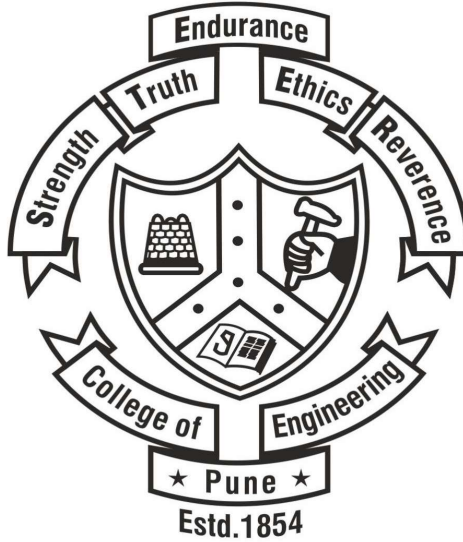


COLLEGE OF ENGINEERING, PUNE



DEPARTMENT OF MECHANICAL ENGINEERING

2021-2022

MINI-PROJECT REPORT

Milling Cutter Health Monitoring Framework for Segregating Undefined Conditions: A Statistical & Machine Learning Approach

Guide: **Dr. Abhishek Patange**

Group Members:

Sr.No.	Name	MIS
1.	Aishwarya Belagali	111910009
2.	Nihar Apte	111910063
3.	Jayesh Raut	111911018
4.	Ketan Joshi	111910052

Contents of the Report

Chapter 1: Introduction

- 1.1. Motivation
- 1.2. Problem Definition
- 1.3. Scope and Objectives
- 1.4. Novelty

Chapter 2: Experimentation and Data Collection

- 2.1. Experimental setup
- 2.2. Faults considered
- 2.3. Data acquisition

Chapter 3: Data Pre-processing and Representation

- 3.1. Pre-processing and Feature extraction

Chapter 4: ML framework for segregating “Unknown Class” and outlier data

- 4.1. Theory
- 4.2. GUI
- 4.3 Using GUI

Chapter 5: Results and Discussion

- 5.1. Earlier/status quo results
 - 5.1.1. 7 class classification without unknown class
 - 5.1.2. 7 class classification with unknown class
- 5.2. Classification with unknown class using model (8 classes)

Chapter 6: Conclusion and Future Scope

Chapter 1: Introduction

1.1 Motivation

Tool condition monitoring is considered necessary for durability of the tools in the machines used in the manufacturing industry today. Significant information regarding tool health can be extracted from the vibration data of the tool during its working. This data collected, is generated in high volumes and is very difficult to process. Statistical sampling and machine learning assists in the extraction of statistical features and in the process of analysing and prediction of tool wear.

During this classification process, the data belonging to undefined class is misclassified into predefined class as there is no provision for data classification beyond the one considered for training of machine learning model. This misclassification may introduce new problems and defies the whole purpose of tool health monitoring. Similar problem arises when outlier data is passed through the model as it will again be classified into the predefined classes while in reality, outlying data signifies some major issue such as misalignment of tool or workpiece, inaccurate working conditions, etc.

Thus, there is a need to classify and segregate this undefined class and the outliers and thus improve the overall accuracy of the framework.

1.2 Problem Definition

To characterize anomalous moments of real-time spindle vibrations owing to in-process milling cutter faults through Machine Learning and to segregate undefined data class and outlier data using statistical approach.

1.3 Scope and objectives

1. To classify the tool as healthy or faulty.
2. To separate data from undefined class and outlier data from predefined classes

2.1 Outlier Data

Outlier data is defined as data that belongs to ranges outside of data from known classes. It is typically characterised by data that comes from a damaged sensor, erroneous set up or a different dataset altogether.

2.2 Alien Data

Alien data is data that is not part of any of the known classes, but is a category of tool wear which is not present in the training set. This signifies that there is a lack of data and more datapoints need to be added to solve misclassification arising out of this.

This work proposes milling cutter health monitoring for segregating undefined conditions with the help of a statistical & machine learning approach based on real-time spindle vibrations as data input. Existing methods misclassify data belonging to undefined classes into a predefined class as there is no provision for data classification beyond the one considered for training a machine learning model. In an attempt to address this problem, the framework aiming at identifying such data and classifying it into a separate category is advocated herein. Initially, the time-domain vibration data corresponding to predefined classes (one healthy class and seven faulty classes) is represented in terms of descriptive statistics, followed by the determination of relevant statistical feature spans for each class. A confidence value derived from a thorough understanding span of faults is assigned to the classification output for each class. A supervised machine learning model is then trained using these statistical features. This allows the understanding of mappings between specific classes and corresponding features. The confidence value assigned assists segregation of data belonging to undefined classes into a separate category. The vibrations caused by factors outside the trained classes are fed into the classifier; owing to a low confidence value for each class, the classifier successfully segregates this undefined condition as an unknown category.

3. To classify the condition of the tool into one of the eight categories:

- i. Flank wear
- ii. Notch wear
- iii. Nose wear
- iv. Edge fracture
- v. Built up edge
- vi. Crater Wear
- vii. All wear

4. To develop and implement a User Interface for the framework.

1.4 Novelty

Extensive research has been done on machine learning approach for tool condition monitoring. This work proposes milling cutter health monitoring for segregating undefined conditions with the help of a statistical & machine learning approach based on real-time spindle vibrations as data input. Existing methods misclassify data belonging to undefined classes into a predefined class as there is no provision for data classification beyond the one considered for training a machine learning model. In an attempt to address this problem, the framework aiming at identifying such data and classifying it into a separate category is advocated herein. Initially, the time-domain vibration data corresponding to predefined classes (one healthy class and seven faulty classes) is represented in terms of descriptive statistics, followed by the determination of relevant statistical feature spans for each class. A supervised machine learning model is then trained using the statistical features.

Thus, a statistical approach is used to tackle the problems and further process the signals.

