EFFECTIVE APPROACHES FOR PROGNOSTICS

M. Tech. Thesis

By

SURAJ KUMAR

DISCIPLINE OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2018

EFFECTIVE APPROACHES FOR PROGNOSTICS

A THESIS

Submitted in partial fulfilment of the

Requirements for the award of the degree

of

Master of Technology

With specialization in

PRODUCTION AND INDUSTRIAL ENGINEERING

by

SURAJ KUMAR

DISCIPLINE OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2018

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled EFFECTIVE APPROACHES FOR PROGNOSTICS in the partial fulfilment of the requirements for the award of the degree of MASTER OF TECHNOLOGY and submitted in the DISCIPLINE OF MECHANICAL ENGINEERING, Indian Institute of Technology Indore, is an authentic record of my own work carried out during the time period from August 2016 to July 2018 of M. Tech. Thesis submission under the supervision of **Dr. Bupesh Kumar Lad**, Associate Professor, (PhD) IITD The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Signature of the student with date
(SURAJ KUMAR)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Signature of supervisor
(Dr. B.K. Lad)

Suraj Kumar has successfully given his M. Tech. Oral Examination held on 4th July 2018.

Supervisor Signature
Date:

Convener, DPGC
Date:

Signature of PSPC Member
Date:

Signature of PSPC Member
Date:

ACKNOWLEDGEMENTS

I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through in the completion of this work. I would like to express my sincere gratitude to my supervisor **Dr. Bhupesh Kumar Lad** for his guidance and support throughout the journey of my project work. His inputs, many insightful conversations during the development of the idea in this thesis, and helpful comments on the text have contributed substantially to the completion of my work. I would like to thank my seniors Dr. Amit, Dr. Sandeep for their kind supports and guidance throughout the whole project. I specially thanks to Mr. Manish for his consistent encouragement and guidance. In addition, special thanks to Mr. Maharshi and Mr. Kshitij for helping with programming related work in my project. Finally, I am thankful to my friends and colleagues here at IIT Indore for their help and encouragement.

Suraj Kumar

M. Tech. (**Production & Industrial Engineering**)

Discipline of Mechanical Engineering

IIT Indore

Dedicated

To

My family

and

My guide

Abstract

Today's manufacturing industries aims at reducing time and cost for maintenance of the products. As far as data driven approaches are considered for prognostics of life of the tool component, different prognostics models have been proposed so far. Although these models have aided the advancement of the discipline, they have made only a limited contribution to developing an effective machinery health prognostic system. One major cost incurred in production industry is the tool cost. In the past and even currently, continuous efforts have been made in this field. Prognostic which refers to determining the remaining useful life of the component play a significant role in bringing down the tool cost. Two most widely used approaches in this direction are the analytical physics based model approach and data based approach. Former lacks accuracy due to the number of assumptions and latter approach varies widely when switching from one model to another or when there is a certain variation in dataset which is always a case in a real world problem. Recent advances in condition monitoring technologies have given rise to many prognostic models for forecasting machinery health based on condition data. So there lies a need for more robust and reliable model for prognostics of machine tool component.

Most of the prognostic models are not intuitive as far as their computational strategy is considered. The set of logics and rules followed by these models are intuitive only to a lower dimensionally cases like 2-dimentional cases. As we switch to higher dimensional problem the representation of the problem becomes difficult hence it gets less intuitive. One cannot wonder the logics behind the classification or regression results proposed by a machine learning models based on a given dataset but it is believed they often provide accurate results. There always lies little insecurity about the results proposed by given prognostic model. To deal with uncertainty in the results proposed by the model we have defined a novel approach so that we can get more intuitive results in the prognostics for tool failure life.

This work presents a novel approach in addressing the above-mentioned challenges. In our present novel approach, we have tried many machine learning models on our dataset to predict the life of the tool. Also, we have tried to devise a novel approach which involves the combination of different machine learning approaches into the single model of machine learning.

The new novel approach proposed tries to add more confidence to the prediction compared to the traditional approaches. .

1. TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AI – Artificial Intelligence

PHM – Prognostic Health Management

JIT – Just in time

MRP – Material Resource Planning

CHAPTER 1. INTRODUCTION

1.1 BACKGROUND

Condition monitoring is the process of monitoring a parameter of condition in machinery, such that a significant change is indicative of a developing failure. It is a major component of Predictive Maintenance. The use of conditional monitoring allows maintenance to be scheduled, or other actions to be taken to avoid the consequences of failure, before the failure occurs. The center of point of discussion will be the tool condition monitoring not entire machinery. Operational reliability of the machinery components like cutting tool is of vital importance in almost every field of industries majorly including production based on machines. Unexpected tool failure not only lead to expensive downtime but it also leads to other unwanted situations like safety concerns, environment detriment. In some worst cases it also leads to enormous legal expenses. Breakdown of machine unexpectedly affects the implementation of present advanced production strategies like Just-in-Time (JIT) and Material Resource Planning (MRP) leading to waste of operational facilities and inventory loss [1].

Spending too much time on maintenance related works brings down the effective productive hours of the labor employs at the production site. Craftsmen spend as little as two hours a day doing actual hands-on work activities [2], similarly in the production department, workers waste too much time due to unwanted breakdown of tool components. Many organizations due to the same reason try to relocate to different parts of the world in search of cheaper labor to cut down their production cost to a large extend. It certainly brings our concern to one the major problem i.e. how to reduce dependencies over these unexpected tool failures.

Currently, many models have been proposed in the research area of machinery fault prognosis. The physics-based models attempt to combine system-specific mechanistic knowledge, defect growth formulas and CM data to predict the propagation of a fault, whereas the data-driven approaches derive models directly from the acquired CM data. After an extensive literature review, several limitations of the existing models have been identified. For example, many of the existing models mostly the work found is on the diagnostics of machine tool components

This research is aimed at developing new practical methods for addressing these limitations. The rest of this chapter will define diagnostics and prognostics, scope of the research, problem of statement and the contribution of this novel approach methodology. Finally, a brief overview of the thesis is presented

1.2 DIAGNOSTICS AND PROGNOSTICS

Diagnosis is the identification of the nature and cause of a certain phenomenon. It helps in identifying the surety if something is wrong with given machine. Fault detection, isolation, and recovery (FDIR) is a subfield of control engineering which concerns itself with monitoring a system, identifying when a fault has occurred, and pinpointing the type of fault and its location. Two approaches can be distinguished: direct pattern recognition of sensor readings that indicate a fault and an analysis of the discrepancy between the sensor readings and expected values, derived from some model. In the latter case, it is typical that a fault is said to be detected if the discrepancy or residual goes above a certain threshold. It is then the task of fault isolation to categorize the type of fault and its location in the machinery. Machine fault diagnosis is a field of mechanical engineering concerned with finding faults arising in machines. A particularly well-developed part of it applies specifically to rotating machinery, one of the most common types encountered. To identify the most probable faults leading to failure, many methods are used for data collection, including vibration monitoring, thermal imaging, oil particle analysis, etc. A machinery prognostic is the forecast of the remaining operational life, future condition, or probability of reliable operation of an equipment based on the acquired condition monitoring data. Prognostics [4] is an engineering discipline focused on predicting the time at which a system or a component will no longer perform its intended function.[5] This lack of performance is most often a failure beyond which the system can no longer be used to meet desired performance. The predicted time then becomes the remaining useful life (RUL), which is an important concept in decision making for contingency mitigation. Prognostics predict the future performance of a component by assessing the extent of deviation or degradation of a system from its expected normal operating conditions. The science of prognostics is based on the analysis of failure modes, detection of early signs of wear and aging, and fault conditions. An effective prognostics solution is implemented when there is sound knowledge of the failure mechanisms that are likely to cause the degradations leading to eventual failures in the system. It is therefore necessary to have initial information on the possible failures (including the site, mode, cause and mechanism) in a product. Such knowledge is important to identify the system parameters that are to be monitored.

The notation of prognosis has been addressed widely in the literatures by various authors. Prognosis research is done in areas such as, mechanical systems (e.g., rail transport, automotive, and aircraft), power systems (e.g.,
fossil-fuelled power plants), and continuous-time production processes (e.g. chemical and petrochemical plants, and pulp and paper mills) where structural durability and operational reliability are critical, also prognosis work found to be high in Department of Defence (DOD), including Navy, Air Force, Army and DARPA, is approaching development of prognosis architectures and technologies in different ways [6]. Prognosis results are used for proactive decisions about preventive actions with the economic goal of maximizing the service life of replaceable and serviceable components while minimizing operational risk [7], as main concept of prognosis is to measure the remaining useful life of a component.

1.3 TOOL CONDITION MONITORING

The tool condition monitoring of the end milling tool is a very complicated process and it involves errors at various stages. The data which is processed for the analysis possess lot of trouble because of signals generation at a very high sampling rate, also the noises attached with the parent signals further increases the problem of signal analysis. Whenever carrying out a certain tool condition monitoring based on some sort of signal monitoring like vibration, force, acoustic signals, it is very much necessary to have enough data to comment upon the inherent characteristic trend hidden inside the data. We have number of monitoring system which aims at monitoring the state of the tool health based on the past data but mostly lacks in updating the models on a regular basis. Before starting any of the signal monitoring techniques for any tool or machinery one must spend a considerable time to justify the type of signals which have potential to govern the state of tool with considerable amount of confidence. The vibration, force and acoustic signals gave quiet promising results because of its high relevance from the source of failure in our case of monitoring condition of end milling tool. The dynamometer signals can be of great importance as they can provide a three-dimensional view of the machining process (x, y and z-axis). Also, the dynamometer could detect which part of the tool is the most deteriorated since most of the wear in the tool affects the forces exerted by the tool in multiple directions. The current sensor gave the same results as the dynamometer and because of their high correlation value; these two sensors can be used to validate their respective results. The accelerometer signal was very difficult to filter and it was prone to error.

1.4 SENSOR IMPORTANCE AND DATA COLLECTION

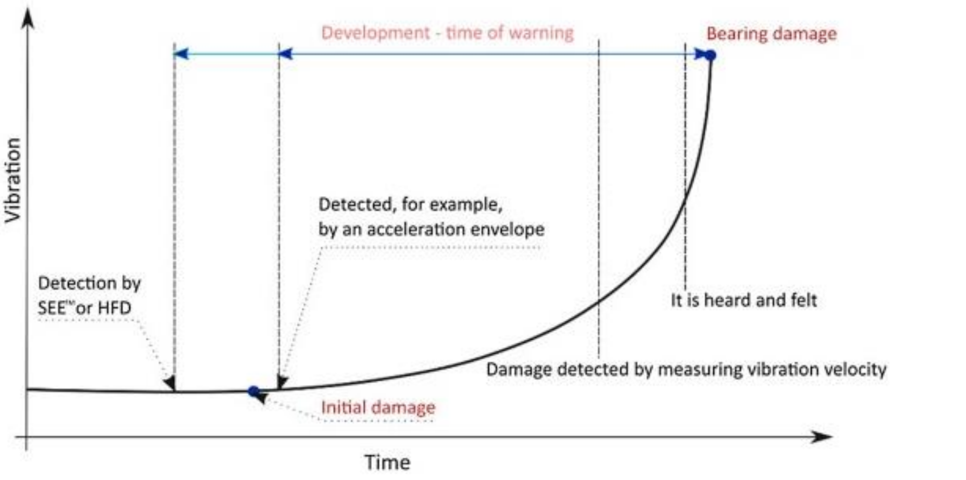


Figure 1 Vibration fault identification trend [8]

Among all the signal monitoring features for machine health prognostics vibration data, is a kind of signal which is widely used for prognostics. In Fig 1 it is clearly depicting that machine operator experienced based identification of the failure leave very less or no time for taking any preventive action to avoid the failure of the machine. So, the more reliable source is some machine learning model which proves to be more sensitive compared to the visual or human heard indemnification of the fault. In most of the failures the identification made by such signal monitoring techniques allows considerable time to fix up the issue before the actual break occurs which might to other consequences like increase in labor cost, production cost, downtime etc. This brings up to the question to what extent these prediction models are reliable. Various research has been carried out in this area identifying the search of better and more reliable extracted features devoid of unnecessary noises Wide variety of techniques have been proposed in past using various signal processing techniques utilizing analysis in both frequency domain and time domain of number of signals extracted from machine running in the present state. These techniques are intuitive only to the person who is expert of this field but to the real operating manager or worked of real machines these complex procedures are very tough to comprehend. The present world of Artificial Intelligence (AI) is trying to bring down these complexities to a lower level and also claiming to give more reliable outcomes compared to prediction models based on traditional signal processing techniques. But in this case the actual working of these AI based models is not known even by experts. It is just the black box computation but the past results has shown the results based on these models proves to be very reliable in most cases but still lacks the sufficient degree of confidence backing up these predictions.

The data collected by a number of sensors is the backbone of any model which is generated latter on for diagnostics or prognostic purposes. The present data collection techniques are not much advanced that they arrange data as per their utility to a specific section. So this disorganized collection of data finally poses difficulty to the data analyst team in extracting the valuable information out of that data. It not only results in improper utilization of resources but also lead waste of time during the preprocessing of data. In most of the cased the more closely governing features are often being ignored because of improper handling of data. In our present research we try to present a new novel approach to divide the data collection technique to specific classes which ease the analysis process of the data collected.

1.5 PROBLEM STATEMENT

Before arriving at the exact problem statement lets first try to understand the most widely used term ‘Data’. Data is defined as the facts and statistics collected together for reference or analysis. Everything around us from your presence to your absence at work, our meal eating time, sleeping time etc can be documented and this record is nothing but the data. Any information collected from the specific system has the potential to be interpreted in some valuable information. Let’s take the example of a turbine plant, numbers of sensors are installed to monitor the various components health status but here lies the major drawback in terms of data collection techniques. It is a usual tradition that data collected is not always well labeled based on the specific state of the machine. At one it ease the data collection process but latter on it leads to number of difficulties in terms of it is analysis. Data analysis team further spends much of their time in sorting the data though the same process of sorting the data to specific group or class could be done right the data collection step which is the first steps towards data analysis.

