-Term Memory,” Neural Comput., 9(8), pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

157. Houdt, G. V., Mosquera, C., and Nápoles, G., 2020, “A review on the long short-term memory model,” Artif. Intell. Rev., 53, pp. 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

158. Gers, F., Schmidhuber, J, and Cummins F., 2000, “Learning to forget: continual prediction with LSTM”. Neural Comput 12:2451–71. https://doi.org/10.1049/cp:19991218

159. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J., 2017, “LSTM: A Search Space Odyssey”. IEEE Transactions on Neural Networks and Learning Systems, 28(10), pp. 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

160. Elsheikh, A., Yacout, S., and Ouali, M. S., 2019, “Bidirectional handshaking LSTM for remaining useful life prediction,” Neurocomputing, vol. 323, pp. 148–156. https://doi.org/10.1016/j.neucom.2018.09.076

161. Wang, Z, Qu, J, Fang, X, Li, H, Zhong, T, and Ren, H., 2020, “Prediction of early stabilization time of electrolytic capacitor based on ARIMA-Bi_LSTM hybrid model,” Neurocomputing, 403, pp. 63–79. https://doi.org/10.1016/j.neucom.2020.03.054

162. Xia, T., Song, Y., Zheng, Y., Pan, E., and Xi, L., 2020, “An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation,” Comput. Ind., vol. 115, p. 103182. https://doi.org/10.1016/j.compind.2019.103182

163. Yuan, M., Wu Y., and Lin, L., 2016, "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network," 2016 IEEE International Conference on Aircraft Utility Systems (AUS), pp. 135-140. https://doi.org/10.1109/AUS.2016.7748035

164. Zheng, S., Ristovski, K., Farahat, A., and Gupta, C., 2017, “Long Short-Term Memory Network for Remaining Useful Life estimation,” 2017 IEEE Int. Conf. Progn. Heal. Manag. ICPHM 2017, pp. 88–95. https://doi.org/10.1109/ICPHM.2017.7998311

165. Wu, Y., Yuan, M., Dong, S., Lin, L., and Liu, Y., 2018, “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,” Neurocomputing, 275, pp. 167–179. https://doi.org/10.1016/j.neucom.2017.05.063

166. ElSaid, A. E. R., El Jamiy, F., Higgins, J., Wild, B., and Desell, T., 2018, “Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration,” Appl. Soft Comput. J., 73, pp. 969–991. https://doi.org/10.1016/j.asoc.2018.09.013

167. Bruneo D., and De Vita, F., 2019, "On the Use of LSTM Networks for Predictive Maintenance in Smart Industries," 2019 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 241-248. https://doi.org/10.1109/SMARTCOMP.2019.00059

168. Kayode O., and Tosun, A. S., 2019, “LIRUL: A Lightweight LSTM based model for Remaining Useful Life Estimation at the Edge,” Proc. - Int. Comput. Softw. Appl. Conf., 2, pp. 177–182. https://doi.org/10.1109/COMPSAC.2019.10203

169. Zhao, S., Zhang, Y., Wang, S., Zhou, B., and Cheng, C., 2019, “A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method,” Meas. J. Int. Meas. Confed., 146, pp. 279–288. https://doi.org/10.1016/j.measurement.2019.06.004

170. Wang, F., Liu, X., Deng, G., Yu, X., Li, H., and Han, Q., 2019, “Remaining Life Prediction Method for Rolling Bearing Based on the Long Short-Term Memory Network,” Neural Process. Lett., 50(3), pp. 2437–2454. https://doi.org/10.1007/s11063-019-10016-w

171. Zhang, B., Zhang, S., and Li, W., 2019, “Bearing performance degradation assessment using long short-term memory recurrent network,” Comput. Ind., 106, pp. 14–29. https://doi.org/10.1016/j.compind.2018.12.016

172. He, M., Zhou, Y., Li, Y., Wu, G., and Tang, G., 2020, “Long short-term memory network with multi-resolution singular value decomposition for prediction of bearing performance degradation,” Meas. J. Int. Meas. Confed., 156, p. 107582, https://doi.org/10.1016/j.measurement.2020.107582

173. Cabrera D., et al., 2020, “Bayesian approach and time series dimensionality reduction to LSTM-based model-building for fault diagnosis of a reciprocating compressor,” Neurocomputing, 380, pp. 51–66, https://doi.org/10.1016/j.neucom.2019.11.006

