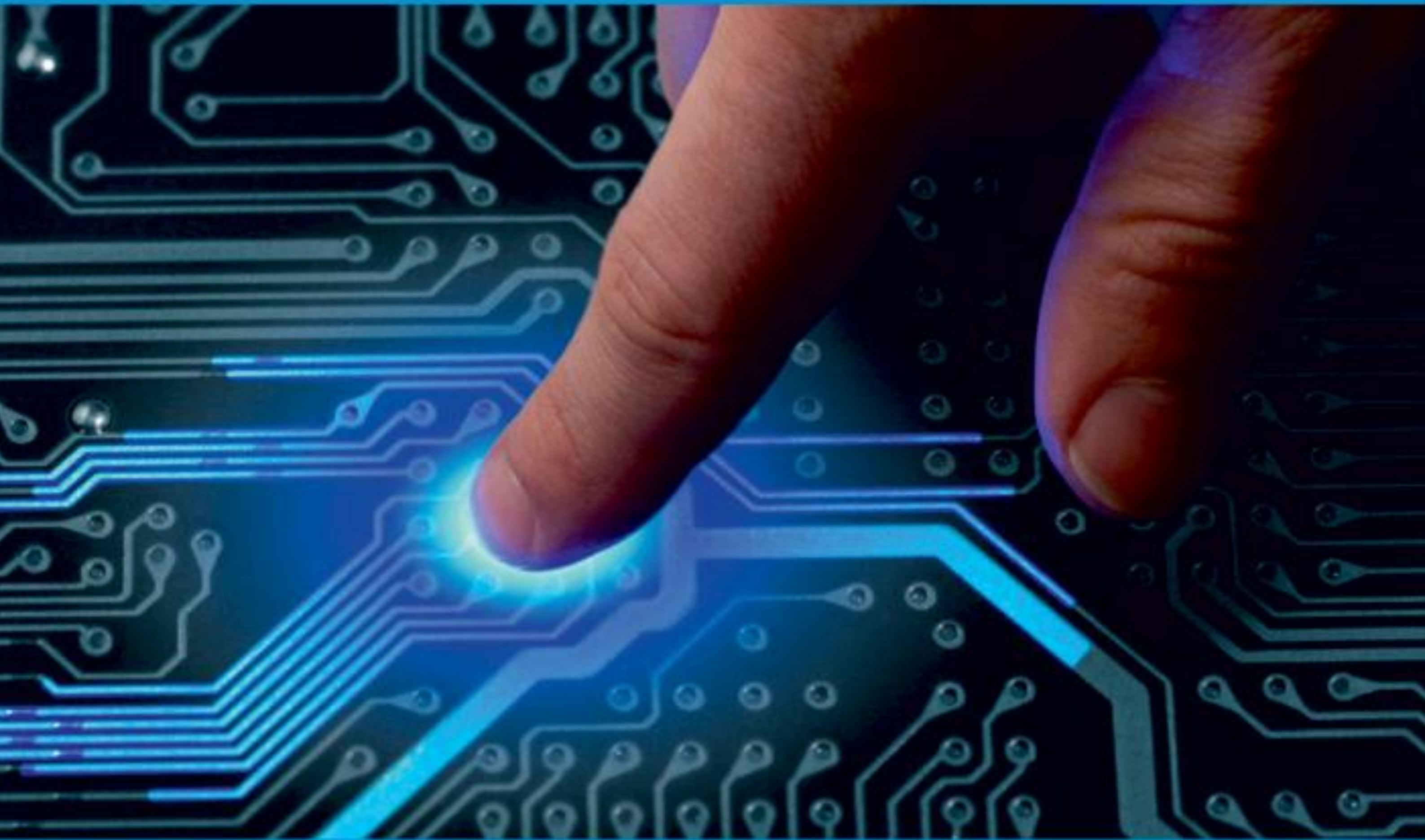
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 5, May 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

Fault Prediction in CPU FanUsing Machine Learning

Minugu B[1], Prajna Vishwanath[2], Pooja M R[2], Shree Lakshmi[2], Sinchana S[2]