Chapter 3: Experimentation and data collection

3.1 Experimental setup

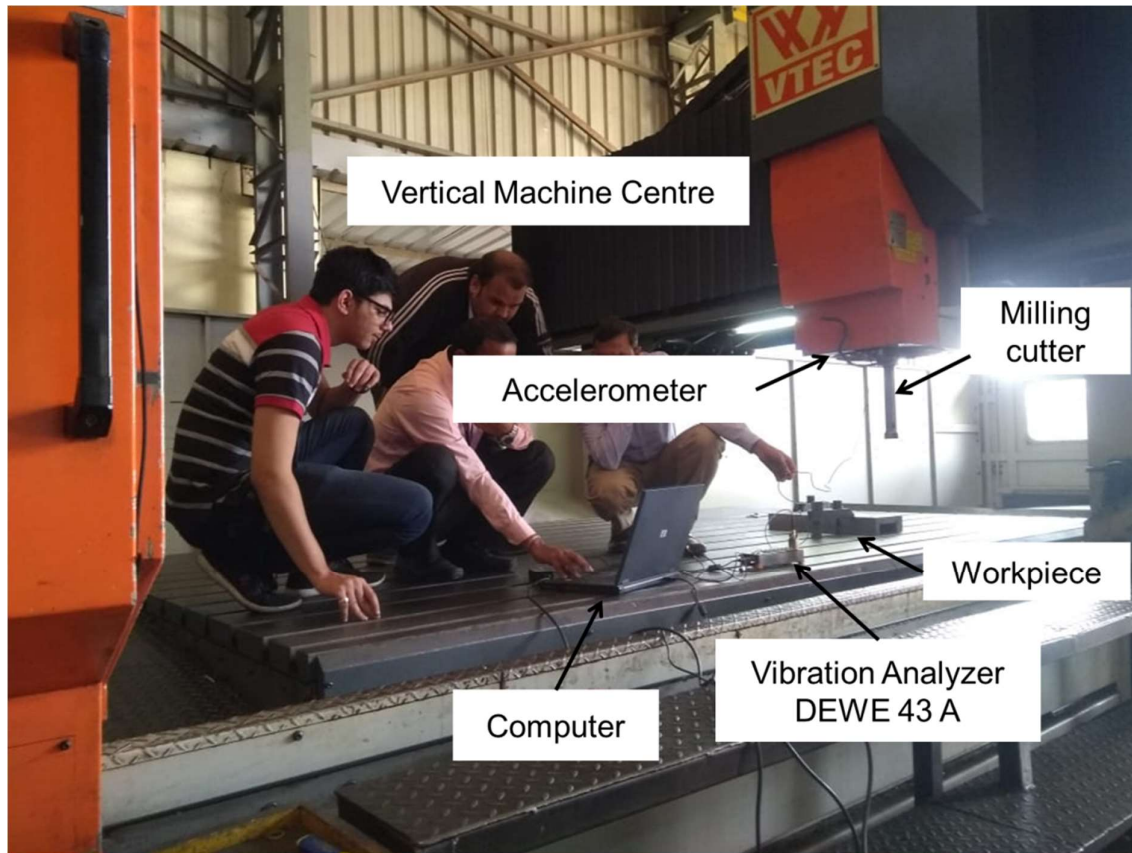


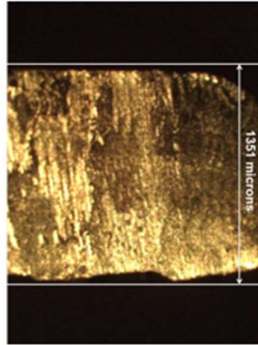
Figure 1: Experimental Setup

- Machine: VMC Mech: VTEC Model CCS00605 (X: 4200 mm, Y: 2600 mm, Z: 920 mm)
- Cutting tool: Face milling cutter diameter 63 mm with 4 inserts
- Workpiece: Mild Steel C shaped hollow cuboid (650mm*250mm*100mm)
- Machining operation to be operation: Face milling
- Location: Axis Metal-cut Technologies
- Vibration Analyzer: DEWE 43 A
- Accelerometer: Piezoelectric Make: PCB, Model: 352C03 ICP of sensitivity factor = 10 mV/g & acceleration with +/- 500g range

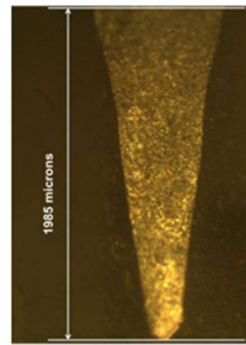
3.2 Faults considered



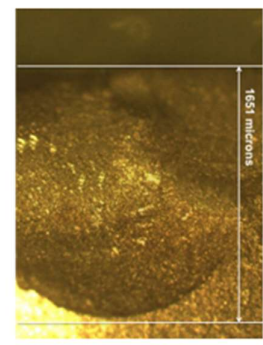
Meta. Microscope



Flank wear

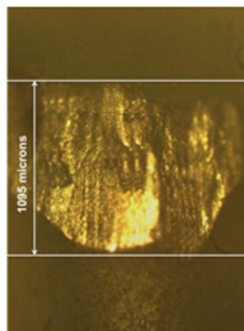


Notch wear

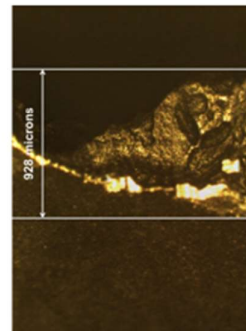


Crater wear

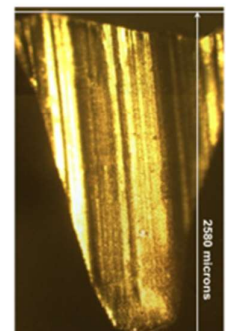
Sr. No.	Type of fault	Size (μm)
1	Flank wear	1351
2	Notch wear	1985
3	Crater wear	1651
4	Nose wear	1095
5	Built-up edge	928
6	Edge fracture	2580



Nose wear



Built-up edge



Edge fracture

Figure 2: Types of wear

3.3 Data acquisition

- At most other times, in order to collect data for a particular fault, healthy inserts are employed and used until the desired fault is observed in the insert.
- Unlike this process, inserts which were in use and developed the faults in the course of machining have been used for data collection here.
- To maintain the authenticity of the faulty inserts, they were used for only 15 seconds while the data was collected.
- Accelerometer was attached to the spindle holder.
- Each time, one insert having a certain fault among crater wear, flank wear, edge fracture, nose wear and notch wear was used.
- There were also 2 scenarios when the insert had all types of faults.

Chapter 4: Data Pre-processing and Representation

The time-domain data was split into samples with 3000 data points per sample, owing to which 17 statistical features like mean, median, mode, standard deviation, etc. are extracted.

	Mean	Median	Mode	Kurtosis	Skewness	Standard Deviation	Variance	Standard Error	
0	0.004117	-0.000421	0.047621	0.705523	0.116939	0.298621	0.089175	0.005453	
1	0.001398	-0.001684	-0.093198	0.612556	0.111255	0.301458	0.090877	0.005505	
2	-0.006598	-0.009079	-0.086464	0.806305	0.134979	0.296204	0.087737	0.005409	
3	-0.000164	-0.003307	-0.117009	0.900254	0.088911	0.318043	0.101151	0.005808	
4	0.000251	-0.005712	-0.118572	0.574849	0.102585	0.310773	0.096580	0.005675	
...
121	0.000478	0.002886	-0.073236	0.239421	-0.073099	0.094935	0.009013	0.001734	
122	-0.001749	-0.000241	-0.026697	0.120556	-0.128240	0.089388	0.007990	0.001632	
123	-0.001059	0.001082	-0.001563	0.246164	-0.139028	0.093249	0.008695	0.001703	
124	-0.000066	-0.000782	-0.016114	0.350961	-0.071369	0.092538	0.008563	0.001690	
125	-0.000169	0.002465	0.007576	0.116950	-0.063157	0.091879	0.008442	0.001678	
126 rows × 18 columns									

Max	Min	Sum	Range	2nd Quartile	RMS	Shape Factor	Impulse Factor	K Factor	Condition
1.107072	-1.161548	12.352286	2.268620	-0.000421	0.298650	1.309393	5.092664	0.257114	Healthy
1.205682	-0.994994	4.195474	2.200676	-0.001684	0.301461	1.291062	5.163557	0.250033	Healthy
1.268696	-1.127275	-19.794779	2.395971	-0.009079	0.296277	1.312865	5.621850	0.233529	Healthy
1.425629	-1.135212	-0.491003	2.560841	-0.003307	0.318043	1.317040	5.903633	0.223090	Healthy
1.311627	-1.258714	0.752559	2.570341	-0.005712	0.310773	1.288958	5.440088	0.236937	Healthy
...
0.325892	-0.353190	1.433927	0.679083	0.002886	0.094936	1.268475	4.719098	0.268796	All Wear
0.273461	-0.314829	-5.248071	0.588290	-0.000241	0.089405	1.265661	4.456850	0.283981	All Wear
0.322165	-0.323848	-3.176068	0.646013	0.001082	0.093255	1.272094	4.417600	0.287960	All Wear
0.328177	-0.387463	-0.197099	0.715641	-0.000782	0.092538	1.274554	5.336661	0.238830	All Wear
0.322165	-0.369305	-0.505554	0.691469	0.002465	0.091879	1.260204	5.065350	0.248789	All Wear

Figure 3: Statistical Features

The features extracted are listed as follows:

- | | |
|-----------------------|------------------------------|
| 1. Mean | 9. Maximum |
| 2. Median | 10. Minimum |
| 3. Mode | 11. Sum |
| 4. Kurtosis | 12. Range |
| 5. Skewness | 13. 2 nd Quartile |
| 6. Standard Deviation | 14. RMS |
| 7. Variance | 15. Shape Factor |
| 8. Standard Error | 16. Impulse Factor |
| | 17. K factor |

Figure 4: Graphical representation of the statistical features and their ranges:

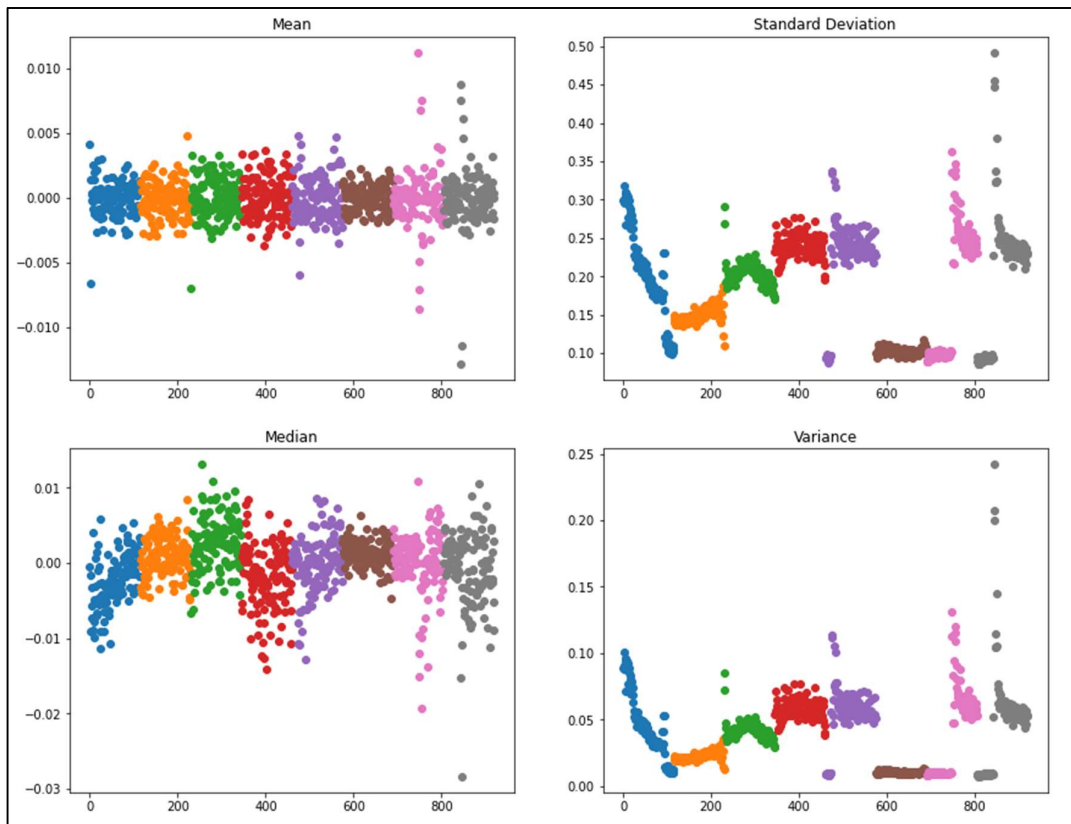


Figure 4.1

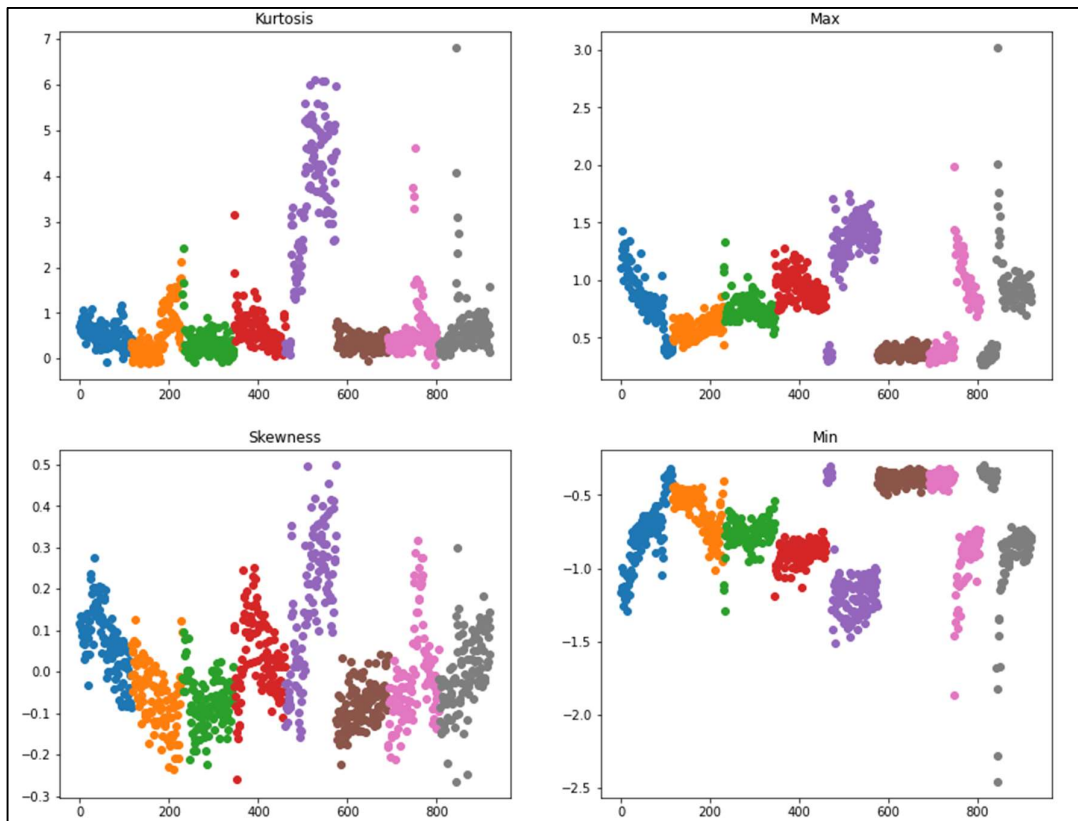


Figure 4.2

Chapter 4: ML framework for segregating “Unknown Class” and Outlier data

4.1 Theory:

For the detection and proper classification of outlier and alien data, we propose a modification of the existing framework of classification. There are two modifications we propose:

- 1) Recognising a new class called “Spam” where alien data will be classified into.
- 2) Adding two discriminators, one before the classifier (Discriminator I) and the other after the classifier (Discriminator II). In essence, Discriminator I will expel outlier data before it enters the classifier, and Discriminator II will make sure that alien classes get classified separately under the “Spam”.

The detailed working is explained below:

- 1) Discriminator I:
The ranges of all features were found out manually. Then, the Discriminator I was set up to expel any data points in the test data whose features had values outside these ranges. That is how using statistical means outlier data was expelled before it entered classifier. The significance of Discriminator I and finding outlier is two-fold:
 - a) Outlier data represents data that is very far off from classes. Thus, in real life, it represents data that was obtained from a damaged sensor, or wrong data set, or an erroneous set up. All these represent huge problems in the setup which were not possible to detect earlier with the data only.
 - b) This outlier data does not enter classifier and thus does not reduce the accuracy of the classifier.
- 2) Discriminator II:
During classification, alien data is classified into one of the known classes. Earlier it did not have a remedy; but this is exactly what the Discriminator II is trained to catch. It checks the probability of classification of datapoints (which is provided by the classifier). If it falls below a certain threshold, it classifies that data point into “Spam”. The threshold is decided using break-even analysis. The costs of failure of the tool and the cost of checking it are compared, and the ratio is determined. This ratio tells us the inaccuracy acceptable for the model. Thus, the threshold is set. The significance of this is:
 - a) Alien classes, while their features may individually fall within the known classes’ ranges, they cumulatively have lower confidence (or probability of classification) which Discriminator II exploits to recognise them.