174. Zhou, J. T., Zhao, X., and Gao, J., 2019, “Tool remaining useful life prediction method based on LSTM under variable working conditions,” Int. J. Adv. Manuf. Technol., 104(9–12), pp. 4715–4726, https://doi.org/10.1007/s00170-019-04349-y

175. Yan, H., Qin, Y., Xiang, S., Wang, Y., and Chen, H., 2020, “Long-term gear life prediction based on ordered neurons LSTM neural networks,” Meas. J. Int. Meas. Confed., 165, p. 108205, https://doi.org/10.1016/j.measurement.2020.108205

176. Ji, S., Han, X., Hou, Y., Song, Y., and Du, Q., 2020, “Remaining useful life prediction of airplane engine based on PCA–BLSTM,” Sensors, 20(16), pp. 1–13, https://doi.org/10.3390/s20164537

177. Shi Z., and Chehade, A., 2021, “A dual-LSTM framework combining change point detection and remaining useful life prediction,” Reliab. Eng. Syst. Saf., 205, p. 107257. https://doi.org/10.1016/j.ress.2020.107257

178. Chui, K. T., Gupta, B. B., and Vasant, P., 2021, “A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine,” Electron., 10(3), pp. 1–15. https://doi.org/10.3390/electronics10030285

179. Song, Y., Shi, G., Chen, L., Huang, X., and Xia, T., 2018, “Remaining Useful Life Prediction of Turbofan Engine Using Hybrid Model Based on Autoencoder and Bidirectional Long Short-Term Memory,” J. Shanghai Jiaotong Univ., 23, pp. 85–94. https://doi.org/10.1007/s12204-018-2027-5
180. Zhang, J., Wang, P., Yan, R., and Gao, R. X., 2018, “Long short-term memory for machine remaining life prediction,” J. Manuf. Syst., 48, pp. 78–86, https://doi.org/10.1016/j.jmsy.2018.05.011
181. Wang, J., Wen, G., Yang, S., and Liu, Y., 2019, “Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network,” Proc. - 2018 Progn. Syst. Heal. Manag. Conf. PHM-Chongqing, pp. 1037–1042. https://doi.org/10.1109/PHM-Chongqing.2018.00184
182. Essien A. E., and Giannetti, C., 2020, “A Deep Learning model for Smart Manufacturing using Convolutional LSTM Neural Network Autoencoders,” IEEE Trans. Ind. Informatics, pp. 1–10. https://doi.org/10.1109/TII.2020.2967556
183. Zhang, H., Zhang, Q., Shao, S., Niu, T., and Yang, X., 2017, “Attention-Based LSTM Network for Rotatory Machine Remaining Useful Life Prediction,” IEEE Access, 8, pp. 132188–132199. https://doi.org/10.1109/ACCESS.2020.3010066
184. Heimes, F.O., 2008, “Recurrent neural networks for remaining useful life estimation,” in Prognostics and Health Management. PHM 2008 International Conference on IEEE 1–6. doi: 10.1109/PHM.2008.4711422.
185. Peng, Y., Wang, H., Wang, J., Liu, D., and Peng, X., 2012, “A modified echo state network based remaining useful life estimation approach”. 2012 IEEE Conference on Prognostics and Health Management, Denver, CO, 1-7. doi: 10.1109/ICPHM.2012.6299524.
186. Liu, D., Xie, W., Liao, H., and Peng, Y., 2014, An integrated probabilistic approach to lithium-ion battery remaining useful life estimation. IEEE Trans Instrum Meas 64:660–670. doi: 10.1109/TIM.2014.2348613
187. Zhao, R., Wang, J., Yan, R., and Mao, K., 2016, “Machine health monitoring with LSTM networks”. IEEE International Conference on Sensing Technology 1–6. doi: 10.1109/ICSensT.2016.7796266.
188. Zhao, R., Yan, R., Wang, J., and Mao, K., 2017, “Learning to monitor machine health with convolutional bi-directional LSTM networks”. Sensors 17(2):273. doi: 10.3390/s17020273.

189. Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., and Wang, J., 2017, “Machine health monitoring using local feature-based gated recurrent unit networks”. IEEE Trans Industr Electron 65(2):1539–1548. doi: 10.1109/TIE.2017.2733438.
190. Malhotra, P., T.V.V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G., 2016, “Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder”. arXiv:1608.06154 [cs.LG]
191. Chen, Y., Peng, G., Zhu, Z., and Li, S., 2020, “A novel deep learning method based on attention mechanism for bearing remaining useful life prediction”. Appl Soft Comput J 86:105919. https://doi.org/10.1016/j.asoc.2019.105919
192. Li, J., and He, D., 2020, “A Bayesian Optimization AdaBN-DCNN Method with Self-Optimized Structure and Hyperparameters for Domain Adaptation Remaining Useful Life Prediction,” IEEE Access, 8, pp. 41482–41501. https://doi.org/10.1109/ACCESS.2020.2976595
193. Sahal, R., Breslin, J. G., and Ali, M. I., 2020, “Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case.” Journal of Manufacturing Systems, Elsevier, 54(December 2019), 138–151. https://doi.org/10.1016/j.jmsy.2019.11.004
194. Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T., 2017, “Intelligent Manufacturing in the Context of Industry 4.0: A Review.” Engineering, Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company, 3(5), 616–630. http://dx.doi.org/10.1016/J.ENG.2017.05.015
195. Gu, Y. J., Dong, X. F., and Yang, K., 2009, “Study on maintenance method intelligent decision support system used for power plant equipment.” Asia-Pacific Power and Energy Engineering Conference, APPEEC, pp. 1-4. https://doi.org/10.1109/APPEEC.2009.4918824
196. Lee, H., 2017, “Framework and development of fault detection classification using IoT device and cloud environment.” Journal of Manufacturing Systems, The Society of Manufacturing Engineers, 43, 257–270. http://dx.doi.org/10.1016/j.jmsy.2017.02.007
197. Lee, S. M., Lee, D., and Kim, Y. S., 2019, “The quality management ecosystem for predictive maintenance in the Industry 4.0 era.” International Journal of Quality Innovation, International Journal of Quality Innovation, 5(1, https://doi.org/10.1186/s40887-019-0029-5

198. Elattar, H. M., Elminir, H. K., and Riad, A. M., 2018, “Towards online data-driven prognostics system.” Complex & Intelligent Systems, Springer Berlin Heidelberg, 4(4), 271–282. http://link.springer.com/10.1007/s40747-018-0082-z

199. Ni, J., and Jin, X., 2012, “Decision support systems for effective maintenance operations.” CIRP Annals - Manufacturing Technology, CIRP, 61(1), 411–414. http://dx.doi.org/10.1016/j.cirp.2012.03.065

200. Wu, S., Gebraeel, N., Lawley, M. a, and Yih, Y., 2007, “A Neural Network Integrated Decision Support System for Condition-Based Optimal Predictive Maintenance Policy.” Ieee Transactions on Systems, Man, and Cybernetics, 37(2), 226–236. https://doi.org/10.1109/TSMCA.2006.886368

201. Ayvaz, S., and Alpay, K., 2021, “Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time.” Expert Systems with Applications, Elsevier Ltd, 173(November 2020), 114598. https://doi.org/10.1016/j.eswa.2021.114598

202. Wang, Y., Jia, Y., Yu, J., and Yi, S., 2017, “Field failure database of CNC lathes”. Int. J. Qual. Reliab. Eng. Manag. 16, 330–343 https://doi.org/10.1108/02656719910266532

203. You, D., Pham, H., 2016, “Reliability Analysis of the CNC System Based on Field Failure Data in Operating Environments”. Qual. Reliab. Eng. Int. 32, 1955–63 https://doi.org/10.1002/qre.1926

204. Zadeh, L.A. 1965, “Fuzzy Sets”, Inf. Control 8, 338–353 https://doi.org/10.1016/S0019-9958(65)90241-X

205. Abadi, D.N.M., Khooban, M.H., Alfi, A., and Siahi, M., 2014, “Design of Optimal Self-Regulation Mamdani-Type Fuzzy Inference Controller for Type I Diabetes Mellitus”. Arab. J. Sci. Eng. 39 977–986 https://doi.org/10.1007/s13369-013-0673-3

206. Tay, K. M., and Lim, C. P., 2006, “Fuzzy FMEA with a guided rules reduction system for prioritization of failures.” International Journal of Quality and Reliability Management, 23(8), 1047–1066. https://doi.org/10.1108/02656710610688202

207. Liao, L., and Pavel, R., 2013, “Machinery time to failure prediction - Case study and lesson learned for a spindle bearing application.” PHM 2013-2013 IEEE International Conference on Prognostics and Health Management, Conference Proceedings, (June 2013, https://doi.org/10.1109/ICPHM.2013.6621416

208. Zhang, L., Yan, R., Gao, R. X., and Lee, K., 2007, “Design of a Real-time Spindle Health Monitoring and Diagnosis System Based on Open Systems Architecture.” International

