

Original Research

Received 08 May 2026

Revised 07 June 2026

Accepted 20 June 2026

Natural Resources for Human Health 2026, Vol. 6 (5): 1397–1408

KEYWORDS:

CNC machines, Monitoring, MSME sector, Sensors.

Conditional Monitoring of Industrial Machines

Dr. V. Arulkumar¹, Dr. R. Lathamaju², Dr. R. Ranjana³, Ms. S R Indurekaa⁴, Dr. Christeena Joseph⁵, Dr. K. Thilagavathi⁶

¹ Associate Professor, Department of CSE, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India. arulkumar.v@eec.srmmp.edu.in

² Associate Professor, Department of ECE, SRM Institute of Science and Technology, Ramapuram Campus, Chennai, Tamil Nadu, India. lathamar@srmist.edu.in

³ Professor, Department of Artificial Intelligence and Data Science, Dr Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India. ranjana.g@gmail.com

⁴ Assistant Professor, Department of Artificial Intelligence and Data Science, Dr Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India. indurekaa@gmail.com

⁵ Professor, Department of ECE, SRM Institute of Science and Technology, Ramapuram Campus, Chennai, Tamil Nadu, India. chrstej@srmist.edu.in

⁶ Assistant Professor, Department of ECE, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India. thilagavathi.k.ece@kct.ac.in

ABSTRACT: The MSME industries manufacture spare parts for different industrial sectors at major quantities. This huge production is attained by Computer Numerical Controller (CNC) machines. The downtime of these CNC machines induces a major loss in their production line. The major symptoms of malfunctioning of the CNC machine are observed by increased radiator temperature and reverberations (irregular vibrations). It is essential to monitor these vibrations to predict the failure and go for periodical maintenance. The aim of our project is to help automate the Real-time condition monitoring of a CNC machine, which has been a challenging task for many operators in MSME sectors. It also proposes to monitor the maintenance of the CNC machine and alert the user (operator) if there are any abnormality in the vibration and temperature pattern. Temperature Sensor and Vibration Sensor are used to monitor the radiator temperature and CNC machine vibrations respectively. These sensors are connected to Raspberry Pi board with the help of Arduino UNO and the sensed information are constantly read. These observed data are simultaneously pushed to a cloud for further storage and analysis. The stored data are analysed statistically using control charts methods for validating the process control. The results of the analysis process are presented in a graphical representation to the user (operator) via Android application. Any abnormal condition of the CNC machine are immediately indicated to the operator via an alert message.

1. INTRODUCTION

For a long time, machine tools have served as the foundation of various production sectors. These days, they are vital to the world economy. These sectors produce large quantities of spare components for several industrial sectors. It is the Computer Numerical Controller (CNC) equipment that achieve this enormous productivity. CNC machines are crucial to the manufacturing sector. These specialized devices, which are operated by computers, offer a degree of uniformity, accuracy, and efficiency that is not feasible with manual labor. These devices, which process materials with varying degrees of hardness to create goods with complicated patterns, require maintenance for optimal functionality, dependability, and—above all—profitability.

Their production line suffers a significant loss when these CNC machines are not in use. The stability, overall production efficiency, and dependability of CNC lathes are all impacted by the spindle unit. The spindle is exposed to high loads and temperatures when the machine is in operation. In the manufacturing sector, spindle problems are a major cause of CNC lathe downtime.

Data gathered from 250 machine tools in the automotive industry indicates that four main variables contribute to machine tool stoppages, according to German research. 26 percent of the overall downtime was caused by malfunctions in the tool changer and spindle equipment. Furthermore, this is the second most serious issue after drive axle failures, as the graph illustrates.

In this project, Real-time condition monitoring of a CNC machine, has been done with the help of a Temperature Sensor and Vibration Sensor. This section of the introduction discusses the basics and drawbacks behind the CNC machines. The following are the sections that will be presented in this paper: Section 2 discusses related works done on the Monitoring of different Industrial machines. Section 3 presents the system design and the implementation of the model in detail, Section 4 represents the experimental results and analysis and Section 5 presents a conclusion and recommendations for future directions.

2. RELATED WORKS

Using IoT for predictive maintenance, Yahya Mohammed Al-Naggara, Norlida Jamila, Mohd Firdaus Hassan, and Ahmad Razlan Yusoffa talked about the conditional monitoring of the state of four CNC machines put in different locations at the same time. The accelerometer's MMA8451 sensor, which is used to check machine condition, is utilized for machine monitoring. The Arduino UNO microcontroller, which is linked to the Raspberry Pi microcomputer, is coupled to this sensor. This makes it easier to read sensor data and upload it to the cloud. The backend database is a Firebase Realtime Database. It is possible to calculate acceleration and, by extension, velocity, by using signal processing techniques such as FFT. Ultimately, the machine condition is categorized as good, satisfactory, unsatisfactory, or unacceptable based on the RMS velocity. The findings demonstrated that the CNC machine's condition could be determined by the acceleration signal in both the time and frequency domains.[1]

Wan Rahiman and Mohamad Hazwan Mohd Ghazali give a thorough review of modern vibration analysis for machine monitoring and diagnostics. It involves gathering data (with tools like analyzers and sensors), extracting features, and applying artificial intelligence (AI) to defect detection techniques. The data acquisition stage includes a variety of sensor mounting techniques, including permanent installation, adhesive mounting, magnetic mounting, and non-mounting, as well as the various vibration sensor and analyzer types that are used to gather and analyze vibration data. Over the years, researchers have accomplished signal processing or feature extraction using the time, frequency, and time-frequency domains. Numerous AI-based techniques, including support vector machines (SVM), neural networks (NN) with deep learning capabilities, fuzzy logic, and genetic algorithms (GA), are investigated during the defect recognition stage. The purpose of this publication is to address several research questions (RQs). It is anticipated that deep learning algorithms and time domain statistical properties will be widely used in the future to automatically extract defect features from unprocessed vibration data. Vibration monitoring and diagnosis will face a new and major challenge with the development of smart machines, as various sensors and communication devices are integrated.[2]