The proper collection of data specific group not only helps in easy prior classification of data but it also contributes to the improved accuracy of models which are based on these models. In addition to the improved accuracy it also make them more reliable because models generated are most likely based on features which shows most of the variance lowering the scope of noise contributed features included in the model.

1.6 ORIGINALITY AND CONTRIBUTION

Specifically talking about the cutting tools, the working mechanism of these tools used in machining have been proposed from tons of models, each model backed up with number of assumptions making them less reliable to state the actual working conditions in practical conditions. Later on simulation based methods were introduced which ease the visualization capability of the system to see post working conditions of the tool under the given working conditions. Unfortunately, these simulations methods need to have a backbone to work with which is nothing but the same analytical proposed models. Hence these simulation methods also carry inaccuracies along with them inherited by the analytical methods. In the recent past years machine learning models are been used extensively which monitors and analyses the tool working conditions and state the condition of tool in future. On one end these models are quite accurate but cannot be relied blindly. There not a single universal model which will fit the all type of problems. One model may work well with a given data extracted from tool's past states but the same model may not work with another. This puts the analyst under dilemma that which model to use whose results can be trusted with considerable reliability.

To deal with this problem of lack of reliability in the results proposed by various algorithms we try to come up with a novel approach to state the prognostics results for tool condition in a more appealing way. The level of risk involved due to the failure of tool varies continuously from the beginning till the end of its life. We try to present the results for tool prognostics in the groups of conservative and non-conservative approaches so depending on the level of risk involved at the given state, the operator can decide which results he has to go along. In addition to this we also try to present the methodology to group the data as per their category right at first step of data processing which is data collection. It helps in building the model more robust as the data in the given group become denser quantitatively. The results of predictions also helps in grouping of the data to the specific class it belongs.

CHAPTER 2. LITERATURE REVIEW

2.1 BACKGROUND

The increase in awareness regarding the need to optimize manufacturing process efficiency has led to a great deal of research aimed at machine tool condition monitoring. The applications of tool condition monitoring techniques to the detection of cutting tool wear and breakage during the milling process. Established approaches to the problem are considered and their application to the next generation of monitoring systems is discussed. Two approaches are identified as being key to the industrial application of operational tool monitoring systems. Multiple sensor systems, which use a wide range of sensors with an increasing level of intelligence, are providing long-term benefits, particularly in the field of tool wear monitoring. Such systems are being developed by many researchers in this area. The second approach integrates the control signals used by the machine controller into a process monitoring system which is capable of detecting tool breakage. Findings in literature have shown that both these approaches can be of major benefit. It is finally argued that a combination of these approaches still lacks in providing ultimate robust systems which can operate in an industrial environment.

Accurate tool condition monitoring (TCM) is essential for the development of fully automated milling processes. The complexity of milling processes continues to complicate the implementation of TCM. The review strictly considers the state-of-the-art methods which are employed for conducting

TCM in milling processes. Under the review we are following some of the extreme important key components of Tool Condition Monitoring (TCM) namely sensors, feature extraction, monitoring different models used for the categorization of cutting tool states in the decision-making process and predicting the life of the tool by implementing the prognostic models. In addition, the primary strengths and weaknesses of current practices are presented for these three components. Finally, this chapter concludes with a list of recommendations which can take the research in Tool Condition Monitoring (TCM) to the new level.

2.2 ANATOMY OF END MILLING TOOL

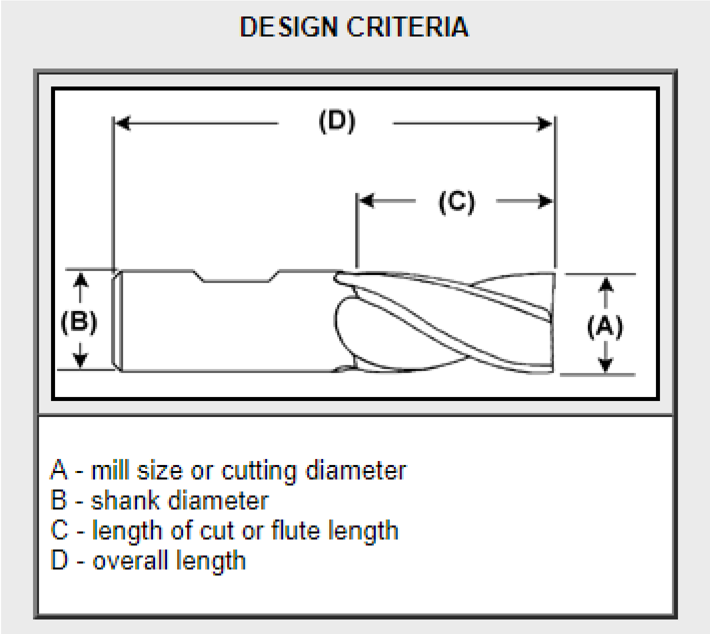


Figure 2 Basic Geometry of the end milling tool [9]

End milling tool has high resemblance with the drill bits but it actually differs many aspects. Figure 2 shows the main components of the end milling tool. More detailed components of end milling tool will be shown in the latter part of this chapter. An end mill is a type of milling cutter or a cutting tool used in industrial milling applications. It is distinguished from the drill bit in many aspects like its application, geometry, and manufacture. While a drill bit can only cut in the axial direction, a milling bit can generally cut in all directions, though some cannot cut axially. Several broad categories of end milling tools exist, such as center-cutting versus non-center-cutting (whether the mill can take plunging cuts); and categorization by number of flutes; by helix angle; by material; and by coating material. Each category may be further divided by specific application and special geometry. End mills feature many different dimensions that can be listed in a tool description. It is important to understand how each dimension can impact tool selection, and how even small choices can make all the difference when the tool is in motion.

2.2.1 Flutes

Flutes are the easiest part of the end mill to recognize. These are the deep spiraled grooves in the tool that allow for chip formation and evacuation. Flutes are the part of the anatomy that allows the end mill to cut on its edge. One consideration that must be made during tool selection is flute count. Lower the flute count, the larger the flute valley – the empty space between cutting edges. This void affects tool strength, but also allows for larger chips with heavier depths of cut, ideal for soft or gummy materials like aluminum. When machining harder materials such as steel, tool strength becomes a larger factor, and higher flute counts are often utilized.

2.2.2 Profile

The profile refers to the shape of the cutting end of the tool. It is typically one of three options: square, corner radius, and ball.

a) **Square Profile:** Square profile tooling features flutes with sharp corners that are squared off at a 90° angle. In our experiment we have used this tool as the experiment demands the wear out of the tool and with this given profile of end milling tool it ease the process of wear out of the tool.

b) **Corner Radius:** This type of tooling breaks up a sharp corner with a radius form. This rounding helps distribute cutting forces more evenly across the corner, helping to prevent wear or chipping while prolonging functional tool life. A tool with larger radii can also be referred to as "bull nose".

c) **Ball Profile:** This type of tooling features flutes with no flat bottom, rounded off at the end creating a "ball nose" at the tip of the tool.

2.2.3 Helix angle

The helix angle of a tool is measured by the angle formed between the centerline of the tool and a straight line tangent along the cutting edge. A higher helix angle used for finishing (45°, for example) wraps around the tool faster and makes for a more aggressive cut. A lower helix angle (35°) wraps slower and would have a stronger cutting edge, optimized for the toughest roughing applications.

A moderate helix angle of 40° would result in a tool able to perform basic roughing, slotting, and finishing operations with good results. Implementing a helix angle that varies slightly between flutes is a technique used to combat chatter in some high-performance tooling. A variable helix creates irregular timing between cuts, and can dampen reverberations that could otherwise lead to chatter.

2.2.4 Pitch

Pitch is the degree of radial separation between the cutting edges at a given point along the length of cut, most visible on the end of the end mill. Using a 4-flute tool with an even pitch as an example, each flute would be separated by 90°.

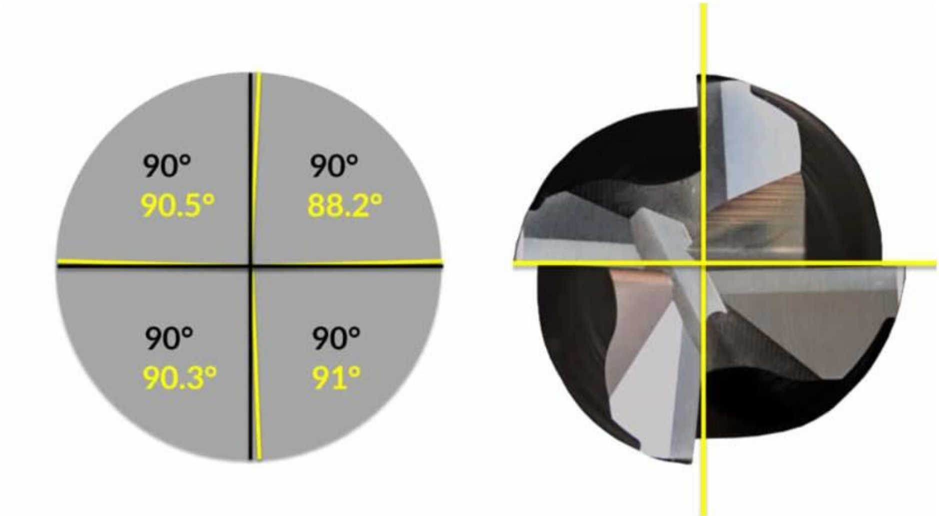


Figure 3 Pitch in the End Milling tool [9]

Similar to a variable helix, variable pitch tools have non-constant flute spacing, which helps to break up harmonics and reduce chatter. The spacing can be minor but still able to achieve the desired effect. Using a 4-flute tool with variable pitch as an example, the flutes could be spaced at 90.5 degrees, 88.2 degrees, 90.3 degrees, and 91 degrees (totaling 360°).

2.2.5 LITERATURE REVIEW

Milling is a machining process which is very common and efficient cutting operation. A typical end milling tool that uses a end milling cutter with one or more teeth to intermittently cut work pieces into flat surfaces, grooves, threads, and many geometrically complex components. Highly efficient milling processes are suitable for mass production, and have been employed widely in industrial manufacturing. Cutting tools are considered to be the primary component of the milling process [10], and tool breakage is a major reason of unscheduled stoppage in industrial settings employing milling. Tool breakage is generally the result of an accumulation of tool damage over time, and has negative indirect (time loss) and direct (capital) effects. In terms of these effects, tool failure accounts for 7–20% of total milling-machine downtime [11, 12], and the costs of tools and tool changes account for 3–12% of the total processing cost [13]. In an effort to limit the indirect effects of tool breakage, milling tools employed in industrial milling processes are replaced prior to breakage. However, conventional tool replacement strategies employ uniform time periods that are determined by the subjective experience of operators. Such strategies inevitably result in either the early replacement of workable tools, which increases tools costs and downtime, or the late replacement of worn tools, which results in lower work piece quality and increased production costs [14]. In fact, research [15, 16] has determined that only 50–80% of the effective life of milling tools is typically used. As such, monitoring the varying conditions of milling tools over time to facilitate a timely detection of tool damage is critical for limiting the indirect effects of tool breakage while maximizing the usable life of milling tools. Thus, tool condition monitoring (TCM) systems have been developed to generate better work piece surface quality and extend tool life by diagnosing cutting tool deficiencies using appropriate signal processing and pattern recognition techniques. An accurate and reliable TCM system can reduce costs by 10–40% by reducing downtime and maximizing the usable life of milling tools [16, 17]. The application of TCM in milling processes has been studied for over 30 years, and has been based on two types of methods: direct monitoring and indirect monitoring. Direct monitoring methods employ optical equipment and machine vision technology to directly monitor tool condition. For example, optical microscopes are used to capture tool images and the tool condition is evaluated with image analysis technology [18]. Direct methods are advantageous because they do not affect the machining process and offer high recognition accuracy under ideal conditions. However, direct methods are generally unsuitable for manufacturing settings because (1) the high expense of the required equipment and software can unacceptably increase manufacturing costs, and (2) the recognition accuracy is easily disturbed by the presence of cutting fluid and cutting chips on tool surfaces [19, 20]. Therefore, indirect monitoring methods, which estimate tool condition based on an analysis of signals derived from one or more sensors, have been widely adopted in manufacturing settings. These signals can be representative of numerous characteristics, such as cutting force, vibration, motor current and acoustic emission (AE). The signal analysis conducted seeks to extract significant features from a signal that are indicative of tool condition, and includes numerous methods, such as those based on the time domain, frequency domain, wavelet transform (WT), empirical mode decomposition (EMD), and multi-domain analysis. Finally, the cutting tool condition could be evaluated with extracted feature parameters based on one certain pattern recognition method, such as artificial neural network (ANN), hidden Markov model (HMM), and support vector machine (SVM). Compared with direct methods, indirect methods are less expensive and more adaptable to practical applications.

Indirect TCM is data-driven, and can be divided into the three phases illustrated in Fig. 1: model training, model testing and online monitoring. The model training phase consists of three modules: setting up the sensor configuration, feature extraction, and monitoring model. The sensor configuration module provides a sensor signal, the feature extraction module extracts features in the sensor signal that are related to tool condition (e.g., wear, damage, and breakage), and the monitoring model module builds a decision support model for online monitoring. The online monitoring phase consists of three modules: online sensor monitoring and decision making.

