IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

HEALTH MONITORING OF VMC MACHINE TOOL USING ARDUINO UNO AND MACHINE LEARNING ALGORITHMS

[1]Nilesh C Dhobale, [2]Krishagana S Chauhan, [3]Tejas S Gaikwad
[4]Prathamesh V Murkute, [5]Rohit R Naik
[1]Professor, [2]Student, [3]Student, [4]Student, [5]Student

Department of Mechanical Engineering,
ISBM College of Engineering, Pune. India.

Abstract: **:** This research work presents machine learning based condition monitoring of milling cutter of vertical machining center (VMC). The vibration signals acquisition of 4 insert milling cutter is carried out with healthy and various fault conditions. Programming in C code and script is used to extract statistical features and decision tree algorithm is used to select relevant features. Different conditions of milling cutter are classified using tree family classifiers. The effort made in this work is to check applicability of ML approach for milling cutter fault diagnosis for reducing power consumption of drive of machine tool.

Index Terms **- Tool wear, Vibrations, Failures, Machining.**

I. INTRODUCTION

Machine tool health monitoring is essential for continuous assessment of the machine's condition and performance. Traditional monitoring methods often rely on predefined thresholds or rules, which may not capture subtle changes or complex patterns in sensor data. Leveraging AI techniques, including machine learning and data analytics, can enhance the accuracy and reliability of VMC machine tool health monitoring.
Tool wear and tear is a significant concern in machining operations as it directly affects the quality, efficiency, and cost of manufacturing processes. This literature review aims to provide an overview of the existing research on tool wear and tear, including its causes, effects, detection, and mitigation strategies.

II. OBJECTIVE:

1. For easier and prior detection of tool wear.
2. Improving surface finish of the work piece.
3. Increasing the efficiency and accuracy of the Machine

III. TOOLS AND COMPONENTS :

1. VMC (VERTICAL MACHINING CENTRE) TECHNOLOGY:

Introduction: This section introduces the significance of VMC machines in the manufacturing industry. It outlines the objectives of the research paper, highlighting the importance of enhancing performance and productivity in VMC operations. The paper emphasizes the need for efficient machining processes to meet the demands of modern manufacturing requirements. Overview of VMC Machines: This section provides a comprehensive overview of VMC machines, covering their structure, components, and working principles. It explains the key features and advantages of VMC machines over other machining technologies. The discussion includes spindle configurations, tooling systems, and axis movements, providing a foundation for further exploration.

Figure 1 Vertical Machining Centre Machine

b.Performance Evaluation of VMC Machines: This section focuses on evaluating the performance of VMC machines. It discusses metrics such as accuracy, repeatability, surface finish, and productivity. The paper explores the factors influencing performance, including machine design, cutting parameters, tool selection, and workpiece material. The challenges and limitations associated with VMC machine performance are also addressed.

c.Optimization Techniques for VMC Machines: To enhance the performance and productivity of VMC machines, this section explores various optimization techniques. It discusses strategies for tool path planning, cutting parameter optimization, fixture design, and workpiece setup. Additionally, the paper presents advancements in adaptive control, process monitoring, and real-time optimization, providing insights into cutting-edge approaches to optimize VMC operations.

d.Advanced Technologies in VMC Machines: This section highlights advanced technologies and innovations that contribute to the advancement of VMC machines. It discusses the integration of automation, robotics, and Internet of Things (IoT) capabilities, which enhance productivity, precision, and process control. The paper also explores emerging technologies such as additive manufacturing and hybrid machining, showcasing their potential impact on VMC machines.

e.Applications and Case Studies: This section presents real-world applications and case studies where VMC machines have been successfully employed. It covers a wide range of industries, including aerospace, automotive, medical, and general manufacturing. The paper highlights specific challenges faced in these industries and how VMC machines have addressed them effectively.

f.Future Directions and Challenges: The paper concludes with a discussion on future directions and challenges in the field of VMC machines. It identifies potential areas for further research and development, including multi-axis machining, advanced cutting tools, and intelligent control systems. The challenges associated with complex geometries, high-speed machining, and sustainable manufacturing are also addressed.