Assistant Professor, Department of Electronics and Communication Engineering, PES College of Engineering, Mandya, India[1]

Final Year UG Student, Department of Electronics and Communication Engineering, PES College of Engineering, Mandya, India[2]

ABSTRACT: In modern computing environments, the reliability and performance of hardware components such as CPU fans are integral to maintaining system integrity and preventing costly downtimes. However, the occurrence of faults in these components can lead to critical failures if left undetected. Traditional maintenance approaches often lack the foresight to predict these faults before they manifest, resulting in inefficiencies and increased risks of system disruptions. Leveraging the capabilities of machine learning (ML), this study explores the potential of predictive maintenance techniques in forecasting faults in CPU fans. By analyzing historical performance data and employing advanced ML algorithms, our research aims to develop models capable of early fault detection, thus enabling proactive maintenance interventions and minimizing system downtime. Through a comprehensive review of existing methodologies, challenges, and empirical insights, this paper contributes to the advancement of predictive maintenance strategies inthe domain of computer hardware, fostering greater resilience and efficiency in contemporary computing infrastructures.

KEYWORDS: Machine Learning, Predictive Maintenance, Reactive Maintenance, TinyML, Edge impulse.

I. INTRODUCTION

A promising maintenance paradigmcalled Predictive Maintenance (PdM) uses models to forecast equipment breakdown.Preventive and reactive maintenance are anticipated to be replaced by PdM. Reactive maintenance involves fixing equipment after it breaks down, which can result in costly production downtime in factories, for example. Preventive maintenance involves replacing equipment on a prearranged schedule. This may be wasteful when equipment that is in fine operating order may be replaced. Furthermore,there is no assurance that malfunctions won't happen before repairs. Under idealcircumstances, PdM might anticipate a breakdown before it happens and schedule and carry out maintenance beforehand to prevent it. There are three common methods for putting PdM systems into practice. The first is a knowledge-based strategy that makes use of things like laws or tangible models. In the contemporary landscape of computing, ensuring the reliability and efficiencyof hardware components is paramount. Among these components, the CPU fan plays a crucial role in maintaining optimal operating temperatures, safeguarding the integrity of thecentral processing unit (CPU), and preventing system failures due to overheating. However, likeany mechanical device, CPU fans are susceptible to faults and failures over time, posing risks of system downtime, performance degradation, and potentially costly repairs. Addressing these challenges necessitates proactive maintenance strategies that can anticipate and mitigate potential faults before they escalate into critical failures. Traditional methods of fan monitoring and maintenance often rely on periodic inspections or reactive responses to evident symptoms of malfunction, leading to inefficiencies and heightened risks of system disruptions. In recent years, the advent of machine learning (ML) techniques has revolutionizedpredictive maintenance practices across various industries. By harnessing the power of data analytics and pattern recognition, ML algorithms can analyze historical performance data, identify subtle precursors to faults, and forecast impending failures with remarkable accuracy. Applied to CPU fan fault prediction, ML offers the promise of early detection, enhancedreliability, and optimized maintenance schedules, thereby minimizing downtime, reducing operational costs, and prolonging the lifespan of computing systems.

This paper explores the application of machine learning in predicting faults in CPU fans, aiming to elucidate its potential benefits, methodologies, challenges, and future directions. Through a comprehensive review of relevant literature, analysis of existing approaches, andempirical insights, this study seeks to contribute to the advancement of predictive maintenance strategies in the realm of computer hardware, fostering greater resilience and efficiency in contemporary computing environments.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

II. LITERATURE REVIEW

The paper titled "TinyML-enabled frugalsmart objects: Challenges and opportunities[11]" by R. Sanchez-Iborra and A. Skarmeta investigates that offers a thorough analysis of the advantages, difficulties, applications, libraries, and models related to Tiny Machine Learning (TinyML). In the case study that concludes the survey, a decision tree model is recommended as the best option for TinyML applications. It is pointed out that the case study's artificial data construction, which mimics decision tree logic, might have contributed to decision trees' betterperformance. Although the survey coversTinyML basics in great detail, it does notgo into great detail about methods for optimizing TinyML models and does notprovide insights into commonly used datasets or hardware for TinyML implementation. This emphasizes the necessity of additional study to address these.

