Machine Learning Framework for Tool Condition Monitoring in Milling

by

Omer Faiz

Submitted to the Department of Management and Production Engineering

Master of Management Engineering

at the

POLITECNICO DI TORINO

04 2021

Author : Omer Faiz

Supervised By : Dr. Giulia Bruno
Assistant Professor,DIGEP

Abstract

In the smart manufacturing era, the dynamics of monitoring and maintenance of the machines are changed. After the 4th industrial revolution, artificial intelligence and machine learning techniques are proven to be beneficial for carrying out predictive maintenance of machines. Internet of Things (IoT) along with the Cyber-Physical Systems (CPS) has made it possible to conduct a data-driven prognosis of a system. Predictive maintenance techniques have been developed in order to monitor an in-service machine for estimating when maintenance should be performed. Big data analysis and machine learning techniques enable the detection of the current health state and the remaining useful life of the equipment.

Above mentioned developments have played an efficient role in increasing production efficiency by minimizing downtime during the manufacturing processes. This study describes the application of a tool condition monitoring (TCM) framework to a real-time milling data-set, with the aim of classifying the tool condition (worn, unworn) of the milling tool during a running process. The application of the framework can help in optimizing the maintenance operations and preventing the breakdown of the equipment.

Acknowlegdement

Finally, I am concluding the journey of my master's which started two years back, it's been an engrossing journey that enabled me to polish my personal and management skills. Politecnico di Torino provided me with a great opportunity to internationalize, where I met people from all around the world and learned from them. This is the time when I would like to thank the people who stood by me during this expedition.

Firstly, I would like to thanks Dr. Giulia Bruno, who guided me professionally during the whole thesis work.

I am grateful to my parents and siblings for their kind support and prayers which remained me focused at all times.

I would like to thank my friends Hassaan (Hashmi and Baig) who were always there to support me with their expertise.

My deep gratitude goes to some friends Abdullah, Faseeh, Izza, and Noor who provided their exceptional support during my whole master's degree program.

In the end, I would like to thank everybody who kept me motivated and added fun to this endeavor.

Regards,

Omer Faiz

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Manufacturing processes these days are becoming more complex, dynamic, and connected. Industry operations face difficulties with strongly non-linear and stochastic action due to the numerous complexities and inter-dependencies that occur. The manufacturing industry is entering a time of incredible progress and transition led by increased integration of sensors and the Internet of Things (IoT), improved availability of data, and developments in robotics and automation. This leads to extensive digitization of the factories and challenges industrial firms to analyze, evaluate, and reassess their existing activities and potential strategic directions in the modern age [1].

Predictive Maintenance systems are developed to track, diagnose and forecast defects and degradation of system components before criticality. The ultimate aim is to avoid disruption, recognize root causes for follow-up operation, and allow effective evidence-based maintenance preparation and optimization [3]. In this regard machine learning techniques can inductively learn useful data patterns and refer to the processing state and outcome reinforcing them. The idea that data is a representation of the features of the physical machine [4]. The purpose of the prognosis is to estimate the time variation of system output degradation from its current state to its final malfunction [5].

1.1 Background

The background of manufacturing is closely related to the history of increasing the efficiency of industrial machinery to remove unnecessary downtime. Research on condition monitoring, root cause analysis and remaining useful life (RUL) prognosis lays out the fundamental expertise for maintenance and Prognostics Health Management (PHM) [7].A discipline that measures a health condition and remains a useful life dependent on past and present operational conditions is also referred to as PHM [6]. The idea is to allow timely detection and isolation of precursors or emerging faults of the components, forecast their advancement, and promote responsible decision-making [2].

Classical prognostic approaches fall into two categories: model-based and data-driven prognostics. Model-based prognosis refers to theories based on statistical models of system actions resulting from physical laws or the distribution of probabilities. For example,traditional model-based prognostics include methods based on Wiener and Gamma processes[8], hidden Markov models[9], Kalman filters[10] and particle filters[11]. One of the drawbacks of model-based prognosis is that an in-depth analysis of the underlying physical mechanisms contributing to device failures is needed.

Compared to model-based prognostics, Data-driven prognostics refers to methods that construct a predictive model using a learning algorithm and a huge volume of historical data. The specific advantage of data-driven approaches is that an in-depth interpretation of the physical actions of the device is not necessary. In comparison, data-driven approaches do not consider any fundamental distributions of probability [12].

Data-driven prognostics is based on several machine learning algorithms which is used for the tool condition monitoring. These machine learning algorithms predict tool wear and the estimate remaining useful life (RUL) of a tool. In this regard Kothuru [12] did audio-based tool condition monitoring in milling of the work-piece material With the Hardness Variation Using Support Vector Machines(SVM) and Convolutional Neural Network CNNs. The proposed SVM and CNN models have

shown a significant overall prediction accuracy of 98% and 96.3% in tool wear monitoring and 68% of overall prediction accuracy in work-piece hardness variation monitoring [12]. Sohyung Cho [14] did tool breakage detection using a support vector machine for an end milling operation, and used cutting force and spindle power as input parameters. The algorithm is computationally efficient and robust in higher dimensions [14]. Clayton Cooper [15] used convolutional neural network-based tool condition monitoring in vertical dry milling operations using acoustic signals. They achieved a reduction in tool wear detection time and demonstrated the untapped potential of acoustic signal monitoring [15].

A. G, Mebrahitom Yuan [16] did remaining tool life prediction of an end milling operation based on the force sensor signal, they used Support Vector Regression (SVR) and neural network for tool wear estimation by using cutting force, acoustic emissions and vibrations as input parameters. It is found that the upper/lower limits of predicted tool wear width are closely following the trend of actual wear width of all the three cutting tools. The maximum error between the median of predicted tool wear width and the actual wear width is around 5%, which confirms the effectiveness of proposed method for long-term tool life prediction [16]. X.Li [17] also used a Fuzzy Neural Network (FNN) model to monitor the tool degradation through tool wear and it was compared with the conventional Multi Regression Models (MRM) [17]. Dazhong Wu [18] used cloud-based machine learning for tool wear prediction in the milling process. The prediction of tool wear in milling operations was performed with the random forest and PRF algorithms with the input parameters of cutting forces, vibrations, acoustic emissions. The performance of the random forest algorithm was evaluated using mean squared error, R-square, and training time. The experimental results have shown that random forests can generate very accurate predictions.[18]

1.2 Thesis Objective

The objective of this thesis is to classify the tool condition (worn,unworn) during a milling process, for predictive maintenance by using a data driven prognosis approach based on several machine learning algorithms.

Predictive maintenance is a process in which the service life of essential components is estimated based on examination or evaluation in order to maintain the quality of components during the span of their service life. The conventional maintenance approach is following run-to-failure management based on the philosophy: "if ain't broken, then don't fix it", this approach leads to the time losses in manufacturing and costly repairs. One of the drawbacks of reactive maintenance is that maintenance resources (e.g. personnel, equipment and spare parts) are hard to forecast for repairs. Hence predictive maintenance is an effective methodology which can optimize the usage of the components by estimating their useful life and predicting the schedule for repairs.

Tool Condition Monitoring (TCM) is one of the essential steps of predictive maintenance. TCM is to monitor pattern values, analyze data on degradation, and using a monitoring system designed to track condition of a component during the process. Mostly tool condition monitoring is done by incorporating sensors with the components and monitoring the activity continuously. These observations are feed into machine learning algorithms which in-turn classify the tool condition (worn, unworn), and predicts the remaining useful life of the components based on historical data.

In today's manufacturing industry, the end milling process is a widely used process of machining, monitoring the quality of the tool is very important since it reflects the quality of the product being produced. This technique can efficiently reduce the cost of the process by reducing the rejections and wastage occurring due to the use of the worn tool. Predictive maintenance can be a great breakthrough to achieve an optimal downtime and finish product quality in a milling process. It can provide the economies of machines and the dimensional accuracy in the machining process which is of the foremost importance in the manufacturing industry.

1.3 Thesis Structure

- Chapter one contains the background as well as the objective of the research. Moreover it gathers the past study and explained why a data-driven approach for monitoring of machines could provide better results.

- Chapter two contains a detailed discussion about industrial evolution and emerging smart manufacturing industries. A detailed guide about the maintenance techniques which are being used in the industry.

- Chapter three reviews the literature regarding Machine Learning and Artificial intelligence. Also it contains a discussion about Prognosis Health Management (PHM), the detailed description of Tool Condition Monitoring (TCM). Moreover this chapter includes a discussion on cloud-based predictive maintenance.

- Chapter five includes a review of Machine Learning algorithms that can be used for data-driven prognostics for predictive maintenance.

- Chapter four includes the topologies of manufacturing processes, a detailed discussion on milling operation and the possible maintenance operations required in a milling process.

- Chapter six contains the details of data preparation.

- Chapter seven contains the methodology which is being proposed to train and test the data-set. Moreover, it will provide the significance of algorithm selected to conduct prognosis.

- Chapter eight contains detailed discussion of all obtained results of the analysis.

- Chapter nine includes the conclusion of the thesis with the the limitations we faced during the research work, and the recommendations for the future research.

Chapter 2

Industry 4.0 and Smart Manufacturing

2.1 Industrial Revolution

2.1.1 Industry 4.0

Manufacturing technologies have rapidly become widespread in the manufacturing sector. The transition towards intelligent and interconnected manufacturing processes, widely recognized as the 4th industrial revolution and referred to as industry 4.0, is the manifestation of three previous revolutions based on mechanization, large-scale production, and automation, respectively. While the ability to access these technologies is a significant step towards improving the manufacturing sector, the transformation from the general capabilities of digital manufacturing to actionable implementation decisions for manufacturing companies is yet another obstacle.The automation of manufacturing processes in third industrial revolution acted as a breakthrough by offering the infrastructure for gathering useful information and providing utilization mechanisms [19].

In addition , the ability of hardware and manufacturing devices to communicate with software systems has led to a new classification of equipment, Cyber-Physical Systems (CPS) or systems that are designed from and based on the synergy of soft-

ware and physical components [20]. Cyber-Physical Systems includes components like multiple sensory input/output devices, touch screens, cameras, GPS chips, speakers, microphone, light sensors, proximity sensors, which enhance the man-machine communication. Such systems have become an important part of lean manufacturing by offering deep insights into the production process. Established closed-loop management, process optimization and quality control mechanisms are among the most common applications for CPS [19].

2.1.2 Industrial IoTs

Industrial IoT (IIoT) is the network of intelligent and highly connected industrial components that are deployed to achieve high production rate with reduced operational costs through real-time monitoring, efficient management and controlling of industrial processes,assets and operational time [22].

Internet of Things (IoT) is a growing paradigm which interconnects humans and objects via internet. Thus, these objects can be identified and communicate over the internet. With IoT in manufacturing, devices on the factory floor, manufacturing processes, production systems and people are connected through the internet. IoT capable devices used in the manufacturing process to improve the efficiency and productivity, which is usually called as smart manufacturing. IoT devices installed in the smart manufacturing systems keep on gathering information. IoT also has diverse applications ranging from the integration of control rooms, asset monitoring, problem-proof maintenance and process planning, additive manufacturing and augmented reality [21].Industrial Internet Consortium (IIC) is an open membership organization founded by several US-based companies with an idea to catalyze the concept of industrial internet which has further potential to transform to business in the industrial sector.

With IIoT, anything from major machines to transformers can be linked to the Internet, providing status alerts and performance data. In this way, operators will take preventive action on a possible issue until it costs the company and consumers billions of dollars.In health care, IIoT and Big Data analytics can drive a variety of

positive aspects. Results such as enhancing patient flow, tracking and controlling the usage of health care equipment, monitoring clinical, financial, and operational steps, and managing efficiency of the workforce. GE Healthcare reports that these advancements will reduce the cost of hospital equipment by 15 to 30 percent by an additional hour of efficiency per shift [23].

2.2 Smart Manufacturing

The Industry 4.0 framework in the manufacturing sector covers a wide range of applications ranging from product design to logistics. The function of mechatronics, a basic concept in the design of the manufacturing system, has been modified to make it compatible with cyber physical systems. Smart product design based on personalized specifications targeting individualized goods has been proposed. Figure 1 presents, industry 4.0 framework for smart manufacturing systems. The horizontal axis shows the typical issues in industry 4.0, including smart design, smart machining, smart monitoring, smart control smart scheduling, and industrial applications. The vertical axis shows challenges in another dimension of Industry, ranging from sensor and actuator deployment to data collection, data analysis, and decision-making.

CPS and IoT-based manufacturing systems require the generation of large quantities of data in industry 4.0, and big data processing is essential for the design and operation of smart manufacturing systems. For example, using the big data analytics methodology, a comprehensive framework for data-driven risk assessment for industrial manufacturing systems has been performed based on real-time data. Such a subject has been widely documented in support of production optimization and production of cyber-physical systems [24].