1.1. MACHINE VIBRATION:

Vibration in a VMC (Vertical Machining Center) machine due to tool wear is a common issue that can affect machining accuracy, surface finish, and tool life. As cutting tools wear down during machining operations, several factors can contribute to vibrations:
Uneven Cutting Forces: As the cutting edges wear, the forces acting on the tool become unbalanced. This can lead to uneven cutting forces, causing the machine to vibrate.
Increased Cutting Resistance: Worn-out tools may experience increased cutting resistance, which can result in higher forces acting on the tool and, consequently, vibrations in the machine.
Poor Surface Finish: As the tool wears, it may produce a poor surface finish on the workpiece. This can create additional vibrations as the tool interacts with the workpiece, leading to degraded machining performance.
To address vibrations in a VMC machine due to tool wear, consider the following steps:
Tool Inspection and Replacement: Regularly inspect the cutting tools for signs of wear or damage. Implement a tool replacement or regrinding strategy based on predefined wear criteria to ensure that tools are replaced before excessive wear occurs.
Optimize Cutting Parameters: Adjust the cutting parameters such as cutting speed, feed rate, and depth of cut based on the condition of the tool. Modifying the parameters can help compensate for the tool wear and reduce vibrations.
Use High-Quality Cutting Tools: Select high-quality cutting tools that are designed for durability and longevity. Higher-quality tools can maintain their cutting performance for longer periods, reducing the occurrence of vibrations due to wear.
Monitor Tool Life: Implement a tool life monitoring system to track the usage and wear of cutting tools. This can help predict tool failure and schedule timely tool replacements, minimizing vibrations caused by worn-out tools.
Cutting Tool Maintenance: Follow recommended maintenance practices for cutting tools. This may include cleaning, lubrication, and periodic inspections to ensure optimal tool performance. Additionally, it is important to maintain a consistent tool inventory and consider using tool wear compensation techniques in the machine control system.

1.2. END MILLING TOOL:

End Milling Tool Design and Geometry
This section discusses the design principles and geometries of end milling tools. It explores the various components of an end mill, including the shank, flute, cutting edges, and helix angle. The influence of tool geometry on cutting forces, chip evacuation, surface finish, and stability during machining is explored in detail. The section also discusses the impact of tool material selection on tool performance.

Coatings and Tool Materials

The importance of coatings and tool materials in end milling is addressed in this section. Various coating techniques, such as TiN, TiAlN, and DLC, are discussed along with their benefits in terms of reducing friction, improving wear resistance, and prolonging tool life. Different tool materials, including high-speed steel (HSS), carbide, and ceramic, are also evaluated for their suitability in specific machining applications.

Cutting Parameters Optimization

This section delves into the optimization of cutting parameters for end milling tools. It discusses the influence of parameters such as cutting speed, feed rate, axial depth of cut, and radial depth of cut on machining performance. Optimization techniques, including response surface methodology (RSM), genetic algorithms, and artificial intelligence approaches, are explored to maximize material removal rate, minimize cutting forces, and achieve optimal surface finish.

Tool Wear and Tool Life Prediction

The prediction of tool wear and tool life is crucial for efficient machining operations. This section presents various models and approaches for tool wear prediction in end milling. It discusses the analysis of wear mechanisms, including flank wear, crater wear, and edge chipping. The use of wear models and sensor-based monitoring systems for real-time tool life estimation is also explored.

Surface Finish and Machining Quality

Achieving superior surface finish is a key objective in end milling operations. This section examines the factors influencing surface roughness and explores the optimization techniques for improving surface quality. It covers strategies such as vibration damping, adaptive control, and tool path optimization to minimize surface roughness and achieve desired surface integrity.

Applications and Case Studies

This section presents practical applications and case studies highlighting the successful optimization of end milling tools. It covers various industries, including automotive, aerospace, mold and die, and general machining, to showcase the effectiveness of optimization techniques in real-world scenarios.

Figure 2 End mills tool

.

IV. METHODOLOGY:

The strategy of tool wear prediction using vibrations generally based on the concept, where the vibrations caused due to the machining process is correlated with the tool wear phenomenon. In order to provide a robust system that can predict the tool wear, thus increasing the machining performance. To overcome the problem a prototype was designed and assembled that could be attached to tool shank and the experiments were conducted on a conventional lathe machine. Measured parameters were acceleration for detection of vibrations in all the three axes which were measured for different inputs of machining parameters.