If there happen to be a lot of datapoints in “Spam”, it represents that our data is lacking and there a type of wear that has not been accounted for by the data.

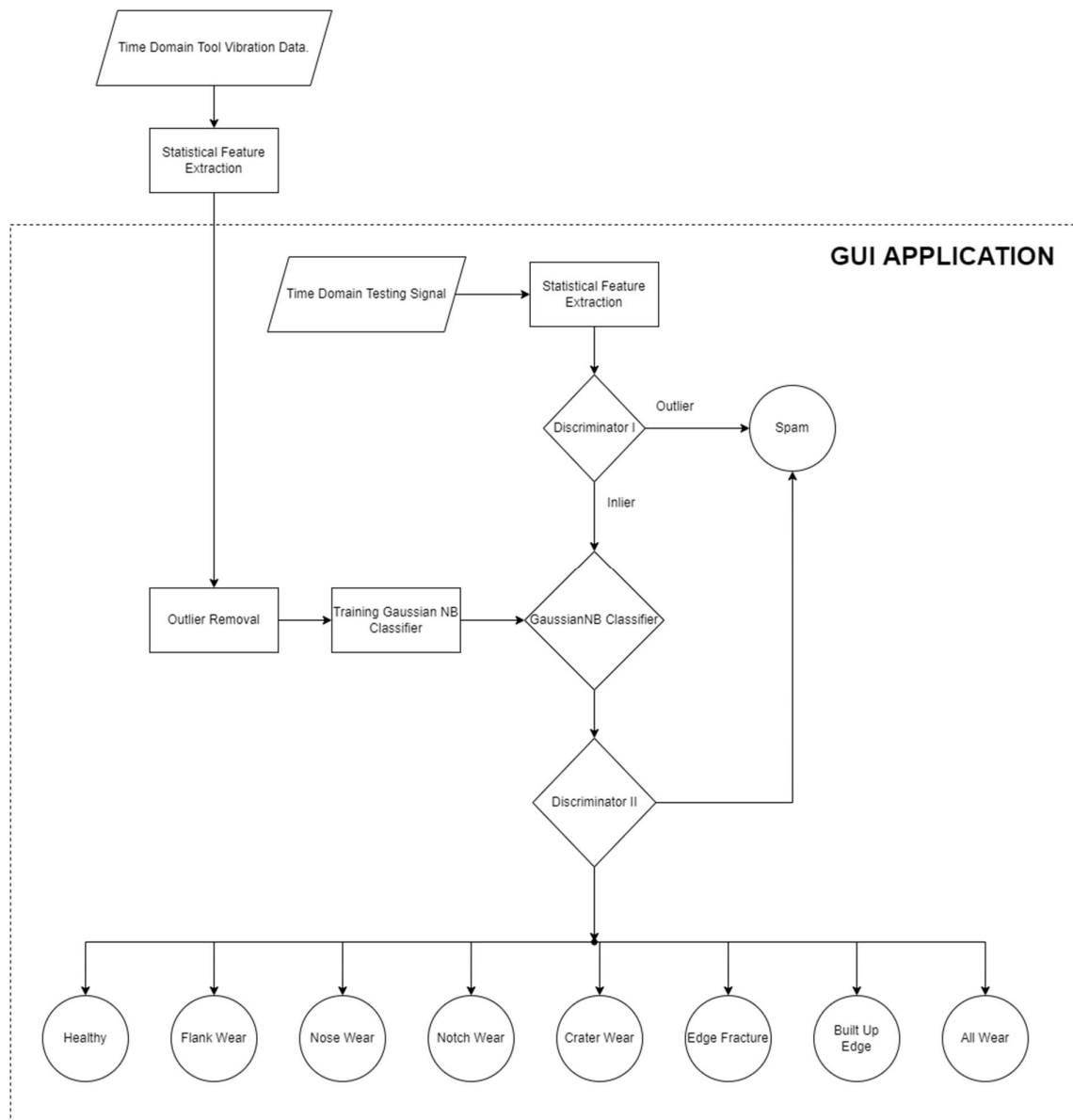


Figure 5: Flowchart of the entire process

The data enters the model at discriminator I where it determines whether it is outlier or not. If not, then it enters the classifier. The classifier is a Navier- Bayes classifier which works on the Navier- Bayes methods. Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. (For theoretical reasons why naive Bayes works well, and on which types of data it does, see the references below.)

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each

distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. Post classification, the data enters Discriminator II, where datapoints with low probability of classification are classified into “Spam”.

4.2 GUI

The Graphical User Interface Application implemented for the framework allows the user to import different training datasets, remove the outliers from the training data, and train the Gaussian Naive Bayes Classifier. The application also facilitates model scoring using different testing datasets.

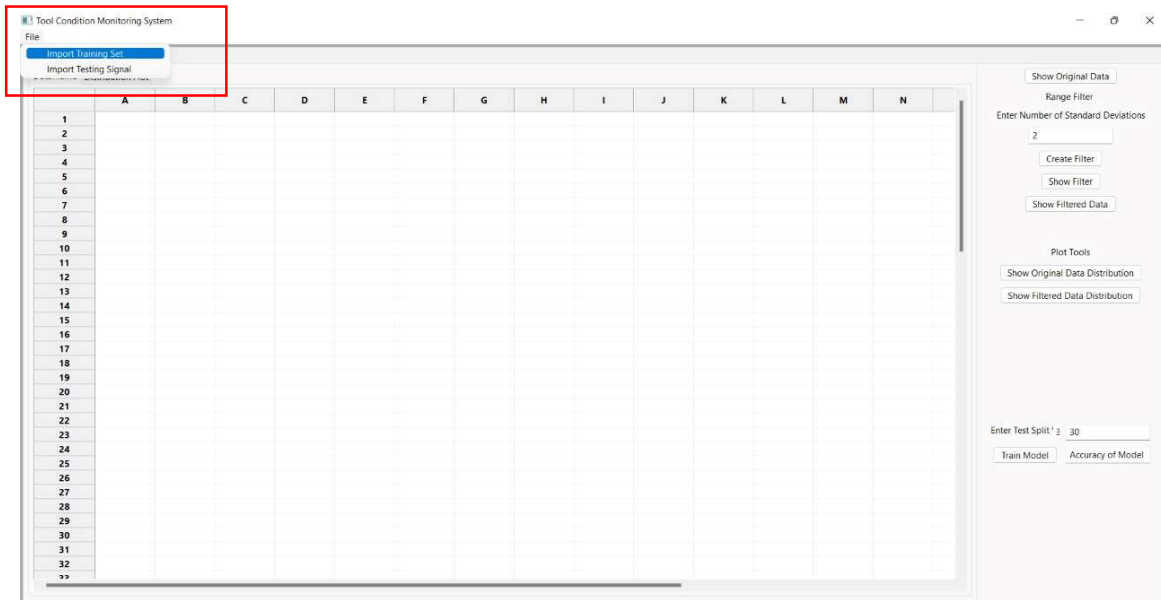
The application further allows the user to classify time domain data into the aforementioned 8 classes as well as detect whether the signal is from a previously undefined class or an outlier.

Data visualization tools are also integrated within the application along with the ability to change various parameters like the span for Discriminator I, percentage of the train test split and probability threshold for Discriminator II.

The application was developed using wxPython while the machine learning algorithm was implemented using SciKit Learn.