Kovač, P., Mankova, I., Gostimirović, M., Sekulić, M., & Savković, B. written their study in response to a recent review of the literature on machining monitoring systems. The study looks at earlier work in these areas and provides a current, in-depth analysis of methodology, sensor technologies, signal processing, decision-making strategies for process monitoring, and integrated workpiece quality assessment. It discusses the direct and indirect monitoring methods, which are two conventional machine monitoring techniques. We deduce that the most prevalent kind of vibration sensing in machining operations is piezoelectric transduction. To measure tool wear, a vision system is also employed. One application that is mentioned is sensor systems. It comes to the conclusion that the instrumentation technology used to measure these qualities has developed and become more sophisticated, making them feasible, practical, affordable, long-lasting, and simple to install. It also notes that the reaction time required to detect changes for on-line machining process monitoring has gotten shorter. Future issues and advancements in sensor-based machining operation monitoring are examined.[3]

It has been mostly overlooked by academics, according to Sheng Zhang, Joseph Mathew, Lin Ma, Yong Sun, and Avin Mathew, that designing control limits for condition monitoring is an essential component of creating maintenance plans. In this study, a new statistical process control tool for identifying condition variation is proposed: the Weighted Loss function CUSUM (WLC) chart. Using data from the baseline condition, an autoregressive model was fitted to the process, and the residuals were used as

the chart statistic to create the control limit. Condition variation can be seen by comparing the changes in the statistic's mean and variance against the baseline condition, which can be seen on a single WLC chart. The method was tested using a case study, and the results showed that the chart can identify defective circumstances and their severity. One advantage of the suggested approach is that condition classification only needs data from healthy baselines. In many situations where data from troublesome conditions is not available, it can be applied. Setting control limits is facilitated by the Weighted Loss the CUSUM (WLC) control chart's ability to identify shifts in both mean and variance. It was successfully used for condition monitoring with shaft misalignment as the example study. It has been demonstrated that the chart can both identify healthy and defective conditions and the severity of the problem.[4]

3. SYSTEM DESIGN

The design of the proposed system is shown below:

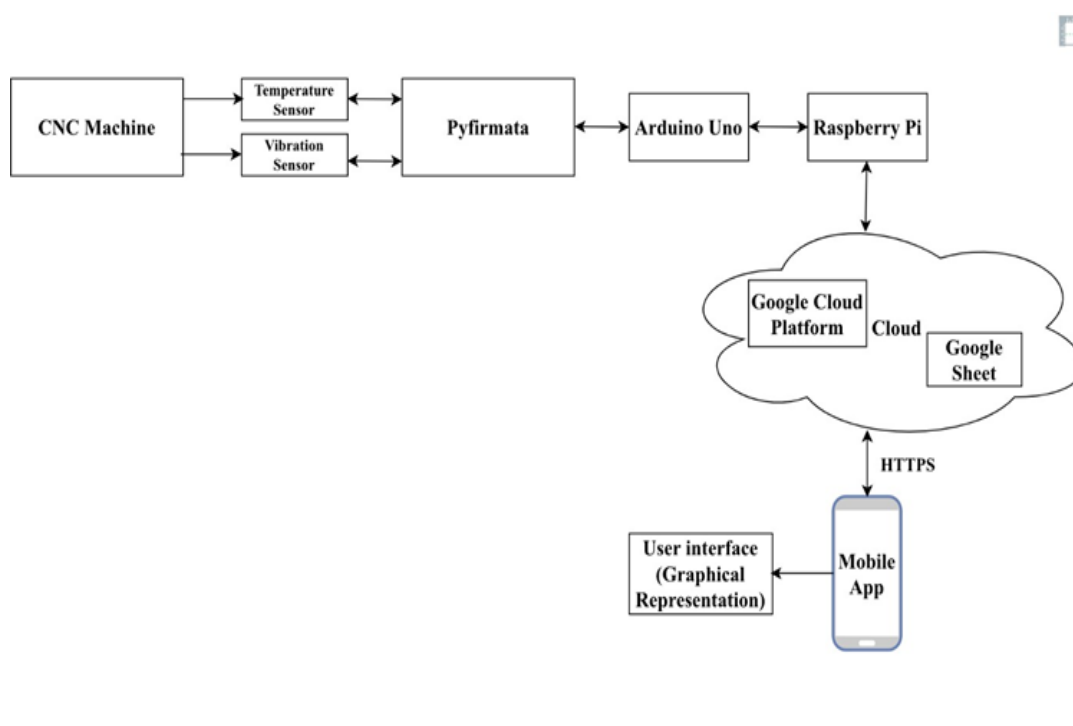
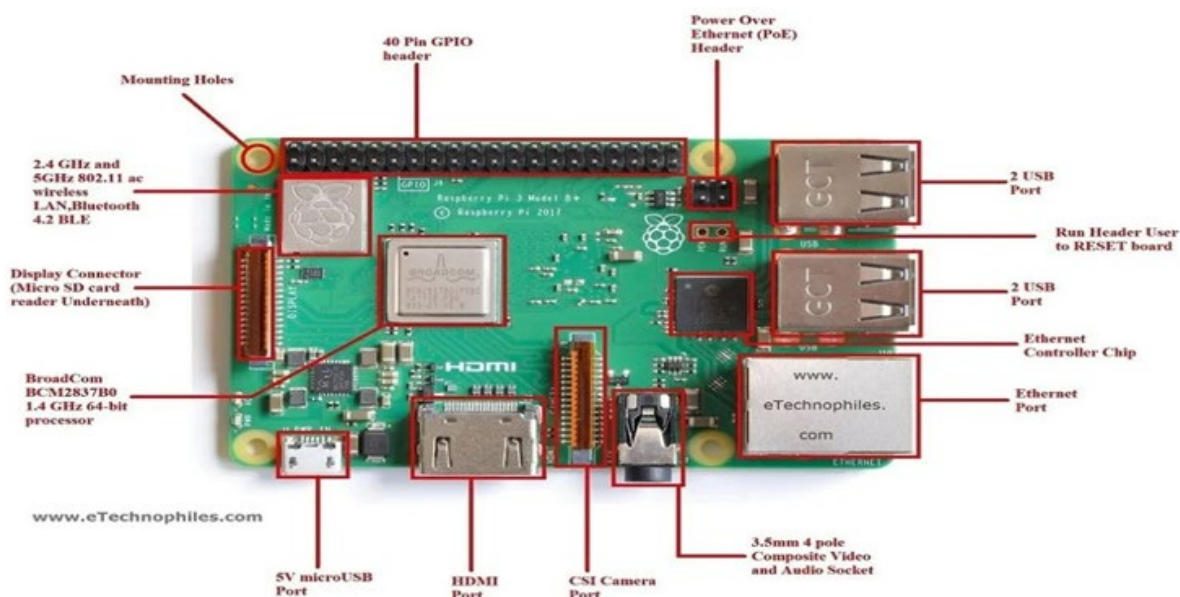


Fig.1. Description of system Design Diagram

3.1 RASPBERRY PI

This 1.4GHz 64bit ARM quad-core Cortex A53 CPU, known as the BCM2837B0 CPU, is the main component of the Raspberry Pi Model 3b+. The R-Pi 3 B+ has a 1GB LPDDR2 SDRAM. OpenGL ES 2.0, an OpenGL API that is hardware-accelerated, and 1080p30 H.264 high-profile decoding are used. 24 GFLOPs, 1.5 Giga texel/s, or up to 1 Gigapixel/s can be achieved by a general-purpose computer. This device contains four USB ports. Additional improvements include the B+ edition's micro-SD memory card slot in place of the SD memory port. The status LEDs on the board are now only red and green after being moved to the other end.