The second paper, "TinyML Meets IoT: A Comprehensive Survey. Internet of Things[14]" by L. Dutta et al. present adetailed analysis of the junction of TinyMachine Learning (TinyML) with the Internet of Things (IoT) in their paper titled"TinyML Meets IoT: A Comprehensive Survey" published in the Internet of Things Journal. The survey investigates the benefits of using TinyML, compares it to other approaches for processing sensordata, and delves into topics such as hardware-software co-design, optimization techniques, libraries and tools, recent advances, and industry participation inTinyML research. While the survey provides samples of datasets and results obtained using TinyML applications, it fallsshort of providing specific insights into regularly used datasets for TinyML projects. This discovery emphasizes the need for more research to understand prevalent datasets relevant to TinyML, particularly in the context of fault.

The third paper, "A review on TinyML:State-of-the-art and prospects[15]" by P. Ray's provides a thorough overview of the state of Tiny Machine Learning (TinyML) today, covering topics such as libraries, hardware, optimizations, and use cases. Thesurvey also presents the authors' opinions on a potential roadmap for TinyML technology advancement. Although the survey covers all of these areas in greatdetail, it doesn't go into the identification ordiscussion of typical datasets used in the TinyML space. This exclusion highlights the necessity for additional research intowidely used datasets, especially in light of their applicability to machine learning algorithms-based failure prediction in fans.

The fourth paper, "A systematic literature review of machine learning methods applied to predictive maintenance" by Thyago P. Carvalhoa, Fabrizzio A. A. M. N. Soares d, Joao P. Bastoc, Symone G. s. Alcalab these review approach involved methodically finding and choosing relevant studies from two major scientific databases. Studies were selected based on predefined criteria for their applicability to predictive maintenance, machine learning approaches,and specifically, fault prediction in CPU fans. This part sets the stage for the literature review. It introduces the broader context of predictive maintenance (PdM) and specifies the focus of the paper, which is fault prediction in CPU fans using machine learning. Here, the author explains how they conducted the literature review. This includes detailing the systematic approach used, such as how they searched for and selected relevant studies based on predefined criteria, and how they analyzed and synthesized the chosen research. This section summarizes the findings from the literature review. It discusses the machine learning techniques commonly used for fault prediction in CPU fans, including supervised and deep learning approaches, and provides insights into the effectiveness of these techniques based on different contexts. The author reflects on the challenges encountered in applying machine learning to predictive maintenance, specifically in the context of CPU fan fault prediction. They also explore potential future research directions, such as hybrid machine learning techniques and leveraging emerging technologies. Finally, the author wraps up the literature survey by summarizing the key findings and their implications. They emphasize the value of the literature review in providing insights for researchers and practitioners aiming to advance predictive maintenance methodologies, particularly particularly in the domain of CPU fan fault prediction.

III. SYSTEM ARCHITECTURE& IMPLEMENTATION

1. EDGE IMPULSE

Fig. 1: Edge impulse

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

Edge Impulse is a platform for developing and deploying machine learning models on edge devices, like microcontrollers and other low- power hardware. It provides tools for collecting, processing, and training data, as well as building and deploying machine learning models optimized for edge computing. Edge Impulse is popular in various industries, including IoT, wearables, and smart devices, where real-time processing and inference are crucial.