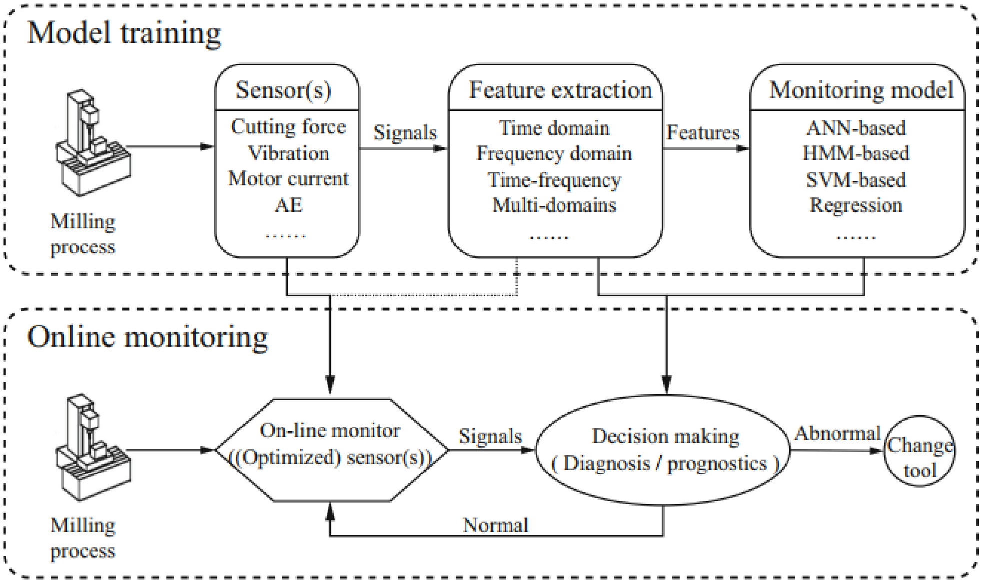


Figure 4 Basic process flow of tool condition monitoring (TCM) in milling processes [21]

The sensor configuration employed during online monitoring is based on that employed during the model training phase, and is generally equivalent. However, if the sensor configuration is altered (it is possible when using multiple sensors initially), the online monitoring phase will not consider irrelevant sensors when evaluating tool condition. In the decision-making module, the sensor signals are firstly subjected to an equivalent feature extraction as that employed during the model training phase, and then the cutting tool condition can be evaluated in real time based on the trained monitoring model obtained during the model training phase.

2.3 SENSOR SIGNALS

1. Cutting forces

During milling processes, tools wear increases surface roughness, and leads to a corresponding increase in cutting force. Many studies [10, 22–24] have demonstrated that the cutting force is very sensitive to changes in tool condition, and can therefore accurately estimate the tool state. For example, Wang et al. [25] determined that the cutting force signal is the most stable and reliable signal among commonly employed sensor signals that are closely related to tool wear. Huang et al. [26] employed a piezoelectric dynamometer to monitor the tool state of an end milling operation according to cutting force. Bulent et al. [27] adopted a rotary dynamometer to capture the cutting forces in three dimensions and the torque of the drive moment on a rotating tool. However, cutting force sensors are difficult to apply in industrial environments because their physical properties limit the physical size of a workpiece, which is not appropriate when milling medium and large workpieces [28]. In addition, Koike et al. [29] established that cutting force monitoring interferes with the motion control of the spindle and stage in a milling machine, and reduces its rigidity. Moreover, the expense of commercial dynamometers can unacceptably increase manufacturing costs [30, 31].

2. Vibration

Vibration sensors are widely employed in TCM because they are inexpensive, easy to install, and provide a similar periodic signal shape to that of the cutting force [32–34]. Besmir et al. [35] established that low levels of vibration are generated with sharp cutting tools, while the levels of vibration increase with increasing deterioration in the tool condition. Numerous studies have demonstrated the feasibility of adopting vibration signals for TCM in milling processes

[36–37]. For example, Hsieh et al. [38] demonstrated that the spindle vibration acceleration signal can distinguish different tool conditions during micro-milling when used in conjunction with appropriate feature extraction and classifiers. Madhusudana et al. [39] installed a tri-axial integrated electronic piezoelectric (IEPE) accelerometer on the spindle housing to capture the spindle vibration acceleration signal during face milling. Gao et al. [40] achieved positive tool condition diagnostic accuracy by adopting a laser vibrometer to acquire the vibration displacement of a tool holder. However, the characteristics of milling processes limit the accuracy of TCM employing vibration signals. First, vibrations are generated during machine operation even when the tool is not engaged in cutting, as during an air-cut operation. In fact, effectively distinguishing between entity-cut and air-cut operations remains an open challenge. Second, vibration signals are difficult to filter, and are therefore prone to providing erroneous data [35]. Finally, the position of sensor installation and the use of cutting fluid can affect the vibration signal [20].

3. Acoustics

Sensors based on AE are particularly suitable for conducting TCM in milling processes because the resulting signals are not mechanically disturbed, have a superior sensitivity to the those of cutting force and vibration signals, and propagate at a frequency much greater than the characteristic frequency caused by cutting, which reduces interference

[43–44]. A study conducted by Vetrichelvan et al. [2] demonstrated that the AE signals obtained from sensors located on the top surface of the tool holder can effectively monitor crater wear. Mathew et al. [45] conducted experiments with 1-tooth, 2-tooth, and 3-tooth milling cutters, and demonstrated that AE signals exhibit marked responses to changes in tool condition such as tool breakage and tool chipping. Ren et al. [46] established that AE signals are easily recorded, and provide very rapid responses to changing conditions in the contact between the tool and workpiece; thus, AE sensors are well suited for TCM in micro-milling processes. However, intermittent cutting during milling processes results in AE signal spikes when individual teeth enter or exit the workpiece, which complicates the analysis of AE signals [28]. In addition, AE sensors are highly sensitive to environmental noise [47], which increases the difficulty of extracting valid signal feature information.

2.4 SUMMARY

Monitoring changing tool conditions in the milling process is exceedingly complicated owing to the influence of many factors, such as the cutting conditions, work piece material, and environmental parameters involved in specific milling processes (e.g., face milling or end milling). This challenge has so far interfered with the development of a general reference model for TCM appropriate for all milling processes. Although much progress has been made in TCM research for milling processes, the following important questions remain open, and require further study.

1. There should be specific TCM models for specific milling conditions because the characteristic of each type of milling like end milling, face milling etc are very different from one another.
2. The configuration of sensor for signal monitoring should be carefully handled as the accuracy during the monitoring of the signals affect the data set hence the developed model based on that data lacks in accuracy as well.
3. Use of monitoring models for prognosis rather than diagnosis. Predicting the RUL rather than diagnosing tool condition is of particular interest in industrial settings because the RUL and failure probability of a milling tool are more meaningful than the diagnosis of tool wear.
4. Application of more advanced AI-based monitoring methods. The AI field is rapidly developing, resulting in the emergence of new advanced algorithms.

2.5 RESEARCH OBJECTIVES

1. Developing the different machine learning models to identify which model works the best for our data set generated from end milling machining.
2. Proposing the novel methodology for data collection techniques aiming at keeping the data collected in their distinct groups.
3. Making use of LSTM for Tool condition monitoring. So far LSTM models have been widely used in the field of the speech recognition. I have tried to utilize the same for predicting the remaining useful life of the tool.
4. Generalizing the prognostic methodology for any new machine.

CHAPTER 3. PERFORMANCE EVALUATION OF VARIOUS MACHINE LEARNING MODELS

3.1 INTRODUCTION

Machine learning is an idea to learn from examples and experiences, without being explicitly programmed. Instead of writing code, you feed data to the genetic algorithm, and it builds logic based on the data given. Broadly there are three types of Machine learning Algorithms: [48]

a) Supervised Learning: This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, Logistic Regression etc.

Artificial neural network: An ANN is based on a collection of connected units or nodes called artificial neurons which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

b) Unsupervised Learning:

In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups. Examples of Unsupervised Learning: K-means.

c) Reinforcement Learning:

Using this algorithm, the machine is trained to make specific decisions. It trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge

to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process.

As the part of this chapter our discussion will be limited to supervised learning algorithms. For optimization of various machine learning models I have tried to optimized the number of hyper parameters of various machine learning models as universal approximation methods states that any machine learning models can represent the wide variety of functions when they are given the approximate value to the hyper parameters.

3.2 DEEP NEURAL NETWORKS

Neural networks can be recurrent or feed forward. Feed forward networks are the one that do not have any loops in their graph and can be organized in layers. If there are "many" layers, then we say that the network is deep.
How many layers does a network should have in order to qualify as deep? There is no definite answer to this (it's a bit like asking how many grains make a heap), but usually having two or more hidden layers counts as deep. In contrast, a network with only a single hidden layer is conventionally called "shallow". Informally, "deep" suggests that the network is tough to handle.
As the part of this thesis we restrict our discussion to neural network having only one hidden layer and optimization of model is tried by varying the number of nodes in that hidden layer.

3.2.1 RECURRENT NEURAL NETWORK

Recurrent neural network are formally known as RNN. As name suggests they uses recursion in a small network of few layers and maintains a hidden state that is being reused in every recursion. Like other deep learning models, RNN has layers, learnable parameters (weights), a loss function, and optimization algorithm like SGD, etc. (hyper parameter). However the RNN network is reused several times with hidden state and either with previous recursion output, or a new input instance, or nothing. The number of recursions depends on the hidden state as well as the provided input. This way of forward computation, large deep executions can be done with few layers and their hidden states. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.

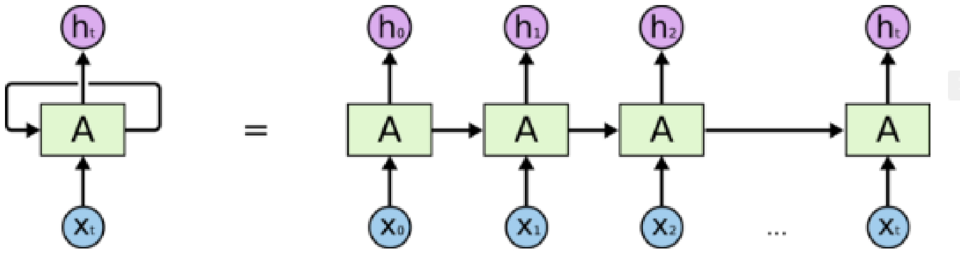


Figure 5 An unrolled recurrent neural network. [49]

3.2.2 LONG SHORT TERM MEMORY NETWORK (LSTM)

LSTM is a very special kind of recurrent neural network which works, for most of the tasks it performs much better than the standard version of traditional RNN. LSTM network finds its huge valuable importance in the field of word recognition or prediction. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in “the clouds are in the sky,” we don’t need any further context – it’s pretty obvious the next word is going to be sky. In such cases, where the gap between the relevant information and the place that it’s needed is small, LSTMs can learn to use the past information. I have tried to use this same unique feature of LSTM and use it to our advantage to predict the RUL of cutting tool based on its past state. The further details about the LSTM models will be discussed in the latter chapter during its implementation on our end milling data set

3.2.3 EXPERIMENT DETAILS

Milling operation is chosen is this research. EMCO MILL E350 CNC three-axis high-speed vertical milling machine is utilized as the test bed. Normal degradation of end milling cutting tool (Figure 7, 8) is carried out to study the degradation behavior of the cutting tool. A high speed steel 6 mm milling tool is chosen for analysis. Experiment was carried out for a range of operating conditions which was used by varying feed, depth of cut and speed. The work piece used is of mild steel (figure 8, 10) of dimension 165mm x 100mm.

Figure 6 End milling tool 6mm utilized in experiment

Figure 7 flute of end milling positioned at 90 degree

Figure 8 Mild steel plate (165mm X 100mm)

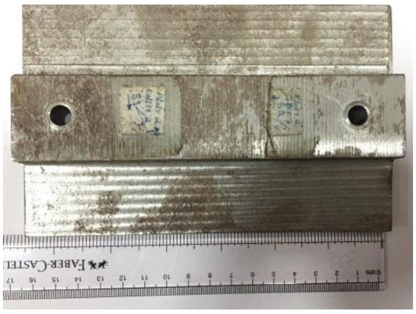


Figure 9 Mild steel plate after machining

3.2.4 BASIC ARCHITECTURE OF MACHINE LEARNING MODELS

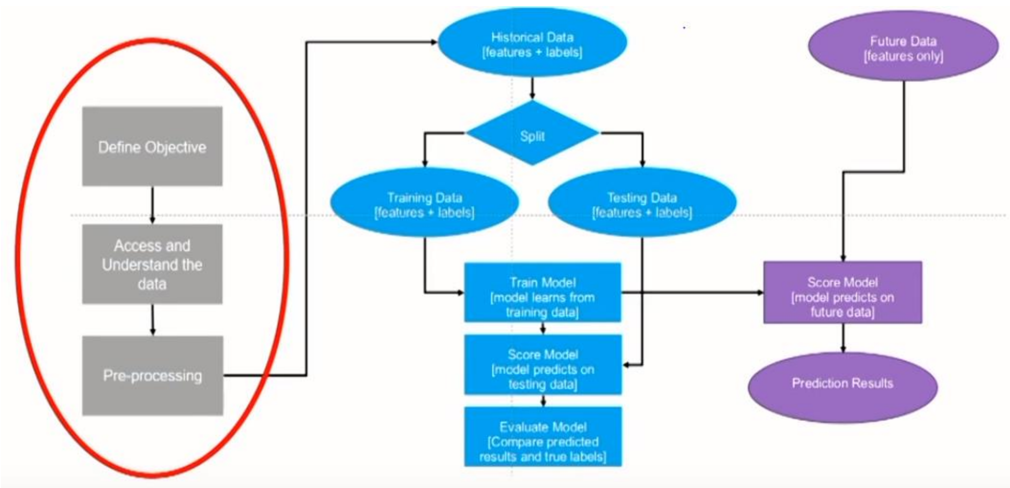


Figure 10 Data acquisition

The signals for force, vibration, and acoustics are constantly monitored throughout the experiment. The data collected for each cut is processed for extracting the number of statistical features to approximate distribution of each signal during each cut precisely. Every data point consists of two parts: first is number of features, second is the label i.e. remaining useful life (RUL) of the component. RUL is defined as the number of cuts a given tool can make for a given operating condition each cut refers to transversing the tool over mild steel for a length of 1320 mm at the fixed operating conditions.

3.2.5 Data Processing

Machine learning algorithms learn from data. It is critical that you feed them the right data for the problem you want to solve. Even if you have good data, you need to make sure that it is in a useful scale, format and even those meaningful features are included. This step is very crucial in machine learning as it helps in the more consistent and better results. Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be removed from the data entirely. Data preparation is a large subject that can involve a lot of iterations, exploration and analysis. Getting good at data preparation will make you a master at machine learning. There are number of statistical features which are calculated to comment upon the feature importance like Pearson’s coefficient, Anova, LDA, chi-square etc. the value of pearson coeeficient lies between 1 and -1.