Increased manufacturing complexity, dynamic economics, and dramatically different performance goals would require the widespread use of networked, real-time information-based technologies that turn facilities into knowledge-based facilities, a reactive, predictive approach to function, threat detection to incident prevention, and vertical decision-making to distributed intelligence and local decision-making with

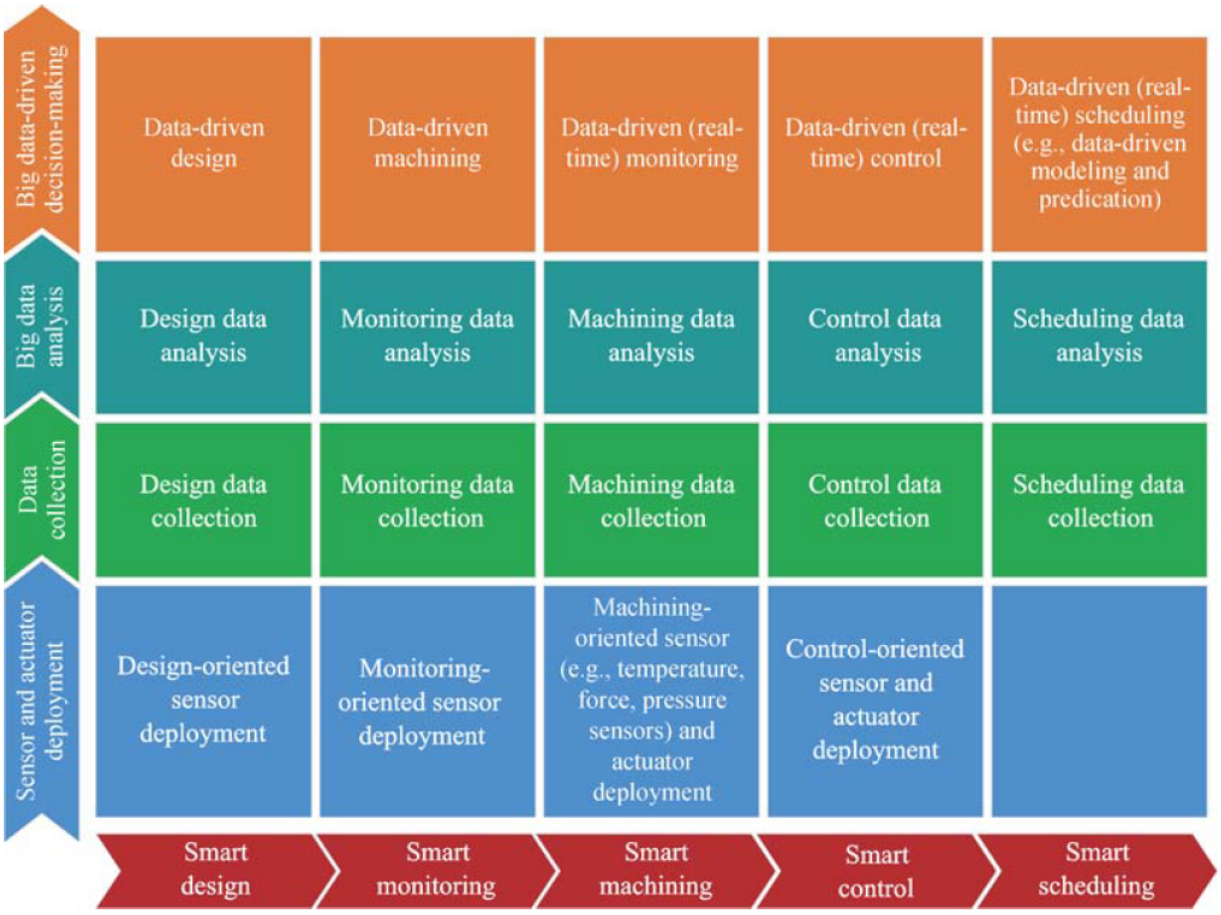

Figure 2-1: Smart Manufacturing Framework

Smart Machining	In Industry 4.0, smart machining can be done with the help of smart robots and other forms of smart objects that can sense and communicate with each other in real-time.
Smart Monitoring	Monitoring is an important part of the operation, maintenance, and optimal scheduling for Industry 4.0 manufacturing systems. The generalized deployment of different sensors has made smart monitoring possible.
Smart Control	High-resolution adaptive production control (i.e. smart control) can be accomplished in Industry 4.0 by creating cyber-physical production control systems. Smart control is primarily performed to handle various smart machines or resources physically through a cloud-enabled platform.
Smart Scheduling	Smart scheduling primarily uses advanced models and algorithms to derive information from data captured by sensors. Data-driven techniques and an advanced decision architecture can be used for smart scheduling.
Smart Design and Manufacturing	Research at the stage of smart design and manufacturing includes smart architecture, smart prototyping, smart controllers, and smart sensors. Real-time control and monitoring help the introduction of smart manufacturing. Supporting technologies include IoT, STEPNC, 3D printing, industrial robotics, and wireless communications. Big data analysis plays a key role in smart manufacturing.
Smart Decission-Making	Smart decision-making is at the core of Industry 4.0. The ultimate aim of deploying large-scale sensors is to achieve smart decision making via a detailed set of data. Realizing smart decision-making needs exchanging knowledge and communication in real-time.

Table 2.1: Smart Manufacturing Framework [24]

global impact.Smart Manufacturing is an organization that combines the knowledge of the consumer, its partners, and the public. It responds as an organized,performance-oriented organization, minimizing energy and material use while optimizing environmental sustainability, health and safety, and economic competitiveness. The company, operations management, labor, and manufacturing innovations lead to new ways of thinking about the manufacturing process [25].

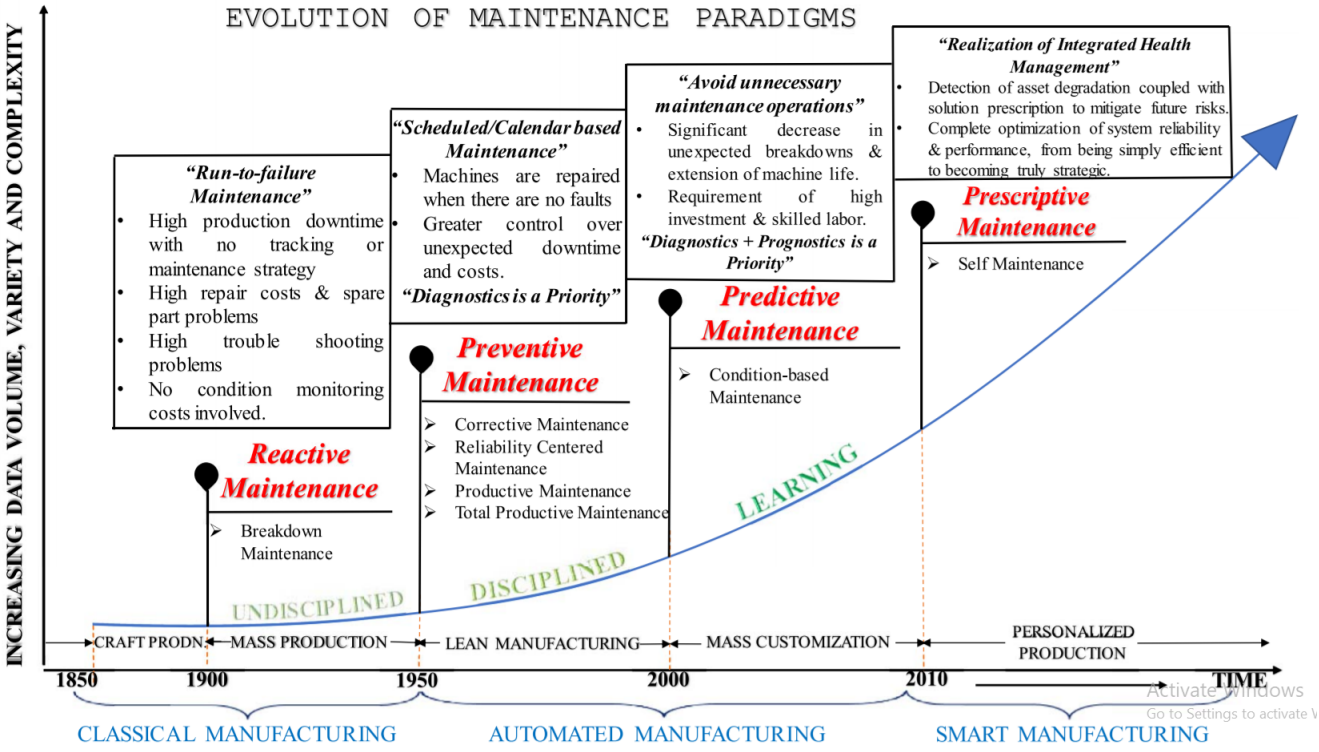

Figure 2-2: Evolution of Maintenance Paradigm

2.3 Smart Maintenance

Modern manufacturing systems typically consist of several machines to satisfy the demand for good quality and high functional complexity goods. The probability of machine failure is aggregated as the number of machines increases in the system. In any industry, sudden failure can lead to huge economic losses due to machine/production downtime. For example, a standard car assembly line experiences a $20,000 loss for every minute of downtime.Therefore it is necessary for large and complex manufacturing systems to have successful maintenance operations that improve the status of machines.During the technological revolutions, maintenance techniques have undergone a radical progression and are currently underway Fig[2].

Usually, the maintenance management is categorized into different policies [26]:

- Corrective maintenance is an unscheduled repair, where the equipment is allowed to operate until it fails unless a maintenance operation is carried out.

- Preventive maintenance, perhaps the most common maintenance policy, A periodic collection of operations on the equipment shall be carried out to minimize the risk of failure. This form of maintenance is carried out while the equipment is still in operation.

- Predictive maintenance includes the use of sensor technology and analytical tools to predict when equipment failures can occur and to prevent failures from occurring through maintenance.

Corrective maintenance is the most straightforward technique, among others, where activities are carried out due to failure to restore the machine to a particular state. Corrective maintenance can be very expensive, as it is often combined with unnecessary downtime, which often costs three or four times as much as planned downtime.Alternatively, preventive maintenance requires steps taken to sustain an object in a particular condition by providing systematic inspection, assessment, and prevention of incipient failures. The simple implementation of preventive maintenance is to perform periodic acts based on assumed behavior, such as the mean time between failures.To prevent unnecessary downtime, a preventive maintenance interval is needed. It is also conservatively set to be much shorter than the average to allow for differences between machines. Besides, the deterioration process of machines can be strongly linked to operations and external influences, making it difficult to calibrate the actions of machines.

Nowadays, industrial maintenance is primarily reactive and preventive, the predictive approach is applied only to severe circumstances. Traditionally, these maintenance techniques do not take into account the vast amount of data generated on the shop floor and the evolving Information and Communication Technology (ICT) available, e.g. Internet of Things (IoT), Big Data, Advanced Data Analytics, Cloud Computing, and Augmented Reality.However, the maintenance paradigm is shifting and industrial maintenance is now seen as a strategic factor and a profit-making factor to ensure efficiency in industrial systems.

Chapter 3

Artificial Intelligence Based Prognosis for Predictive Maintenance

3.1 Artificial Intelligence

Nowadays, manufacturing objectives are becoming extremely challenging, with a multitude of demands arising from a growing product and process complexity, increased uncertainty in consumer demand and expectations, and steadfast competitive pressures from competitors in the market to remain profitable. Seen from a positive perspective these extreme conditions provides an opportunity for the unique capabilities of Artificial Intelligence (AI) over the daily used conventional tools. AI tools capable of defining and classifying multivariate, non-linear trends in operating and output data that are hidden from the plant engineer.AI provides the opportunity to turn large volumes of complex manufacturing data, which have become commonplace in today's factory, into actionable and insightful knowledge. AI tools which establishing core competencies in maintenance are Prognostic and Health Management (PHM) and Condition Based Monitoring (CBM). The objective is to allow timely identification and isolation of precursors or incipient faults of components, predict their development, and promote reasonable decision-making [1].

3.2 Predictive Maintenance

Predictive Maintenance in its broad sense is an application of AI in the maintenance which apply data analysis techniques to information generated in shop floor processes for the identification of anomalies in the behavior of properties. This approach extends PHM and CBM maintenance approaches by considering machine learning and augmented reality technology to assist maintenance technicians during maintenance interventions by offering guided intelligent decision-making support articulated by the use of human-machine interaction technologies [26]. While the main objective of PHM is to provide the health status and predict the Remaining Useful Life (RUL) of components or facilities, financial benefits such as operating and repair cost savings and increased lifespans are also achieved [26].

3.2.1 Condition Based Maintenance

Condition-based maintenance (CBM) makes maintenance decisions based on information about the current deterioration of the machine and its evolution. It uses system degradation information derived from online monitoring systems to optimize CBM decisions by balancing the probability of failure and achievable profits.It is important to understand how the dynamics of production shift under the CBM stop events and build an optimal CBM control system to minimize the negative effect of CBM stoppage on output [27].The idea of an AI-based diagnosis is to formulate a task as a classification and to relate the details contained in the data to the types of fault and the degree of severity. The framework of CBM is shown in the Fig [3-1] which describes that CBM is further categorized into three types 1) Physics based 2) Model based 3) Data-driven/AI.