The Raspberry Pi board can be connected to external input/output devices using GPIO pins. Model B+, like the previous model, has a 40-pin GPIO. A GPIO pin that is set to input receives the signal that the device connected to this pin sends to the Raspberry Pi. Any voltage less than 1.8V is interpreted by the Raspberry Pi as LOW, and any value between 1.8V and 3.3V as HIGH. The voltage signal is transmitted as either high (3.3V) or low (1.5V) (0V) when a GPIO pin is set up as an output pin. The output voltage is 3.3V when this pin is set to HIGH and 0V when it is set to LOW.



Parts of Raspberry Pi 3 B+

Fig.2. Raspberry Pi

3.2 ARDUINO UNO

The Arduino UNO is an open-source, programmable microcontroller board that is inexpensive, versatile, and simple to use. It can be used in a wide range of electronic projects. This board can control relays, LEDs, servos, and motors as an output and can interface with other Arduino boards, Arduino shields, and Raspberry Pi boards. The ATmega328 microcontroller belongs to the Atmel family and is a single chip microcontroller. Its processor code is eight bits long. It includes registers, I/O lines, SPI serial ports, memory (SRAM, EEPROM, and Flash), oscillator, timer, and external and internal interrupts. By using the Arduino board's firmware, users can program using the In-Circuit Serial Programming (ICSP) pin.

The LED's ON state indicates that the power is on. The LED won't turn on when the power is off. The values of digital I/O pins are either HIGH or LOW. Digital pins have numbers ranging from D0 to D13. The TX and RX LEDs show that the data flow is successful. The Arduino UNO board receives a reference voltage from the external power supply via the Analog Reference (AREF) pin. The board and computer can connect thanks to USB. It is necessary in order to program the Arduino UNO board. The purpose of Analog Pins A0 through A5 is to read the analog sensor that is connected to the device. Also, it is capable of functioning as GPIO (General Purpose Input Output) pins.

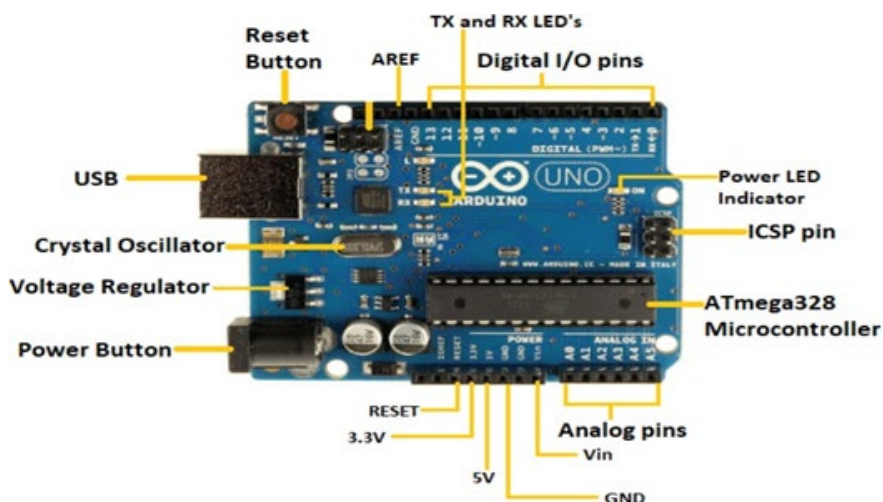


Fig.3. Parts of Arduino UNO

3.3 DHT11 (TEMPERATURE SENSOR)

A cheap digital temperature and humidity sensor is the DHT11. To detect humidity and temperature in real time, this sensor can be easily connected to any microcontroller, such as an Arduino, Raspberry Pi, or other similar device. The DHT11 sensor is composed of a thermistor for temperature detection and a capacitive humidity detecting element. The resistance value decreases as the temperature increases because this sensor uses a Negative Temperature Coefficient Thermistor to measure temperature. Semiconductor ceramics or polymers are frequently used in its construction to achieve a higher resistance value even at very small temperature changes. With a 2-degree accuracy, the DHT11 can measure temperatures between 0 and 50 degrees Celsius. Put another way, it provides one reading every second. The DHT11 is a tiny transistor that operates between three and five volts. The maximum current used for measurement is 2.5 mA. The four pins on the DHT11 sensor are connected to VCC, GND, Data, and a not connected pin. A 5k–10k ohm communication resistor is offered.

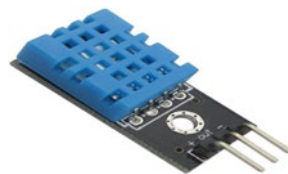


Fig.4. DHT11 - Temperature Sensor

3.4 PIEZOELECTRIC SENSOR (VIBRATION SENSOR)

A piezoelectric sensor is a gadget that uses the piezoelectric effect to convert changes in force, strain, temperature, acceleration, pressure, or other physical variables into an electrical charge. A charge is generated across some materials when a mechanical stress is applied to them. By measuring the voltage produced by the applied pressure across a piezoelectric element, piezoelectric pressure sensors make use of this phenomenon. They can be found in many different industrial settings and are very durable.

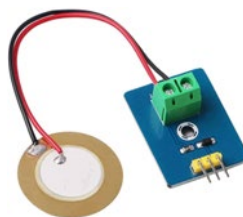


Fig.5. Piezoelectric Vibration sensor

3.5 PYFIRMATA PROTOCOL

The Firmata protocol has a Python interface called PyFirmata. Firmata is a protocol that allows software on a host computer to communicate with microcontrollers. The protocol may be implemented in both firmware and software on any microcontroller architecture and any host computer software package. Although the Raspberry Pi and Arduino are two different hardware platforms in terms of applications and construction, they are both regarded competitors in the open-source hardware space. They both have a large and supportive community. Using a Raspberry Pi python script, PyFirmata firmware is utilized to send commands to Arduino. PyFirmata is a python library package that can be loaded in Arduino to allow serial communication between a python script running on any computer and an Arduino.