Fig. 2: Develop edge ML application with edge impulse

- **Data Collection:** Begin by collecting data relevant to CPU Fan. This could be sensor data from devices like accelerometers, gyroscopes, or microphones. Edge Impulse provides tools to collect and label this data efficiently.
- **Data Preprocessing:** Clean and preprocess the collected data to remove noise, normalize values, and prepare it for training. This step ensures that machine learning model can learn effectively from the data.
- **Model Training:** Use Edge Impulse's platform to train machine learning models on the preprocessed data. We can choose from various algorithms and configurations depending on our project requirements. For CPU Fan fault prediction using Gradient Boosting Machine (GBM) algorithm. Gradient Boosting Machine (GBM) is a powerful machine learning algorithm used for both regression and classification tasks. GBM can identify the most important features contributing to CPU faults, helping prioritize which factors to focus on for prediction and mitigation.
- **Model Optimization:** Once the model is trained, optimize it for deployment on edge devices. This may involve reducing the model size, quantizing parameters, or converting it to a format suitable for deployment on resource-constrained hardware.
- **Deployment:** Deploy the optimized model onto edge devices. Edge Impulse provides tools and libraries to integrate the model into your firmware or software stack seamlessly.
- **Inference:** Use the deployed model to perform real-time inference on the edge device. This allows the device to make predictions or decisions based on new data it receives, without needing to send that data to the cloud for processing.
- **Monitoring and Iteration:** Continuously monitor the performance of deployed model and collect feedback data. Use this data toiteratively improve model's accuracy and efficiency over time.

IV. BLYNK MOBILE APPLICATION

Fig. 3: Blynk mobile application

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

Blynk is a platform for building IoT applications, allowing users to control hardware remotely. Integrating machine learning into Blynk projects could involve tasks like predictive maintenance, anomaly detection, or optimizing energy usage. Implementing machine learningfor fault notification in Blynk projects can significantly enhance efficiency and reduce downtime. By analyzing data collected from sensors or connected devices, ML algorithms can detect patterns indicative of potential faults or anomalies. When a fault or anomaly is detected, the system can trigger notifications through the Blynk app, alerting users or maintenance personnel in real-time. This proactive approach allows for timely intervention, reducing the risk of equipment failure and minimizing costly downtime. Additionally, ML models can continuously learn from new data, improving their accuracy over time and enabling early detection of emerging issues beforethey escalate into critical faults.

V. BLOCK DIAGRAM

- This project demonstrates how to use machine learning to create a multi-class classification model for detecting anomalies in CPU fanvibration data. For this application, we will make use of the Arduino Nano33, one of the many embedded hardware platforms that have been integrated into Machine Learning
- The proposed fault prediction system for a CPU fan combines multiple components, mostly based on a dual-core Arm-based Microcontroller (MCU) with TinyML functionality. The MCU functions as the primary processing unit for gathering, analyzing, and executing data from models. Now let's examine the essential elements:

1. Arm-based Microcontroller (MCU): TinyML algorithms can be executed by the dual-core Arm-based MCU, which is the heart of the system. Efficient multitasking is made possible by its dual-core architecture, which is essential for real-time data processing and model execution.

2. Sensors Interface: A variety of sensors can be connected to the MCU using I2C connection, which makes sensor data collecting easier. The following sensors are essential for the CPU fan's fault prediction:

- Accelerometer: Measures acceleration in order to identify vibration patterns that could point to a problem.
- Gyroscope: This tool helps identify abnormal fan behavior by providing information on rotational motion.
- Temperature Sensor: Monitor temperature variations and provides information about potential overheating risks.

3. Data Acquisition and Processing: Sensor data collected and processed by the Microcontroller Unit (MCU). The MCU organizes the following tasks:

- Data Collection: Raw sensor data is continuously collected and processed.
- Comparison with Training Dataset: The MCU compares incoming data with a pre-trained dataset to identify anomalies.
- Anomaly Detection: It occurs when there are differences between the original and training datasets.

4. Display Output: An OLED display is included in the system to provide visual feedback on detected irregularities. When an anomaly is detected, the display displays a message such as "Abnormality found, service required," instructing operators to take appropriate action.