- **Physics-based**
- **Model-based**
- **Data-driven/AI**

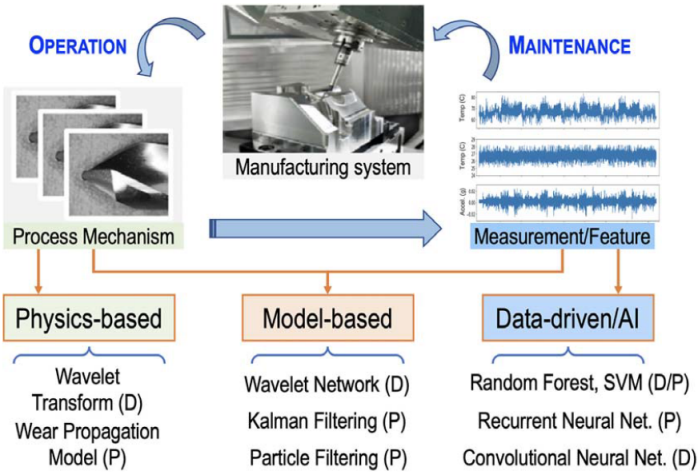

Figure 3-1: Framework of Condition Based Maintenance

3.2.2 Cloud-based Predictive Maintenance

Cloud-based prognosis, which reflects a new form of service-oriented technology to help several organizations to deploy and operate prognosis services over the internet can be imagined, powered by the ability of cloud computing and cloud development. First, the data is collected by monitoring the system state of the machine. Sensors are connected on the shop floor, remotely and dynamically. Based on in-situ measurements, remote data analysis, and root cause failure diagnosis and prognosis are then performed.Secondly, collaborative technical teams can have expert knowledge in the cloud, which is known as a knowledge base that can be accessed across the internet. The results of the forecasting service and the calculation of time-to-failure would lead to predictive maintenance planning, which can be used remotely and dynamically on the factory floor.

Cloud-based prognosis have following characteristics [28]:

- **Service Oriented ability**: All prognostic system functions are provided as cloud-based services that are accessible through web browsers or the internet. Local installation and servicing do not require costly software packages.

- **Accessibility and promotion of robustness**: Cloud-based framework as a

flexible and integrated solution, Configurable prognosis services can increase the robustness of existing manufacturing processes. If required or applicable, the cloud can choose alternative and pay-as-you-go prognostic services and various maintenance options.

- **Resource-aware ability**: Cloud-based prognosis allows for tracking and forecasting system usage and conditions, locally or remotely, so that maintenance decisions can become resource-conscious and well-informed.

- **Collaboration and distribution**: Cloud-based prognosis facilitates seamless and collaborative sharing of process and system data between different applications at different locations.

3.3 Tool Condition Monitoring

Tool failure is a dynamic phenomenon involving one or more failure modes, including excessive wear, brittle fracture (chipping), and breakage.In high-speed cutting processes, late replacement of damaged tools can lead to machine breakdowns and have a negative effect on the quality of the product, which will lead to scrapping and high process costs. Accurate tool condition detection is important to achieve a high level of competitiveness by increasing process efficiency and standardizing the quality of the parts made. Tool Condition Monitoring (TCM) systems have therefore been widely emphasized as an important concept for achieving these industrial requirements. Due to the difficulties of implementing direct methods online in harsh cutting environments, more intensive efforts have been made towards indirect methods that define the tool fault by detecting indicative process-borne features derived from different signal measurements [29].

TCM techniques include direct measurement and indirect measurement of tool wear. In direct measurement methods, the tool's wear parameters are measured by a microscope surface profiler, etc. These direct methods have the benefit of acquiring high accuracy Dimension shifts due to tool wear, but they are not technically and

economically desirable. The direct methods are inappropriate to field conditions due to the continues flow of the cutting fluid and disturbance due to offline operations, which severely limits the application of direct measurement.

Indirect methods are designed on the basis of specific parameters which are directly correlated with tool wear. These parameters includes machining process data such as operational parameters and sensor signals (force signal, vibration signal, acoustic signal, current/power signals, etc) are acquired and the relevant features are extracted from the data. Moreover, Artificial intelligence techniques are applied on these extracted features to predict the tool condition. Generally, the indirect TCM system consists of hardware and software components for signal acquisition, signal pre-processing, features extraction, features selection and decision making.

Chapter 4

Machine Learning Algorithms for Predictive Maintenance

4.1 Introduction

Machine learning is a branch of artificial intelligence that tends to allow machines to perform their jobs skillfully through the use of intelligent software. Statistical learning approaches are the foundation of intelligent software used to create machine intelligence. Since machine learning algorithms need data to be trained, the discipline must be related to the discipline of the database. Similarly, there are familiar concepts such as Information Discovery from Data (KDD), data mining, and pattern recognition.

Machine learning algorithms are helpful in bridging this gap of understanding. We are not aiming to grasp the fundamental mechanisms that help us learn. We write computer programs that will make machines learn and allow them to perform tasks such as prediction. The purpose of learning is to create a model that takes feedback and generates the desired outcome. Often we can grasp the model, but at other times it can also be like a black box for us the work of which cannot be completed. The model can be seen as an approximation of the mechanism that we want machines to imitate. In such a case, we will get errors for some input, but most of the time, the model is right. The accuracy of the results would therefore be another indicator of

the efficiency of the machine learning algorithm.

4.1.1 Learning Strategies

Machine learning is divided into two major learning strategies:

Unsupervised Learning

Unsupervised learning methods are used against data that have no labels.The main objective of the unsupervised learning method is to explore the data and find some hidden structure among them.The target of such unsupervised learning problems may be to discover groups of similar examples within data called clustering, or to determine the distribution of data within input space, known as density estimation, or to project data from high-dimensional space to low-dimensional space for the purpose of dimensional reduction and data visualization. Conventional unsupervised learning methods in the process industry include main component analysis, independent component analysis, k-means clustering, kernel density estimation, self-organizing map, Gaussian mixture models, manifold learning, vector data description support, and so on [31].

Supervised Learning

Supervised learning deals with the cases where the data is labeled, discrete and continuous. The main applications of the supervised machine learning method include process monitoring, fault classification and identification, online operating mode localization, soft sensor modeling and online applications, quality prediction and online estimation, key performance index prediction and diagnostics, etc. Supervised learning is majorly categories into following:

- **Regression** If the desired output consists of one or more continuous variables, then the task of supervised learning is called regression. A typical example of the problem of data regression is the prediction of key performance in a process

in which inputs contain routinely recorded process variables such as temperature and pressure.

- **Classification** Classification is a type of supervised learning. It specifies the class to which data elements belong to and is best used when the output has finite and discrete values. It predicts a class for an input variable as well.e.g. fault classification, or operating mode classification

In our case we are going to train classification models to predict the state of milling tool i-e (a) worn or (b) unworn. Moreover in the following sections several classification models are explained which will be further applied on our data.

4.2 Logistic Regression

Logistic Regression is a Machine Learning technique used for classification problems, a predictive analysis algorithm centered on the idea of probability. In logistic regression instead of predicting the exact value for the variable, the probability of happening of that event is predicted.

The logistic function, also known as the sigmoid function, was developed by statisticians to explain the characteristics of population growth in ecology, gradually growing and optimizing environmental efficiency. It's a shaped curve that can take any real-valued number and map it to a value between 0 and 1, but never precisely within those limits.

$$f(value) = 1/(1 + e^{-}value) \tag{4.1}$$

Where e is the origin of the natural logarithms (Euler number or EXP() feature in your spreadsheet) and the value is the real numerical value that you want to transform.

Logistic regression is a classification algorithm used to allocate observations to a discreet collection of classes. Examples of labeling issues include e-mail spam or not spam, internet transactions fraud or not fraud, tumor malignant or benign. Logistic

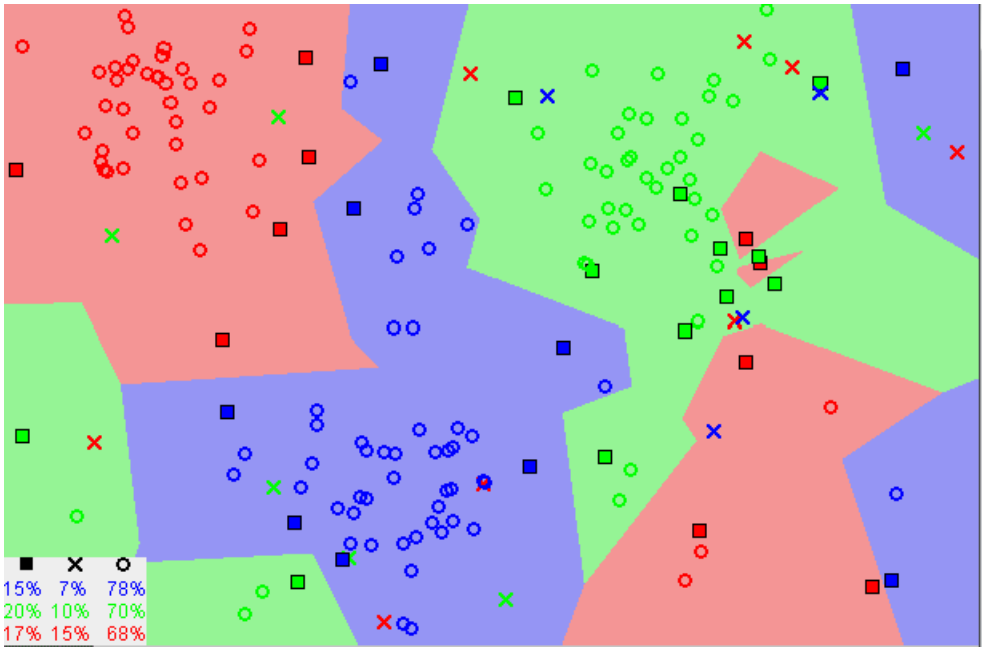

Figure 4-1: K-Nearest Neighbour

regression transforms the output by using the logistic sigmoid function to return the likelihood value.

4.3 K-Nearest Neighbors (K-NN)

KNN can be used for both classification and regression predictive problems. However, it is most commonly used for classification issues in the industry. In order to assess any methodology, we usually look at three critical aspects:

1. Simple to understand performance
2. Calculation of time
3. Predicting Strength

The KNN algorithm assumes that identical objects occur in close proximity to each other. In other terms, related objects are close to each other. Note in the picture above that, much of the time, identical data points are close to each other. The KNN algorithm believes that this assumption is valid enough for the algorithm to be useful. . KNN captures the concept of similarity (sometimes called distance,

similarity, or proximity).There are several methods to measure the distance, and one way may be preferable based on the problem we solve. The straight-line distance (also called the Euclidean distance) is, however, a common and familiar choice.

k is therefore just the number of neighbors "voting" on the proximity of test example's class.To choose the K that is appropriate for your results, we run the KNN algorithm multiple times for various values of K and choose the K that decreases the amount of errors we find while retaining the algorithm's ability to correctly make predictions when it's given data that it hasn't seen before.

4.4 Support Vector Machine (SVM)

Support vector machine algorithm finds out a hyperplane in n-dimensional plane that clearly classifies the data points. There are several potential hyperplanes that could be chosen to differentiate the two types of data points. Our goal is to locate a plane with the highest margin, i.e. the maximum difference between the data points in both groups. Maximizing the margin gap gives some reinforcement such that potential data points can be classified more confidently.

Hyperplanes are judgment boundaries that help to distinguish data points. Data points falling on either side of the hyperplane can be assigned to various groups. Also, the scale of the hyperplane depends on the number of functions. If the number of input features is 2, the hyperplane is just a line. If the number of input features is 3, the hyperplane can become a two-dimensional plane. It's hard to think when the number of features reaches 3. Support vectors are data points that are closer to the hyperplane and change the direction and orientation of the hyperplane. Using these support vectors, we optimize the range of the classifier. Removing the support vectors would shift the direction of the hyperplane. These are the things that enable us to develop our SVM.