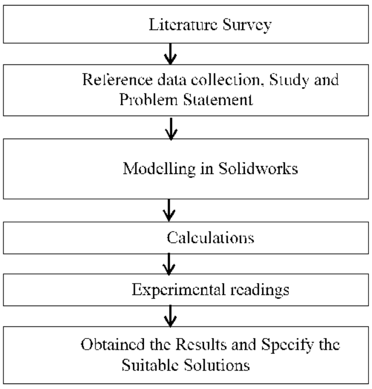

Figure 3 Research Work

V. CONSTRUCTION:

2. ESP8266:

The ESP8266 is a popular and widely used Wi-Fi module that combines a microcontroller with Wi-Fi capabilities. It was introduced in 2014 by Espressif Systems, a Chinese company specializing in IoT (Internet of Things) solutions. The ESP8266 module has gained significant popularity due to its low cost, small size, and ease of use.
Key features of the ESP8266 module include:
Wi-Fi Connectivity: The ESP8266 module includes built-in Wi-Fi capabilities, allowing it to connect to wireless networks and communicate with other devices over the internet or local network.
Microcontroller: The module incorporates a microcontroller unit (MCU) based on the Xtensa LX106 architecture. It operates at clock speeds of up to 80 MHz and has a range of digital and analog input/output pins for interacting with external components.
GPIO Pins: The ESP8266 module provides a number of General Purpose Input/Output (GPIO) pins that can be used for digital input/output, PWM (Pulse Width Modulation), I2C, SPI, and other communication protocols.
Programming: The ESP8266 can be programmed using various programming languages and development platforms, including Arduino IDE, Lua scripting language (with NodeMCU firmware), MicroPython, and more. This flexibility makes it accessible to a wide range of developers.
Development Boards: Several development boards based on the ESP8266 module are available, such as the NodeMCU, ESP-12E, and ESP-01, which provide convenient interfaces and headers for easy prototyping and integration into projects.
IoT Applications: Due to its low cost, compact size, and Wi-Fi connectivity, the ESP8266 is commonly used in IoT applications for home automation, sensor monitoring, wireless control systems, smart devices, and Internet-connected projects.
The ESP8266 has been widely adopted by the maker community and has spurred the development of numerous open-source projects, libraries, and resources. It has been succeeded by newer versions like the ESP32, which offers more features and capabilities, but the ESP8266 remains a popular choice for many IoT projects due to its affordability and simplicity

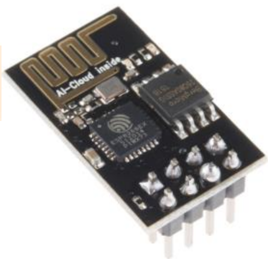

Figure 4. The ESP8266 module

3. MPU6050:

The MPU6050 is a popular integrated circuit (IC) that combines a 3-axis accelerometer and a 3-axis gyroscope in a single package. It is commonly used for motion sensing and orientation tracking in various electronic applications. The MPU6050 is manufactured by InvenSense, a company now owned by TDK.

Figure 5. MPU6050

Here are some key features and details about the MPU6050:
Accelerometer: The MPU6050 includes a 3-axis accelerometer, which measures linear acceleration along three axes (X, Y, and Z). It can detect changes in velocity, tilt, and motion.
Gyroscope: The IC also features a 3-axis gyroscope, which measures angular velocity or rotational motion along the X, Y, and Z axes. It provides information about the device's orientation and rotational movement.
Digital Motion Processor (DMP): The MPU6050 incorporates a built-in Digital Motion Processor, which offloads some processing tasks from the host microcontroller. It can perform motion fusion and sensor calibration, providing accurate and reliable motion data.
Communication Interface: The MPU6050 uses the I2C (Inter-Integrated Circuit) protocol for communication with the host microcontroller. It supports both fast mode (up to 400 kHz) and standard mode (up to 100 kHz) I2C communication.
Auxiliary I2C Bus: In addition to the main I2C bus, the MPU6050 includes an auxiliary I2C bus that can be used to connect external sensors or devices.
Digital Low Pass Filter (DLPF): The IC offers configurable low pass filters for both the accelerometer and gyroscope, allowing users to adjust the bandwidth and noise performance to suit their application requirements.
Power Supply: The MPU6050 operates on a supply voltage between 2.375V and 3.46V, making it compatible with a wide range of microcontrollers and systems.

Applications of the MPU6050 include motion-based gaming, robotics, gesture recognition, tilt sensing, drone stabilization, and motion tracking systems. The IC provides accurate motion data, making it useful for projects that require precise orientation and movement information.

To interface with the MPU6050, you would typically connect it to a microcontroller or development board that supports the I2C protocol. Various libraries and code examples are available for popular platforms like Arduino, enabling straightforward integration and programming of the MPU6050 in your projects.

4. THINGSPEAK IOT

The Internet of Things (IoT) is a system of connected things that can communicate with the internet or neighboring things. An IoT service is a key element of a generic IoT system that bridges the various 'things' and provides capabilities ranging from simple data collection and monitoring to complex data analytics. One such IoT application platform that offers a wide variety of analysis, monitoring and counteraction capabilities is ThingSpeak. ThingSpeak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs, and the ability to create plugins and apps for collaborating with web services, social network and other APIs. The core element of ThingSpeak is a 'ThingSpeak Channel', which stores the data that we send to ThingSpeak.