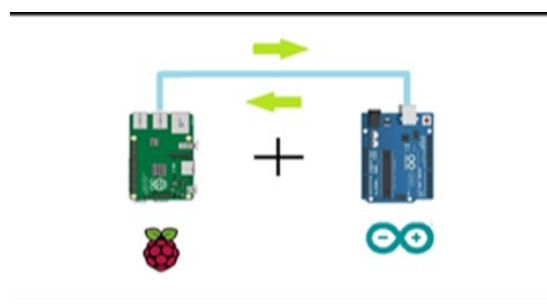


Fig.6. PyFirmata Protocol Working

3.6 DATA IN JSON FORMAT

Using human-readable text (or other serializable values), JSON (JavaScript Object Notation) is an open standard file format and data exchange format that stores and transmits data objects made up of attribute-value pairs and arrays. It's a widely used data format that can be used for a variety of data transfers, including server and web app communication. Language-neutral data formats include JSON. Code to generate and parse data in JSON format is included in many modern computer languages, despite being derived from JavaScript. JSON files are named with the .json extension. The extracted data in JSON format is entered into the database, where the values are represented as tuple data and the key is the column name.

3.7 ANDROID STUDIO

Android Studio is the official Android application development integrated development environment (IDE). The platform it is built upon is IntelliJ IDEA, an integrated development environment (IDEA) for Java that comes with developer tools and code editing capabilities. It lets you to create apps in one environment for Android Wear, Android TV, Android Auto, Android phones, and tablets. Using structured code modules, you can divide your project into manageable chunks that you can work on independently and test, develop, and debug.

3.8 STATISTICAL PROCESS CONTROL CHARTS

A graph used to examine how a process changes over time is the control chart. It's a control mechanism for figuring out whether a business or manufacturing process is under control. The information is presented chronologically. On a control chart, the average is always shown by the center line, the upper line for the upper control limit, and the lower line for the lower control limit. By comparing the current data to these lines, you can determine whether the process variation is consistent (under control) or unexpected (out of control, affected by special causes of variation).

The process mean and standard deviation are often analyzed over time using control charts, such as X Bar S charts. S charts are used when subgroups have a large sample size because they provide a clearer understanding of the distribution of subgroup data than range charts do. Plotting the subgroup standard deviation on X bar S charts and the subgroup range on R charts is the primary difference between the two types of control charts. Choosing the appropriate control chart is essential when mapping control charts; otherwise, the data's control limits will be incorrect.

The standard deviation is a more accurate indicator of variance than the range when the subgroup has a large sample size because it includes all of the data rather than just the lowest and maximum values. This example of statistical process control uses two charts to track the process mean and process range over time (as represented by the standard deviation of the data). These combination charts are useful for identifying special causes of variation and for understanding process stability.

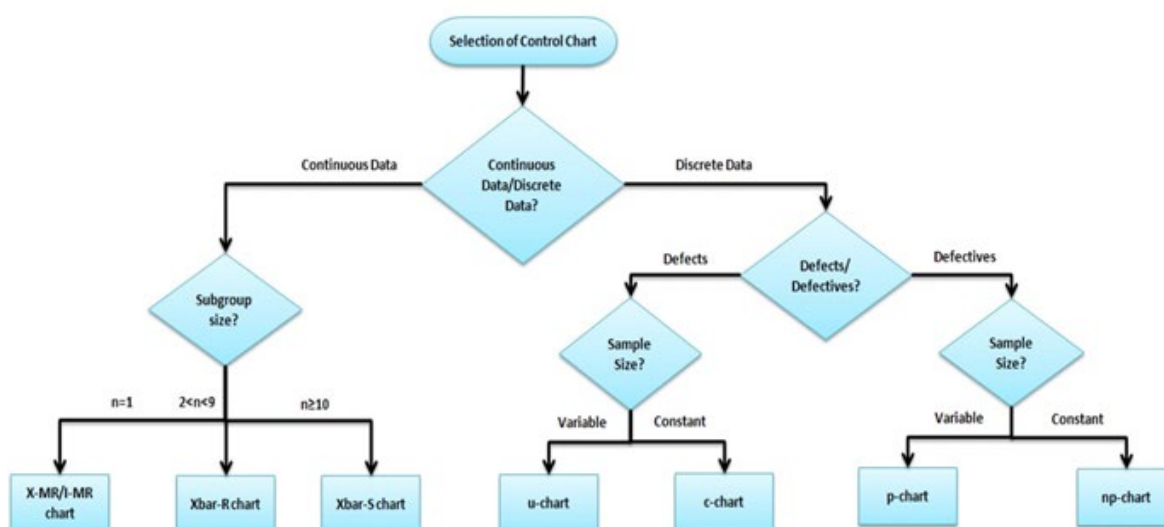


Fig.7. Control chart selection tree

In this paper the algorithm for calculating the various control limits for both the S Chart and X bar charts is done without using of control chart constants. The principle of control charts is followed to achieve a new algorithm. Both the X bar chart(to represent mean) and S chart(to represent standard deviation) used in this model requires parameters like LCL, UCL, CL, Hv, Lv. The algorithm to calculate these parameters for the charts are as follows: The data is collected in real-time is grouped at a standard sample size and sorted in ascending order. The median of this sorted data is found and the entries above the median constitute the LCL band and the entries above the median constitute the UCL band. The average of the LCL band entries is found and the deviation of each data entry in this band from this average is found. The overall average of all these deviations constitute to the LCL value. Similarly, the average of deviation of the UCL band entries from the UCL band average form the UCL value. The overall deviation of the sampled data forms the CL value. The highest deviated value forms the Hv and the lowest deviated value forms the Lv.

The below shown is the formula for mean calculation for a population.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$
$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

σ : Population standard deviation
 x : Datapoint value
 μ : Population mean
 N : Population size

where, n = sample size the below shown is the formula for standard deviation calculation of population.

The mean or average process change over time from subgroup values is displayed in the X bar chart. The sample mean and center are taken into account by the control limits on the X-Bar. The deviation of each entry from the mean is displayed on the S chart. The S bar (average standard deviation) values are used to determine the X bar chart control limits; if the values in the S chart are out of control, the X bar chart control limits are also out of date.