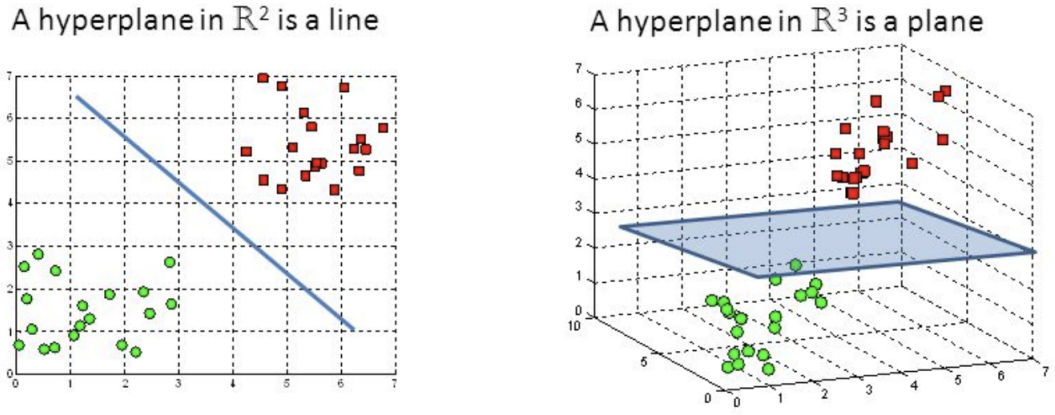

Figure 4-2: Support Vector Machine

4.5 Decision Tree Classification

The Decision Tree algorithm is a family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can also be used to solve problems of regression and classification. It breaks down the data set into smaller and smaller subsets as, at the same time, the related decision tree is gradually created. The end product of this is a tree with decision nodes and leaf nodes. The decision node (for example, Outlook) has two or three branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g. Play) is a classification or a decision. The largest decision node in the tree that corresponds to the greatest indicator known as the root node. Decision trees can handle both categorical and numeric results. Fig [5-3] shows an example of decision tree classification.

Compared to other algorithms, decision trees require less time to prepare data during pre-processing. The decision tree does not require data normalization. The decision tree does not even include the scaling of results. Missing values in the data often do NOT greatly impact the method of creating a decision tree. The decision tree model is very intuitive and easy to communicate to professional teams and stakeholders.

Figure 4-3: Decision Tree Classification

4.6 Random Forest Classification

The Random Forest is a supervised learning algorithm. The "forest" that it creates is a series of decision trees, usually trained by the "bagging" process. The general principle of the baggage approach is that a mixture of learning models would maximize the total result.

A benefit of the random forest is that it can be used for both classification and regression problems, which make up the majority of modern machine learning models. Let's look at random forest classification, as classification is often used as the building block of machine learning. In Fig[5-4] you can see how a random forest with two trees would look like.

Random forest has almost the same hyperparameters as a decision tree or a baggage classifier. The Random Forest adds additional randomness to the model as the trees expand. Instead of looking for the most important feature when splitting a node, it looks for the strongest feature in a random subset of features. This results in a large range that usually results in a stronger model.

Another high benefit of the random forest algorithm is that it is very straightforward to calculate the relative value of each function to the forecast. Sklearn is a fantastic way to calculate the value of a feature by looking at how often the tree nodes that use the feature decrease impurity among all trees in the forest. It dynamically

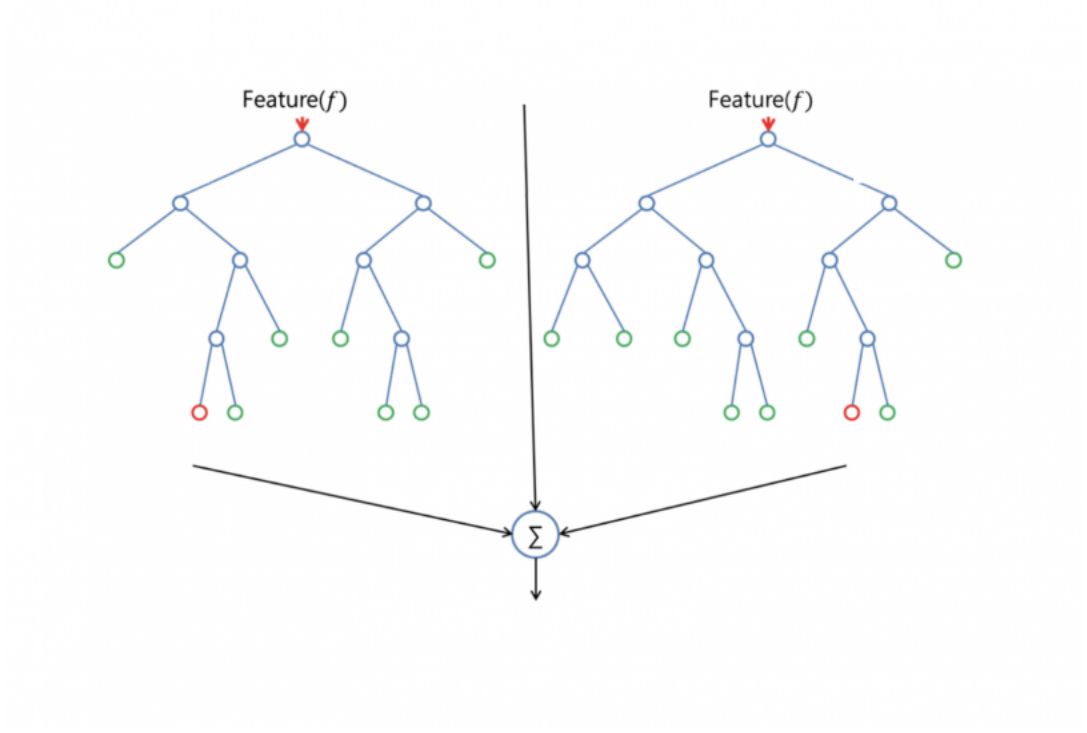

Figure 4-4: Random Forest

measures this score for each training element and averages the scores such that the sum of all value is equal to one.

4.7 Artificial Neural Network

ANN is a human nervous system-inspired simulation technique that enables learning, for example from representative data representing a physical event or decision-making mechanism. The special characteristic of ANN is that it is capable of defining observational associations between independent and dependent variables and extracting hidden details and detailed insights from representative data sets.

ANNs consist of a layer of input nodes and a layer of output nodes attached to one or more layers of hidden nodes. Input layer nodes transfer information to hidden layer nodes by firing activation functions, and hidden layer nodes fire or stay dormant based on the proof given. Hidden layers attach weighting functions to the proof, and when the value of a single node or group of nodes in the hidden layer exceeds a

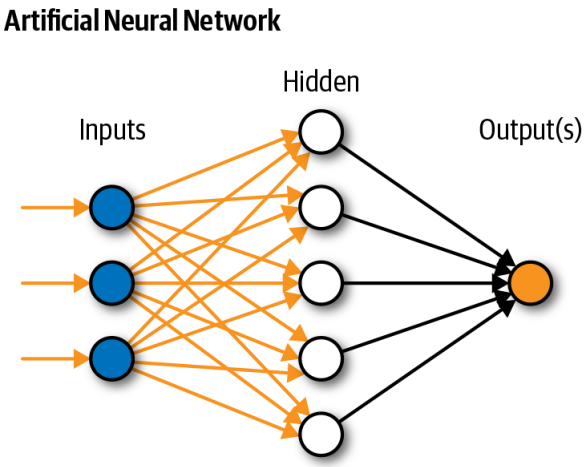

Figure 4-5: Structure of ANNs

threshold, the value is transferred to one or more nodes in the output layer. ANNs must be qualified in a significant number of instances (data) [32].

Fig [5-5] shows how a normal neural network feels like. After training the model, you need to look at two things: bias and variance. The bias-variance tradeoff is a kind of problem that implies that low bias models would have high variance and vice versa.High bias leads the model to ignore the relevant correlation between essential features that can contribute to under-fitting. High variance may cause the model to be susceptible to random noise in the data set contributing to over-fitting. Our mission is to eliminate bias and variance and to find an optimum fit line as shown in the Fig [5-6]

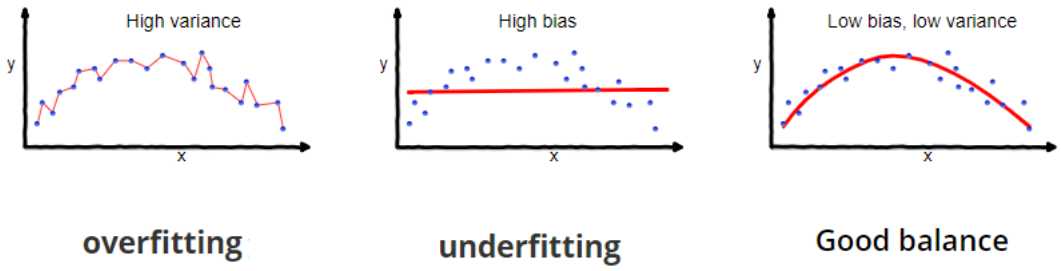

Figure 4-6: Tuning of ANNs

Chapter 5

Manufacturing Processes and control

5.1 Use-Case Definition

The use-case selected for the thesis is a "Milling Data Set",A series of machining experiments were run on 2" x 2" x 1.5" wax blocks in a CNC milling machine in the System-level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan. Machining data was collected from a CNC machine for variations of tool condition, feed rate, and clamping pressure. Each experiment produced a finished wax part with an "S" shape - S for smart manufacturing - carved into the top face.

5.2 Milling Processes

Among all the manufacturing processes the largely used is milling operation. The milling process is a type of manufacturing process that allows the production of complex parts and trimming of several materials such as metal,wood,plastics and composites. Unlike the other manufacturing processes milling process is performed on a dedicated milling machined equipped with a multi-sharp tool, mounted on a spindle, and the process is carried out by the simultaneous movements of motorized axes. The cutting head of the tool removes the material from the work piece when comes in contact with the cutter.

5.2.1 Types of Milling

There are two major types of milling processes:

- **Face Milling**, in general, is defined as the process of cutting surfaces that are perpendicular to the cutter axis, or the faces of a part.Face milling generates a surface normal to the axis of rotation. It is used for wide flat surfaces. The peripheral portions of the teeth do most of the metal cutting. There are several forms of face milling:(a) conventional face milling, in which the cutter width extends beyond the work piece on both sides,so the cutter overhangs the work on the both sides;(b) partial face milling , where the cutter overhangs the work on one side only; (c) End milling, In which the cutter diameter is less than the working distance, the slot is cut into the part;(d) Profile milling, it is a type of end milling which is used to create a profile if the end of a work piece;(e) Pocket milling, it is a type of end milling in which tool needs to cut shallow into the work piece;(f) Surface contouring, which uses a ball-nose cutter along a curvilinear trajectory to create a three-dimensional surface.

- **Peripheral milling** Peripheral milling generates a surface parallel to the rotation axis. Both flat and formed surfaces can be produced by this method, with a cross-section of the resulting surface corresponding to the axial contour of the cutter. The process is sometimes referred to as 'slab' milling. There are various forms of peripheral milling: (a) slab milling, the basic form of peripheral milling in which the diameter of the cutter is greater than the work part width; (b) slot milling, in which the diameter of the cutter is less than the workpiece width, creating a slot in the work when the cutter is very thin; (c) side milling, in which the cutter machines the side of the workpiece; (d) straddle milling, similar to side milling but the cutting action occurs on both sides of the item; (e) form milling, in which the teeth of the milling cutter have a shape which corresponds to the profile of the surface to be produced [25].

5.2.2 CNC milling machine

All the milling operations described above are used in conjunction with milling machines, which provide rotary movement to the cutters, and feed to the workpiece and arrangement for clamping, automatic feed etc. Milling machines come in three basic models:

- Horizontal Milling Machine
- Vertical Milling Machine
- Universal Milling Machine

CNC stands for computer numeric controlled. It refers to any machine tool (i.e. mill, lathe, drill press, etc) that uses a device to electronically regulate the motion of one or more axes on the machine.CNC Milling Machinery is an incredibly useful milling machine for both commercial and industrial production. The aerospace industry, the medical industry and the electronics industry can all benefit from the products of CNC Milling.

Components of CNC milling machine

- **Frame** The frame protects the unit and provides stiffness to withstand cutting forces. Usually, the base has a detachable column. CNC Milling Machine Frames are most commonly made from cast iron. Other options include epoxy granite filling and aluminum welding.
- **Table** The table is where some kind of workholding solution retains the workpiece for machining. Most of the milling machine tables use T-Slots to connect the workholder to the bench. By mounting a Fixture Plate on it you can make the T-Slot table much more convenient and flexible.
- **Spindle** The spindle is the foundation of every milling machine. It consists of a revolving taper assembly where the holders of the tool can be mounted.

- **Axes** The axes of the CNC Milling Machine make motion with Cartesian coordinates programmed through g-code and manual jogging from the control panel. Generally there are three axes corresponding to X, Y, and Z. The optional 4th Dimension is a CNC Milling Machine accessory. Five axis milling machines are feasible but not very popular in the DIY CNC world. The spindle rotates a motor with an optional transmission of some kind.

- **CNC Controller** The CNC Controller is the brain of the system. It includes the electronics that drive the axle motors to shift the axes. CNC Controllers are responsible for receiving G-Code and manual inputs from the CNC Control Panel and translating them to the correct signals to control the Axis Stepper or Servo Motors.

Accessories of CNC milling machine

- **Coolant** CNC Coolant plays many roles in cutting and comes in various varieties. There are many forms of coolant device typical to DIY CNC including: – Mist Coolant – Flood Coolant – Air Blast or Cool Air Gun – Dust Collector Although the Dust Collector does not "cool," it satisfies the primary coolant that is the chip evacuation.