4.3 Using GUI:

Step 1: Open File Menu to import Training Dataset and Testing Signal

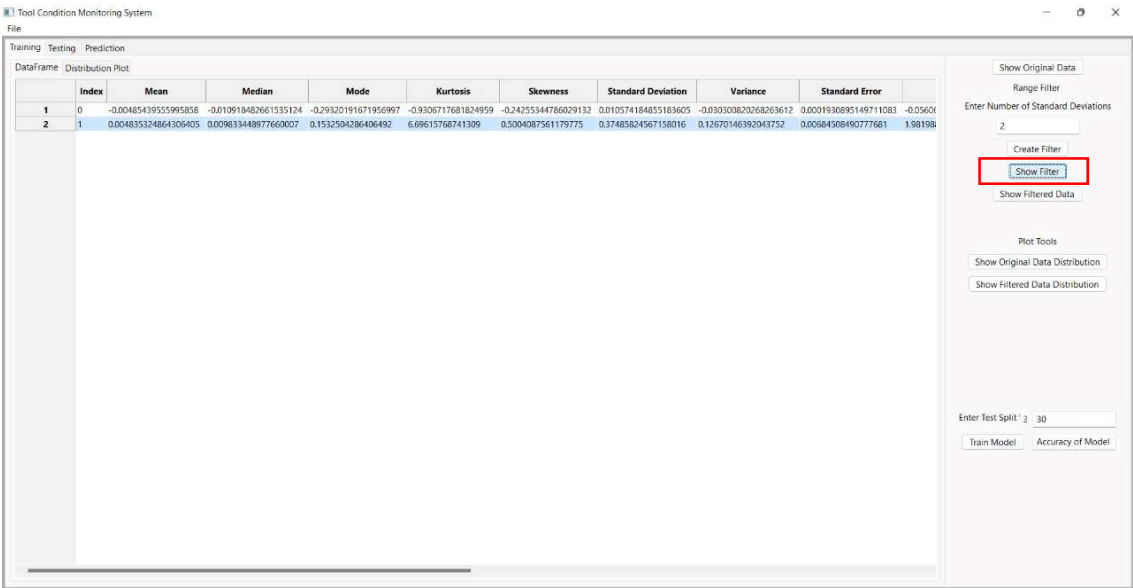


Step 2: Display original data by clicking “Show Original Data” button

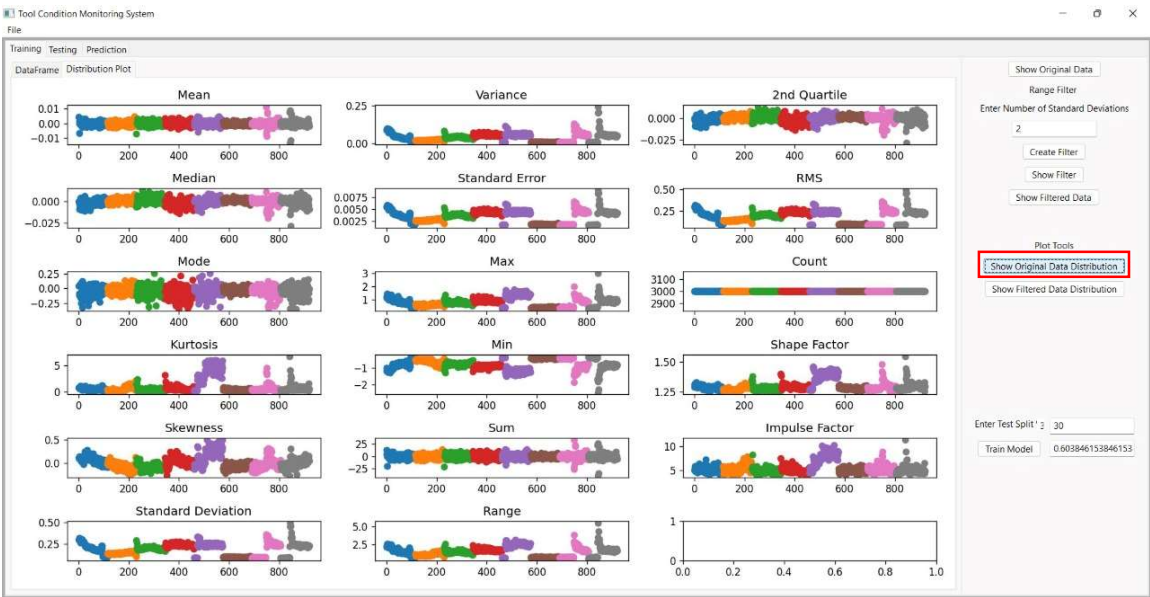
The screenshot shows the 'Tool Condition Monitoring System' window with the 'Show Original Data' button highlighted in the 'Plot Tools' section. The main area of the window displays a table with 33 rows and 18 columns. The columns are labeled: Index, Mean, Median, Mode, Kurtosis, Skewness, Standard Deviation, Variance, Standard Error, Max, Min, Sum, Range, 2nd Quartile, R, and R. The table contains numerical data for each row. On the right side, there are controls for 'Range Filter' (with a dropdown set to 2) and 'Plot Tools' (with buttons for 'Show Original Data Distribution' and 'Show Filtered Data Distribution'). At the bottom right, there are buttons for 'Train Model' and 'Accuracy of Model'.