3.2.6 Machine learning models

A brief discussion on a number of machine learning models is carried in this section. Python as a tool is utilized for implementing the various machine learning models. Every model is presented in a very stepwise manner stating the commands executed in python as well.

Each of the model is trained on the basis of first four tool data and tested on the last two tool data.

3.3 REGRESSION MODELS

Results for each of the model is discussed in the following section of the chapter and the corresponding detailed program code for same can be found in the appendices.

3.3.1 Logistic Regression

<u>Multiple Linear Regression model</u>

Check Appendix A for entire code

The results obtained after fitting the multiple linear regression model is as follows:

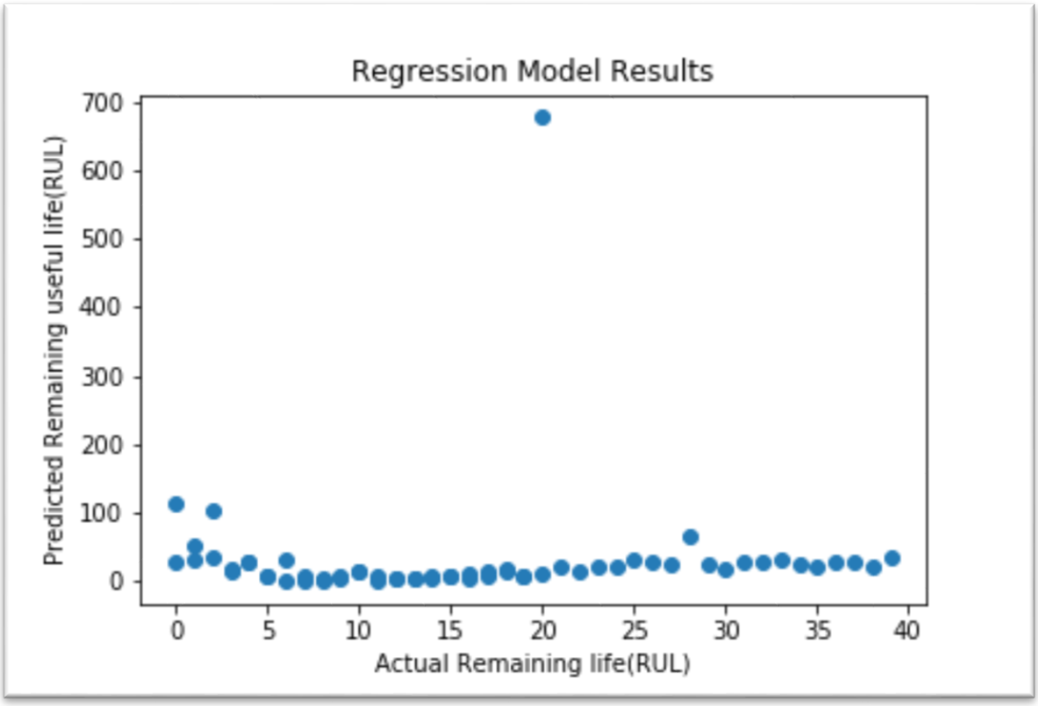


Figure 11 Multiple Linear Regression Results

The root mean square recorded for the same is 144.25

Table 1 Optimization for Multiple Linear Regression model

S. No.	PCA components	Root Mean Square Error
1	1	14.80352153
2	2	14.74470065
3	3	14.68492943
4	4	14.68940348
5	5	16.75085931
6	6	16.77168749
7	7	16.72766364
8	8	16.70004986
9	9	16.54790075

The most optimum value is obtained at PCA equal to 3 which reduced the RMSE value from 144.25 to 14.68.

The results obtained after fitting the multiple linear regression model along with PCA is as follows:

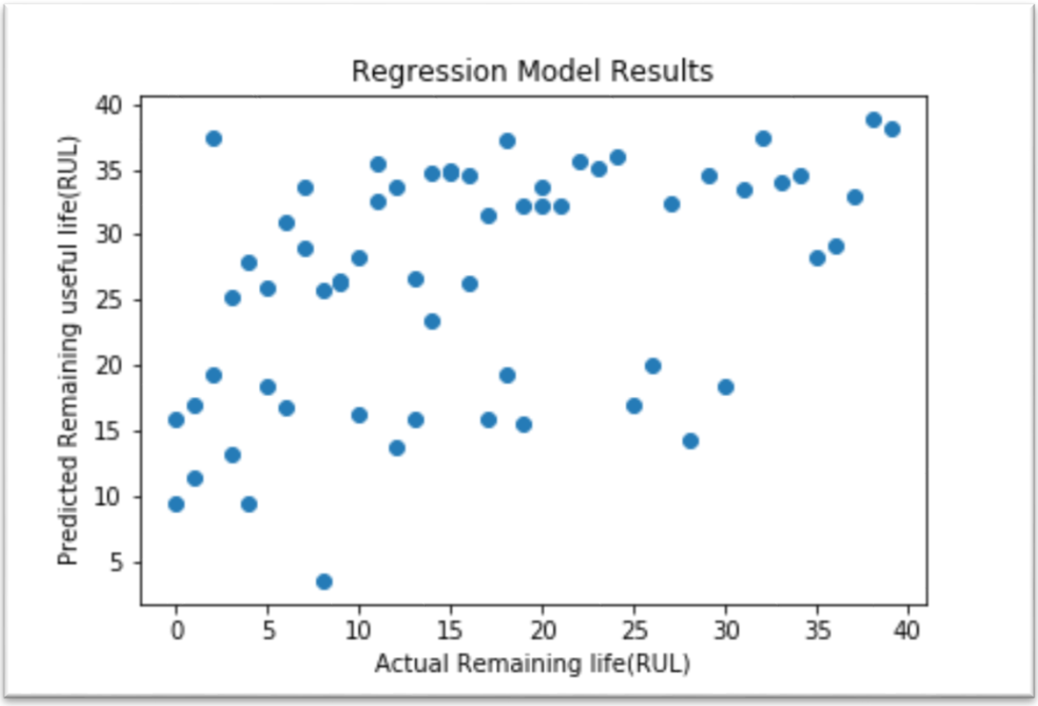


Figure 12 Multiple Linear Regression with PCA Results

As the scatter plot between actual RUL and predicted RUL is not resembling the ideal 45-degree line hence there is a scope to improve over our approach with the help of other algorithms.

<u>Polynomial Linear Regression model</u>

Check Appendix B for entire code

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \dots + b_n x_1^n$$

The governing equation for polynomial linear regression model is shown above. The given equation is still called linear because of the presence of constants. Each of the constant present above is linearly independent of each other.

Table 2 Optimization for Multiple Linear Regression model

S. No.	Input matrix	RMSE
1	1	149.64
2	2	4397
3	3	9734
4	4	7244

The results obtained reveal the nonexistence of nonlinear relationship between the data points due to the same reason the error goes exceedingly high with the increase in the degree of the polynomial.

Table 3 Optimization for Polynomial Linear Regression model with PCA

S.N0	Degree of input vector	Principle component	RMSE
1	1	1	14.80352153
2	1	2	14.74470065
3	1	3	14.68492943
4	1	4	14.68940348
5	1	5	16.7508593
6	1	6	16.77168746
40	2	1	13.7677071
41	2	2	13.83365406
42	2	3	15.70710413
43	2	4	15.65754061
44	2	5	15.69225961
45	2	6	15.61645184
79	3	1	13.6471629
80	3	2	13.78975036
81	3	3	13.47260456
82	3	4	13.39405344
83	3	5	13.6580942
84	3	6	13.65983337

The RMSE value decreased by only a minor amount with help of polynomial linear regression method with PCA.

3.3.2 Decision Tree Regression

Check Appendix C for entire code

Classification and Regression Trees or CART for short is a term. It is one of its unique type regression model which predicts values in steps. The results obtained after fitting the decision tree regression model with and without PCA is the figures below:

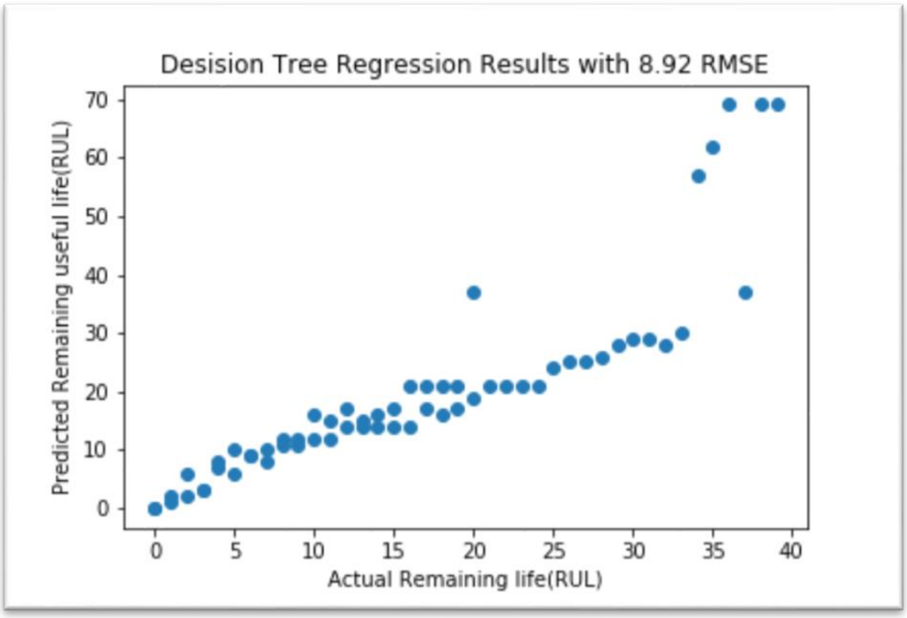


Figure 13 Decision Tree Regression without PCA

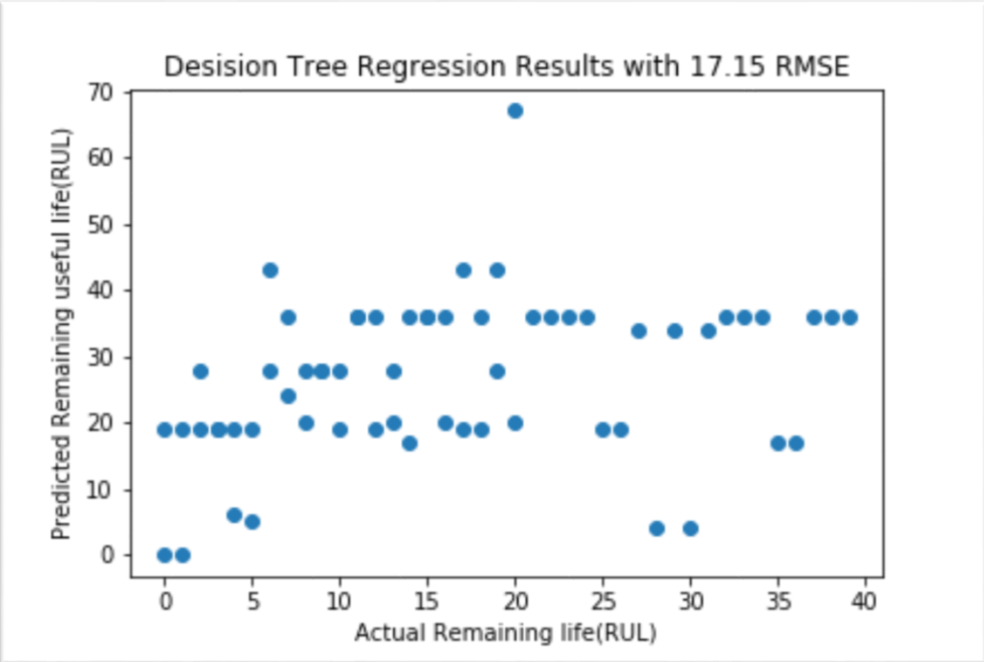


Figure 14 Decision Tree Regression with PCA

3.3.3 Random Forest Regression

Check Appendix D for entire code.

The RMSE value decreased considerably, after proper optimization of all possible hyperparameters the best RMSE. value obtained is equal to 6.57. The results degraded after applying the PCA as in the case of decision tree regression.

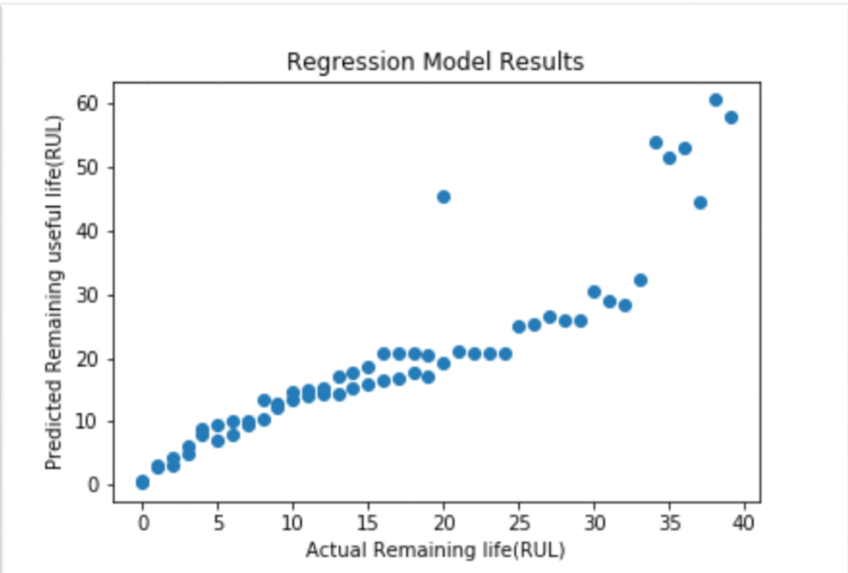


Figure 15 Random Forest Regression Results

3.3.4 DEEP LEARNING MODELS

After obtaining some benchmark values with the basic machine learning models now let's try to improve your results with the help of some deep neural networks.

3.3.4.1 Artificial Neural Network

Keras is a powerful easy-to-use Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in a few short lines of code. We have particularly worked on TensorFlow as one of the backend library for all our deep learning models. The best result obtained after optimizing a number of hyperparameters is shown in the table below. The RMSE value achieved in the most optimized model is recorded as 12.46.