4. IMPLEMENTATION AND RESULTS

This chapter explains the implementation of the project with the proposed system architecture.

Step 1: SENSING DATA VIA TEMPERATURE SENSOR

Adafruit's version of Micro Python, called Circuit Python, is intended to make learning and experimenting with inexpensive microcontrollers easier. Rather than needing to use intricate low-level languages like Arduino, C, or C++, you can write clear and straightforward Python code to control hardware with Circuit Python. We are able to read the temperature sensor's input by utilizing this Adafruit library.

Step 2: SENSING DATA VIA VIBRATION SENSOR

Step 3: PUSHING DATA TO GOOGLE SPREADSHEET

Google Cloud Platform enables you to create, launch, and scale apps, websites, and services on Google's infrastructure. The google cloud is used as the platform to store the sensed data. The read data is populated in a google spreadsheet in the linked google drive account via google sheet API.

Step 4: FETCHING DATA FROM GOOGLE CLOUD

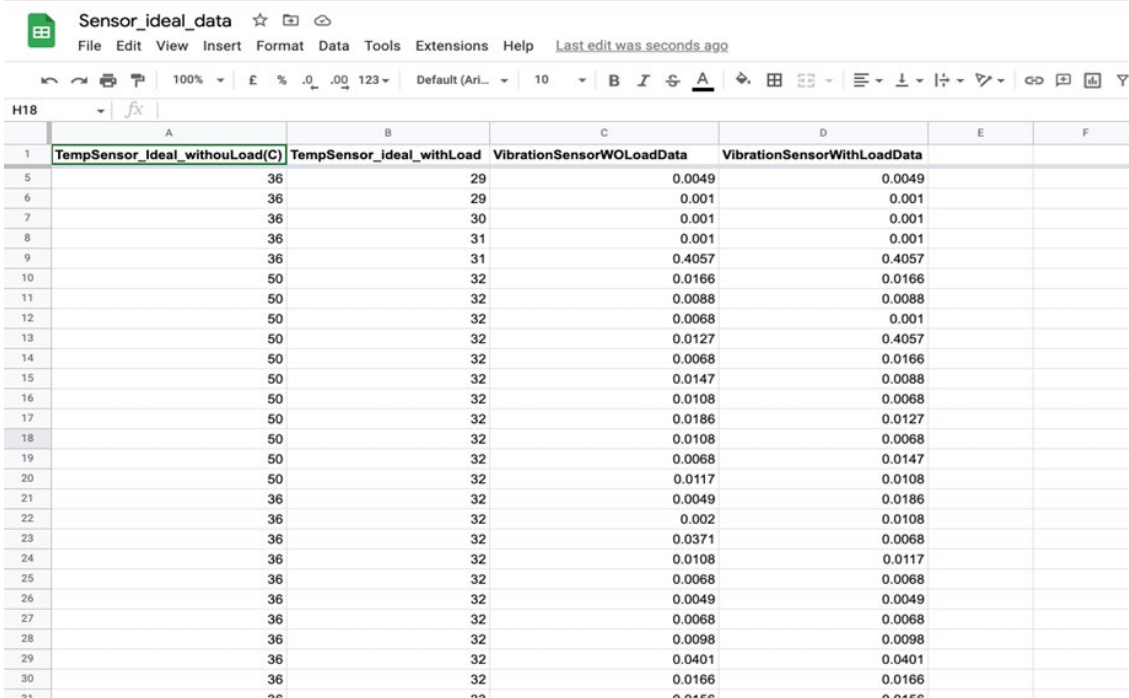
Step 5: STANDARD DEVIATION CALCULATION

Step 6: PLOTTING CONTROL CHARTS

MPAndroidchart - A robust Android chart view and graph view library, MPAndroidchart supports line, bar, pie, radar, bubble, and candlestick charts in addition to scaling, panning, and animations.

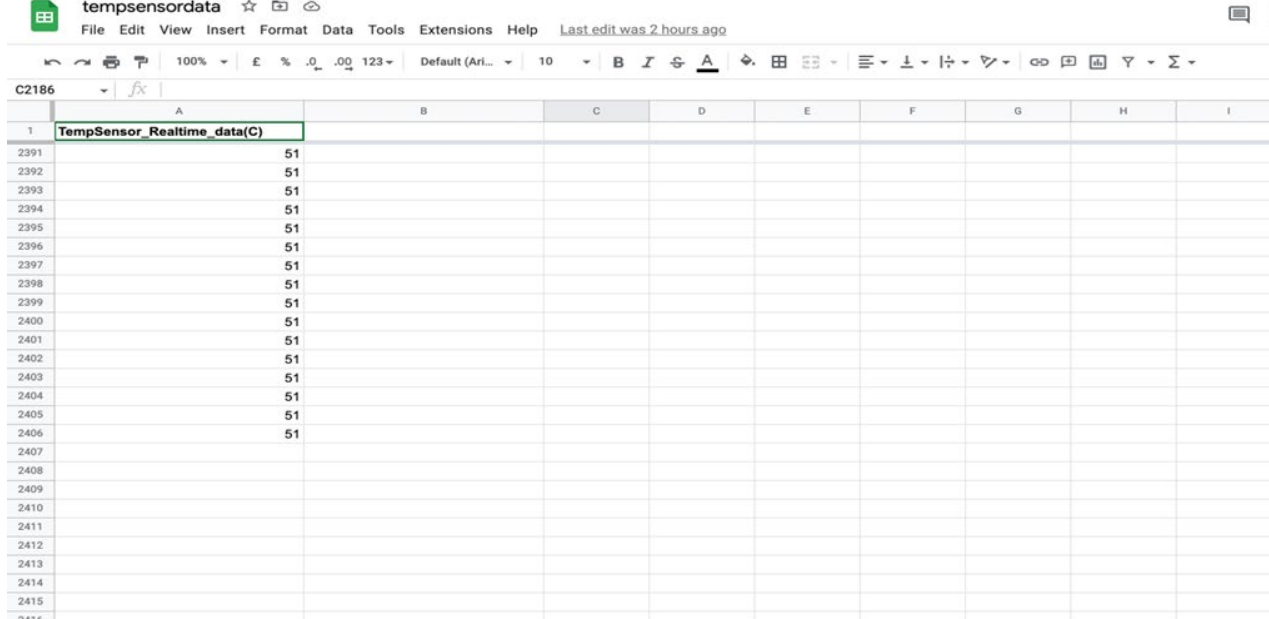
MPAndroidChart - An intuitive chart library for Android is called MPAndroidChart. Information that changes over time is displayed using a line chart. Plotting a number of points and connecting them with a straight line produces line charts. To monitor changes over both short and long time periods, line charts are utilized. Line graphs are preferable to bar graphs when there are smaller changes. Comparing changes for multiple groups over the same time span can also be done with line graphs.