Smart Machining Systems Conference Design. [online], https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=823023

209. Dong, Y., Zhou, Z., and Liu, M., 2017, “Bearing preload optimization for machine tool spindle by the influencing multiple parameters on the bearing performance.” Advances in Mechanical Engineering, 9(2), 1–9. https://doi.org/10.1177%2F1687814016689040

210. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., 2019, “Deep learning for time series classification: a review,” Data Min. Knowl. Discov., 33(4), pp. 917–963. https://doi.org/10.1007/s10618-019-00619-1

211. Snoek, J., Larochelle, H., and Adams, R. P., 2012, “Practical Bayesian optimization of machine learning algorithms.” Advances in neural information processing systems 25, pp. 1-12. arXiv:1206.2944

212. Abbasimehr, H., Shabani, M., and Yousefi, M., 2020, “An optimized model using LSTM network for demand forecasting,” Comput. Ind. Eng., 143, p. 106435. https://doi.org/10.1016/j.cie.2020.106435

213. Chen, Z., Cao, S., and Mao, Z., 2018, “Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach.” Energies, 11(1) 28. https://doi.org/10.3390/en11010028

214. Louen, C., Ding S.X., and Kandler, C., 2013, "A new framework for remaining useful life estimation using Support Vector Machine classifier," 2013 Conference on Control and Fault-Tolerant Systems (SysTol), pp. 228-233, https://doi.org/10.1109/SysTol.2013.6693833.

215. Du, X., Xu, H., and Zhu, F. (2021). “Understanding the Effect of Hyperparameter Optimization on Machine Learning Models for Structure Design Problems.” CAD Computer Aided Design, Elsevier Ltd, 135, 103013. https://doi.org/10.1016/j.cad.2021.103013

216. Hertel, L., Collado, J., Sadowski, P., Ott, J., and Baldi, P. (2020). “Sherpa: Robust hyperparameter optimization for machine learning.” SoftwareX, Elsevier B.V., 12, 100591. https://doi.org/10.1016/j.softx.2020.100591

217. Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., and Deng, S.H., 2019, “Hyperparameter optimization for machine learning models based on Bayesian optimization,” J. Electron. Sci. Technol., vol. 17, pp. 26–40, https://doi.org/10.11989/JEST.1674-862X.80904120

218. Rai A., and Upadhyay, S.H., 2018, "Intelligent bearing performance degradation assessment and remaining useful life prediction based on self-organising map and support vector regression," vol. 232, pp. 1118–1132, https://doi.org/10.1177/0954406217700180
219. Tsirikoglou, P., Abraham, S., Contino, F., Lacor, C., and Ghorbaniasl, G., 2017, "A hyperparameters selection technique for support vector regression models," Appl. Soft Comput. J., vol. 61, pp. 139–148, https://doi.org/10.1016/j.asoc.2017.07.017
220. Laref, R., Losson, E., Sava, A., and Siadat, M., 2019, "On the optimization of the support vector machine regression hyperparameters setting for gas sensors array applications," Chemom. Intell. Lab. Syst., vol. 184, pp. 22–27, https://doi.org/10.1016/j.chemolab.2018.11.011
221. Jardine, A. K. S., Lin, D., & Banjevic, D., 2006, "A review on machinery diagnostics and prognostics implementing condition-based maintenance". Mechanical Systems and Signal Processing, 20, 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012
222. Gu, M., & Chen, Y., 2018, "A multi-indicator modeling method for similarity-based residual useful life estimation with two selection processes". International Journal of System Assurance Engineering and Management, 9, 987–998. https://doi.org/10.1007/s13198-018-0708-y
223. Zhang, Q., Tse, P. W. T., Wan, X., & Xu, G., 2015, "Remaining useful life estimation for mechanical systems based on similarity of phase space trajectory". Expert Systems with Applications, 42(5), 2353–2360. https://doi.org/10.1016/j.eswa.2014.10.041
224. Liu, Y., Hu, X. & Zhang, W., 2019, "Remaining useful life prediction based on health index similarity". Reliability Engineering & System Safety, 185, 502–510. https://doi.org/10.1016/j.ress.2019.02.002
225. Bejaoui, I.; Bruneo, D.; Xibilia, M.G., 2021, "Remaining Useful Life Prediction of Broken Rotor Bar Based on Data-Driven and Degradation Model". Appl. Sci. 11, 7175. https://doi.org/10.3390/app11167175
226. Banerjee, A., Gupta, S. K., and Putcha, C., 2021, "Degradation Data–Driven Analysis for Estimation of the Remaining Useful Life of a Motor." ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, 7(2), 04021012.
227. Liao, L. & Lee, J., 2009, "A novel method for machine performance degradation assessment based on fixed cycle features test". Journal of Sound and Vibration, 326(3):894-908. https://doi.org/10.1016/j.jsv.2009.05.005