- **Lubricating Mechanism** CNC Milling Machines require lubrication for their conductors and in particular, for their pathways. Ways are any device used to cause axes to slip. Way Oilers can be either manual or automatic. Automatic electric way oiler contributes dramatically to reliability by maintaining smooth friction and less road wear.

- **Powered Draw-bar** Powered or Power Drawbars are a great comfort for CNC Milling Machines and an absolute necessity if the Automatic Tool Changer is to be used. They effectively allow the button to capture and release tool holders from the spindle. There are three different styles, including Impact Wrench, Tormach TTS Style and Pull Stud. Machines without a Power Drawing bar

enable the operator to loosen the drawbar (or spindle-dependent collet) with a key that is much longer than the push-button type.

- **Enclosure** CNC Milling Machines are fundamentally a messy beast. They're going to chuck chips away long ways. If they are fitted with a flood coolant, the mess is much worse. With the Enclosure, the mess is contained inside, where it doesn't get all over the workshop.

Operational Parameters of CNC milling machine

- **Spindle Speed of Revolution** Spindle revolution speed determines the cutting edge velocity relative to the workpiece, i.e. the cutting speed. Since cutting speed has a significant influence on tool life, the selection of cutting speed is closely related to the reliability of the tool. Too low or too high cutting speed can lead to a drastic reduction in tool life. In the meanwhile, the speed of the spindle revolution in the milling of thin-walled workpieces has a direct influence on the stability of the cutting process. The spindle speed of the revolution should therefore be chosen discreetly in the milling process.

- **Cutting depth and Cutting width** The cutting depth and the cutting width are constrained by the spindle strength, the transmitting power of the machine tool, the material type, the tool specifications, the coolant, the machining process, and the rigidity of the machine tool-tool-workpiece device. And they're having a huge influence on tool life. They should then be chosen fairly according to the machining standard, the machining efficiency, and the machining process. Generally, machining performance is the first objective of roughing machining, such that a greater cutting depth and a larger cutting width are chosen. The consistency of the workpiece surface is the main objective of the finishing process so that the depth of cutting and the width of cutting can be minimized.

- **Feed Rate** The feed rate is the speed at which the cutting tool travels relative to the workpiece in the milling process. In general, the linear feed rate is used in

practical processing and is defined as feed per minute. The feed rate of milling can directly impact machining precision, surface quality, workpiece deformation, and tool life. It is also limited by tool parameters, workpiece material, tool direction, machine tool stiffness, and feed device efficiency.In the machining process, the feed rate of milling is selected based on component material, geometry characteristics, quality specifications, and machine tool capacity.

- **Tool material** In milling operation cutting tools should have following characteristics:

  - **Hardness** is the ability of a material to resist deformation. Cutting tool material must be harder than the material of workpiece. For this reason hardness and stength of the cutting tool is maintained at relatively high temperatures.

  - **Toughness** is the ability of a material to absorb energy and plastically deform without fracturing. Toughness is necessary in order for the cutting tool not to chip or fracture, especially during the cutting operation.

  - **Wear Resistance**means the attainment of acceptable tool life before tools need to be replaced. Hardness is the most important property that resist abrasive wear.

- **Workpiece Material** Mechanical properties of the workpiece material are important requirements influencing the working conditions. In the case of low cutting forces, low hardness and tensile strength usually provide better machinability.Common materials that are used in milling involve steel, cast iron, aluminium, nickel, zinc, magnesium, titanium and thermoset plastics. Properties of the work material have a significant influence on the success of milling operation and the selection of the workpiece material must take into consideration several factors as cost, strength, resistance to wear and especially its machinability.

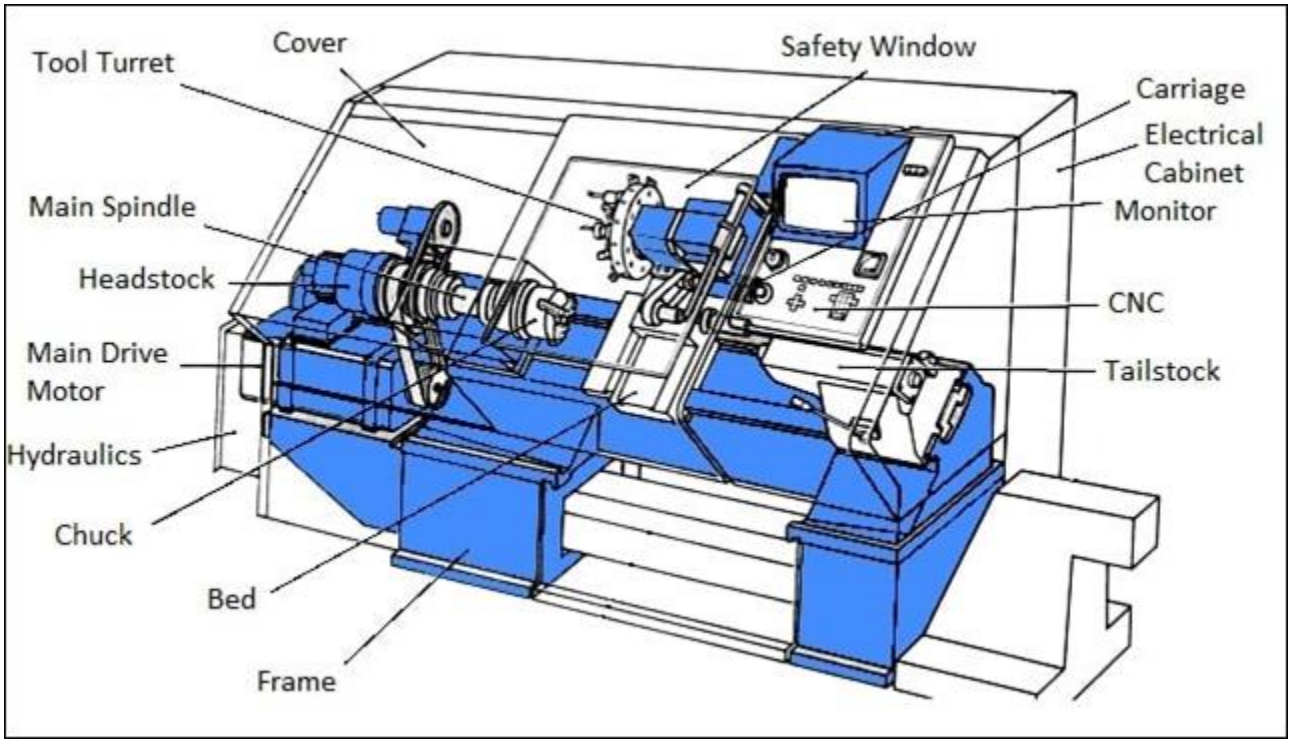

Figure 5-1: CNC Milling Machine

5.3 Tool Wear Monitoring

During a milling operation the replacement of a worn tool is a major maintenance required. Tool wear monitoring in very important and a delicate task to insure the on-time tool replacement and moreover the quality of the finish product. Sensor-based approaches, which are considered here, can be divided into mainly two methods in which we classify tool wear monitoring: (a) Direct method;(b) Indirect Methods.

Direct methods calculate the individual values of such wear parameters (e.g. the size of the wear area), whereas indirect methods measure relevant process parameters that are associated with tool wear (e.g. cutting forces or vibrations). It is possible to conclude the values of these method parameters on the current tool state using an effective analytical or empirical model. Direct method measuring equipment is very costly and these systems are vulnerable to errors due to environmental factors in the machine tool (e.g. chips, coolant, etc.)

Another relevant classification criteria for sensor-based methods depends on the

monitoring period: significant parameters are measured in continuous or on-line methods during the cutting process, while significant parameters are measured in intermittent or off-line methods only throughout the cutting process intervals.Altogether, a trend towards indirect, continuous methods can be detected. These methods will be investigated in the table.

Wear or process parameters	Examples for measurement procedures and transducers
Shape or position of the cutting edge or the wear area	*Direct methods* Shape or position of the cutting edge or the wear area Measurement with optical methods (e.g. CCD camera or fibre optic sensor) or integration of thin film sensors into the coating of a cutting tool
Volumetric overall loss of the tool	Measurement of size and concentration of wear particles in the coolant (and electrochemical analysis) or measurement of radioactivity (for specifically prepared tools)
Changes of the electrical resistance at the junction of tool and workpiece	Voltage measurement at a specific, conductive tool coating
Changes of workpiece dimensions	Dimension measurement by means of micrometers or optical, pneumatic, ultrasonic, or electromagnetic transducers
Change of distance between tool (or toolholder) and workpiece	Distance measurement by means of micrometers, pneumatic gauges, displacement transducers (e.g. inductive or capacitive), or ultrasonic sensors
Cutting forces	*Indirect Method* Force measurement with strain gauges or piezoelectric sensors at the tool or at (or in) the toolholder, piezoelectric force measuring plates or rings at the turret, force-measuring bearings, torque measurement at the main spindle
Electrical current, power, or energy	Measurement of current or power consumption of spindle or feed motors (e.g. ampere meter or dynamometer)
Cutting temperature	Temperature measurement by means of thermocouples or pyrometers, reflectance of chip surface or chip color
Roughness of the machined surface	Measurement with a mechanical stylus or optical methods (e.g. CCD camera or fibre optic sensor)

Table 5.1: Examples for direct and indirect tool wear sensing methods

Chapter 6

Data preparation

6.1 Data Description

Data is the essential component for training a machine learning algorithm. For the purpose of tool condition monitoring data related to the operational features of CNC milling machine is required. The data-set used here for the classification of the worn and unworn tool taken from System-level Manufacturing and Automation Research Testbed (SMART), University of Michigan. The data-set comprises of a series of machining experiments were run on 2" x 2" x 1.5" wax blocks in a CNC milling machine. There are total 18 experiments each experiment produced a finished wax part with an "S" shape - S for smart manufacturing - carved into the top face, as shown in Fig [6-1] Eight experiments were run with an unworn tool while ten were run with a worn tool. Moreover experiments were run with pressures of 2.5, 3.0, and

Figure 6-1: 'S' shaped test artifact

4.0 bar. The data-set obtained from the CNC milling machine is consist of two files. One data-set contain the general data from 18 different experiments which are given in the "train.csv" and includes following features Table [6-1]

The other part of data-set is a time series data was collected from 18 experiments with a sampling rate of 100 ms and are separately reported in files experiment_01.csv to experiment_18.csv. Each file has measurements from the 4 motors in the CNC (X, Y, Z axes and spindle). The features available in the machining data-sets are listed in table [6-2].

Material	Feedrate	Clamp pressure	Tool Condition	MachiningFinalize
wax	6	4	unworn	yes
wax	20	4	unworn	yes
wax	6	3	unworn	yes
wax	6	2.5	unworn	no
wax	20	3	unworn	no
wax	6	4	worn	yes
wax	20	4	worn	no
wax	20	4	worn	yes
wax	15	4	worn	yes
wax	12	4	worn	yes
wax	3	4	unworn	yes
wax	3	3	unworn	yes
wax	3	4	worn	yes
wax	3	3	worn	yes
wax	6	3	worn	yes
wax	20	3	worn	no
wax	3	2.5	unworn	yes
wax	3	2.5	worn	yes

Table 6.1: Machining data-set of a CNC milling

```
frames = list()
results = pd.read_csv("train.csv")
for i in range(1,19):
    exp = '0' + str(i) if i < 10 else str(i)
    frame = pd.read_csv("experiment_{}.csv".format(exp))
    row = results[results['No'] == i]
    if row.iloc[0]['machining_finalized'] == 'yes':
        frame['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1
        frames.append(frame)
    else:
        i+=1
```

Figure 6-2: Python Code: Data Cleaning and Labeling

6.2 Data Pre-processing

Data pre-processing is a data mining strategy that involves transforming raw data into an usable form. Real-world data is often unreliable, contradictory and/or deficient in certain behaviors or patterns and is expected to include a variety of mistakes. Pre-processing data is an established way of tackling these problems. Notice that excessive data pre-procesing can reduce the quality of data, so a careful data pre-processing procedure will be adopted.

6.2.1 Data Cleaning and Labeling

Data-cleaning is the method of identifying and fixing (or deleting) corrupt or faulty information from a record set, table or archive and refers to the detection of missing, incorrect, corrupted or irrelevant parts of the data and the substitution, alteration or deletion of contaminated or gross data. Data cleaning can be done interactively with data wrangling tools, or as batch processing by scripting.

Data labeling is an important part of data pre-processing for ML, particularly for supervised learning, in which both input and output data are labeled for classification to provide a learning basis for future data processing.