4.1 OUTPUT



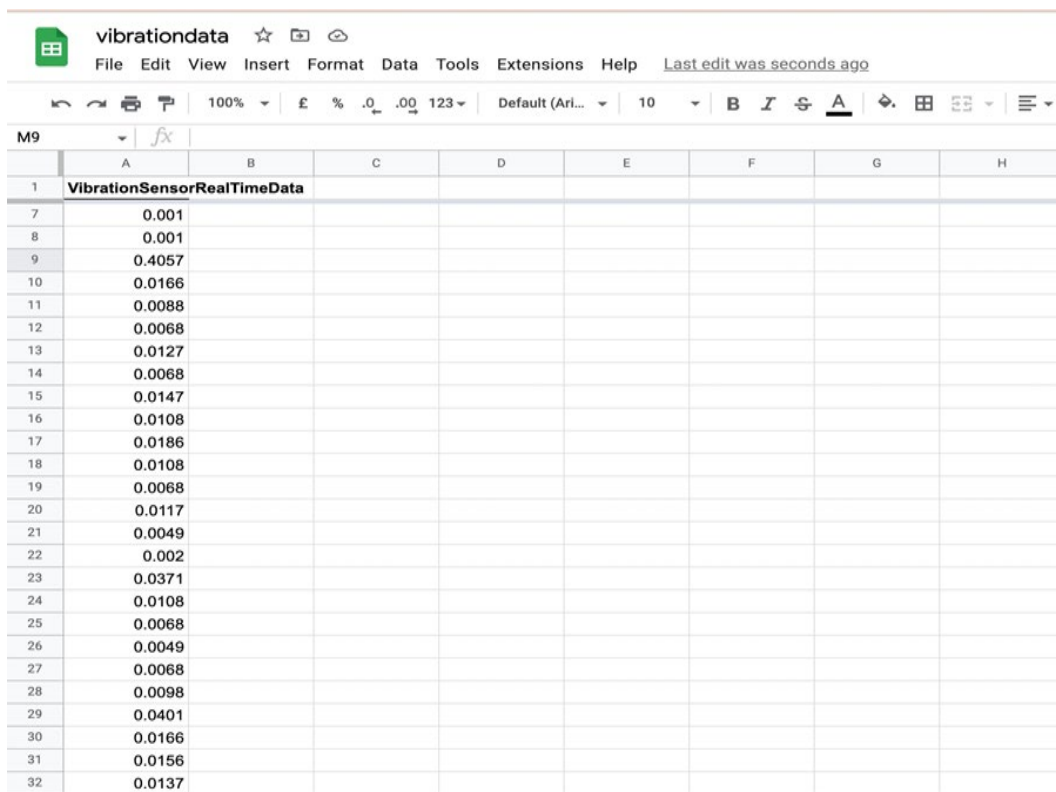
	A	B	C	D	E	F
1	TempSensor_ideal_withoutLoad(C)	TempSensor_ideal_withLoad	VibrationSensorWOLoadData	VibrationSensorWithLoadData		
5	36	29	0.0049	0.0049		
6	36	29	0.001	0.001		
7	36	30	0.001	0.001		
8	36	31	0.001	0.001		
9	36	31	0.4057	0.4057		
10	50	32	0.0166	0.0166		
11	50	32	0.0088	0.0088		
12	50	32	0.0068	0.001		
13	50	32	0.0127	0.4057		
14	50	32	0.0068	0.0166		
15	50	32	0.0147	0.0088		
16	50	32	0.0108	0.0068		
17	50	32	0.0186	0.0127		
18	50	32	0.0108	0.0068		
19	50	32	0.0068	0.0147		
20	50	32	0.0117	0.0108		
21	36	32	0.0049	0.0186		
22	36	32	0.002	0.0108		
23	36	32	0.0371	0.0068		
24	36	32	0.0108	0.0117		
25	36	32	0.0068	0.0068		
26	36	32	0.0049	0.0049		
27	36	32	0.0068	0.0068		
28	36	32	0.0098	0.0098		
29	36	32	0.0401	0.0401		
30	36	32	0.0166	0.0166		
31	36	32	0.0156	0.0156		

Fig.8. Collected Ideal value dataset for CNC Machine



	A	B	C	D	E	F	G	H	I
1	TempSensor_Realtime_data(C)								
2391	51								
2392	51								
2393	51								
2394	51								
2395	51								
2396	51								
2397	51								
2398	51								
2399	51								
2400	51								
2401	51								
2402	51								
2403	51								
2404	51								
2405	51								
2406	51								
2407									
2408									
2409									
2410									
2411									
2412									
2413									
2414									
2415									
2416									

Fig.9. Realtime Temperature Data of CNC Machine



	A	B	C	D	E	F	G	H
1	VibrationSensorRealTimeData							
7	0.001							
8	0.001							
9	0.4057							
10	0.0166							
11	0.0088							
12	0.0068							
13	0.0127							
14	0.0068							
15	0.0147							
16	0.0108							
17	0.0186							
18	0.0108							
19	0.0068							
20	0.0117							
21	0.0049							
22	0.002							
23	0.0371							
24	0.0108							
25	0.0068							
26	0.0049							
27	0.0068							
28	0.0098							
29	0.0401							
30	0.0166							
31	0.0156							
32	0.0137							