5. Sound Notification: A buzzer component enhances visual feedback by providing audio alerts. When an abnormality is detected, the buzzersounds a notification, ensuring that the observed defect is addressed immediately.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

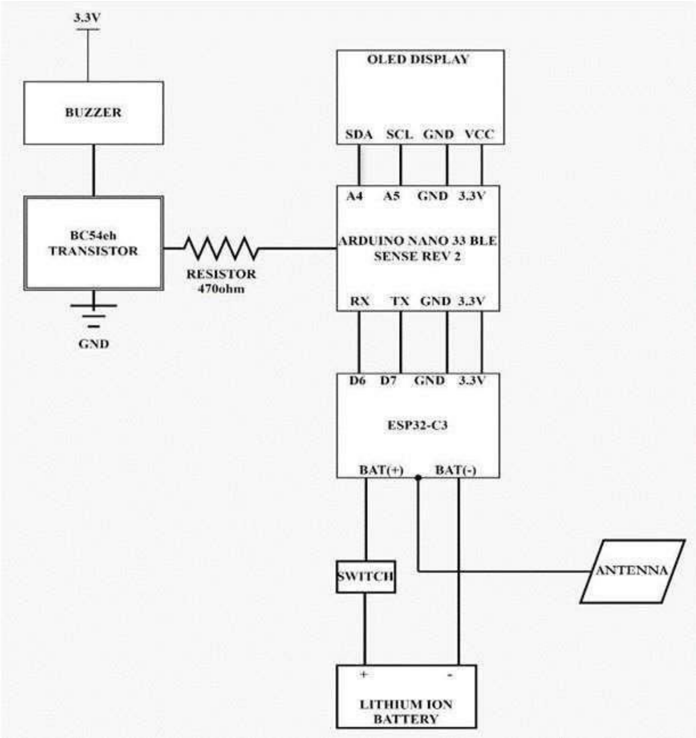


Fig. 4: Block diagram

VI. HARDWARE REQUIREMENTS

1. ARDUINO NANO 33 BLE SENSE REV 2

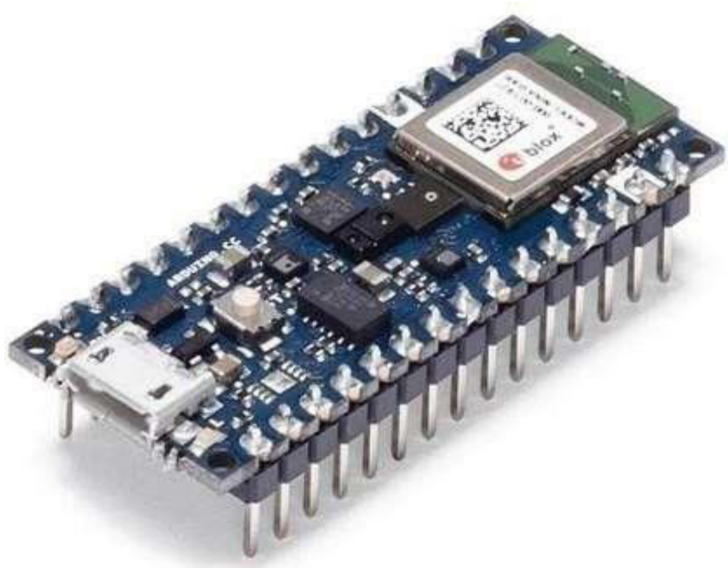


Fig. 5: Arduino Nano 33 BLE Sense Rev 2

The Arduino Nano 33 BLE Sense Rev 2 is a small board with Bluetooth Low Energy connectivity and a variety of sensors, including an accelerometer, gyroscope, pressure, magnetometer, microphone, temperature, humidity, light, and color. It's ideal for IoT projects, wearable devices, and sensor-based applications. It also supports machine learning through its integrated hardware AI capabilities, as well as its compatibility with TensorFlow Lite, which allows you to

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

deploy machine learning models directly onto the board for edge computing applications. It can be powered via USB or an external power supply. It also has a built-in LiPo charging circuit, which allows you to power it with a rechargeable battery.

2. WIFI MODULE

Fig. 6: ESP32-C3 WIFI Module

Espressif Systems produced the ESP32-C3, a low-power Wi-Fi and Bluetooth module. It is based on the RISC-V architecture, which provides a balance of performance and power efficiency. The module is intended for IoT applications, offering connectivity alternatives for a variety of devices. It supports the Wi-Fi 802.11b/g/n and Bluetooth 5.0 LE protocols. Its small size and low power consumption make it appropriate for a wide range of embedded applications. It includes wifi connection, which enables devices to connect to wireless networks. We have protective ICs within the ESP32-C3 that help with battery charging. It can serve as a gateway for deploying machine learning models to edge devices, thereby dispersingcomputational load and enabling more efficient resource utilization.