6.2.2 Encoding Categorical Data

Feature encoding is one of the most important preprocessing steps in any machine learning project. It is the method of converting categorical data into numerical data in a data-set. We need to perform feature encoding since most models of machine learning can only interpret numerical data in text form and not data. In our data-set we have a "Machining Positioning" which is a categorical feature which is needed to be converted into numerical values. OneHOTEncoder is used to convert the categorical features into numerical one.

```
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer([('encoder', OneHotEncoder(), [47])], remainder='passthrough')
x= np.array(ct.fit_transform(x))
```

Figure 6-3: Python Code: OneHotEncoding

6.2.3 Feature Scaling

Feature scaling is essential for machine learning algorithm that tune all the features in the data to a same scale. It refers to putting the values to the same scale so that no feature is dominated by the other. There are two major scaling techniques which are used for feature scaling.

Standardization

Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance). By computing the relevant statistics on the samples in the training set, centering and scaling occur independently on each feature.

$$z = (x - \mu)/\sigma \tag{6.1}$$

For example, several elements used in a learning algorithm's objective function (such as the Support Vector Machines RBF kernel or linear model L1 and L2 regularizers) assume that all features are centered around 0 and differ in the same order. If a function has a variance that is greater than others in order of magnitude, it could overpower the objective function and make the estimator unable to correctly learn from other characteristics as expected.

Normalization

Database normalization is the structuring of the relational database, in order to reduce the redundancy and improve data integrity. The aim of normalization is to adjust the numeric column values in the dataset to use a common scale, without distorting variations in the value ranges or losing information.

Min-max Normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

$$x_{scaled} = (x - x_{min})/(x_{max} - x_{min}) \tag{6.2}$$

There is one drawback of Min-max normalization that it does not handle outliers very well.

6.2.4 Data Splitting

One of the first decision in developing a model is how to utilize the existing data. A common technique to split the data is as training and test set. To build models and feature sets, the training set is used; it is the substrate for estimating parameters, comparing models, and all the other activities needed to achieve a final model. The test set is used for estimating a final, unbiased evaluation of the model's success only at the end of these activities. It is critical that the test set not be used prior to this point. Looking at the test sets results would bias the outcomes since the testing data

will have become part of the model development process. There is no rule of thumb that how much data should be in the training and test sets. The proportion of data can be driven by many factors, including the size of the original pool of samples and the total number of predictors.

Wear or process parameters	Examples for measurement procedures and transducers
X1_ActualPosition	actual x position of part (mm)
X1_ActualVelocity	actual x velocity of part (mm/s)
X1_ActualAcceleration	actual x acceleration of part (mm/s/s)
X1_CommandPosition	reference x position of part (mm)
X1_CommandVelocity	reference x velocity of part (mm/s)
X1_CommandAcceleration	reference x acceleration of part (mm/s/s)
X1_CurrentFeedback	current (A)
X1_DCBusVoltage	voltage (V)
X1_OutputCurrent	current (A)
X1_OutputVoltage	voltage (V)
X1_OutputPower	power (kW)
Y1_ActualPosition	actual y position of part (mm)
Y1_ActualVelocity	actual y velocity of part (mm/s)
Y1_ActualAcceleration	actual y acceleration of part (mm/s/s)
Y1_CommandPosition	reference y position of part (mm)
Y1_CommandVelocity	reference y velocity of part (mm/s)
Y1_CommandAcceleration	reference y acceleration of part (mm/s/s)
Y1_CurrentFeedback	current (A)
Y1_DCBusVoltage	voltage (V)
Y1_OutputCurrent	current (A)
Y1_OutputVoltage	voltage (V)
Y1_OutputPower	power (kW)
Z1_ActualPosition	actual z position of part (mm)
Z1_ActualVelocity	actual z velocity of part (mm/s)
Z1_ActualAcceleration	actual z acceleration of part (mm/s/s)

Table 6.2: Examples for direct and indirect tool wear sensing methods

Wear or process parameters	Examples for measurement procedures and transducers
Z1_CommandPosition: reference z position of part (mm)	
Z1_CommandVelocity	reference z velocity of part (mm/s)
Z1_CommandAcceleration	reference z acceleration of part (mm/s/s)
Z1_CurrentFeedback	current (A) Z1_DCBusVoltage
voltage (V)	
Z1_OutputCurrent	current (A)
Z1_OutputVoltage	voltage (V)
S1_ActualPosition	actual position of spindle (mm)
S1_ActualVelocity	actual velocity of spindle (mm/s)
S1_ActualAcceleration	actual acceleration of spindle (mm/s/s)
S1_CommandPosition	reference position of spindle (mm)
S1_CommandVelocity	reference velocity of spindle (mm/s)
S1_CommandAcceleration	reference acceleration of spindle (mm/s/s)
S1_CurrentFeedback	current (A)
S1_DCBusVoltage	voltage (V)
S1_OutputCurrent	current (A)
S1_OutputVoltage	voltage (V)
S1_OutputPower	current (A)
S1_SystemInertia	torque inertia (kg*m^2)

Table 6.3: Examples for direct and indirect tool wear sensing methods

Chapter 7

Methodology Proposed

The proposed approach is to classify the tool as worn and unworn as described in figure 7-1

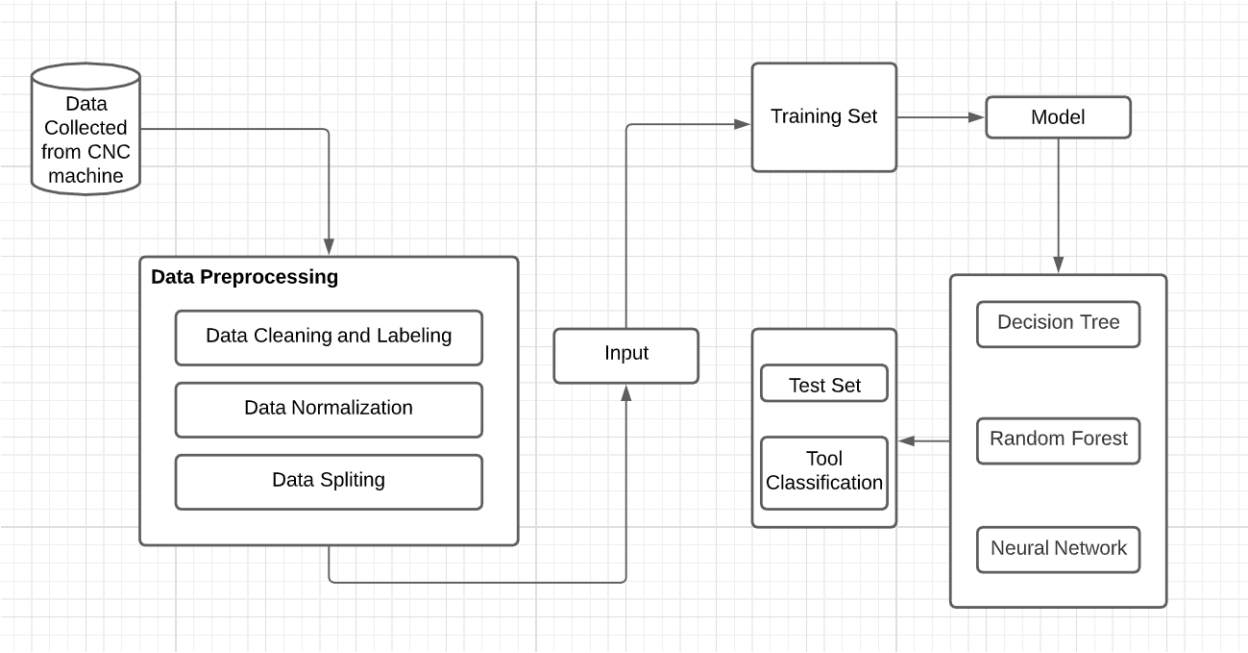

Figure 7-1: Python Code: Flow Chart

7.1 Data Pre-Processing

As explained in the chapter 6, the raw Data collected from System-level Manufacturing and Automation Research Testbed (SMART), University of Michigan is pre-processed for the analysis.

The whole data-set obtained from the CNC milling machine is exported to python by using pandas library. Firstly the required data from both the files is gathered and integrated in a single data-frame. Tool condition is assigned to all 18 experiments correspondingly. In the data cleaning part only those experiments are picked from the data-set in which machining is finalized.

In the data-set tool condition is given by the categorical labels, for the ease applying the machine learning models the data has given binary labels i-e

worn	unworn
0	1

7.2 Data Splitting

Firstly, the data splitting is done by using the train_test_split function of the scikit learn library. It splits the whole data-set, the size of test set is given as 0.3 which means 30% of the data will be in the test set and remaining 70% data will be in train set randomly.

```
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,random_state=0)
```

Figure 7-2: Python Code: Flow Chart

Secondly, the data is split manually in such a way that test set will take all the experiments one by one and all remaining will go in the train set. Similarly another split is done by assigning 2 consecutive experiments to the test set and remaining in

the train set. In this way accuracy of the models can be checked for several situations instead of just one random split which was done firstly.

7.3 Feature Scaling

Feature scaling is done by standardization, using the StandardScaler function of scikit learn library. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using transform.

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
x_train[:,11:] = sc_x.fit_transform(x_train[:,11:])
x_test[:,11:] = sc_x.transform(x_test[:,11:])
```

Figure 7-3: Python Code: Feature Scaling by Standardization

While doing the standardization, the columns which are generated during the OneHotEncoding are already in the binary form, so these columns are skipped from the standardization.

7.4 Feature Extraction

Feature extraction is the process of transforming raw data into meaningful features more suitable for a machine data mining task, which act as inputs for machine learning algorithms and help in improving the overall predictive model performance. Generally, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be explanatory and essential, simplifying the subsequent learning and modelling phases. A feature is simply an attribute on which the prediction is done.

7.5 Model Application

Machine Learning models are trained for the data-set containing containing the pre-processed data. Machine Learning algorithms which are usually used to classify the tool condition are decision tree, random forest and neural network. They are applied and accuracy of the model is tested by comparing the predicted values from the test set values. Results of the analysis are given in the following chapter.

7.6 Principal Component Analysis

Principal component analysis is the mathematical algorithm that reduces the dimensionality of the data by keeping most of the variations in the data. It reduces the features by identifying the distances, which are called principal components, along which the variation of the data is maximal. By using a few components, each sample can be represented by relatively few numbers instead of by values for thousands of variables.

Principal component analysis (PCA) is a technique that is useful for the compression and classification of data. The purpose is to reduce the dimensionality of a data set (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample's information. By information we mean the variation present in the sample, given by the correlations between the original variables. The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Chapter 8

Results

The criteria selected to evaluate the classification model is accuracy score and cross-validation score.

- **Accuracy Score**: Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$AccuracyScore = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions}$$

- **Cross-validation score**: Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

  The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation. In our reseach we have used 5-fold cross validation to validate the machine learning models.

8.1 By using train-test split for 18 experiments

In the first analysis we have concatenated all eighteen machining experiments in a single data-frame. After the concatenation of data data normalization, feature encoding and data splitting is applied which is already explained in chapter 6. Dataframe in this case includes 25286*49 of matrix. We are going to split this whole data frame, firstly into dependant and independent variable, secondly we used the function of train-test split from the model selection module of sklearn library.

After splitting the data-set into train-test sets, machine learning algorithms are applied whose results are as under:

Machine Learning Algorithm	Accuracy
Random Forest	99.193%
Decision Tree	98.956%
Neural Networks	88.755%
KNN	86.776%
kernel SVM	73.014%
Logistic Regression	53.732%

Table 8.1: By using train-test split

8.2 By using cross-validation approach

In the second analysis the data-set is distributed into training and test sets by creating a loop. During this analysis CNC machine experiments are divided in such a way that: 1) test set will get 1 experiment and the other 17 experiments will be in the train set i-e when the loop starts experiment # 1 will go to test set and the remaining other experiments will go to the training set. In the next loop experiment # 2 will go to the test set and the other experiments will be in the training set. Moreover this loop will go all over 18 times, providing us 18 accuracy values for the machine learning model.

2) test set will get 2 experiments and the other 16 experiments will be in train set.i-e when the loop starts experiment # 1 and experiment # 2 will go to test set and the remaining other experiments will go to the training set. In the next loop experiment # 2 and experiment # 4 will go to the test set and the other experiments will be in the training set. Moreover this loop will go all over 18 times, providing us 9 accuracy values for the machine learning model. Hence we will be getting 18 and 9 accuracies of one machine learning models at a time respectively.

In this method all possible classification accuracies in a model are extracted and their mean value will be considered as the model accuracy.

This analysis is done by considering all 18 experiments. In those 18 experiments 8 experiments are with "unworn" tool and 10 experiments are with "worn" tool.

Machine Learning Algorithm	Mean Accuracy	Mean Cross Validation Score
Random Forest (segmentation 17,1)	78.35%	90.50 %
Random Forest (segmentation 16,2)	66.47%	89.46%
Decision Tree (segmentation 17,1)	85.40%	89.49 %
Decision Tree (segmentation 16,2)	76.78%	90.08%
Neural Networks (segmentation 17,1)	44.63%	61.21%
Neural Networks (segmentation 16,2)	51.57%	54.11%

Table 8.2: By using cross-validation approach

8.3 Analysis with machining finalized data

In the third analysis strategy followed is the same as first and second analysis but experiments are reduced on the basis of finalization of machining operation. In this analysis only those experiments are considered for which the machining operation in finalized. Experiments are excluded on the basis of feature "Machining Finalize".

The analysis is done by considering 14 experiments for which machining is finalized. In those 14 experiments 6 experiments have unworn tool and other 8 experiments have worn tool.

Machine Learning Algorithm	Mean Accuracy	Mean Cross Validation Score
Random Forest (segmentation 13,1)	86.97%	95.74%
Random Forest (segmentation 12,2)	81.80%	95.49%
Decision Tree (segmentation 13,1)	89.89%	92.70%
Decision Tree (segmentation 12,2)	80.47%	93.60%
Neural Networks (segmentation 13,1)	49.07%	55.67%
Neural Networks (segmentation 12,2)	49.46%	56.71%

Table 8.3: Analysis with Machining Finalized experiments

8.4 Feature extraction

Firstly, feature extraction is done by considering 14 experiments for which machining is finalized. In those 14 experiments 6 experiments have unworn tool and other 8 experiments have worn tool. Mean, median, kurtosis and skewness are selected at the

same time. Iterations are done with a data-frame of 14×189.

Machine Learning Algorithm	Mean Accuracy	Mean Cross Validation Score
Random Forest (segmentation 13,1)	42.85%	73.33%
Random Forest (segmentation 12,2)	42.85%	46.66%
Decision Tree (segmentation 13,1)	42.5%	46.66%
Decision Tree (segmentation 12,2)	50%	56.66%
Neural Networks (segmentation 13,1)	42.85%	50%
Neural Networks (segmentation 12,2)	28.57%	36.66%

Table 8.4: Results after Feature Extraction

8.5 Feature Reduction

Feature reduction, also known as reduction of dimensionality, is the method of reducing the number of features in a heavy computing resource without losing essential details. Reducing the number of characteristics requires reducing the number of factors, making it simpler and quicker for the machine to operate.There are several methods by which feature reduction is done. Generalized discriminant analysis, auto-encoders, non-negative matrix factorization, select-from-model, and principal component analysis are some of the most common. In our analysis we have used principal component analysis and SelectfromModel techniques for applying feature reduction.

8.5.1 Principal component analysis

Principal component analysis is applied to the machine learning model by setting the component variability as 95%. Results of principal component analysis are shown in the table below.