Fig.10. Realtime Vibration Data of CNC Machine

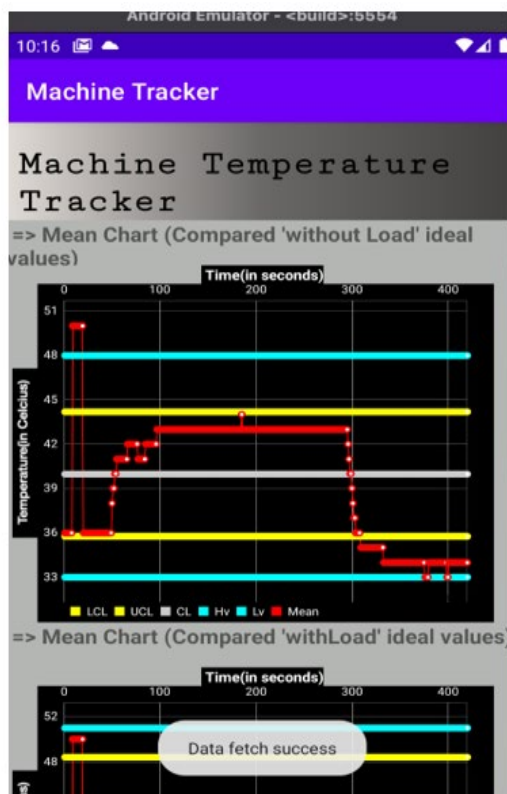
4.2 MOBILE APPLICATION OUTPUT SCREENS



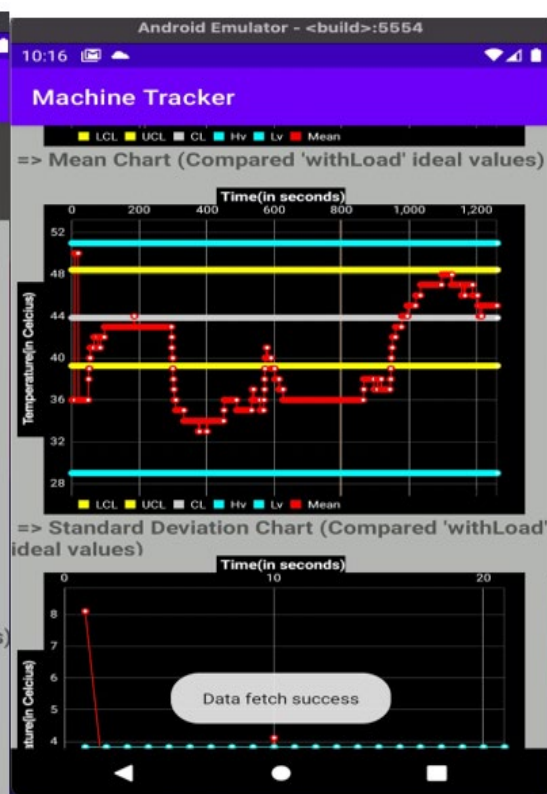
Screen 1



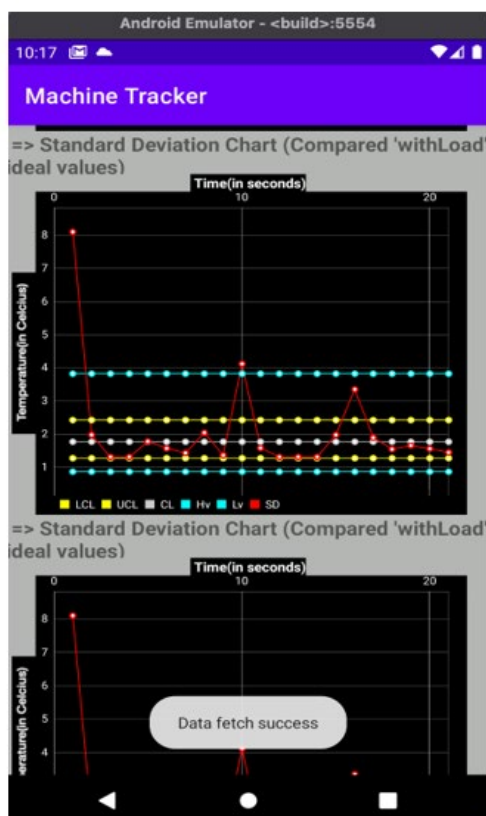
Screen 2



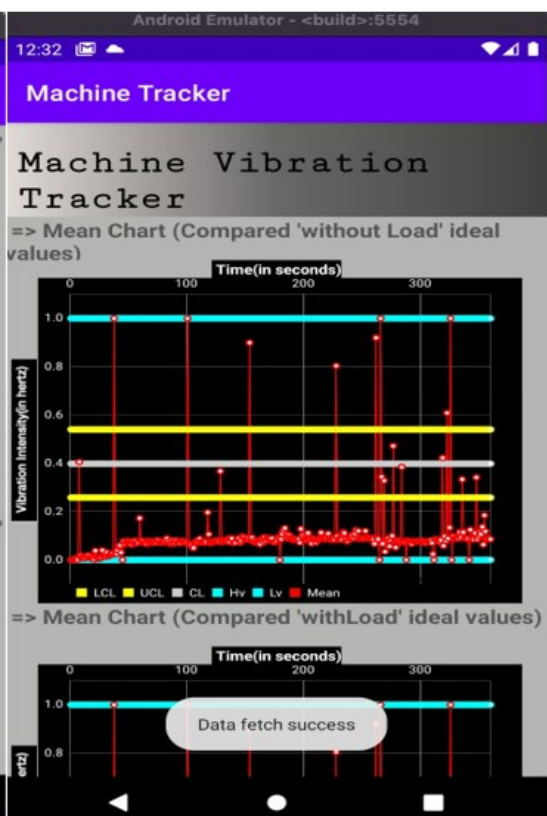
Screen 3



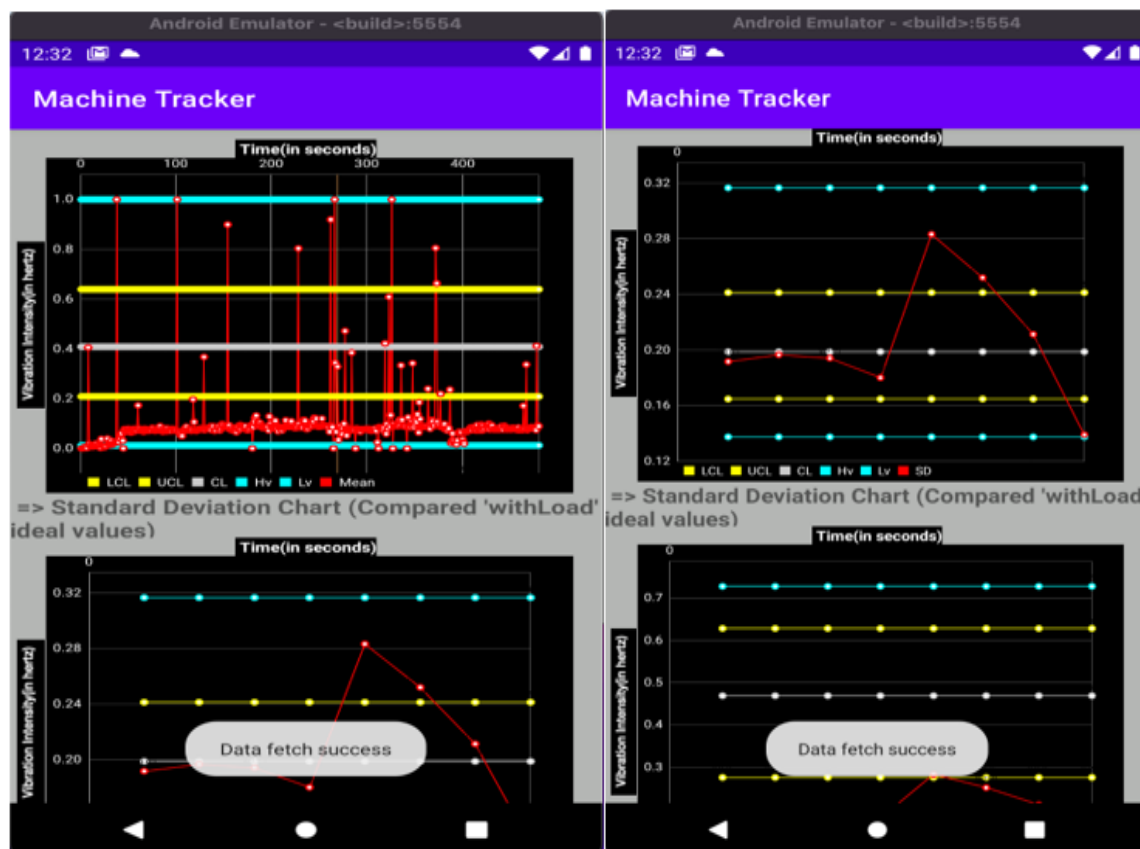
Screen 4



Screen 5.



Screen 6



Screen 7

Screen 8

Fig.11. Mobile Application Output screens

5 CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The goal of this project is to help automate the Real-time condition monitoring of a CNC machine, which has been a challenging task for many operators in MSME sectors and also to monitor the maintenance of the CNC machine and alert the user (operator) if there are any abnormality in the vibration and temperature pattern. The control charts are an unsupervised way of identifying the conditions that may be used in situations where defective condition data is missing because their design only requires data from healthy circumstances. Consequently, the Control Chart method yields a good result when it comes to identifying mean and standard deviation shifts for the control limit design. The anticipated results were obtained with the suggested model.

5.2 FUTURE WORK

If standardized sensors are made available, the algorithm can be made to achieve increased accuracy. Use of IoT platforms like Node MCU can help us make this setup as a standalone device. Invention future IOT boards with high computational capacity and GPU facilities, it will enable us to apply ML and Deep Learning methods for the real time systems.

AUTHOR CONTRIBUTIONS

Dr V Arulkumar: Conceptualization, Investigation.

Dr R Lathamanju: Methodology, Writing—Review & Editing

Dr R Ranjana: Data Curation

Ms.S. R Indurekaa: Formal Analysis.

Dr Christeena Joseph: Supervision

Dr K Thilagavathi: Laboratory Analysis.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

REFERENCES

- [1] Mohamad Hazwan Mohd Ghazali, Wan Rahiman, "Vibration Analysis for Machine Monitoring and Diagnosis: A Systematic Review", *Shock and Vibration*, vol. 2021(2021).
- [2] Kovač, P., Mankova I., Gostimirović, M., Sekulić, M., Savković, B. "A Review of Machining Monitoring Systems" *A Journal of Production Engineering* Vol.14 No.1(2011).