3. BATTERY

Fig. 7: Lithium-ion battery

Lithium-ion batteries (Li-ion) are rechargeable batteries that are widely utilized in portable electronics and electric vehicles. Li-ion batteries have various advantages over other types of rechargeable batteries, including high energy density, a low self-discharge rate, and no memory effect (they do not need to be entirely depleted before recharging). Li-ion batteries require specific charging algorithms to ensure safe and efficient operation. Overcharging can shorten battery life and even pose safety risks, hence most Li-ion batteries include built-in protective circuits to prevent overcharging and overdischarge. While Li-ion batteries are generally safe when used correctly, they can cause fires or explosions if they are damaged, mistreated, or made incorrectly. Manufacturers incorporate numerous safety features and standards to limit these dangers, such as thermal protection, pressure relief.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

4. OLED DISPLAY

Fig. 8: OLED Display

The acronym OLED stands for Organic Light Emitting Diode. It's a display technique that use organic compounds to emit light when an electric current is applied. OLED displays are distinguished by their brilliant colors, deep blacks, and high contrast ratios. Because of their flexibility and energy economy, they are utilized in a wide range of products, including smartphones, televisions, and wearable devices. OLED displays can be used to continuously monitor the performance of machine learning models in real time, showing metrics like accuracy, loss, and other pertinent factors. OLED displays can be used as intuitive user interfaces for machine learning applications, allowing users to interactwith models, enter data, and receive feedback immediately on the screen.

5. BUZZER

Fig. 9: Buzzer

Electronic buzzers generate audible alerts or indications. They usually consist of a piezoelectric device or an electromechanical component that produces sound when an electric current is applied. Electronic buzzers are commonly employed in a variety of applications, including alarms, timers, industrial equipment, and electronic gadgets, to indicate occurrences or draw user attention.

VII. METHODOLOGY

- **System Overview:** The proposed failure prediction system detects irregularities in fan operation using machine learning (ML) algorithms running on an Arduino Nano33 BLE Sense Rev2 microcontroller unit (MCU). The system consists of sensors, an embedded ML model, and additional parts wrapped in an unique 3D-printed enclosure.

- **Data Collection and Preprocessing:** An accelerometer, gyroscope, temperature, and other important features are among the sensor data that are gathered by the Arduino Nano33 through its integrated sensors. To prepare the data for model training, preprocessing is used, which includes segmentation and labeling.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

- **Model Training:** A multi-class classification model is trained using preprocessed sensor data. The model is intended to distinguish normal and pathological fan operation conditions based on sensor data. Various machine learning techniques, such as decision trees and neural networks, can be investigated and evaluated for maximum performance.

- **Hardware Implementation:** The hardware setup include installing the Arduino Nano33 and other components inside the 3D-printed outer covering. A General Purpose PCB allows for secure attachment of electronic components and ensures electrical isolation to preventinterference. Berg pins are used to provide firm connections between the MCU and the PCB.

- **Integration of Components:** The system also incorporates other parts, such as an ESP32-C3 WiFi module for wireless communication, a buzzer for alarm messages, and a transistor for signal amplification. For mechanical stability, a solar board supports the ESP32-C3 module.

- **Power Management:** A Li-ion battery provides power, which is controlled by protective integrated circuits in the ESP32-C3 module. To guarantee that every system component operates consistently, these ICs control charging and discharging. The ESP32-C3 is equipped with a 2.4GHz antenna to facilitate wireless communication.

- **Display Interface:** The container is equipped with an OLED display that offers instantaneous feedback on various environmental parameters, including temperature, pressure, and humidity. I2C connection connects the display to the MCU, enabling smooth data transfer.

- **Software Integration:** The built ML model is installed on the Arduino Nano33, allowing for real-time anomaly detection based on incoming sensor data. When the system detects deviations from regular fan operation, it sends an alarm message via the buzzer, indicating probable defects or maintenance requirements.