```
from sklearn.decomposition import PCA
pca = PCA(0.95)
pca.fit(x)
x= pca.transform(x)
x_test= pca.transform(x_test)
```

Figure 8-1: Python Code: Principal Component Analysis

Machine Learning Algorithm	Mean Accuracy	Mean Cross Validation Score
Random Forest (segmentation 13,1)	42.24%	62.25%
Random Forest (segmentation 12,2)	47%	57.12%
Decision Tree (segmentation 13,1)	50.3%	60.2%
Decision Tree (segmentation 12,2)	49.4%	57.2%
Neural Networks (segmentation 13,1)	46.8%	63.4%
Neural Networks (segmentation 12,2)	44.9%	55.8%

Table 8.5: Principal Component Analysis

8.5.2 Feature Reduction by using SelectFromModel

SelectFromModel is an effective tool of scikit learn library, which selects features on the basis of importance wiegths. Decision Tree and Random Forest models have a

built in method which generates the importance of the features according to their weights in the data-set.

By applying SelectFromModel, features are reduced from 48 to 12. Results for the analysis are given in the table below.

Machine Learning Algorithm	Mean Accuracy	Mean Cross Validation Score
Random Forest (segmentation 13,1)	86.94%	96.60%
Random Forest (segmentation 12,2)	82.77%	96.56%
Decision Tree (segmentation 13,1)	88.08%	92.47%
Decision Tree (segmentation 12,2)	84.10%	94.86%

Table 8.6: Feature Reduction by using SelectFromModel

Chapter 9

Conclusion, Limitations and Future Recomendations

In the light of the above-stated results, we can conclude, that we have achieved the objective of this research which was to classify the tool condition (worn, unworn) during a milling operation, by using machine learning models. We have mainly used the following models to classify the tool condition:

- Random forest
- Decision tree
- NeuralNeural networks
- K-Nearest Neighbor
- Kernel SVM
- Logistic Regression

Finally, looking at the evaluation parameters, mean accuracy scores, and mean cross-validation scores we can conclude that Random forest, Decision tree, and Neural network approach are providing more accurate predictions as compared to others. While conducting the analysis with all 18 experiments concatenated to form a data-set and splitting them by using train-test split method, we have acquired the predicting

accuracy of 99.193%. The machine learning model is efficient but assuming it in a real case scenario it may be overfitting or giving biased results.

we have applied the cross-validation approach for getting the unbiased predicting accuracies for all the experiments as test set, and giving a single mean accuracy value for the analysis. This analysis also concluded with a higher mean accuracy value for Random Forest algorithm, with an accuracy value of 78.35% and cross-validation score of 90.50%. It shows that the for predicting the tool condition in a milling operation random forest algorithm can be beneficial.

The research has concluded that the strength of features is more important than the number of features for training a machine learning model. Principal component analysis shows that our features are weak to extract any of them as the significant features. It's beneficial to have features who have a quite significant variation with the tool condition at any point in time.

Machine learning framework as discussed above can play an important role in decreasing the downtime during a milling process and in return can increase the production efficiency and product quality. It could be possible by the implementation cyber-physical systems more precisely for attaining data-set with some strong features.

Now-a-days for any machine learning model the main limitation is the availability of the data-set with some strong features. Due to the COVID most of the companies has closed or temporarily stopped their ongoing projects. For the same reason i did not get the data-set which i was supposed to get from an italian car prototyping company. The unavailability of the data with required features was the main hurdle in my research.

Further improvements for the smart manufacturing system for tool condition and machining process monitoring involves the development of effective data preparation methodology to produce more general data in the corresponding machining process and the advanced signal processing methods can be applied for feature generation. Acoustic and vibration signals can be used for generating strong features. In addition to the Neural Networks, convolution neural networks (CNN) and reinforced machine

learning techniques can be used to get more accurate predictions. Work also remains regarding tool wear regression-based classification in place of discrete classification; such an investigation will require close monitoring of tool condition at various points during tool life. Another important development could be the prediction of the remaining useful life of the tool (RUL), which can be beneficial for optimizing the maintenance operation.

Appendix

Phython Code

8.1

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


    frames = list()
results = pd.read_csv("train.csv")
for i in range(1,19):
exp = '0' + str(i) if i < 10 else str(i)
frame = pd.read_csv("experiment_.csv".format(exp))
row = results[results['No'] == i]
frame['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1
frames.append(frame)
df = pd.concat(frames, ignore_index = True)
df.head()
df.replace('Machining_Process': 'Starting':'Prep','end':'End',
inplace=True)
print(df['Machining_Process'].value_counts().sort_index())


    x = df.iloc[:,:-1].values
```

```
y = df.iloc[:,48].values

    from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer([('encoder', OneHotEncoder(), [47])], remainder='passthrough')
x= np.array(ct.fit_transform(x))

    from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.25,random_state=0)

    from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
x_train[:,8:] = sc_x.fit_transform(x_train[:,8:])
x_test[:,8:] = sc_x.transform(x_test[:,8:])

    from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion= 'entropy', random_state=0) classifier.fit(x_train,y_train)
from sklearn.model_selection import cross_val_score
clf = cross_val_score(classifier,x_train,y_train)
y_pred = classifier.predict(x_test)

    from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)

    print (cm)
ac = accuracy_score(y_test,y_pred)
print (ac)
imp= classifier.feature_importances_
```

8.2

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import random
```

imported all the libraries

```
    frames = list()
results = pd.read_csv("train.csv")
for i in range(1,19):
exp = '0' + str(i) if i < 10 else str(i)
frame = pd.read_csv("experiment_.csv".format(exp))
row = results[results['No'] == i]
if row.iloc[0]['machining_finalized'] == 'yes':
frame['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1
frames.append(frame)
else:
i+=1
```

imported all experiments in which machining is finalized
giving the class labels worn = 0 and unworn = 1
appended all the experiments in the list called "frames"

```
    accuracy = list()
cross_val_scorelist= list()
```

Def a variable "accuracy" in the list type to store the accuracy scores.
Def a variable "cross_val_scorelist" in the list type to store the cross val-

idation scores.

for c in range(0,14):

generated a "for loop" in the range 0-14 for spliting the train and test sets in each iteration"

df_testlist = list()

df_trainlist = list()

Def two list type variables "df_testlist" and "df_trainlist" to store train and test experiments at each instance.

for a in range (0,14):

if a==c:

print('c', a)

df_test= frames[a]

if a==0:

df_test.replace('Machining_Process': 'Starting':'Prep','end':'End', inplace=True)

df_testlist.append(df_test)

appended the experiment which is test set for the instance to the list

else:

print(a,' ')

df_train= frames[a]

if a==0:

df_train.replace('Machining_Process': 'Starting':'Prep','end':'End', inplace=True)

df_trainlist.append(df_train)

appended all other experiments in the train set list.

a+=1

print('———')

df = pd.concat(df_trainlist, ignore_index = True)

concatenated all the experiments in the train list for applying the model.

```
x = df.iloc[:,:-1].values
```

assigned the columns for the independent variable x

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer([('encoder', OneHotEncoder(), [47])], remainder='passthrough')
x= np.array(ct.fit_transform(x))
```

OneHotEncoding is done to convert the categorial features into numerical one.

```
y = df.iloc[:,48].values
```

assigned the column for dependent variable "Tool Condition"

```
x_test = df_test.iloc[:,:-1].values
x_test= np.array(ct.fit_transform(x_test))
y_test = df_test.iloc[:,48].values
```

Feature scaling by using Standardization

```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
x= sc.fit_transform(x)
x_test= sc.transform(x_test)
```

Applying machine learning model to the data set i-e x and y will be independent and dependent variables respectively.

Classifier is trained for the respective model (Random Forest, Decision Tree, and Neural Networks) with the given dataset.

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 50, criterion= 'entropy', random_state= 0)
classifier.fit(x,y)
```

cross_val_score returns score of test fold where cross_val_predict returns predicted y values for the test fold

```
from sklearn.model_selection import cross_val_score
clf = cross_val_score(classifier,x,y)
cross_val_scorelist.append(clf)
```

Machine learning model(Random Forest) predict the value of y for the corresponding x_test.

```
y_pred = classifier.predict(x_test)
```

Accuracy is one metric for evaluating classification models.
Informally, accuracy is the fraction of predictions our model got right.

```
from sklearn.metrics import accuracy_score
ac = accuracy_score(y_test,y_pred)
accuracy.append(ac)
```

```
c+=1
```

```
import statistics
```

mean accuracy score for all iterations.

```
mean= statistics.mean(accuracy)
print (mean)
```

mean cross validation score for all iterations.

```
cvm = statistics.mean(clf)
print (cvm)
```

8.4

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import random
```

imported all the libraries required.

```
    means = list()
medians = list()
kurtosiss = list()
skewnesss = list()
stds=list()
frames = list()
results = pd.read_csv("train.csv")
```

imported all experiments in which machining is finalized giving the class labels worn = 0 and unworn = 1 appended all the experiments in the list called "frames"

```
    for i in range(1,19):
exp = '0' + str(i) if i < 10 else str(i)
frame = pd.read_csv("experiment_.csv".format(exp))
row = results[results['No'] == i]
mean = frame.mean(axis=0)
median = frame.median(axis=0)
kurtosis = frame.kurtosis(axis=0)
skewness = frame.skew(axis=0)
std= frame.var(axis=0)
```

Calculated the features (mean, median,kurtosis and skewness) for the feature extraction

mean['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1

means.append(mean)

median['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1

medians.append(median)

kurtosis['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1

kurtosiss.append(kurtosis)

skewness['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1

skewnesss.append(skewness)

frame['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1

frames.append(frame)

i+=1

making the lists of required features respectively.

df_me = pd.concat(means, axis=1 , join='inner')

df_md = pd.concat(medians, axis=1 , join='inner')

df_ku = pd.concat(kurtosiss, axis=1 , join='inner')

df_skw = pd.concat(skewnesss, axis=1 , join='inner')

df_std = pd.concat(stds, axis=1 , join='inner')

Concatenated all the features to convert list into a dataframe.