Table 4 Optimization parameters for artificial neural network

Parameters	Value
Neurons in first layer	16
Neurons in second layer	17
Batch size	40
Epochs	500
RMSE	12.46

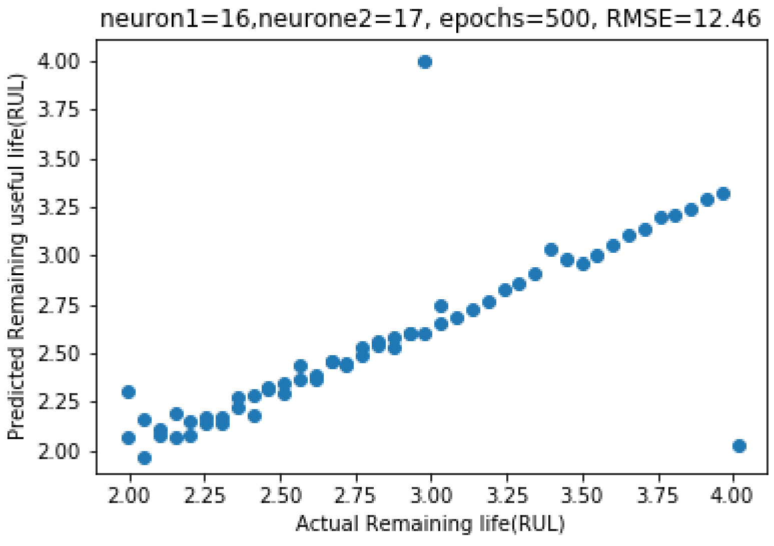


Figure 16 Artificial Neural Network Results

The results obtained from the deep neural network are not much improved compared to the conventional algorithms. Now we will try to fit a new algorithm which belongs to the same class of deep neural network known as recurrent neural network.

3.3.4.2 Long Short Term Memory (LSTM)

When we arrange our calendar for the day, we prioritize our appointments right? If in case we need to make some space for anything important we know which meeting could be canceled to accommodate a possible meeting.
Turns out that an RNN doesn't do so. In order to add a new information, it transforms the existing information completely by applying a function. Because of this, the entire information is modified, on the whole, i. e. there is no consideration for 'important' information and 'not so important' information.
LSTMs on the other hand, make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a cell state has three different dependencies. These dependencies are taken care by forget gate, output gate and input gate. Let's understand the basic architecture of a LSTM cell involving the mathematical operation occurring on each cell.

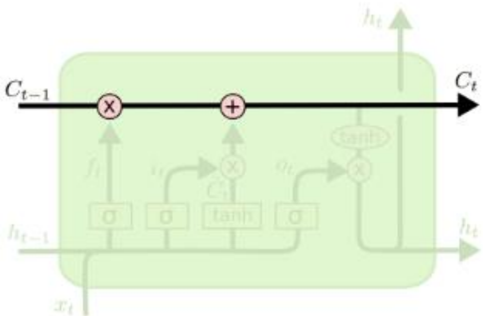
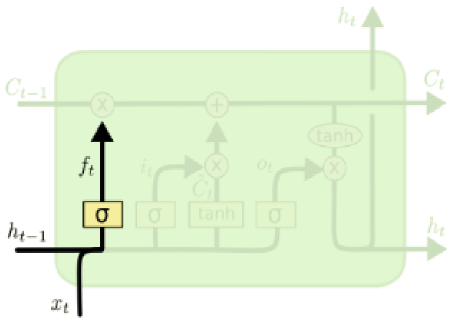


Figure 17 LSTM cell, cell state [50]

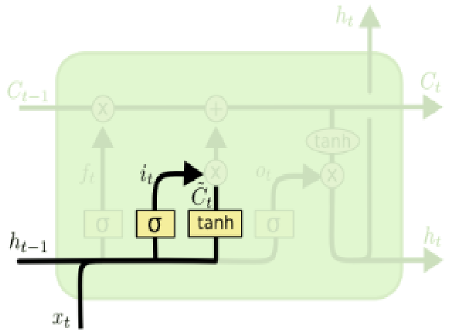
The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 18 Figure 19: LSTM Forget Gate [50]

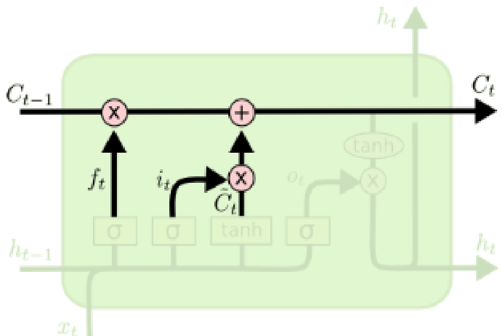
The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 19 LSTM cell, Input cell [50]

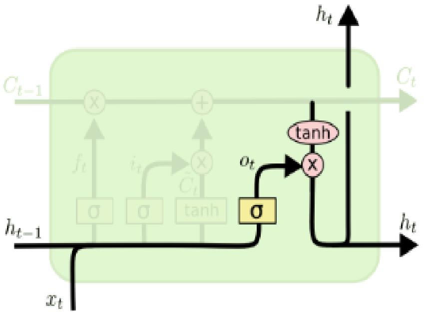
The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values, $C^{\sim}{}_t$, that could be added to the state.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 20 LSTM cell, updating the information [50]

It's now time to update the old cell state, Ct−1, into the new cell state Ct. The previous steps already decided what to do, we just need to do it.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 21 LSTM cell, Decision Step [50]

Results for LSTM

Several hyperparameters were optimized for LSTM model and most optimized results were obtained for following configuration.

Table 5 Optimization for LSTM model

Best configuration	
No. of layers	1
LSTM nodes	5
Batch Size	3
No. of epochs	980
RMSE	2.0455

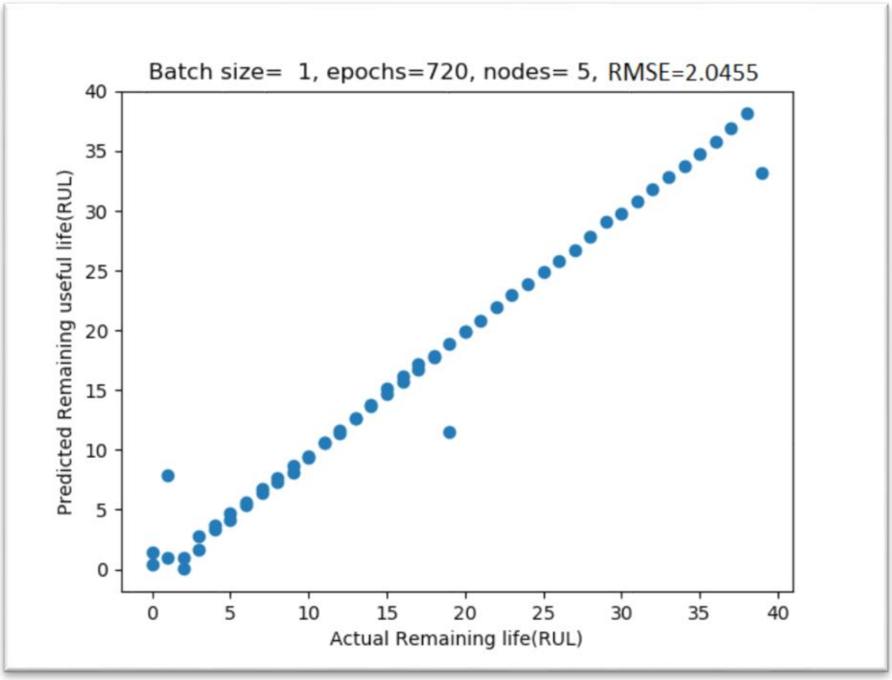


Figure 22 LSTM Results

3.4 SUMMARY

After implementing many regression models on given data set we reached to a conclusion different machine learning models results in varying results. Hence one cannot rely on given machine learning models for a given data set. We need to have some additional information along with the results predicted so that we can rely on the results of RUL predicted by a given model to a higher accuracy level. This brings us to the advent of our new approach regarding predicting the RUL discussed in the next chapter.

CHAPTER 4. DEVELOPING A NOVEL PROGNOSTIC MODEL

4.1 INTRODUCTION

Every prognostic model end results greatly depend on number of factors; one important factor is the data set which is input to the model. Due to the same reason data preprocessing step is the integral part of the prognostic model generation. More labelled and organized data often lead to the generation of better model. When we talk about any machine learning based models aiming to predict the Remaining Useful Life (RUL) we must know all model are wrong as they deviate from actual results in some or other way but some models are useful. Due to this inherent inaccuracy of the models we try to propose a novel methodology which aims to provide increase the reliability of resulting model and helps in storing the data in more organized fashion which further helps in improving the accuracy of the model and reducing the computational time.

4.2 EXPERIMENTAL SETUP

Milling operation is chosen is this research. EMCO MILL E350 CNC three-axis high-speed vertical milling machine is utilized as the test bed. Normal degradation of end milling cutting tool (Figure 7, 8) is carried out to study the degradation behavior of the cutting tool. A high-speed steel 6 mm milling tool is chosen for analysis. Experiment was carried out for a range of operating conditions which was used by varying feed, depth of cut and speed. The work piece used is of mild steel (figure 8, 10) of dimension 165mm x 100mm.
Six end milling tools run to failure data was generated for the following operating condition shown in table. During the machining, force, vibration and acoustic signal were constantly monitored during every cut of machining the mild steel plate for length of cut equal to 1320. The table below shows the brief summary of collecting along with the operating conditions:
E1 – Operating conditions
feed=250mm, speed=1050rpm, depth of cut=0.20

Table 6 **Summary of tools in E1 operating condition**

Cutter	Time (Cuts)	Failure Modes
1	27	Breakage
2	70	Worn Out
3	39	Breakage
4	40	Breakage
5	40	Breakage
6	21	Breakage

E2 – Operating conditions

feed=250mm, speed=1300rpm, depth of cut=0.35

Table 7 **Summary of tools in E2 operating condition**

Cutter	Time (Cuts)	Failure Modes
1	30	Worn Out
2	21	Breakage
3	16	Breakage
4	16	Breakage
5	19	Breakage
6	31	Worn Out

E3 – Operating conditions

feed=350mm, speed=1050rpm, depth of cut=0.20

Table 8 **Summary of tools in E3 operating condition**

Cutter	Time (Cuts)	Failure Modes
1	16	Breakage
2	20	Breakage
3	26	Breakage
4	23	Worn Out
5	19	Breakage
6	30	Worn Out

E4 – Operating conditions

feed=350mm, speed=1300rpm, depth of cut=0.20

Table 9 Summary of tools in E4 operating condition

Cutter	Time (Cuts)	Failure Modes
1	22	Worn Out
2	18	Breakage
3	42	Worn Out
4	37	Breakage
5	30	Breakage
6	29	Breakage

E5 – Operating conditions(force data only)

feed=300mm, speed=1000rpm, depth of cut=0.25

Table 10 Summary of tools in E5 operating condition

Cutter	Time (Cuts)	Failure Modes
1	66	Worn Out
2	63	Worn Out
3	35	Breakage
4	47	Breakage
5	71	Worn Out
6	52	Worn Out + Breakage

4.3 EXTRACTION OF FEATURES

Data collected for each sensor during machining is collected at a high frequency of 2000 hz. This makes sure that at the data collection step we should not miss out any important information in terms of varying signal signature trend during the machining process. Once machining is completed for every cut the data corresponded to every cut is reduced to data points for simplified processing of data. Number of different features were extracted from the data collected corresponding to each cut. Most of the features extracted are mean, standard deviation, kurtosis, range etc. Calculating these range of features helps in approximation the true distribution of signals which were generated during the process. In addition to that it helps in reducing the size of the dataset which further helps in saving the computational time during model generation.

4.4 NOVEL MODEL APPROACH

Six end milling tools were used in machining the mild steel plate till their end of life for each operating condition. Unlike the traditional machine learning model, we are not diving the data into two sets of training and testing despite we use data for training our Machine Learning (ML) model in a step wise manner as shown by the detailed description in figure. Every tool has a stochastic behavior in terms of its failure. Sensor signals captured for one failure mode may be different than the other. Prognostic models aim at predicting these failures due to different modes of failure well in advance before the actual failure. But if the model which is trained on data consisting of multiple failure modes leads to inaccuracies in the resulting models. As data corresponding to each failure mode acts like noise for data belonging to other failure mode. During our experiment we try to generate as many prognostic models as the number of failure mode. It is assumed that tool undergoes a different failure mode if the Machine Learning (ML) model based on past tools histories fails to predict the life of the tool accurately. So, for every tool we can have predictions generated from multiple prognostic models. Further these predictions about tool life are grouped into conservative approach or non-conservative approach. This freedom completely lies on the user end which value he/she want to rely on as per the current working condition and amount of risk involved at that time of machining. It is believed during the initial stage of machining the operator will prefer to go for the non-conservative and during the later part giving preference to conservative results as the risk involved towards the failure of tool increases at the end.