- **Evaluation and Testing:** The fault prediction system's performance is tested using rigorous testing under a variety of operational conditions. Accuracy, precision, and recall are metrics used to evaluate the performance of anomaly detection. Real-world certification offan equipment offers information about the system's reliability and practical utility.

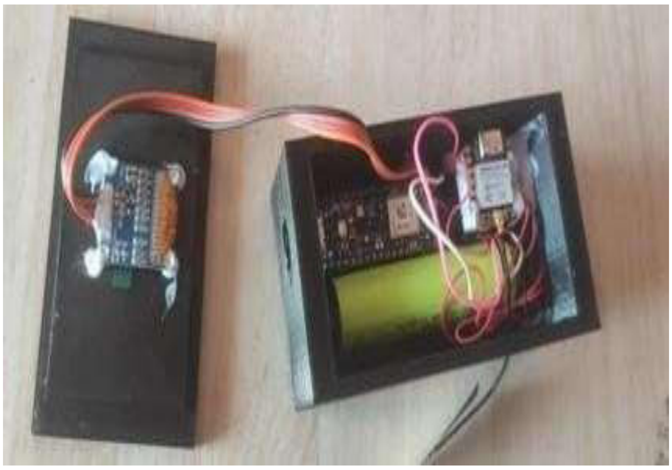


Fig. 10: Circuit Connection

Fig. 11: Abnormality detection

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205284 |

Fig. 12: Final Model

VIII. CONCLUSION

- In this work, we looked into how machine learning techniques can be used to detect problems in CPU fans, with the goal of improvingcomputer system dependability and efficiency. Several major results have arisen froma thorough analysis of existing literature and
- methodology, as well as empirical insights gleaned from our research.

- To begin, our findings highlight the importance of predictive maintenance procedures in reducing the risks associated with CPU fan failures, such as system downtime and performance deterioration. We demonstrated the capability of early defect identification and proactive
- maintenance interventions by using machine learning techniques to historical performance data.

- Furthermore, our analysis demonstrated that several machine learning models, such as decision trees, neural networks, and ensembleapproaches, can reliably predict CPU fan problems. However, it is critical to recognize the impact of data quality, feature selection, and
- model interpretability on the effectiveness of predictive models.

- Furthermore, while our results provide illumination for the use of machine learning for defect prediction in CPU fans, various areas forfuture research remain unexplored. These include the creation of more robust predictive models, the incorporation of real-time sensor data
- for dynamic fault diagnosis, and the investigation of unique feature engineering techniques specific to CPU fan activity.

REFERENCES

[1]. Riera-Guasp, M.; Antonino-Daviu, J.A.; Capolino, G.A. Advances in the electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art. IEEE Trans. Ind. Electron. 2015, 62, 1746–1759.

[2]. Imam, A.M.; Divan, D.M.; Harley, R.G.; Habetler, T.G. Real-time condition monitoring of the electrolytic capacitors for power electronics applications. In Proceedings of the IEEE Applied Power Electronics Conference and Exposition— APEC, Anaheim, CA, USA, 25 February–1 March 2007; doi:10.1109/APEX.2007.357646.

[3]. Sanchez-Iborra, R., Skarmeta, A.F.: Tinyml-enabled frugal smart objects: Challenges and opportunities. IEEE Circuits and Systems Magazine20, 4–18 (7 2020).

[4]. Ray, P.P.: A review on tinyML: State-of-the-art and prospects. Journal of King Saud University - Computer and Information Sciences (2021).

[5]. Fafoutis, X., Marchegiani, L., Elsts, A., Pope, J., Piechocki, R., Craddock, I.: Extending the battery lifetime of wearable sensors with embeddedmachine learning. In: IEEE 4th World Forum on Internet of Things (WF-IoT). pp. 269–274 (2018).

[6]. Dutta, D.L., Bharali, S.: TinyML Meets IoT: A Comprehensive Survey. Internet of Things (Netherlands) 16 (12 2021).