Taking transpose of features for further analysis.

df_me=df_me.transpose()

df_md=df_md.transpose()

df_ku=df_ku.transpose()

df_skw=df_skw.transpose()

df_std=df_std.transpose()

Concatenated all the features in a single dataframe. df = pd.concat([df_me.drop('TC',axis=1),d

axis=1 , join='inner')

```
df.insert(0, "No", [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17], True)
```

Def a variable "accuracy" in the list type to store the accuracy scores. Def a variable "cross_val_scorelist" in the list type to store the cross validation scores.

```
accuracy = list()
cross_val_scorelist= list()
```

generated a "for loop" in the range 0-18 for spliting the train and test sets in each iteration"

```
for c in range(0,18):
```

Def two list type variables "df_testlist" and "df_trainlist" to store train and test experiments at each instance.

```
df_testlist = list()
df_trainlist = list()
for a in range (0,18):
if a==c:
print('c', c)
df_test= df[df["No"] == c]
df_test=df_test.drop(labels= 'No',axis= 1)
df_testlist.append(df_test)
```

appended the experiment which is test set for the instance to the list

```
else:
print(a,' ')
df_train= df[df["No"] == a]
df_train= df_train.drop(labels='No',axis= 1)
df_trainlist.append(df_train)
```

```
a+=1
print('———')
me = pd.concat(df_trainlist, ignore_index = True)
```

concatenated all the experiments in the train list for applying the model

```
x = me.iloc[:,:-1].values
```

assigned the columns for the independent variable x

```
y = me.iloc[:,188].values
y=y.astype('int')
x_test = df_test.iloc[:,:-1].values

y_test = df_test.iloc[:,188].values
y_test=y_test.astype('int')
```

Feature scaling by using Standardization

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
x = sc_x.fit_transform(x)
x_test = sc_x.transform(x_test)
```

Applying machine learning model to the data set i-e x and y will be independent and dependent variables respectively. Classifier is trained for the respective model (Random Forest, Decision Tree, and Neural Networks) with the given dataset.

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion= 'entropy', random_state=0)
```

```
classifier.fit(x,y)
```

cross_val_score returns score of test fold where cross_val_predict returns predicted y values for the test fold

```
    from sklearn.model_selection import cross_val_score
clf = cross_val_score(classifier,x,y)
cross_val_scorelist.append(clf)
```

Machine learning model(DecisionTreeClassfier) predict the value of y for the corresponding x_test.

```
y_pred = classifier.predict(x_test)
```

```
    from sklearn.metrics import accuracy_score
ac = accuracy_score(y_test,y_pred)
accuracy.append(ac)
```

```
    c+=1
import statistics
```

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

```
mean= statistics.mean(accuracy)
print (mean)
```

mean cross validation score for all iterations.

```
cvm = statistics.mean(clf)
print (cvm)
```

8.5

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import random
```

imported all the libraries

```
    frames = list()
results = pd.read_csv("train.csv")
for i in range(1,19):
exp = '0' + str(i) if i < 10 else str(i)
frame = pd.read_csv("experiment_.csv".format(exp))
row = results[results['No'] == i]
if row.iloc[0]['machining_finalized'] == 'yes':
frame['TC'] = 0 if row.iloc[0]['tool_condition'] == 'worn' else 1
frames.append(frame)
else:
i+=1
```

imported all experiments in which machining is finalized
giving the class labels worn = 0 and unworn = 1
appended all the experiments in the list called "frames"

```
    accuracy = list()
cross_val_scorelist= list()
```

Def a variable "accuracy" in the list type to store the accuracy scores.
Def a variable "cross_val_scorelist" in the list type to store the cross val-

idation scores.

for c in range(0,14):

generated a "for loop" in the range 0-14 for spliting the train and test sets in each iteration"

df_testlist = list()

df_trainlist = list()

Def two list type variables "df_testlist" and "df_trainlist" to store train and test experiments at each instance.

for a in range (0,14):

if a==c:

print('c', a)

df_test= frames[a]

if a==0:

df_test.replace('Machining_Process': 'Starting':'Prep','end':'End', inplace=True)

df_testlist.append(df_test)

appended the experiment which is test set for the instance to the list

else:

print(a,' ')

df_train= frames[a]

if a==0:

df_train.replace('Machining_Process': 'Starting':'Prep','end':'End', inplace=True)

df_trainlist.append(df_train)

appended all other experiments in the train set list.

a+=1

print('———')

df = pd.concat(df_trainlist, ignore_index = True)

concatenated all the experiments in the train list for applying the model.

```
x = df.iloc[:,:-1].values
```

assigned the columns for the independent variable x

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer([('encoder', OneHotEncoder(), [47])], remainder='passthrough')
x= np.array(ct.fit_transform(x))
```

OneHotEncoding is done to convert the categorial features into numerical one.

```
y = df.iloc[:,48].values
```

assigned the column for dependent variable "Tool Condition"

```
x_test = df_test.iloc[:,:-1].values
x_test= np.array(ct.fit_transform(x_test))
y_test = df_test.iloc[:,48].values
```

Feature scaling by using Standardization

```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
x= sc.fit_transform(x)
x_test= sc.transform(x_test)
```

```
from sklearn.decomposition import PCA
pca = PCA(0.95)
pca.fit(x)
x= pca.transform(x)
x_test= pca.transform(x_test)
```

imported PCA from sklearn.decomposition for applying Principal Component Analysis.

the algorithm will select the number of components while preserving 95%

of the variability in the data.

Applying machine learning model to the data set i-e x and y will be independent and dependent variables respectively.
Classifier is trained for the respective model (Random Forest, Decision Tree, and Neural Networks) with the given dataset.

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 50, criterion= 'entropy', random_state= 0)
classifier.fit(x,y)
```

cross_val_score returns score of test fold where cross_val_predict returns predicted y values for the test fold

```
from sklearn.model_selection import cross_val_score
clf = cross_val_score(classifier,x,y)
cross_val_scorelist.append(clf)
```

Machine learning model(Random Forest) predict the value of y for the corresponding x_test.

```
y_pred = classifier.predict(x_test)
```

Accuracy is one metric for evaluating classification models.
Informally, accuracy is the fraction of predictions our model got right.

```
from sklearn.metrics import accuracy_score
ac = accuracy_score(y_test,y_pred)
accuracy.append(ac)
```

```
c+=1
```

```
import statistics
```

mean accuracy score for all iterations.

mean= statistics.mean(accuracy)

print (mean)

mean cross validation score for all iterations.

cvm = statistics.mean(clf)

print (cvm)

Bibliography

[1] Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., and Zhang, J. (August 13, 2020). "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook." ASME. J. Manuf. Sci. Eng. November 2020; 142(11): 110804.

[2] Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D., 2014, "Prognostics and Health Management Design for Rotary Machinery Systems —Reviews, Methodology and Applications," Mech. Syst. Signal Process., 42(1–2), pp. 314–334.

[3] ge.com/research/project/predictive-maintenance.

[4] Lecun, Y., Bengio, Y., and Hinton, G., 2015, "Deep Learning," Nature, 521(7553), pp. 436–444.

[5] Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., and Helu, M., 2015, "Cloud-Enabled Prognosis for Manufacturing," CIRP Ann., 64(2), pp. 749–772.

[6] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld, 2010, "Advanced monitoring of machining operations," CIRP Ann-Manuf Techn, 59(2), pp. 717-73.

[7] Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., and Zhang, J. (August 13, 2020). "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook." ASME. J. Manuf. Sci. Eng. November 2020; 142(11): 110804.

[8] X.S. Si, W. Wang, C.H. Hu, M.Y. Chen, and D.H. Zhou, 2013, "A wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation," Mechanical Systems and Signal Processing, 35(1), pp. 219-237.

[9] M. Dong and D. He, 2007, "Hidden semi-Markov model-based methodology for

multi-sensor equipment health diagnosis and prognosis," European Journal of Operational Research, 178(3), pp. 858-878.

[10] Saha, K. Goebel, and J. Christophersen, 2009, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," T I Meas Control.

[11] M. E. Orchard and G. J. Vachtsevanos, 2009, "A particle-filtering approach for on-line fault diagnosis and failure prognosis," T I Meas Control.

[12] B. Sick, 2002, "On-line and indirect tool wear monitoring in turning with artificial neural networks: a review of more than a decade of research," Mechanical Systems and Signal Processing, 16(4), pp. 487-546.

[13] Kothuru, A., Nooka, S. P.,and Liu, R. (August 3,2018). "Audio-Based Tool Condition Monitoring in Milling of the Workpiece Material With the Hardness Variation Using Support Vector Machines and Convolutional Neural Networks." ASME. J.Manuf. Sci. Eng.November 2018; 140(11):111006.

[14] Sohyung Cho, Shihab Asfour, Arzu Onar, Nandita Kaundinya, Tool breakage detection using support vector machine learning in a milling process, International Journal of Machine Tools and Manufacture, Volume 45, Issue 3,2005,

[15] Clayton Cooper, Peng Wang, Jianjing Zhang, Robert X. Gao, Travis Roney, Ihab Ragai, Derek Shaffer, Convolutional neural network-based tool condition monitoring in vertical milling operations using acoustic signals, Procedia Manufacturing, Volume 49, 2020,

[16] Wu, Zhenhua. "Cutting Tool Condition Monitoring and Prediction Based on Dynamic Data Driven Approaches." Proceedings of the ASME 2015 International Manufacturing Science and Engineering Conference. Volume 1: Processing. Charlotte, North Carolina, USA. June 8–12, 2015

[17] Li, X., et al. Fuzzy neural network modelling for tool wear estimation in dry milling operation. in Annual conference of the prognostics and health management society. 2009.

[18] D. Wu, C. Jennings, J. Terpenny and S. Kumara, "Cloud-based machine learning for predictive analytics: Tool wear prediction in milling," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 2062-2069

[19] Kurfess, T., Saldana, C., Saleeby, K., and Parto-Dezfouli, M. (August 28, 2020). "INDUSTRY 4.0 AND INTELLIGENT MANUFACTURING PROCESSES: A REVIEW OF MODERN SENSING TECHNOLOGIES." ASME. J. Manuf. Sci. Eng

[20] I. Dumitrache and S. I. Caramihai, "The Enterprise of Future as a Cyber-Physical System," IFAC Proceedings Volumes, vol. 46, no. 9, pp. 1310-1315, 2013/01/01/2013

[21] Alothman, Hussam Ali, Khasawneh, Mohammad T., and Nagarur, Nagen N. "Internet of Things in Manufacturing: An Overview." Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition. Volume 2: Advanced Manufacturing. Pittsburgh, Pennsylvania, USA. November 9–15, 2018

[22] W.Z. Khan, M.H. Rehman, H.M. Zangoti, M.K. Afzal, N. Armi, K. Salah, Industrial internet of thing: Recent advances, enabling technologies and open challenges, Computers Electrical Engineering, Volume 81,2020, 106522,ISSN 0045-7906,

[23] Bergonzi, L, Colombo, G, Rossoni, M, Furini, F. "Data and Knowledge in IIoT-Based Maintenance Application." Proceedings of the ASME 2017 International Mechanical Engineering Congress and Exposition. Volume 11: Systems, Design, and Complexity. Tampa, Florida, USA. November 3–9, 2017

[24] Zheng, P., wang, H., Sang, Z. et al. Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. Front. Mech. Eng. 13, 137–150 (2018).

[25] Jim Davis, Thomas Edgar, James Porter, John Bernaden, Michael Sarli, Smart manufacturing, manufacturing intelligence and demand-dynamic performance, Computers Chemical Engineering, Volume 47, 2012,

[26] A. Cachada et al., "Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture," 2018 IEEE 23rd International Conference on Emerging Tech-

nologies and Factory Automation (ETFA), Turin, 2018,

[27] Li, Y, Jia, D. "A Real-Time Analysis of Condition-Based Maintenance in a Multistage Production System." Proceedings of the ASME 2018 13th International Manufacturing Science and Engineering Conference. Volume 1: Additive Manufacturing; Bio and Sustainable Manufacturing. College Station, Texas, USA. June 18–22, 2018

[28] Yang, Y, Gao, RX, Fan, Z, Wang, J, Wang, L. "Cloud-Based Prognosis: Perspective and Challenge." Proceedings of the ASME 2014 International Manufacturing Science and Engineering Conference collocated with the JSME 2014 International Conference on Materials and Processing and the 42nd North American Manufacturing Research Conference. Volume 1: Materials; Micro and Nano Technologies; Properties, Applications and Systems; Sustainable Manufacturing. Detroit, Michigan, USA. June 9–13, 2014.

[29] Hassan, M., Sadek, A., Attia, M. H., and Thomson, V. (December 18, 2017). "A Novel Generalized Approach for Real-Time Tool Condition Monitoring." ASME. J. Manuf. Sci. Eng. February 2018

[30] M. P. Groover, Fundamentals of modern manufacturing: Materials, Processes, and Systems,1996.

[31] Z. Ge, Z. Song, S. X. Ding and B. Huang, "Data Mining and Analytics in the Process Industry: The Role of Machine Learning," in IEEE Access, vol. 5, pp. 20590-20616, 2017, doi: 10.1109/ACCESS.2017.2756872.

[32] Rehan Sadiq, Manuel J. Rodriguez, Haroon R. Mian, Empirical Models to Predict Disinfection By-Products (DBPs) in Drinking Water: An Updated Review, Editor(s): Jerome Nriagu, Encyclopedia of Environmental Health (Second Edition), Elsevier, 2019, ISBN 9780444639523,

[33]