4.5 RANDOM FOREST REGRESSION

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it’s simplicity and the fact that it can be used for both classification and regression tasks. Due to the same reason to present this novel approach I have used random forest model for both classification and regression. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

With a few exceptions a random-forest classifier has all the hyperparameters of a decision-tree classifier and all the hyperparameters of a bagging classifier, to control the ensemble itself. Instead of building a bagging-classifier and passing it into a decision-tree-classifier, you can just use the random-forest classifier class, which is more convenient and optimized for decision trees. The random-forest algorithm brings extra randomness into the model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model. Therefore, when you are growing a tree in random forest, only a random subset of the features is considered for splitting a node. You can even make trees more random, by using random thresholds on top of it, for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

4.6 ARCHITECTURE OF PROPOSED PROGNOSTIC MODEL

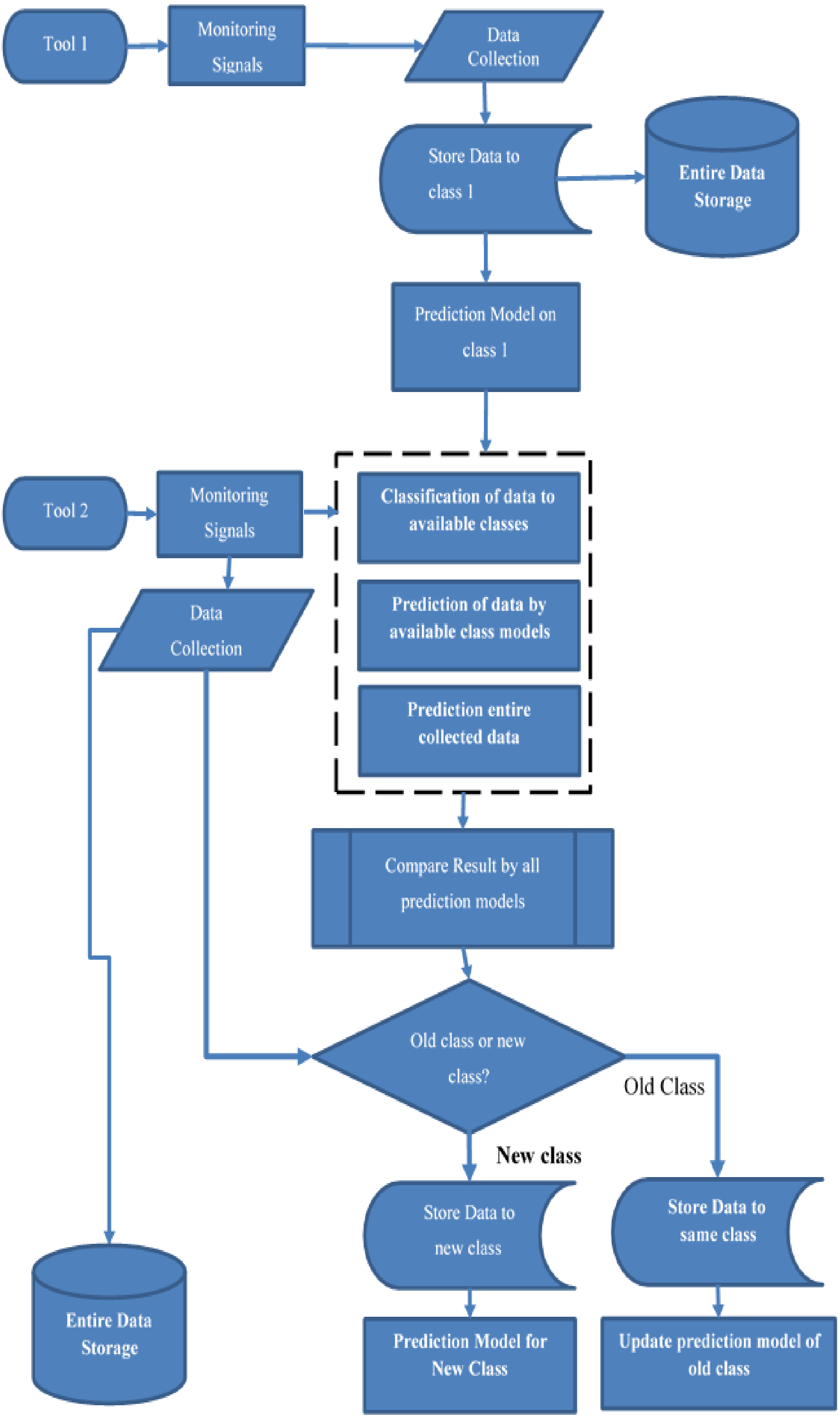


Figure 23 Flow Chart depicting the process step of novel prognostic approach

4.7 RESULTS AND CONCLUSION

Table 11 Results for E1 operating conditions

E1 – Operating conditions

S.No.	M-1	M-2	Conservative(min RUL)	Non-Conservative(max RUL	RUL based on RF classificatio	RUL based on LR classificatio	Conventional Mode
TOOL 2(70)	M1-TOOL-1						
RMSE	22.55						
RMSE 1st 50%	31.02						
RMSE 2nd 50%	7.42						
Tool 3(39)	M1-TOOL-1	M1-TOOL-2					
RMSE	7.497	10.56	7.365	10.66	10.3816	10.485	10.55
RMSE 1st 50%	10.242	14.75	10.39	14.752	14.752	14.705	15.08
RMSE 2nd 50%	2.481	3.323	1.79	3.8596	1.8497	2.9926	1.03
TOOL 4(40)	M1-TOOLS-1,3	M1-TOOL-2					
RMSE	2.212	12.524	2.194	12.527	9.539	8.237	11.02
RMSE 1st 50%	2.7105	17.08	2.7105	17.08	13.4004	10.704	15.53
RMSE 2nd 50%	1.5639	4.663	1.511	4.68	1.564	4.596	1.27
TOOL 5(40)	M1-TOOLS-1,3,4	M1-TOOL-2					
RMSE	1.829	11.79	1.822	11.797	2.9668	7.106	4.85
RMSE 1st 50%	1.23	15.48	1.237	15.481	3.527	9.644	6.45
RMSE 2nd 50%	2.27	6.215	2.2804	6.2191	2.2718	2.82621	2.35
TOOL 6(21)	M1-TOOLS-1,3,4,5	M1-TOOL-2					
RMSE	4.855	15.7456	4.8527	15.7463	10.847	4.855	4.719
RMSE 1st 50%	6.5536	19.5411	6.5536	19.54	14.7817	6.5536	6.3737
RMSE 2nd 50%	2.4396	11.23	2.43	11.234	5.09	2.4396	2.3655

4.7.1 Result analysis for table 10

- **<u>Predicting RUL for tool 2</u>**: The first model is trained as per the data of tool 1. The Root Mean Square (RMSE) results were obtained while predicting the life of tool 2. A very high value of RMSE value is obtained while predicting the Remaining Useful Life (RUL)of tool 2. At this stage based on the higher values of RMSE we assume that tool 2 belongs to a different class or in other words it undergoes a different failure mode compared to the failure mode of tool 1.
- **<u>Predicting RUL for tool 3</u>**: RUL of tool 3 is predicted based on two models i.e. M1 model (trained on tool 1 data) and M2 model (trained on tool 2 data). The results obtained have lower value of RMSE corresponding to M1 model. For further analysis of predicting models. Tool 3 is grouped among tool 1. Hence, subsequent predictions based on model M1 will be trained on combined data of tool 1 and tool 3.
- **<u>Predicting RUL for tool 4</u>**: Prediction for tool 3 showed significantly improved results based on model M1. The possible reason for improved results can be increase in the number of data points hence obtaining the better predicting model.
- **<u>Predicting RUL for tool 5</u>**: Based on model 1 which has data corresponding to tool 1,2 and 3 showed very good results registering the minimum RMSE value of 1.829. for prediction of subsequent tool model M1 data is updated like above cases.
- **<u>Predicting RUL for tool 6</u>**: Model M1 showed better results compared to model 2. But despite of increase number of data points the predicted RMSE is more compared to predictions made for tool 4. The possible reason for deviation is due to variation in failure mode of two tools.
- **<u>Overall analysis</u>**: At the end tool 1,3,4 and 5 are grouped in class 1 and tool 2 is grouped in class 2. Now comparing with table 6 data, we have actually grouped the data as per their failure modes. Lets apply the same approach on another data set for different operating conditions. Using multiple models for prediction gives us the option of considering conservative approach or non-conservative approach based on the criticality of the operation at that instant.

Table 12 Results for E2 operating conditions

Main Experimental Raw Data - E2

S.No.	M-1	M-2	Conservative(min RUL)	Non-Conservative(max RUL	RUL based on RF classificatio	RUL based on LR classificatio	Conventional Mode
TOOL 2(21)	M1-TOOL-1						
RMSE	11.39						
RMSE 1st 50%	7.89						
RMSE 2nd 50%	4.89						
Tool 3(16)	M1-TOOL-1	M2-TOOL-2					
RMSE	7.98	3.85	2.69	7.32	5.45	5.44	7.59
RMSE 1st 50%	9.74	2.94	3.18	9	6.64	6.62	8.72
RMSE 2nd 50%	5.69	2.89	2.097	5.12	3.92	3.92	6.25
TOOL 4(16)	M1-TOOL-1	M2-TOOL-2,3					
RMSE	7.93	4.988	4.37	7.94	8.09	7.26	4.75
RMSE 1st 50%	7.654	5.4	4.217	7.65	7.97	6.16	4.804
RMSE 2nd 50%	8.207	4.53	4.52	8.22	8.207	8.216	4.697
TOOL 5(19)	M1-TOOL-1	M2-TOOL-2,3,4					
RMSE	6.83	1.47	1.43	6.84	2.18	2.3	3
RMSE 1st 50%	7.36	1.76	1.75	7.36	2.92	2.92	4.107
RMSE 2nd 50%	6.32	1.3	1.06	6.33	1.16	1.54	1.5
TOOL 6(31)	M1-TOOL-1	M2-TOOL-2,3,4,5					
RMSE	4.45	8.95	8.99	4.38	4.92	4.92	6.89
RMSE 1st 50%	4.04	12.09	12.09	4.04	4.99	4.98	9.48
RMSE 2nd 50%	4.81	7.89	4.4	4.67	4.85	4.85	3.78

4.7.2 Result analysis for table 11

- **Predicting RUL for tool 2**: The first model is trained as per the data of tool 1. Higher values of RMSE values was recorded while predicting the Remaining Useful Life (RUL).Hence we group tool 2 in a new class.
- **Predicting RUL for tool 3:** RUL of tool 3 is predicted based on two models i.e. M1 model (trained on tool 1 data) and M2 model (trained on tool 2 data). The results obtained have lower value of RMSE corresponding to M2 model. For further analysis of predicting models. Tool 3 is grouped among tool 2. Hence, subsequent predictions based on model M1 will be trained on combined data of tool 2 and tool 3.
- **Predicting RUL for tool 4:** Prediction for tool 4 showed better results for model 2 compared to model 1. The RMSE values in first half and second half of tool life remained very close to each other which signifies the stability of model in both the stages.
- **Predicting RUL for tool 5:** Based on model 2 which has data corresponding to tool 2,3 and 4 showed very good results giving the minimum RMSE value of 1.47.
- **Predicting RUL for tool 6:** Model M1 showed better results compared to model 2. Hence tool 6 is classified to class 1
- **Overall analysis:** At the end tool 1 and 6 are grouped in class 1 and tools 2, 3, 4, 5 are grouped in class 2. The table 7 showed that tools 1 and 6 failed due to breakage hence as per our approach we have classified the models correctly based on their failure. Also obtained the improved RMSE value because of better collection of data in a grouped manner.

Table 13 Results for E3 operating conditions

Main Experimental Raw Data - E3

S.No.	M-1	M-2	Conservative(min RUL)	Non-Conservative(max RUL	RUL based on RF classificatio	RUL based on LR classificatio	Conventional Mode
TOOL 2(19)	M1-TOOL-1						
RMSE	2.147						
RMSE 1st 50%	2.804						
RMSE 2nd 50%	1.75						
Tool 3(25)	M1-TOOL-1,2						
RMSE	3.4						
RMSE 1st 50%	4.7						
RMSE 2nd 50%	0.93						
TOOL 4(22)	M1-TOOL-1,2,3						
RMSE	1.75						
RMSE 1st 50%	1.375						
RMSE 2nd 50%	2.034						
TOOL 5(19)	M1-TOOL-1,2,3,4						
RMSE	1.76						
RMSE 1st 50%	2.01						
RMSE 2nd 50%	1.55						
TOOL 6(29)	M1-TOOL-1,2,3,4,5						
RMSE	4.67						
RMSE 1st 50%	6.36						
RMSE 2nd 50%	1.81						

4.7.3 Result analysis for table 12

The present case the operating condition is comparatively different from earlier cases due to higher magnitude of feed. In each of the case the RMSE value obtained has a smaller magnitude. Despite the combination of both types of failures like in the case of E1 and E2, model failed to classify them to different groups of classes. The average life of tool in this group is only 22.33 with a standard deviation of 4.64. The possible reason for this may be the closeness in mode of failures in both the cases. As the life of the tool is nearly same in both the cases the model fails to classify them as separate group. Practically a worn-out tool should have a relatively higher life compared to breakage failure. In this present case we have four breakage case and 2 worn out each having nearly same life hence all have been grouped in same class.

Table 14 Results for E4 operating conditions

Main Experimental Raw Data - E4							
S.No.	M-1	M-2	Conservative(min RUL)	Non-Conservative(max RUL	RUL based on RF classificatio	RUL based on LR classificatio	Conventional Mode
TOOL 2(17)	M1-TOOL-1						
RMSE	2.86						
RMSE 1st 50%	3.13						
RMSE 2nd 50%	2.615						
Tool 3(41)	M1-TOOL-1,2						
RMSE	12.65						
RMSE 1st 50%	16.89						
RMSE 2nd 50%	5.92						
	M1-TOOL-1,2	M2-TOOL-3					
TOOL 4(36)							
RMSE	9.72	3.85	9.72	3.84	6.98	5.92	4.73
RMSE 1st 50%	13.28	3.604	13.28	3.6	9.03	7.12	5.89
RMSE 2nd 50%	4.11	4.07	4.13	4.06	4.18	4.52	3.27
TOOL 5(29)	M1-TOOL-1,2	M2-TOOL-3,4					
RMSE	6.62	4.49	6.62	4.5	4.68	4.53	3.66
RMSE 1st 50%	8.814	6.18	8.81	6.18	6.36	6.18	4.64
RMSE 2nd 50%	3.17	1.48	3.17	1.48	1.831	1.69	2.3
TOOL 6(28)	M1-TOOL-1,2	M2-TOOL-3,4,5					
RMSE	4.34	6.87	4.34	6.87	5.15	6.87	4.74
RMSE 1st 50%	5.84	7.58	5.84	7.58	6.16	7.58	5.45
RMSE 2nd 50%	2.154	6.14	2.15	6.14	4	6.14	3.96

4.7.4 Result analysis for table 13

- **<u>Predicting RUL for tool 2</u>**: The first model is trained as per the data of tool 1. lower values of RMSE values was recorded while predicting the Remaining Useful Life (RUL).Hence we group tool 1 in the same class.
- **<u>Predicting RUL for tool 3</u>:** As no classes has been formed so far the predicted values of RUL are based on class 1 only. The results obtained have higher value of RMSE. Because of the higher magnitude of RMSE values for the predicted RUL, tool 3 is grouped into the new class hence further tool's life will be predicted based on two models.
- **<u>Predicting RUL for tool 4</u>:** Prediction for tool 4 showed better results for model 2 compared to model 1. The RMSE values in first half and second half of tool life remained very close to each other which signifies the stability of model in both the stages.
- **<u>Predicting RUL for tool 5</u>:** Tool 5 predicted values for RUL do not show much difference in RUL of both the predicted models. The predicted values by M2 model were more close to the actual model hence it has been grouped to class 2.
- **<u>Predicting RUL for tool 6</u>:** Model M1 showed better results compared to model 2. Hence tool 6 is classified to class 1

<u>Overall analysis</u>: At the end tool 1 and 2 are grouped in class 1 and tools 3, 4, 5 and 6 are grouped in class 2. The table 9 showed that tools 1 and 3 failed due to worn-out, while other tools are grouped in breakage. The tool 3 is incorrectly classified tool 1 had life of 22 cuts while tool 3 had life of 42 cuts. The deviation in the results is possible due to this large magnitude of difference in life of both the tools.