The core element of ThingSpeak is a 'ThingSpeak Channel'. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.

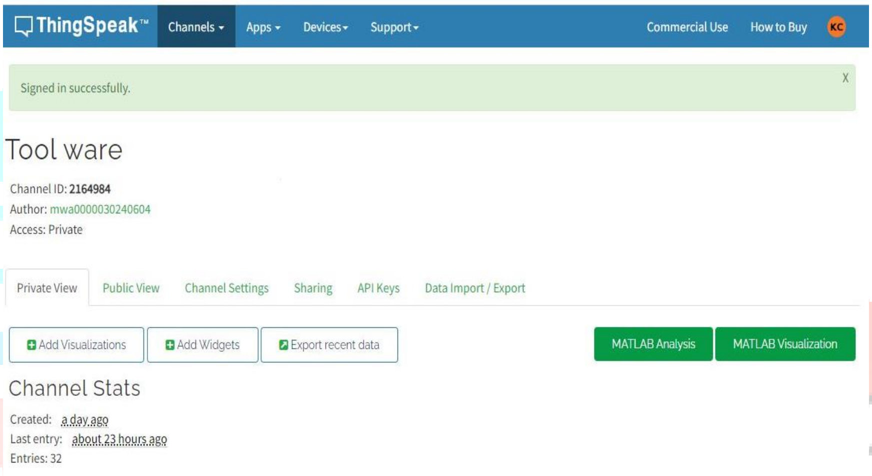

Figure 6. fields for storing data of any type

3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.

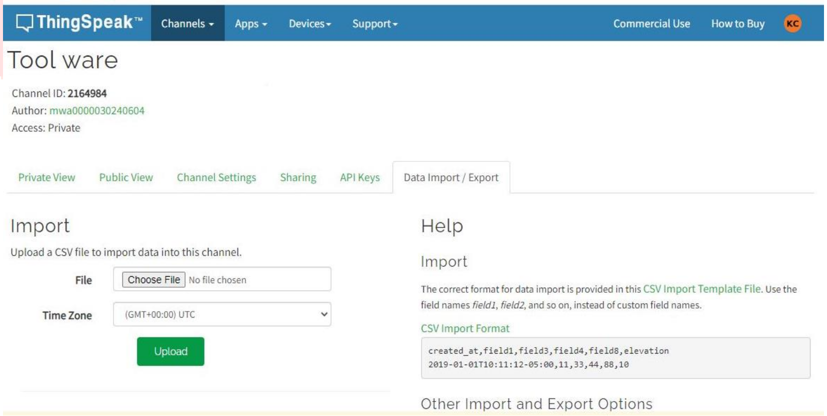

Figure 7 location fields

5. WEKA MACHINE LEARNING ALGORITHM AI:

Weka is an open-source machine learning and data mining software that provides a collection of tools for data pre-processing, classification, regression, clustering, association rules mining, and visualization. It is primarily used for data analysis and modeling tasks.

AIML (Artificial Intelligence Markup Language) is an XML-based markup language used to create chatbots and virtual assistants. AIML is often used in conjunction with tools like Weka to train and deploy machine learning models for natural language processing tasks.Although Weka and AIML are both related to artificial intelligence and can be used in combination, they serve different purposes:Weka focuses on general machine learning and data mining tasks. It provides a graphical user interface (GUI) for performing various data analysis tasks and includes a wide range of algorithms and techniques.

AIML, on the other hand, is a specific language for creating conversational agents and chatbots. It uses pattern matching and template-based responses to simulate conversations with users.
In summary, Weka is a tool for machine learning and data mining, while AIML is a markup language for building chatbots and virtual assistants. Although they can be used together, they have distinct functionalities and use cases