228. Varanis, M. & Pederiva, R., 2015, “Wavelet Packet Energy-Entropy Feature Extraction and Principal Component Analysis for Signal Classification”. Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, 3(1), 1-7. https://doi.org/10.5540/03.2015.003.01.0471

229. Kundu, P., Darpe, A. K., & Kulkarni, M. S., 2019, “Weibull accelerated failure time regression model for remaining useful life prediction of bearing working under multiple operating conditions”. *Mechanical Systems and Signal Processing*, 134, 1-19. https://doi.org/10.1016/j.ymssp.2019.106302

230. Tayade, A., Patil, S., Phalle, V., Kazi, F., & Powar, S., 2019, “Remaining useful life (RUL) prediction of bearing by using regression model and principal component analysis (PCA) technique”. Vibroengineering PROCEDIA, 23, 30-36. https://doi.org/10.21595/vp.2019.20617

231. She, D. & Jia, M., 2019, “Wear indicator construction of rolling bearings based on multi-channel deep convolutional neural network with exponentially decaying learning rate”. Measurement, 135, 368–375. https://doi.org/10.1016/j.measurement.2018.11.040

232. de Myttenaere, A., Golden, B., Le Grand, B., and Rossi, F., 2016, “Mean Absolute Percentage Error for regression models.” Neurocomputing, Elsevier, 192, 38–48. https://doi.org/10.1016/j.neucom.2015.12.114

233. Ashton, Kevin. 2009, "That ‘internet of things’ thing." RFID journal 22(7): 97-114.

234. Mattern F., Floerkemeier C., 2010, “From the Internet of Computers to the Internet of Things”. In: Sachs K., Petrov I., Guerrero P., (eds) “From Active Data Management to Event-Based Systems and More”. Lecture Notes in Computer Science, vol 6462. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17226-7_15

235. Mohamed K.S., 2019, “IoT Cloud Computing, Storage, and Data Analytics. In: The Era of Internet of Things”. Springer, Cham. https://doi.org/10.1007/978-3-030-18133-8_4

236. Parsa, A., Najafabadi, T.A., Salmasi, F.R., 2018, “Implementation of smart optimal and au- tomatic control of electrical home appliances (IoT)”, IEEE Proc. 2017 Smart Grid Conf. SGC 2017, 2018-Janua, pp. 1–6. https://doi.org/10.1109/SGC.2017.8308861

237. Mtshali, P., Khubisa, F., 2019, “A smart home appliance control system for physically disabled people”, Conf Inf. Commun. Technol. Soc. ICTAS (2019) 1–5. https://doi.org/10.1109/ICTAS.2019.8703637

238. Singh, M., and Shimi, S. L., 2018, “Implementation of room automation with cloud based monitoring system.” Proceedings of the 2nd International Conference on Inventive

Systems and Control, ICISC 2018, IEEE, (Icisc), 813–817. https://doi.org/10.1109/ICISC.2018.8398911

239. Marimuthu, R., Deepak, V., Gowtham, S., and Prasad, V. R., 2013, “Remote Heart Rate Monitoring System.” 1(3), 47–52.

240. Mohamad, A. A. H., Jumaa, N. K., and Majeed, S. H., 2019, “ThingSpeak cloud computing platform based ECG diagnose system.” International Journal of Computing and Digital Systems, 8(1), 11–18. http://dx.doi.org/10.12785/ijcds/080102

241. Reviews, C., 2020, “An IoT based approach for smart ambulance service using ThingSpeak.” 7(9), 1697–1703. http://dx.doi.org/10.31838/jcr.07.09.307 I.