Table 15 Results for E5 operating conditions(force data)

Main Experimental Raw Data - Force

S.No.	M-1	M-2	Conservative(min RUL)	Non-Conservative(max RUL	RUL based on RF classificatio	RUL based on LR classificatio	Conventional Mode
TOOL 2(63)	M-1						
RMSE	6.12						
RMSE 1st 50%	4.91						
RMSE 2nd 50%	7.1						
Tool 3(35)	M-1,2	,					
RMSE	29.22						
RMSE 1st 50%	28.697						
RMSE 2nd 50%	29.71						
TOOL 4(47)	M-1,2	M-3					
RMSE	17.84	10.52	17.84	10.53	10.527	17.84	12.138
RMSE 1st 50%	17.93	12.314	17.93	12.314	12.314	17.93	15.314
RMSE 2nd 50%	17.76	8.47	17.76	8.47	8.469	17.76	7.986
TOOL 5(70)	M-1,2	M-3,4					
RMSE	6.26	21.51	21.55	6.124	11.95	14.02	11.05
RMSE 1st 50%	6.82	29.29	29.29	6.82	16.02	18.86	14.51
RMSE 2nd 50%	5.68	8.86	9.06	5.36	5.68	6.46	6
TOOL 6(52)	M-1,2,5	M-3,4					
RMSE	13.56	6.12	6.123	13.576	11.76	12.24	9.11
RMSE 1st 50%	15.41	8.307	8.307	15.41	12.32	13.69	11.53
RMSE 2nd 50%	11.41	2.47	2.34	11.44	11.17	10.6	5.73

4.7.5 Result analysis for table 14

- **Predicting RUL for tool 2**: The model was based on tool 1 data, the predictions made for tool 2 were better predicted in the second half of life of the tool. The average RMSE value for tool life is on lower hence tool 2 has been group to class 1 which was earlier consisting of tool 1 only.
- **Predicting RUL for tool 3:** at this stage we have only one class model M1, hence predictions were made only based on model M1. The predicted values for life of the tool 3 deviate too much from its actual life. This can be verified from the RMSE values obtained which went over 20.
- **Predicting RUL for tool 4:** The prediction made for tool 4 was based on two models i.e., M1 and M2. The RMSE value obtained from either model is having a high magnitude. But comparatively lower magnitude of RMSE value is obtained by model M2 hence tool 4 is grouped in class 2
- **Predicting RUL for tool 5:** The predicted values obtained by model M2 deviated by large magnitude. So compared to model M2 and conventional model results the model M1 predicted the RUL quiet well showing less deviation from the actual life of the tool.
- **Predicting RUL for tool 6:** Compared to conventional model and model M1, the predictions made by model M2 are far better showing a lower magnitude of RMSE values.

Overall analysis: At the end of our analysis the we got two classes. Class 1 consists of tools1,2 and 5 and class 2 consists of tools 3, 4, 6. As per the table 10 the tools 1,2 and 5 are the cases of worn-out and as per our model they have been grouped in the same class. Rest of the tools have been grouped in another class. The last tool has mixed information about its mode of failure.

The tool 6 has its life comparable to the tools belonging to the case of breakage. This is the possible reason it has been grouped in the class belonging to breakage failures.

CHAPTER 5. CONCLUSIONS

5.1 SUMMARY

The main aim of this thesis was to develop a reliable prognostic model for predicting the Remaining Useful Life (RUL) of the tool. The thesis has discussed the data driven prognostics approaches for prognostics of the tool life. As discussed in the literature review there are number of prognostics models have been proposed for predicting the life of the tool based on the past data of the tool but still these prognostic models lacks the desired reliability in the predictions made by these model. The key role of this thesis is to develop robust algorithm capable of producing results which are more reliable. The thesis has aimed to fulfill the following objective as per the gaps discussed in the literature review

1. Developing the different machine learning models to identify which model works the best for our data set generated from end milling machining.
2. Making use of LSTM for Tool condition monitoring. So far LSTM models have been widely used in the field of the speech recognition. We have tried to utilize the same for predicting the remaining useful life of the tool.
3. Proposing the novel methodology for data collection techniques aiming at keeping the data collected in their distinct groups.
4. Generalizing the prognostic methodology for any new machine.

5.2 CONTRIBUTION OF THE THESIS

Keeping the above objective in mind the following contribution is made as the result of this thesis.

1. Every machine learning algorithm does not work well on all set of data without optimization. There are number of hyperparameters which must be tuned as per the demand. A given prognostic model may work well with given data set but the same prognostic model may not work well with the data set obtained as a result from other operating conditions. Different data driven approaches were used on the past data set of multiple tools which were operated on same operating conditions. It was

realized that a varying accuracy is obtained even after trying number of methods of optimization for each of the model.

2. One of the trending machine learning model used in speech recognition is Long Short-Term Memory(LSTM). It finds great application due to its advantage over other machine learning models due to its capability of considering the sequence of data which acts as an advantage for this model compared to another conventional model. LSTM is grouped among the deep learning models. Artificial neural network(ANN) are the most famous model in this group due its diverse applicability in number of application. ANN also consider the data as discrete points but LSTM outperforms the results compared to ANN by its sequence tracking capability.
3. In our thesis we have tried to propose a novel methodology to keep our data in groups. A given failure mode data is grouped into a certain class while data corresponding to other failure mode is grouped into another class. This segregation of data at the very prior stage helps in improving the efficiency of the model to a large extent as model aiming to make correct prediction is not misguided by some unimportant data or noise. This approach further helps in improving the computational time for running a given prognostic model. also it saves lot of time of data analyst during the data preprocessing stage of prediction models. The storing of data gets handy and data stored is easy to interpret as it is more labelled compared to the conventional strategy of storing the data.
4. The prime demand of data driven approaches is the past data. This is often a problem as we must run a machine ideal just for the sake of generating data so that some machine learning models could be trained based on that data to make predictions. In our methodology we suggest a methodology that we should keep our systems ready to fetch the data from the machine from the very early stage and processing them. This has two objectives to serve. First the data collection for the machine is carried out in a more structured form and secondly with time due to the aggregation of more and more data better predictions can be made about the life the tool.

5.3 FUTURE SCOPE

The prognostics of remaining useful life(RUL) of the machine tool component based on machine learning model has wide scope of trying new methodology for better results. The work can be extended further by bringing in better and robust machine learning models to make predictions. In our approach we have introduced our methodology by the help of conventional machine learning model though better prediction were obtained by deep learning models during the early part of the project. It is believed as per the results of deep learning models that we could have obtained better results with the proposed novel approach. The reason behind not choosing the deep learning models like LSTM for our approach is the difficulty in optimization and higher computational time required to arrive at the optimized hyperparameters.

APPENDICES

A. Multiple Linear Regression Model

```
# Multiple linear Regression
#Importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from math import sqrt

#Importing the data set
import tkinter as tk
from tkinter import filedialog
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
print(file_path)
dataset = pd.read_csv(file_path, index_col=0)
dataset.head(5)

#Splitting into input and output
X = dataset.iloc[:, :-1]
Y = dataset.iloc[:, -1]

X.head(5)
Y.head(5)

# Choose the spliting Method

# 1. For ordered sequence
#Split into input and output
X_train = X.iloc[:176,: ]   #this is stored as matrix
Y_train = Y.iloc[:176, ]    #this is stored as vector
X_test = X.iloc[176:, :]
Y_test = Y.iloc[176:,]

# 2. For shuffuled case
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)
```

```
#Fitting the model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model = model.fit(X_train, Y_train)

#Predicting the Test set results
Y_pred = abs(model.predict(X_test))

# Applying K-Fold Validation
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = model, X=X_train, y=Y_train,
                      cv=10, scoring = 'neg_mean_squared_error')

accuracy_model_rmse = sqrt(abs(accuracies.mean()))
accuracy_model_rmse

# variances = sqrt(abs(accuracies))
list =[]
for i in range(0, 10 , 1):
    list.append(sqrt(abs(accuracies[i])))
    pass
std = sum(list)/(len(list)+1)
std

# Comparing the predicted Results with Actual Results
#Scatter plot
plt.scatter(Y_test, Y_pred)
plt.xlabel("Actual Remaining life(RUL)")
plt.ylabel("Predicted Remaining useful life(RUL)")
plt.title("Regression Model Results")
plt.savefig('Regression Model Results.png')
plt.show()

# Model Summary
#RMSE calculation
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse = sqrt(mean_squared_error(Y_pred, Y_test ))
rmse
```

B. Polynomial Linear Regression Model

```
#Importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import time

#Importing the data set
import tkinter as tk
from tkinter import filedialog
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
print(file_path)
dataset = pd.read_csv(file_path)
dataset.head(5)

#Splitting into input and output
X = dataset.iloc[:, :-1]
Y = dataset.iloc[:, -1]
#Split into input and output
X_train = X.iloc[:176,: ]   #this is stored as matrix
Y_train = Y.iloc[:176, ]    #this is stored as vector
X_test = X.iloc[176:, :]
Y_test = Y.iloc[176:,]
Result =[('Degree of input vector','Principle component','RMSE')]
for i in range(4, 6, 1):
   #Transforming X_train and X_test to polynomial expressions
   from sklearn.preprocessing import PolynomialFeatures
   poly_reg = PolynomialFeatures(degree=i)
   X_poly_train =poly_reg.fit_transform(X_train)
   X_poly_test = poly_reg.fit_transform(X_test)

   for j in range(1, 40, 1):
        # Appling PCA
       from sklearn.decomposition import PCA
       pca = PCA(n_components = j)  # n_compoent is set to none so that we can
check how varaince is explained by top features extracted
       X_train = pca.fit_transform(X_poly_train)
       X_test = pca.fit_transform(X_poly_test)
```

```
#Fitting the model polynomial model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, Y_train)

#Predicting the Test set results
Y_pred = abs(model.predict(X_test))

#RMSE calculation
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse = sqrt(mean_squared_error(Y_pred, Y_test ))

Result.append((i, j, rmse))
import pandas as pd
PCA_results = pd.DataFrame(Result)

#Scatter plot
plt.scatter(Y_test, Y_pred)
plt.xlabel("Actual Remaining life(RUL)")
plt.ylabel("Predicted Remaining useful life(RUL)")
plt.title("Regression Model Results")
plt.savefig('Regression Model Results.png')
plt.show()
```

C. Decision Tree regression

```
# Decission tree Regression
#Importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import xlwt
from sklearn.decomposition import PCA
from sklearn.tree import DecisionTreeRegressor
from math import sqrt
from sklearn.metrics import mean_squared_error

#Importing the data set
#Imprt the dataset on which prediction is to be done
import tkinter as tk
from tkinter import filedialog
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
print(file_path)
dataset = pd.read_csv(file_path)
dataset.head(5)

#Splitting into input and output
X = dataset.iloc[:, :-1]
Y = dataset.iloc[:, -1]

#Split into input and output
X_train = X.iloc[:176,: ]   #this is stored as matrix
Y_train = Y.iloc[:176, ]    #this is stored as vector
X_test = X.iloc[176:, :]
Y_test = Y.iloc[176:,]

# Making an spread sheet for storing results of parameter optimization
wb = xlwt.Workbook()
ws = wb.add_sheet('sheet 1')
ws.write(0, 0, "S.no")
ws.write(0, 1, "Degree PCA")
ws.write(0, 2, "root mean square error")
count = 1
```

```
# Appling PCA
from sklearn.decomposition import PCA
from sklearn.tree import DecisionTreeRegressor
from math import sqrt
from sklearn.metrics import mean_squared_error
for degree in range(1, 60, 1):
    X_train = X.iloc[:176,: ]
    X_test = X.iloc[176:, :]
    pca = PCA(n_components = degree)
    X_train = pca.fit_transform(X_train)

    pca = PCA(n_components = degree)
    X_train = pca.fit_transform(X_train)
    X_test = pca.fit_transform(X_test)
    regressor = DecisionTreeRegressor(random_state = 0)
    regressor.fit(X_train, Y_train)

    # Making predictions
    Y_pred = regressor.predict(X_test)
    rmse = sqrt(mean_squared_error(Y_pred, Y_test ))

    # writing data to excel sheet
    z = 0
    ws.write(count, z, count)
    z += 1
    ws.write(count, z, degree)
    z += 1
    ws.write(count, z, rmse)
    wb.save('Decission_tree_results.csv')
    count += 1
    pass

pca = PCA(n_components = 6)
X_train = pca.fit_transform(X_train)
X_test = pca.fit_transform(X_test)
pca = PCA(n_components = 6)
X_train = pca.fit_transform(X_train)
X_test = pca.fit_transform(X_test)
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X_train, Y_train)
```

```
# Making predictions
Y_pred = regressor.predict(X_test)
rmse = sqrt(mean_squared_error(Y_pred, Y_test ))
rmse

# Fitting the decission tree regressor to the training data set
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X_train, Y_train)

#RMSE calculation
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse = sqrt(mean_squared_error(Y_pred, Y_test ))
rmse
```