- [3] Sheng Zhang, Joseph Mathew, Lin Ma, Yong Sun and Avin Mathew "Statistical condition monitoring based on vibration signals" *CRC for Integrated Engineering Asset Management, School of Mechanical, Manufacturing and Medical Engineering, Queensland University of Technology, Brisbane, QLD 4001, Australia*(2004).
- [4] Yahya Mohammed Al-Naggara, Norlida Jamila, Mohd Firdaus Hassanb and Ahmad Razlan Yusoff "Condition monitoring based on IoT for predictive maintenance of CNC machines" *18th CIRP Conference on Modeling of Machining Operations ,Procedia CIRP* Vol.102 (2021).
- [5] Chen, Y.Q., Zhou, B., Zhang, M. and Chen, C.M., 2020. Using IoT technology for computer-integrated manufacturing systems in the semiconductor industry. *Applied Soft Computing*, 89, p.106065.
- [6] I-Scoop., 2020. *IoT spend and evolutions in manufacturing 2017- 2020*. [Online]. Available at: < <https://www.i-scoop.eu/internet-of-things-iot/internet-of-things-in-manufacturing/>>. [Accessed on February 3, 2022].
- [7] Kalsoom, T., Ahmed, S., Rafi-ul-Shan, P.M., Azmat, M., Akhtar, P., Pervez, Z., Imran, M.A. and Ur-Rehman, M., 2021. Impact of IOT on Manufacturing Industry 4.0: A new triangular systematic review. *Sustainability*, 13(22), p.12506.
- [8] Kao, Y.S., Nawata, K. and Huang, C.Y., 2019. Systemic functions evaluation based technological innovation system for the sustainability of IoT in the manufacturing industry. *Sustainability*, 11(8), p.2342.
- [9] Kho, D.D., Lee, S. and Zhong, R.Y., 2018. Big data analytics for processing time analysis in an IoT -enabled manufacturing shop floor. *Procedia Manufacturing*, 26, pp.1411-1420.
- [10] Lu, Y., Min, Q., Liu, Z. and Wang, Y., 2019. An IoT -enabled simulation approach for process planning and analysis: a case from engine re-manufacturing industry. *International Journal of Computer Integrated Manufacturing*, 32(4-5), pp.413-429.
- [11] Shiklo, B., 2018. *IoT in manufacturing: The Ultimate Guide*. [Online]. Available at: < <https://www.scnsoft.com/blog/iot-in-manufacturing>>. [Accessed on February 3, 2022].
- [12] Trappey, A.J., Trappey, C.V., Govindarajan, U.H. and Sun, J.J., 2019. Patent value analysis using deep learning models—The case of IoT technology mining for the manufacturing industry. *IEEE Transactions on Engineering Management*, 68(5), pp.1334-1346.