242. Talha, M., Upadhyay, A., Shamim, R., and Beg, M. S., 2018, “A cloud integrated wireless garbage management system for smart cities.” IMPACT 2017 - International Conference on Multimedia, Signal Processing and Communication Technologies, 175–179. https://doi.org/10.1109/MSPCT.2017.8363999

243. Pandit, S. N., Rohit Mohan Krishna, G. V. L. R., Akash, R., and Moharir, M., 2019, “Cloud Based Smart Parking System for Smart Cities.” Proceedings of the 2nd International Conference on Smart Systems and Inventive Technology, ICSSIT 2019, (Icssit), 354–359. https://doi.org/10.1109/ICSSIT46314.2019.8987592

244. Nasution, T. H., Muchtar, M. A., and Simon, A., 2019, “Designing an IoT-based air quality monitoring system.” IOP Conference Series: Materials Science and Engineering, 648(1). https://doi.org/10.1088/1757-899X/648/1/012037

245. Goyal, N., Goel, V., Anand, M., and Garg, S., 2020, “Smart Vehicle: Online Prognosis for Vehicle Health Monitoring.” Journal of Innovation in Computer Science and Engineering, 9(2), 12–22.

246. Chowdhry, B. S., Shah, A. A., Harris, N., Hussain, T., and Nisar, K., 2020, “Development of a Smart Instrumentation for Analyzing Railway Track Health Monitoring Using Forced Vibration.” 14th IEEE International Conference on Application of Information and Communication Technologies, AICT 2020 - Proceedings, 2–6. https://doi.org/10.1109/AICT50176.2020.9368670

247. Ashifuddinmondal, M., and Rehena, Z., 2018, “IoT Based Intelligent Agriculture Field Monitoring System.” Proceedings of the 8th International Conference Confluence 2018 on Cloud Computing, Data Science and Engineering, Confluence 2018, 625–629. https://doi.org/10.1109/CONFLUENCE.2018.8442535

248. Benyezza, H., Bouhedda, M., Djellout K., and Saidi, A., 2018, "Smart Irrigation System Based Thingspeak and Arduino," 2018 International Conference on Applied Smart Systems (ICASS), pp. 1-4. https://doi.org/10.1109/ICASS.2018.8651993

249. Pathak, A., Uddin, M. A., Jainal Abedin, M., Andersson, K., Mustafa, R., and Hossain, M. S., 2019, "IoT based smart system to support agricultural parameters: A case study." Procedia Computer Science, Elsevier B.V., 155, 648–653. https://doi.org/10.1016/j.procs.2019.08.092

250. Ramachandran, V., Ramalakshmi, R., and Srinivasan, S., 2018, "An Automated Irrigation System for Smart Agriculture Using the Internet of Things." IEEE, 1–6. https://doi.org/10.1109/ICARCV.2018.8581221

251. Danita, M., Mathew, B., Shereen, N., Sharon, N., and Paul, J. J., 2019, "IoT Based Automated Greenhouse Monitoring System." Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018, IEEE, (Iciccs), 1933–1937. https://doi.org/10.1109/ICCONS.2018.8662911

252. Shalini, H., and Aravinda, C. V., 2021, "An IoT-Based Predictive Analytics for Estimation of Rainfall for Irrigation". Advances in Intelligent Systems and Computing, Springer Singapore. http://dx.doi.org/10.1007/978-981-15-3514-7_105

253. Chandrayan B., Kumar R., 2020, "IoT Integration in Industry—A Literature Review". In: Kumar H., Jain P. (eds) "Recent Advances in Mechanical Engineering". Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-15-1071-7_2

254. D. Daji, K. Ghule, S. Gagdani, A. Butala, P. Talele and H. Kamat, "Cloud-Based Asset Monitoring and Predictive Maintenance in an Industrial IoT System," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-5. https://doi.org/10.1109/INCET49848.2020.9154148

255. Ge, Wenbo & Zhong, Ray., 2018, "Internet of things enabled manufacturing: a review". International Journal of Agile Systems and Management. 11. 126. https://doi.org/10.1504/IJASM.2018.092545

256. Awasthi, Y., and Mohammed, A.S., 2019, "IoT- A Technological Boon in Natural Disaster Prediction," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 318-322.