D. Random Forest Regression

```
# Random Forest Classification

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the data set
import tkinter as tk
from tkinter import filedialog
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
print(file_path)
dataset = pd.read_csv(file_path, index_col=0)
dataset.head(5)

# Split the dataset into input and output
X = dataset.iloc[:, :-1]
Y = dataset.iloc[:, -1]

#Split into input and output
X_train = X.iloc[:176,: ]   #this is stored as matrix
Y_train = Y.iloc[:176, ]    #this is stored as vector
X_test = X.iloc[176:,:]
Y_test = Y.iloc[176:,]

X_train = X_train.values
Y_train = Y_train.values
X_test = X_test.values
Y_test = Y_test.values

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
# Fitting tha Random Forest Classification
from sklearn.ensemble import RandomForestClassifier
model_class = RandomForestClassifier(n_estimators = 8, criterion =
'entropy', random_state =0)
model_class.fit(X_train, Y_train)

# Prediction
Y_pred = model_class.predict(X_test)

pd.DataFrame(Y_pred).to_csv('random_forest_prediction_results.csv')

# calculate RMSE
rmse = sqrt(mean_squared_error(Y_pred, Y_test))
rmse
```

F. Artificial Neural Network

```
# Neural Network - Regression Model

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import StandardScaler

# Importing the data set
import tkinter as tk
from tkinter import filedialog
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
print(file_path)
dataset = pd.read_csv(file_path, index_col=0)
dataset.head(5)

dataset = dataset.values

#Splitting into input and output
X = dataset[:,:-1]
Y = dataset[:,-1]

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)

#Split into input and output
X_train = X[:176,: ]   #this is stored as matrix
Y_train = Y[:176, ]    #this is stored as vector
X_test = X[176:, :]
Y_test = Y[176:,]

#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

```
# define base model
def baseline_model():
	# create model
	model = Sequential()
	model.add(Dense(34, input_dim=68, kernel_initializer='normal',
activation='relu'))
	model.add(Dense(1, kernel_initializer='normal'))
	# Compile model
	model.compile(loss='mean_squared_error', optimizer='adam')
	return model

# fix random seed for reproducibility
from keras.wrappers.scikit_learn import KerasRegressor
seed = 7
np.random.seed(seed)
# evaluate model with standardized dataset
estimator = KerasRegressor(build_fn=baseline_model, epochs=200, batch_size=5,
verbose=0)

from sklearn.model_selection import KFold
from sklearn.pipeline import Pipeline
from sklearn.model_selection import cross_val_score
kfold = KFold(n_splits=10, random_state=seed)
results = cross_val_score(estimator, X_train, Y_train, cv=kfold)
print("Results: %.2f (%.2f) MSE" % (results.mean(), results.std()))

# Fitting the ANN to training set
model.fit(X_train, Y_train, batch_size =10  , nb_epoch =100)

# Prediction
Y_pred = model.predict(X_test)

#RMSE calculation linear model
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse = sqrt(mean_squared_error(Y_pred, Y_test ))
rmse
```

LSTM

```
# LSTM
from math import sqrt
from numpy import concatenate
import numpy
from matplotlib import pyplot
import matplotlib.pyplot as plt
from pandas import *
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense, LSTM
import xlwt
import os
import os.path
from sklearn.model_selection import train_test_split
# Importing Data Set
import tkinter as tk
from tkinter import filedialog
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
print(file_path)
dataset = pd.read_csv(file_path)
dataset.head(5)
dataset = dataset.values
```

```
# convert series to supervised learning
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j + 1, i)) for j in range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j + 1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j + 1, i)) for j in range(n_vars)]
    # put it all together
    agg = concat(cols, axis=1)
    agg.columns = names
    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)
    return agg
```

```
values = dataset.astype('float32')

# Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)

# frame as supervised learning
reframed = series_to_supervised(scaled, 1, 1)  #last digit is look back

# drop columns we don't want to predict
list_drop = []
y = values.shape[1]
n_in = 1
k = n_in*y
print(k)
print(k+y)
print(list)
```

```
for i in range(y-1):
    list_drop.append(k+1+i)
reframed.drop(reframed.columns[list_drop], axis=1, inplace=True)
print(reframed.head())
reframed.to_csv('endmilling_data.csv')
```

```
# Setting up excel sheet for data collection
wb = xlwt.Workbook()
ws = wb.add_sheet('sheet 1')
ws.write(0, 0, "S.no")
ws.write(0, 1, "LSTM_nodes")
ws.write(0, 2, "batch_size")
ws.write(0, 3, "epochs")
ws.write(0, 4, "root mean square error")
count = 1

for LSTM_node in range(5, 1000, 5):
    for batch_size in range(10,180,10):
        for epochs in range(100,500,50):

            # split into train and test sets
            values = reframed.values
            # values = np.random.permutation(values)
            # in case of time series data ...uncmmenting the above
comment does not make any sence
            tool_4 = 175
            train = values[:tool_4, :]
            test = values[tool_4:238, :]

            # split into input and outputs
            train_X, train_y = train[:, :-1], train[:, -1]
            print("train_X", train_X)
            print("train_y", train_y)
            test_X, test_y = test[:, :-1], test[:, -1]
```

REFERNCES

[1] G. Abdulnour, R. A. Dudek, and M. L. Smith, *Effect of maintenance policies on the just-in-time production system*, International Journal of Production Research 33 (1995) 565-583.

[2] J. Shuler, Uptime: The magazine for PdM & CBM professionals June (2006).

[3] M. Roemer and C. Byington, *Session 1: Introduction to PHM/CBM design & deployment process analysis*, Proceedings of PHM/CBM Workshop and User's Forum, Miami Beach, Florida, 2007, pp.

[4] Prognostics: The Science of Making Predictions, K. Goebel, M. Daigle, S. Sankararaman, A. Saxena, I. Roychoudhury, J. Celaya, 2017, CreateSpace Independent Publishing Platform.

[5] Vachtsevanos; Lewis, Roemer; Hess, and Wu (2006). Intelligent fault Diagnosis and Prognosis for Engineering Systems. Wiley. ISBN 0-471-72999-X.

[6] Nasser L., Tryon R., Integration of material-based simulation into prognosis architectures, Aerospace Conference, Proceedings. 2004 IEEE, Vol. 6, pp 3742 – 3747, 2004.

[7] Provan G., An open systems architecture for prognostic inference during condition-based monitoring, Aerospace Conference, Proceedings, IEEE Vol. 7, pp 3157 – 3164, 2003.

[8] Š. Beneš, E. Tomeh, Methods of Diagnosis of Rolling Bearings, VŠST Liberec, 1991, pp. 55.

[9] http://www.endmill.com/ pages/training/design.html.

[10] Javed K, Gouriveau R, Li X, Zerhouni N (2016) Tool wear monitoring and prognostics challenges: a comparison of connectionist methods toward an adaptive ensemble model. J Intell Manuf 30(4):1–18

[11] Vetrichelvan G, Sundaram S, Kumaran S, Velmurugan P (2014) An investigation of tool wear using acoustic emission and genetic algorithm. J Vib Control 21(15):3061–3066.

[12] Bhattacharyya P, Sengupta D, Mukhopadhyay S (2007) Cutting force-based real-time estimation of tool wear in face milling using a combination

of signal processing techniques. Mech Syst Signal Process 21(6):2665–2683.

[13] Liu C, Wang G, Li Z (2015) Incremental learning for online tool condition monitoring using ellipsoid ARTMAP network model. App Soft Comput 35:186–198.

[14] Aliustaoglu C, Ertunc H, Ocak H (2009) Tool wear condition monitoring using a sensor fusion model based on fuzzy inference system. Mech Syst Signal Process 23(2):539–546.

[15] Konstantinos S, Athanasios K (2014) Reliability assessment of cutting tool life based on surrogate approximation methods. Int J Adv Manuf Technol 71(5–8):1197–1208.

[16] Karandikar J, Mcleay T, Turner S, Schmitz T (2015) Tool wear monitoring using naïve bayes classifiers. Int J Adv Manuf Technol 77(9–12):1613–1626.

[17] Rehorn A, Jiang J, Orban P (2005) State-of-the-art methods and results in tool condition monitoring: a review. Int J Adv Manuf Technol 26(7–8):693–710

[18] Dutta S, Kanwat A, Pal S, Sen R (2013) Correlation study of tool flank wear with machined surface texture in end milling. Measurement 46(10):4249–4260.

[19] Ghosh N, Ravi Y, Patra A, Mukhopadhyay S, Paul S, Mohanty A (2007) Estimation of tool wear during CNC milling using neural network-based sensor fusion. Mech Syst Signal Process 21(1):466–479.

[20] Drouillet C, Karandikar J, Nath C, Journeaux A, Mansori M, Kurfess T (2016) Tool life predictions in milling using spindle power with the neural network technique. J Manuf Process 22:161–168.

[21] Visariya, Ramesh & Ruparel, Ronak & Yadav, Rahul. (2018). Review of Tool Condition Monitoring Methods. International Journal of Computer Applications. 179. 29-32. 10.5120/ijca2018916853.

[22] Nouri M, Fussell B, Ziniti B, Linder E (2015) Real-time tool wear monitoring in milling using a cutting condition independent method. Int J Mach Tools Manuf 89:1–13.

[23] Azmi A (2015) Monitoring of tool wear using measured machining forces and neuro-fuzzy modeling approaches during machining of GFRP composites. Adv Eng Softw 82:53–64

[24] Zhang H, Zhao J, Wang F, Li A (2015) Cutting forces and tool failure in high-speed milling of titanium alloy tc21 with coated carbide tools. J Eng Manuf 229(1):20–27

[25] Wang M, Wang J (2012) CHMM for tool condition monitoring and remaining useful life prediction. Int J Adv Manuf Technol 59(5–8):463–471

[26] Huang P, Ma C, Kuo C (2015) A PNN self-learning tool breakage detection system in end milling operations. Appl Soft Comput 37:114–124

[27] Kaya B, Oysu C, Ertunc H (2011) Force-torque based on-line tool wear estimation system for CNC milling of inconel 718 using neural networks. Adv Eng Softw 42(3):76–84

[28] Prickett P, Johns C (1999) An overview of approaches to end milling tool monitoring. Int J Mach Tools Manuf 39(1):105–122

[29] Koike R, Ohnishi K, Aoyama T (2016) A sensorless approach for tool fracture detection in milling by integrating multi-axial servo information. CIRP Ann Manuf Technol 65(1):385–388

[30] Ghani J, Rizal M, Nuawi M, Ghazali M, Haron C (2011) Monitoring online cutting tool wear using low-cost technique and user-friendly GUI. Wear 271(9–10):2619–2624

[31] Stavropoulos P, Papacharalampopoulos A, Vasiliadis E,Chryssolouris G (2016) Tool wear predictability estimation in milling based on multi-sensorial data. Int J Adv Manuf Technol 82(1–4):509–521

[32] Sevilla P, Robles J, Jauregui J, Jimenez D (2015a) FPGA-based reconfigurable system for tool condition monitoring in high-speed machining process. Measurement 64:81–88

[33] Sevilla P, Robles J, Muñiz J, Lee F (2015b) Tool failure detection method for high-speed milling using vibration signal and reconfigurable bandpass digital filtering. Int J Adv Manuf Technol 81(5–8):1–8

[34] Sevilla P, Jauregui J, Herrera G, Robles J (2013) Efficient method for detecting tool failures in high-speed machining process. J Eng Manuf 227(4):473–482

[35] Cuka B, Kim D (2017) Fuzzy logic based tool condition monitoring for end-milling. Robot Comput Integr Manuf 47(10):22–36

[36] Wang G, Yang Y, Zhang Y, Xie Q (2014) Vibration sensor based tool condition monitoring using ν, support vector machine and locality preserving projection. Sensors Actuators A Phys 209: 24–32

[37] Zhou Y, Liu X, Li F, Sun B, Xue W (2015) An online damage identification approach for numerical control machine tools based on data fusion using vibration signals. J Vib Control 21(15):2925–2936

[38] Chen B, Chen X, Li B, He Z, Cao H, Cai G (2011) Reliability estimation for cutting tools based on logistic regression model using vibration signals. Mech. Syst. Signal Process 25(7):2526–2537

[39] Zhang C, Yao X, Zhang J, Jin H (2016) Tool condition monitoring and remaining useful life prognostic based on a wireless sensor in dry milling operations. Sensors 16(795):1–20

[40] Hsieh W, Lu M, Chiou S (2012) Application of backpropagation neural network for spindle vibration-based tool wear monitoring in micro-milling. Int J Adv Manuf Technol 61(1–4):53–61

[41] Madhusudana C, Kumar H, Narendranath S (2016) Condition monitoring of face milling tool using k-star algorithm and histogram features of vibration signal. Int J Eng Sci Technol 19(3):1543–1551

[42] Gao C, Xue W, Ren Y, Zhou Y (2017) Numerical control machine tool fault diagnosis using hybrid stationary subspace analysis and least squares support vector machine with a single sensor. Applied Sci 7(4):346 1-12

[43] Jemielniak K (2008) Arrazola P (2008) application of AE and cutting force signals in tool condition monitoring in micro-milling. CIRP J Manuf Sci Technol 1:97–102

[44] Yen C, Lu M, Chen J (2013) Applying the self-organization feature map (som) algorithm to ae-based tool wear monitoring in micro-cutting. Mech Syst Signal Proces 34(1–2):353–366

[45] Pechenin V, Khaimovich A, Kondratiev A, Bolotov M (2017) Method of controlling cutting tool wear based on signal analysis of acoustic emission for milling. Procedia Eng 176:246–252

[46] Mathew M, Pai P, Rocha L (2008) An effective sensor for tool wear monitoring in face milling: acoustic emission. Sadhana 33(3):227–233

[47] Ren Q, Balazinski M, Baron L, Jemielniak K, Botez R, Achiche S (2014) Type-2 fuzzy tool condition monitoring system based on acoustic emission in micromilling. Info Sci 255(1):121–134.

[48] https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

[49] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[50] http://colah.github.io/posts/2015-08-Understanding-LSTMs/