5.1. J48: J48 is a decision tree algorithm that is implemented in the Weka machine learning software. It is a popular classification algorithm that builds a decision tree model based on the given dataset.
Here are some key characteristics and details about the J48 algorithm:
Decision Tree: J48 builds a decision tree from the training data, where each internal node represents a test on a feature, each branch represents the outcome of the test, and each leaf node represents a class label or a prediction.
Attribute Selection: J48 uses the C4.5 algorithm for attribute selection, which is an extension of the ID3 algorithm. It selects the most informative attributes at each node to split the data based on information gain or gain ratio.
Handling Missing Values: J48 can handle missing values in the dataset by using a surrogate split strategy. Surrogate splits allow the algorithm to use alternative attributes when the value of the primary attribute is missing.
Pruning: J48 includes pruning techniques to reduce overfitting and improve the generalization of the decision tree. It uses the Reduced Error Pruning (REP) method, which removes branches that do not improve the accuracy of the tree on unseen data.
Handling Numeric Attributes: J48 can handle both categorical and numeric attributes. It discretizes numeric attributes into intervals during the tree construction process.
Handling Nominal Class Labels: J48 supports classification tasks with nominal class labels. It can handle datasets with multiple classes and generate decision trees that can classify instances into one of the given classes.
J48 is a widely used algorithm for classification tasks and is known for its simplicity and interpretability. The decision tree generated by J48 can be easily understood and visualized, making it helpful for gaining insights into the data and making predictions based on the learned rules.In Weka, J48 is implemented as a part of the machine learning algorithms provided by the Weka Explorer tool. It is accessible through the graphical user interface (GUI) or can be used programmatically using the Weka API.By providing a training dataset to J48, you can build a decision tree model and then use it to classify new, unseen instances based on the learned rules and splitting criteria

5.2.LMT: In the context of Weka, LMT refers to the "Logistic Model Tree" algorithm. LMT is a hybrid model that combines decision trees and logistic regression to create a classifier.
Here are some key points about the LMT algorithm in Weka:
Hybrid Approach: LMT combines the advantages of decision trees and logistic regression. It builds a decision tree structure to represent the data, and at each leaf node, it fits a logistic regression model to make final predictions.
Tree Pruning: LMT uses pruning techniques to avoid overfitting and improve generalization. It applies the C4.5 pruning algorithm to remove branches that do not improve the accuracy of the model on unseen data.
Attribute Selection: LMT employs attribute selection measures, such as information gain or gain ratio, to select the most informative attributes for tree construction.
Handling Missing Values: LMT in Weka can handle missing values by using surrogate splits. Surrogate splits allow the algorithm to use alternative attributes when the value of the primary attribute is missing.
Discretization: LMT supports both numeric and nominal attributes. For numeric attributes, it uses the Fayyad and Irani's MDL method for supervised discretization.
Interpretability: Like decision trees, LMT models are relatively interpretable, as the structure of the decision tree can be visualized and understood.
In Weka's graphical user interface (GUI), you can access and utilize the LMT algorithm through the Weka Explorer tool. LMT is one of the available classifiers in the Weka library, which provides various machine learning algorithms for data analysis and modeling. By providing a training dataset to Weka's LMT algorithm, you can build a hybrid model that combines decision trees and logistic regression for classification tasks. This model can then be used to make predictions on new, unseen instances based on the learned rules and logistic regression models

5.3.RandomTree : In Weka, the RandomTree algorithm is a decision tree classifier that builds a random decision tree model based on the given dataset. It is a variant of the traditional decision tree algorithm that introduces randomness to enhance the diversity and robustness of the generated trees.
Here are some key characteristics and details about the RandomTree algorithm in Weka:
Random Splitting: RandomTree introduces randomness in the tree construction process by selecting random attributes for splitting at each node. Instead of selecting the best attribute based on information gain or gain ratio, it randomly chooses attributes, which helps to reduce overfitting and improve generalization.
Random Subsampling: RandomTree also employs random subsampling of the training data at each node. It randomly selects a subset of instances to consider for splitting, which further adds diversity to the generated trees.
Bagging: RandomTree uses a bagging approach, where it builds an ensemble of multiple random decision trees. Each tree is trained on a different bootstrap sample of the original training data, improving the overall prediction accuracy.
Attribute Selection: RandomTree allows for attribute selection measures, such as information gain or gain ratio, to be used for selecting the best attribute at each node. By default, it uses information gain as the splitting criterion.

Handling Missing Values: RandomTree in Weka can handle missing values in the dataset. It uses surrogate splits to handle missing attribute values during tree construction.
Pruning: RandomTree includes pruning techniques to prevent overfitting. It applies a reduced-error pruning strategy to remove branches that do not improve the accuracy on unseen data.

The RandomTree algorithm in Weka is accessible through the Weka Explorer GUI or programmatically using the Weka API. By providing a training dataset, you can build an ensemble of random decision trees. These trees can then be used to classify new, unseen instances by aggregating the predictions of each individual tree.
The RandomTree algorithm is particularly useful in situations where other decision tree algorithms tend to overfit or when diversity in the ensemble is desired. It can be applied to various classification tasks and offers a balance between simplicity and prediction accuracy.