257. Pillai, A. S., Chandraprasad, G. S., Khwaja, A. S., and Anpalagan, A., 2021, “A service oriented IoT architecture for disaster preparedness and forecasting system.” Internet of Things (Netherlands), 14, 8–15. https://doi.org/10.1016/j.iot.2019.100076
258. Rymaszewska, A., Helo, P., and Gunasekaran, A., 2017, “IoT powered servitization of manufacturing – an exploratory case study.” International Journal of Production Economics, Elsevier B.V., 192(February), 92–105. http://dx.doi.org/10.1016/j.ijpe.2017.02.016
259. Khademi, A., Raji, F., and Sadeghi, M., 2019, “IoT Enabled Vibration Monitoring Toward Smart Maintenance.” 2019 3rd International Conference on Internet of Things and Applications (IoT), IEEE, 1–6. https://doi.org/10.1109/IICITA.2019.8808837
260. Mekid, S., 2021, "IoT for health and usage monitoring systems: mitigating consequences in manufacturing under CBM," 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 569-574. https://doi.org/10.1109/SSD52085.2021.9429296
261. Chandra, A. A., Jannif, N. I., Prakash, S., and Padiachy, V., 2017, “Cloud based real-time monitoring and control of diesel generator using the IoT technology.” 2017 20th International Conference on Electrical Machines and Systems, ICEMS 2017, 0–4. https://doi.org/10.1109/ICEMS.2017.8056222
262. Vishnu Karthik, S., Akshaya, V., and Sriramalakshmi, P., 2021, “IoT based predictive maintenance of electrical machines in aircraft.” Proceedings of the 7th International Conference on Electrical Energy Systems, ICEES 2021, 569–575. https://doi.org/10.1109/ICEES51510.2021.9383669
263. Tan, Y., Yang, W., Yoshida, K., and Takakuwa, S., 2019, “Application of IoT-aided simulation to manufacturing systems in cyber-physical system.” Machines, 7(1). https://doi.org/10.3390/machines7010002
264. Moens, P., Bracke, V., Soete, C., Hautte, S. Vanden, Avendano, D. N., Ooijevaar, T., Devos, S., Volckaert, B., and Van Hoecke, S., 2020, “Scalable fleet monitoring and visualization for smart machine maintenance and industrial iot applications.” Sensors (Switzerland), 20(15), 1–15. https://doi.org/10.3390/s20154308
265. Noyjeen, E., Tanita, C., Panthasarn, N., Chansri, P., and Pukkham, J., 2021, “Monitoring Parameters of Three-Phase Induction Motor Using IoT.” Proceeding of the 2021 9th International Electrical Engineering Congress, iEECON 2021, 483–486. https://doi.org/10.1109/iEECON51072.2021.9440368

266. Santiago, A.R., Antunes, M., Barraca, J.P., Gomes, D., and Aguiar, R.L., 2019, "Predictive Maintenance System for Efficiency Improvement of Heating Equipment," 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 93-98. https://doi.org/10.1109/BigDataService.2019.00019
267. Tao, F., Zuo, Y., Xu, L. Da, and Zhang, L., 2014, “IoT-Based intelligent perception and access of manufacturing resource toward cloud manufacturing.” IEEE Transactions on Industrial Informatics, 10(2), 1547–1557. https://doi.org/10.1109/TII.2014.2306397
268. Pasha, S., 2016, “Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis.” Int. J. New Technol. Res., 2(6), 19–23.
269. Gómez Maureira, M. A., Oldenhof, D., and Teernstra, L., 2014, “ThingSpeak – an API and Web Service for the Internet of Things.”
270. https://thingspeak.com/

Publications

International Journals

1. **Thoppil, N.M.,** Vasu, V. & Rao, C.S.P. "Failure Mode Identification and Prioritization Using FMECA: A Study on Computer Numerical Control Lathe for Predictive Maintenance". *J Fail. Anal. and Preven.* 19, 1153–1157 (2019). https://doi.org/10.1007/s11668-019-00717-8 **(SCOPUS, ESCI)**

2. **Thoppil, N.M.,** Vasu, V. & Rao, C.S.P. "On the Criticality Analysis of Computer Numerical Control Lathe Subsystems for Predictive Maintenance". *Arab J Sci Eng* (2020). https://doi.org/10.1007/s13369-020-04397-7 **(SCIE)**

3. **Thoppil, N.M.,** Vasu, V. & Rao, C.S.P. "Deep Learning Algorithms for Machinery Health Prognostics Using Time-Series Data: A Review". J. Vib. Eng. Technol. 9, 1123–1145 (2021). https://doi.org/10.1007/s42417-021-00286-x **(SCIE)**

4. **Thoppil, N.M.,** Vasu, V. & Rao, C.S.P. "Health indicator construction and remaining useful life estimation for mechanical systems using vibration signal prognostics". *Int J Syst Assur Eng Manag* 12, 1001–1010 (2021). https://doi.org/10.1007/s13198-021-01190-z **(SCOPUS, ESCI)**

5. **Thoppil, N. M.,** Vasu, V., and Rao, C. (October 28, 2021). "Bayesian optimization LSTM/bi-LSTM network with self-optimized structure and hyperparameters for remaining useful life estimation of lathe spindle unit." *ASME. J. Comput. Inf. Sci. Eng.* doi: https://doi.org/10.1115/1.4052838 **(ASME, SCIE)**