	Index	Mean	Median	Mode	Kurtosis	Skewness	Standard Deviation	Variance	Standard Error	Max	Min	Sum	Range	2nd Quartile	R
1	0	0.004117429	-0.000420895	0.047621183	0.705523178	0.116939139	0.259621422	0.089174754	0.005452965	1.1070722	-1.161548	12.35228582	2.2686202	-0.000420895	0.2596
2	1	0.001398491	-0.001683578	-0.093198024	0.612555942	0.111255388	0.301457554	0.090876657	0.005004754	1.2056818	-0.9949941	4.195474011	2.2006759	-0.001683578	0.3014
3	3	-0.000163668	-0.003307027	-0.11700861	0.900254017	0.088910965	0.318043146	0.101151443	0.005807615	1.4256291	-1.1352121	0.491003129	2.5608412	-0.003307027	0.3180
4	4	0.000250853	-0.005712137	-0.118671954	0.574849009	0.102584949	0.310772665	0.09657965	0.005674853	1.3116269	-1.2587144	0.752559423	2.5703413	-0.005712137	0.3107
5	5	0.004406304	-0.007215331	-0.103254	0.79638107	0.102127779	0.307271798	0.094415558	0.005610925	1.1223447	-1.0382303	1.21891056	2.153175	-0.007215331	0.3072
6	6	-0.000895663	-0.00751597	-0.091634698	0.691869264	0.069265328	0.266653453	0.071210765	0.004873967	1.0002854	-1.1374969	2.686989415	2.1337823	-0.00751597	0.2666
7	7	-0.002556797	0.00402856	0.086583972	1.058328414	0.069811829	0.303203319	0.091932252	0.005356633	1.3018862	-1.1269144	7.676390243	2.4288006	0.00402856	0.3032
8	8	-0.000394487	-0.008888908	-0.15849677	0.520874911	0.128019708	0.300629007	0.0903778	0.005486625	1.2689882	-1.1199396	-0.91340861	2.3888378	-0.008888908	0.3006
9	9	0.002001332	-0.00980824	-0.25121376	0.77997019	0.06790644	0.3070195	0.094260973	0.005606318	1.323893	-1.2054412	6.003996882	2.5293342	-0.00980824	0.3070
10	10	-0.001197905	-0.007816609	-0.1354077	0.662711802	0.029551001	0.276667798	0.07654507	0.005052082	1.1367754	-1.1576998	-3.59371505	2.2944752	-0.007816609	0.2766
11	11	0.001385344	-0.004810221	-0.17725663	0.935423302	0.082319107	0.304498003	0.092719034	0.00560274	1.1781433	-1.2167453	4.15603069	2.3948886	-0.004810221	0.3044
12	12	-0.001017722	0.000721533	-0.041367896	0.621865174	0.106242272	0.291762018	0.085125075	0.005327709	1.1753774	-1.0438179	-3.053166995	2.2191953	0.000721533	0.2917
13	14	-0.002638406	-0.006553926	0.13232619	0.841598668	0.140084453	0.296938224	0.072704772	0.004932719	1.200031	-1.2927658	-7.915218246	2.4928968	-0.006553926	0.2969
14	15	0.001214541	0.001443967	0.01940037	0.674023345	0.041687731	0.26388541	0.080137915	0.005168289	1.2021943	-1.0451406	6.522621812	2.2473349	0.001443967	0.2638
15	16	-0.001460553	-0.006914692	-0.18944038	0.799050223	0.08730796	0.263187847	0.090357843	0.004805932	0.9414804	-1.0352798	-3.91750089	1.9767602	-0.006914692	0.2631
16	17	5.72e-05	-0.004689965	-0.014911683	0.456090573	0.123062823	0.288267869	0.093096364	0.005263905	1.0980531	-1.1134458	0.171724864	2.2114989	-0.004689965	0.2882
17	18	-0.001343294	-0.001021272	-0.12157832	0.563121018	-0.032222384	0.274487802	0.075345554	0.005012274	1.0394887	-1.0020553	-4.029882939	2.119744	-0.001021272	0.2744
18	19	0.002966628	0.002405111	0.016475005	0.680617674	0.082828006	0.279941874	0.078367453	0.005111868	1.058489	-0.9069671	8.905882606	1.9654561	0.002405111	0.2799
19	20	-0.00082816	-0.002765877	-0.1362495	0.950239216	0.122912157	0.272922318	0.074486592	0.004903688	1.3424122	-1.1811496	-2.484479074	2.5235618	-0.002765877	0.2729
20	21	-0.000846919	-0.005471626	-0.071912795	0.618546458	0.099855893	0.272119692	0.074049127	0.004960031	1.1697254	-1.0648626	-2.54075827	2.234588	-0.005471626	0.2721
21	22	-6.76e-05	-0.000481023	-0.13300259	0.621484704	0.0422448	0.262161106	0.086172446	0.004787183	1.0050956	-1.009545	-0.202871389	2.0146406	-0.000481023	0.2621
22	23	0.0003446701	0.005772265	0.032829754	0.615103826	0.102818471	0.251551676	0.063278246	0.00499345	1.0199465	-0.81978184	1.046102452	1.83942834	0.005772265	0.2515
23	25	0.00080826	-0.000797291	0.154690284	0.78424483	0.225967221	0.227446325	0.051713831	0.004153275	0.94814211	-0.7509957	2.448781112	1.59915381	-0.000797291	0.2274
24	26	0.00304992	0.002645621	-0.11207814	1.097156681	0.189188001	0.223908954	0.050712522	0.004986861	1.0141147	-0.78057855	9.149780397	1.79469325	0.002645621	0.2239
25	27	0.00027282	-0.00426907	-0.056880858	0.680573605	0.143513738	0.225102137	0.090670972	0.004110469	0.86030793	-1.0506724	0.818436742	1.91098033	-0.00426907	0.2251
26	28	-0.000959599	-0.005351371	0.0719818071	0.514262189	0.155674051	0.236902314	0.056122706	0.004325946	1.0246972	-0.822399076	-2.878796246	1.84868796	-0.005351371	0.2369
27	29	-0.001519789	-0.004990603	0.042697006	0.544531554	0.203634277	0.233671206	0.054602232	0.004266944	0.87521964	-0.8379886	-4.59366659	1.71231824	-0.004990603	0.2336
28	30	-0.002498669	-0.002765877	-0.070108965	0.599437184	0.065906186	0.210969307	0.044580848	0.003852397	0.85116851	-0.96529102	-7.496007147	1.81645953	-0.002765877	0.2109
29	31	0.000238547	-0.006493798	-0.22127014	0.642090906	0.203863555	0.230671311	0.053289254	0.004212165	0.99631691	-0.84154809	0.715640222	1.837865	-0.006493798	0.2306
30	32	-0.000587648	-0.007816609	-0.21742196	0.500747872	0.158732882	0.235681806	0.055545914	0.004303659	0.97743678	-0.91454315	-1.762945174	1.89197993	-0.007816609	0.2356
31	33	0.000549167	-0.009079291	-0.15967779	0.402087314	0.274281668	0.217743634	0.04741229	0.003976099	0.87738425	-0.72718507	1.647500557	1.60456932	-0.009079291	0.2177
32	34	0.000172807	-0.004752093	0.055678301	0.299134646	0.135114217	0.211161141	0.044538914	0.003855865	0.86319405	-0.72995096	0.516421558	1.59314501	-0.004752093	0.2111
33	35	0.000690512	-0.003368009	0.025100837	0.713170482	0.144818538	0.217021442	0.04741229	0.003976099	0.87738425	-0.72718507	1.647500557	1.60456932	-0.003368009	0.2111

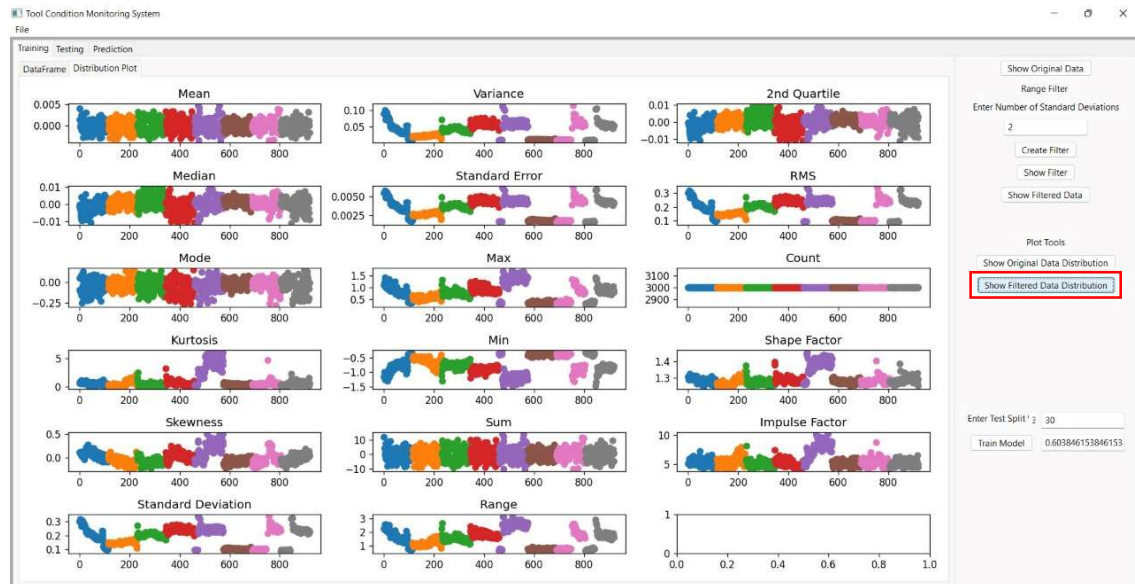
Step 3: Create filter using standard deviation and “Show Filter” button will display the range of filter