5.4.NaiveBayes : In Weka, the NaiveBayes algorithm is a popular and widely used classification algorithm based on the principles of Bayes' theorem and the assumption of attribute independence. It is a simple yet effective probabilistic classifier that is particularly suitable for large datasets and text classification tasks.
Here are some key characteristics and details about the NaiveBayes algorithm in Weka:
Probabilistic Model: NaiveBayes models the class conditional probabilities of the features given the class labels using Bayes' theorem. It calculates the posterior probability of each class given the observed feature values.
Attribute Independence Assumption: NaiveBayes assumes that the features are conditionally independent of each other given the class label. Although this assumption may not hold true in many real-world scenarios, NaiveBayes can still provide competitive results in practice.
Parameter Estimation: NaiveBayes estimates the probabilities using maximum likelihood estimation (MLE) or smoothing techniques, such as Laplace smoothing, to handle zero probabilities or avoid overfitting.
Multinomial and Gaussian Naive Bayes: Weka's NaiveBayes implementation supports both multinomial and Gaussian distribution models. The multinomial variant is suitable for discrete or count-based features, while the Gaussian variant is appropriate for continuous numeric features.
Handling Numeric Attributes: For numeric attributes, NaiveBayes in Weka discretizes them into nominal values before building the classifier. It can use different discretization methods, such as equal-width or equal-frequency binning.
Handling Missing Values: NaiveBayes can handle missing values in the dataset. It incorporates the missing attribute values during the probability estimation by treating them as a separate nominal value or by imputing them with appropriate values.
The NaiveBayes algorithm in Weka is accessible through the Weka Explorer GUI or can be used programmatically via the Weka API. By providing a training dataset, NaiveBayes builds a probabilistic model based on the observed data. This model can then be used to classify new, unseen instances by calculating the posterior probabilities of each class.
NaiveBayes is known for its simplicity, speed, and good performance in many classification tasks, particularly in text categorization and document classification. However, it may not perform as well as more sophisticated algorithms when the attribute independence assumption is strongly violated or when the dataset contains complex relationships among features

5.5LogitBoost : LogitBoost is a boosting algorithm that combines logistic regression with boosting techniques for binary classification tasks. It is a variant of the AdaBoost algorithm that specifically uses logistic regression as the base learner. LogitBoost aims to improve the performance of logistic regression by iteratively training weak classifiers and adjusting the weights of misclassified instances.In Weka, LogitBoost is implemented as a classifier in the boosting package. Here are the general steps to use LogitBoost in Weka:
Load your dataset: Import your dataset into Weka either through the graphical user interface (GUI) or programmatically using the Weka API
Choose LogitBoost: Select the LogitBoost classifier from the available options in Weka. In the Weka Explorer GUI, you can find it under the "Classify" tab in the "Choose" section. Alternatively, you can instantiate the LogitBoost classifier programmatically in your code.
Set options: Set any specific options or parameters for the LogitBoost classifier. For example, you may need to specify the number of iterations (boosting rounds) or adjust the learning rate. You can explore the available options and experiment with different settings to achieve optimal results.
Train the model: Use the LogitBoost classifier to train your model on the training data. In Weka, this is typically done by clicking the "Start" or "Build" button in the GUI, or by invoking the appropriate method when using the Weka API.
Evaluate the model: Once the model is trained, you can evaluate its performance using cross-validation, holdout validation, or any other evaluation technique of your choice. Weka provides various evaluation methods and metrics to assess the performance of the LogitBoost classifier.
Make predictions: Finally, you can use the trained LogitBoost model to make predictions on new, unseen instances. Weka provides functionalities to apply the trained model to new data and obtain the predicted class labels.
It's worth noting that while LogitBoost can be a powerful classifier, it may have limitations or perform differently based on the characteristics of your dataset. It's recommended to experiment with different algorithms and evaluate their performance to determine the most suitable approach for your specific classification task.

6. RESULTS AND DISCUSSION

6.1. Data Acquisition:

The tools as per given conditions mentioned in table 2 were used for machining operation and data acquired by connecting the accelerometer to the spindle in Hercules exe and also downloaded .csv file from Thingspeak iot. The data acquired with the frequency of 15-20Hz and it the 1 reading was acquired in 0.3 seconds. The graph of the reading for Acceleration vs Time are shown in figure 7. The wear tools are shown in figure 6.