International Conferences

1. **Nikhil M Thoppil**, V Vasu, and C S P Rao, "An Integrated Learning Algorithm for Vibration Feature Selection and Remaining Useful life Estimation of Lathe Spindle Unit," *SASM-2021: International Conference on Simulation, Automation & Smart Manufacturing IEEE conference*, India, August 20-21, 2021 **(SCOPUS)**

2. **Nikhil M Thoppil**, V Vasu, and C S P Rao, "Vibration signal analysis for the prognostics of computer numerical control (CNC) lathe spindle unit," *International*

Conference on Precision Meso Micro Nano Engineering (COPEN), India, December 2019.

3. **Nikhil M Thoppil**, V Vasu, and C S P Rao, “Sensor-Based Method for Condition Monitoring of Machine Tools: A Review,” *International Conference on Advanced Functional Materials and Devices ICFMD 2019*, India, 26th- 28th February 2019

National Conference

1. **Nikhil M Thoppil**, V Vasu, and C S P Rao, “Condition-Based Maintenance of Machine Tool Spindle Unit: A Review,” *National Conference on Emerging Trends in Mechanical Engineering, Proceedings of EMTE 2019*, pp. 206-211

Acknowledgements

First and foremost, I would like to express immense gratitude to my supervisors *Dr. V. Vasu* and *Prof. C. S. P. Rao*. It has been a great learning experience working with them and a lot of credit for my achievements and accomplishments goes to both my supervisors. Their constant support and guidance have always motivated me to work and aided me to channelize my energy on the correct path. I have learned a lot of new things, both in my career as well as in life, during the time I spent with my supervisors.

I am very grateful to the doctoral committee chairman *Prof. A. Kumar* and members *Prof. N. Selvaraj*, *Prof. D. Dutta*, *Dr. P. S. C. Bose* for their timely inputs and suggestions. I acknowledge the support offered by *Dr. M Vijaya Kumar*, *Assistant Professor, NIT Warangal,* and *Dr. Thella Babu Rao*, *Assistant Professor, NIT Andhra Pradesh* for providing useful insights into my research. Thanks to the financial assistance from the Department of Mechanical Engineering, NIT Warangal.

I am grateful to *Dr. T. Sadasiva Rao*, *In-charge Mechanical Workshop*, and *Mr. K. Yellaswamy*, *Tech. Officer*, *Mechanical Workshop*, and all other technical assistants on their immense support to fabricate the accelerated run-to-failure experimental setup and perform experiments on it. I am also grateful to *Dr. Hari Kumar V., In-charge Computational Research Laboratory* for providing the workstations to execute algorithms required in the context of research work.

I am thankful to many researchers and scholars who have directly or indirectly helped me in this journey. I should point out the immense support offered by *Dr. Jinoop A. N., Post Doc, Waterloo, Canad*a for his continuous support during the entire span of my research work in making impactful research observations and publishing papers. I am also blessed to have *Mrs. Seenu P. Z., Research Scholar, NIT Warangal,* and *C. Kavin Kumar, Research Scholar, NIT Warangal* for being my best companions during my life at *NIT Warangal* and to share all our happiness and worries. I should, specially thank my senior *Dr. Arun Manohar* and Juniors *Mr. Maurya Upendra K., Mr. P. Jyothish*, and *Mr. Anuj Kumar* for their continued support during my research work. I am also thankful to my batch mates at NIT Warangal, all PG and UG friends for the joyful moments at NIT Warangal Campus.

On a special note, I would like to acknowledge the affection, support, and moral encouragement provided by my father *Mr. T. G. Mohanan*, mother *Mrs. T. Priyamvada*. I am always thankful for their prayers and sacrifice, and for listening to all my issues during my tough phase. Their prayers helped me to move forward in life. I am also thankful to my *in-laws*, *Mr. K. K. Thankraj* and *Mrs. P.J. Lalitha* for their motivation to better and belief in me. I am also thankful to my sister *Dr. Neenu P. Mohan* and the entire *Thoppil family* for respecting my hard work and supporting me in achieving my goals.

Last but not the least, I am eternally grateful to my caring and supportive wife *Dr. Sreelekshmy Raj*. It was a great comfort and relief that she could understand the hardship I am passing through and the encouragement when times got rough and stayed with me achieving my goal. I thank her for continuous prayer and motivation which helped me to make this thesis above my level of satisfaction.

[illegible]

Nikhil M Thoppil