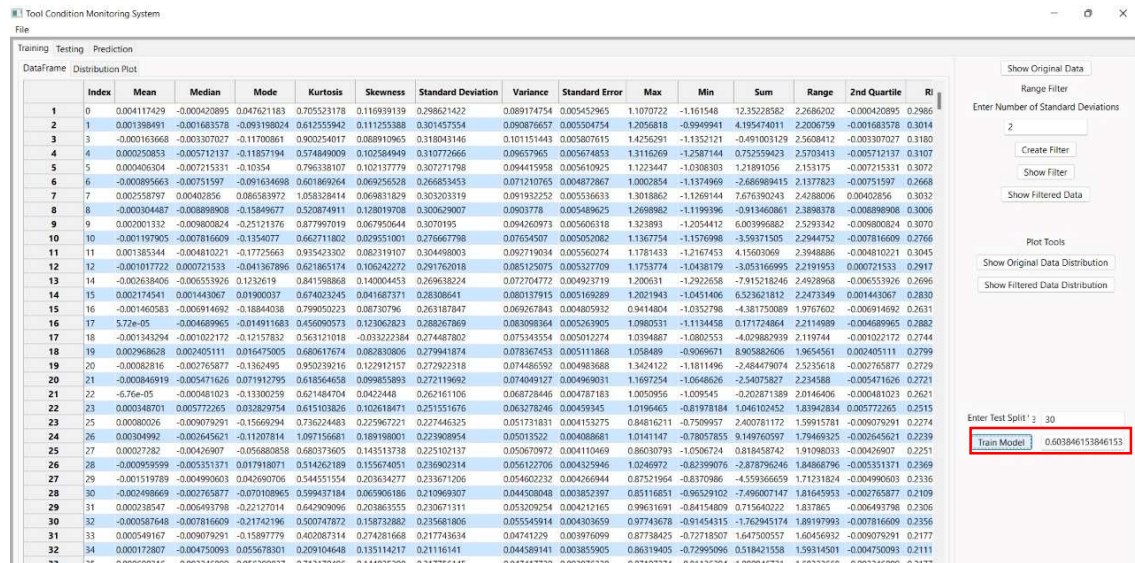
Step 4: Original data is plotted



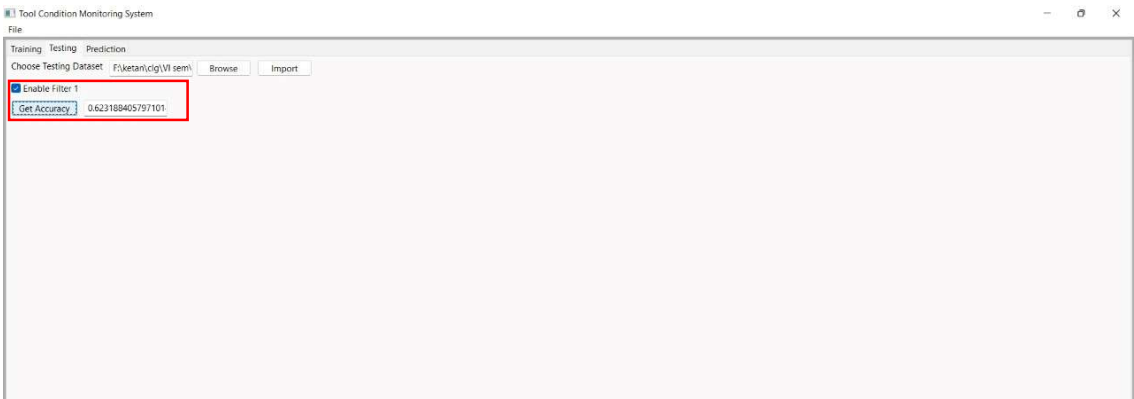
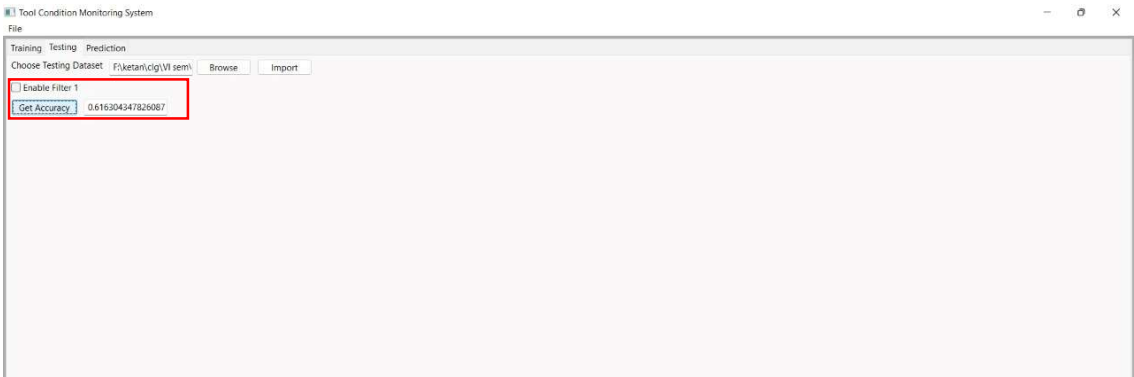
Step 5: Filtered data is plotted



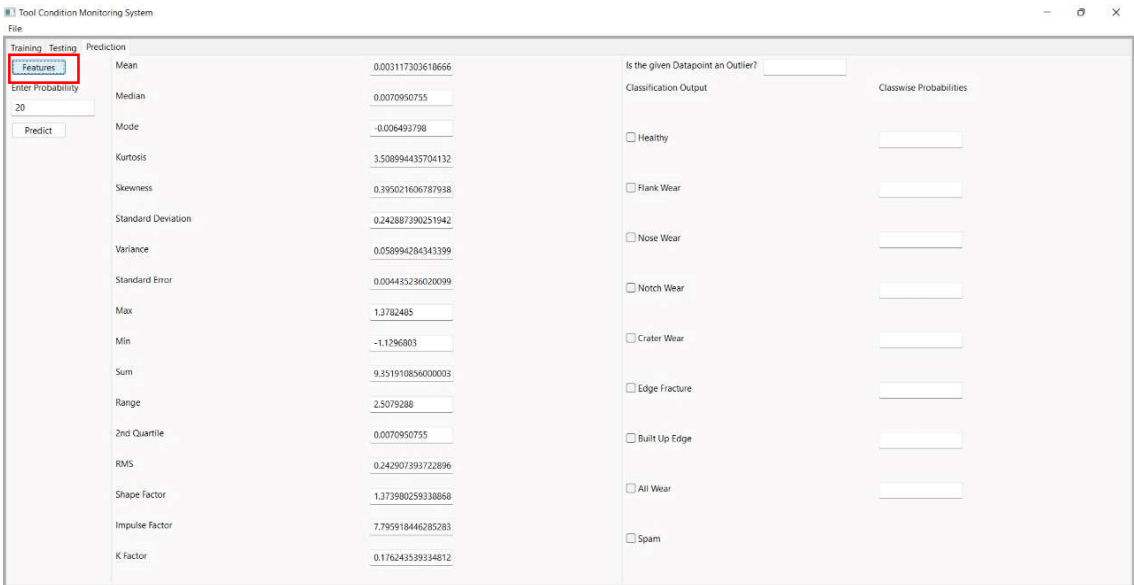
Step 6: Enter test-train split percent and train model, accuracy is displayed



Step 7: Import testing data and find model accuracy without applying filter 1 and applying filter 1



Step 8: Display features of testing data



Step 9: Classify testing data into different classes or spam

Tool Condition Monitoring System

File

Training Testing Prediction

Features

Enter Probability
20
Predict

Mean

311730361866668

Median

0.0070950755

Mode

-0.006493798

Kurtosis

3.508994435704132

Skewness

0.395021606787938

Standard Deviation

0.242887390251942

Variance

0.058994284343399

Standard Error

0.004435236020099

Max

1.3782485

Min

-1.1296803

Sum

9.351910856000003

Range

2.5079288

2nd Quartile

0.0070950755

RMS

0.242907393722896

Shape Factor

1.373980259338868

Impulse Factor

7.795918446285283

K Factor

0.176243539334812

Is the given Datapoint an Outlier?