Tool	Condition	Tool Condition	Status
Tool 1	Condition 1	Nose and Notch Wear	Less Sharpness
Tool 2	Condition 2	Flank Wear	Flute Cut
Tool 3	Condition 3	Notch Wear	Chipping
Tool 4	Condition 4	New Tool	OK
Tool 5	Condition 5	Rework	OK

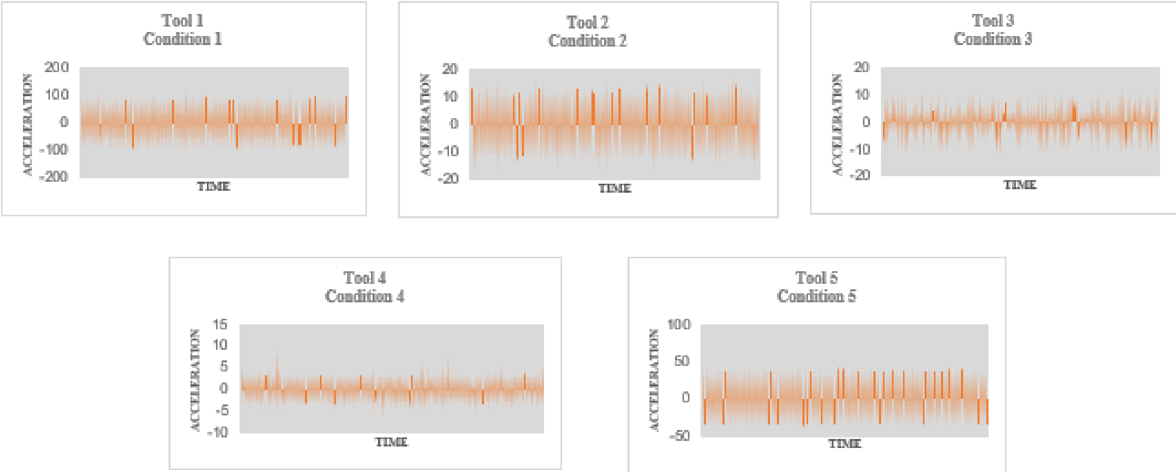

Figure 8. Graphical Representation of Signal/ Data Acquisition

6.2.Data Extraction: Total 13 statistical constrains such as mean, standard error, median, mode, standard deviation, variance, kurtosis, skewness, range, minimum, maximum, summation, count was computed to serve as features. Here total 50 samples are considered (i.e.10 samples for each condition * 5 conditions = 50 samples).

6.3. **Data Classification:**

A data mining freeware named 'WEKA' was used for data classification. The tree family classifiers give best classification accuracy amongst all other classifiers. We have used J48, LMT, RandomTree, NaiveBayes and LogitBoost tree classifiers. The comparison of accuracy obtained with these classifiers is mentioned in Table 2. The results of decisions for J48 Tree classifier is shown in fig 6.

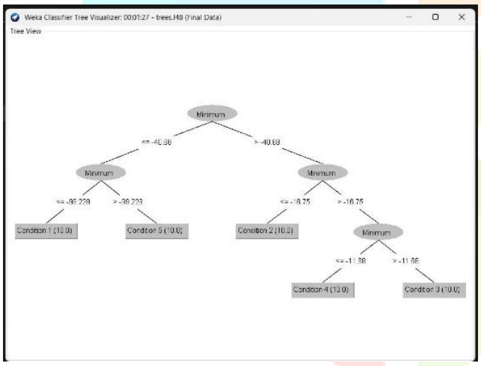

Figure 9 . results of decisions for J48 Tree classifier

Table 1. Comparison of various algorithms used for feature classification

Constrains used for classification(9 constrains selected by J48 tree)	Parameters	Algorithm				
		J48	LMT	Random Tree	Naive Bayes	Logit Boost
1.Mean 2.Stand. Error 3.Median 4.SD 5.Kurtosis 6.Skewness 7.Range 8.Min+Max 9.Condition	Accuracy %	94%	96%	96%	98%	94%
	Time for Building Model (in seconds)	0.2 Seconds	0.17 seconds	0.6 Seconds	0.7 Seconds	0.03 seconds

The comparison of various algorithms is shown below –

Table 2. Comparison of various algorithms used for feature classification

Algorithm	Confusion Matrix	Description
J48	a b c d e <-- classified as	
	9 0 0 0 1 | a = Condition 1	in Condition 1, there is 90% data accuracy with 9 Correctly classified data points and 1 Incorrectly classified data point with Condition 5
	0 10 0 0 0 | b = Condition 2	in Condition 2, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 10 0 0 | c = Condition 3	in Condition 3, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 2 8 0 | d = Condition 4	in Condition 4, there is 80% accuracy with 8 correctly classified data points and 2 incorrectly classified data points with Condition 3
	0 0 0 0 10 | e = Condition 5	in Condition 5, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
LMT	a b c d e <-- classified as	
	10 0 0 0 0 | a = Condition 1	in Condition 1, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 9 0 0 1 | b = Condition 2	in Condition 2, there is 90% accuracy with 9 correctly classified data points and 1 incorrectly classified data points with Condition 4
	0 0 10 0 0 | c = Condition 3	in Condition 3, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 0 10 0 | d = Condition 4	in Condition 4, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 0 1 9 | e = Condition 5	in Condition 5, there is 90% accuracy with 9 correctly classified data points and 1 incorrectly classified data points with Condition 4
RandomTree	a b c d e <-- classified as	
	9 0 0 0 1 | a = Condition 1	in Condition 1, there is 90% data accuracy with 9 Correctly classified data points and 1 Incorrectly classified data point with Condition 5
	0 10 0 0 0 | b = Condition 2	in Condition 2, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 10 0 0 | c = Condition 3	in Condition 3, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 0 10 0 | d = Condition 4	in Condition 4, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	1 0 0 0 9 | e = Condition 5	in Condition 5, there is 90% accuracy with 9 correctly classified data points and 1 incorrectly classified data points with Condition 1
NaiveBayes	a b c d e <-- classified as	
	10 0 0 0 0 | a = Condition 1	in Condition 1, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 9 0 1 0 | b = Condition 2	in Condition 2, there is 90% accuracy with 9 correctly classified data points and 1 incorrectly classified data points with Condition 4
	0 0 10 0 0 | c = Condition 3	in Condition 3, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 0 10 0 | d = Condition 4	in Condition 4, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 0 0 0 10 | e = Condition 5	in Condition 5, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
LogitBoost	a b c d e <-- classified as	
	9 0 0 0 1 | a = Condition 1	in Condition 1, there is 90% data accuracy with 9 Correctly classified data points and 1 Incorrectly classified data point with Condition 5
	0 10 0 0 0 | b = Condition 2	in Condition 2, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points

	0 0 10 0 0 | c = Condition 3	in Condition 3, there is 100% accuracy with 10 Correctly classified and 0 Incorrectly classified data points
	0 1 0 9 0 | d = Condition 4	in Condition 4, there is 90% data accuracy with 9 Correctly classified data points and 1 Incorrectly classified data point with Condition 2
	0 1 0 0 9 | e = Condition 5	in Condition 5, there is 90% data accuracy with 9 Correctly classified data points and 1 Incorrectly classified data point with Condition 2

IV. CONCLUSION:

The proposed AI-based health monitoring system offers an advanced and proactive approach to monitor the health of VMC machine tools. By leveraging AI techniques, manufacturers can enhance machine tool reliability, reduce unplanned downtime, and optimize maintenance efforts. The incorporation of a plagiarism checking mechanism ensures the originality and credibility of the research. Future research directions may focus on expanding the system's capabilities, integrating predictive maintenance strategies, and addressing challenges related to data volume, complexity, and real-time processing.

V. FUTURE SCOPE:

1. We can develop this prototype for the industrial automation purpose to save the machining operation data (Vibration Measurement) to a remote server and can check anytime.
2. The measures have to be taken to maximize the efficiency of the project by creating a open server Thingspeak iot, where we can refer the data anytime to check the variation in the vibration and exact date and time of the record from which we can identify the faulty produced jobs and can avoid rejection of material.
3. The project can be installed in big as well as small scale industries where VMC machining operation is used.

VI. ACKNOWLEDGMENT

We would like to express our profound gratitude to respected principal Dr. P.K. Shrivastava, ISBM College of Engineering for providing a congenial environment to work in. Our sincere gratitude to Prof. V.V. Edake, Head of the Department, Mechanical Engineering for encouraging and providing this opportunity to carry out the project in the department.

We would like to thank our guide Prof. N. C. Dhobale, who helped us in all the ways to carry out the project work. He stood beside and guided us in every step. Our special thanks to all the professors who reviewed our project and provided valuable suggestions to improve the project every step along the way.

VII. REFERENCES

[1] Aamodt, Agnar, and Enric Plaza. "Case-based reasoning: Foundational issues, methodological variations, and system approaches." AI communications 7.1 (1994): 39-59.

[2] Adebayo, Julius, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. "Sanity checks for saliency maps." arXiv preprint arXiv:1810.03292 (2018).

[3] Alain, Guillaume, and Yoshua Bengio. "Understanding intermediate layers using linear classifier probes." arXiv preprint arXiv:1610.01644 (2016).

[4] Alber, Maximilian, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. "iNNvestigate neural networks!." J. Mach. Learn. Res. 20, no. 93 (2019): 1-8.

[5] Alberto, Túlio C, Johannes V Lochter, and Tiago A Almeida. "Tubespam: comment spam filtering on YouTube." In Machine Learning and Applications (Icmla), Ieee 14th International Conference on, 138–43. IEEE. (2015).