It is not an outlier

Classification Output

Classwise Probabilities

☐ Healthy

0.0

☐ Flank Wear

0.0

☐ Nose Wear

0.0

☐ Notch Wear

0.0

☒ Crater Wear

100.0

☐ Edge Fracture

0.0

☐ Built Up Edge

0.0

☐ All Wear

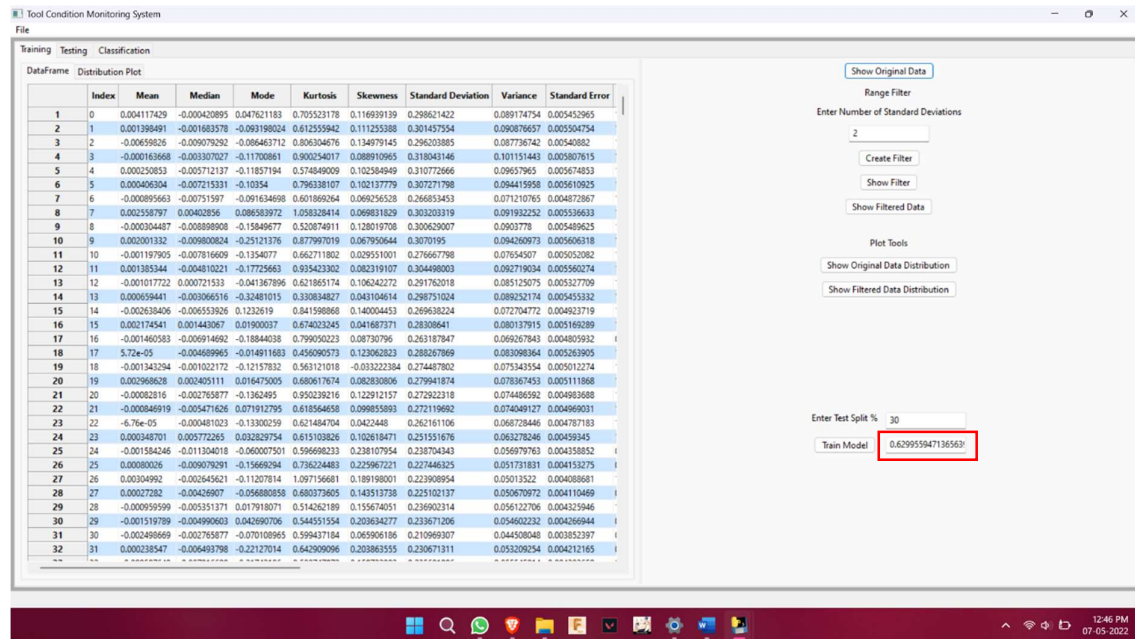
0.0

☐ Spam

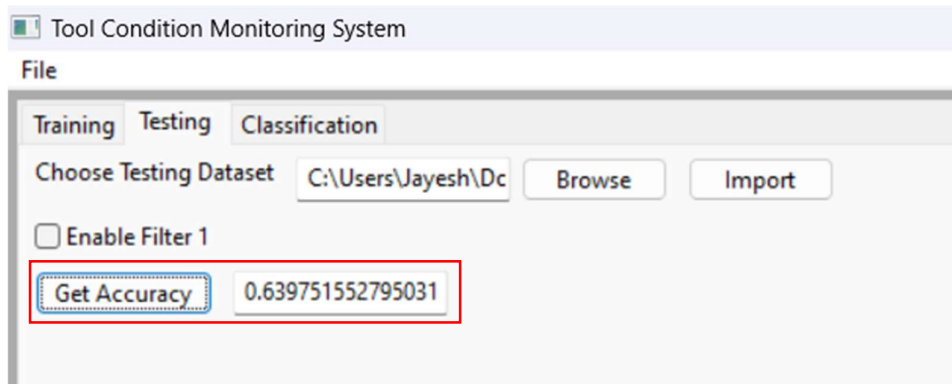
Chapter 5- Results and Discussion

5.1- Results in Status Quo

The training model accuracy of the classifier is 62.995%

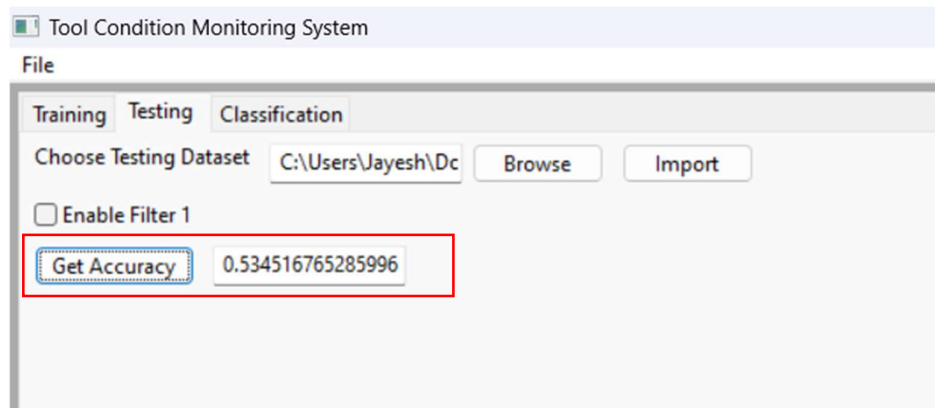


5.1.1: Classification without alien or outlier data



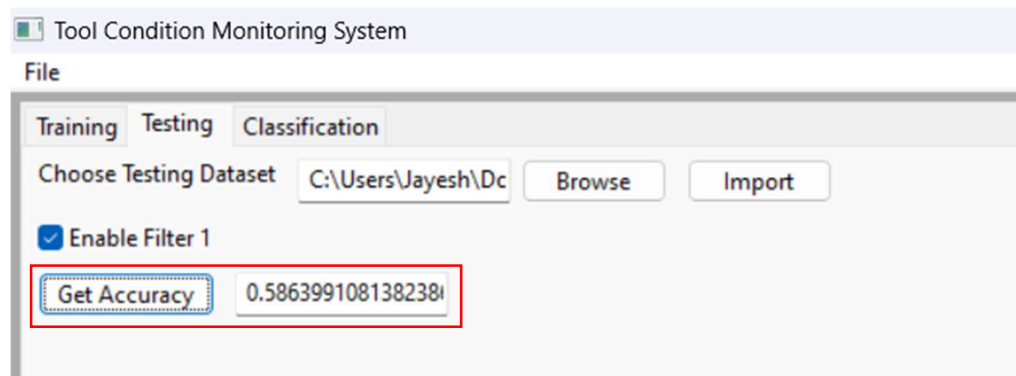
But this kind of data is not feasible in real life as there would always be outliers and alien classes present.

5.1.2: Classification with outlier and alien data



Thus, we see that presence of outlier and alien data reduces the accuracy of the model. (From 63.97% to 53.45%)

5.2- Classification of data with alien and outlier data with our model



Thus, we see that the model does reduce misclassification of data hence leading to more accurate results (from 53.45% to 58.64%)

Chapter 6: Conclusion and future scope

6.1 Conclusion

The presence of outlier and alien data reduces the accuracy of the model and increases misclassification.

Our statistics-based Navier-bayes classifier model not only increases accuracy and reduces misclassification, but also provides quality of life advantages like letting the operators know of damaged set up or about the lack of data.

6.2 Future scope

The future scope of this project lies in increasing the accuracy of the model through more training data.

Another future scope would also be to be able to set a threshold point which not only breaks even at zero cost zero profit, but can shift towards profit optimisation.