[1]P.V.S.Anusha
[2]P.Swapna
[3]D.V.Rama Koti Reddy

Edge Machine Learning-Enabled Predictive Fault Detection System for Conveyor Belt Maintenance Optimization in Industrial Settings

Journal of Electrical Systems

Abstract: - Digital transformation is essential for industrial and manufacturing sectors, especially in conveyor belt systems. On the other hand, Innovative electronic components and software applications offer opportunities for advanced fault prediction, minimizing operational disruptions, reducing inefficiencies, and enhancing overall industrial efficiency. Efficient conveyor belt systems are crucial for industrial operations, requiring early fault detection to prevent costly downtime and ensure safety. The utilization of embedded machine learning via the Edge Impulse platform enhances data collection and spectral feature extraction using wavelet approximation. This article presents a novel methodology for predictive fault detection in conveyor belt systems employing edge machine learning. Leveraging the ESP32 microcontroller and accelerometer sensor for real-time vibration data capture, a four-layer Artificial Neural Network (ANN) model, trained on 336 feature samples, is deployed via Edge Impulse and TensorFlow scripts. TensorFlow Lite compresses the model for microcontroller integration. The ESP32 microcontroller, acting as an edge device, achieves real-time predictions with 97.5% accuracy and 0.20 minimal loss. This work promises significant strides in predictive maintenance, offering cost savings, operational reliability, and efficiency gains in industrial settings, revolutionizing maintenance strategies.

Keywords: Conveyor Belt Systems Fault Detection, Edge Impulse, Embedded Machine Learning, ESP32.

I. INTRODUCTION

Nowadays, belt conveyors are crucial for bulk material transportation, and are used in various industries. They are evolving to be faster and smarter with help of the modern sensing and AI technologies, which are enabling the self-perception and autonomous operation. Moreover, the intelligent transportation system enhances the safety, and efficiency by focusing on the intelligent mining equipment. The researchers in[1, 2] presented the different methods, which are focus on the field of energy-efficient systems and expert-based fault diagnosis by using noise and vibration, and vision-based monitoring. They also focused on various aspects like deviation, speed, material flow, foreign objects, tears and roller temperature. In addition to that, they specifically address visual monitoring of conveyor belt damage. Furthermore, the design of a heavy-duty coal processing belt conveyor system, which was adhering to Indian standards (IS-11592) with a capacity of 4400TPH and a speed of 4.65m/s is designed. The design calculations and finite element analysis in ANSYS is confirmed that its safety and efficiency under varying load conditions, where it gives for beneficial for designing the conveyor bed [3]. Furthermore, the recent advancements in belt conveyor technology were discussed in [4], where it addressed the need for higher tonnage, longer distances, and complex routes. Also, it explores how traditional components are adapted for non-traditional uses, such as horizontal curves and intermediate drives, which expand the conveyor capabilities and reliability through numerical simulation. Many other authors disclosed the conveyor belt applications in various industries, where it emphasizes the design improvements for reducing failures, maintenance costs and accidents. Additionally, it discussed the importance of efficient design across different sectors, and mentioned about online monitoring for cost reduction and safety in coal transport via belt conveyors [5].

In the other hand, Internet of Things (IoT), machine learning (ML) and deep learning techniques are promising tools for monitoring the conveyor belt health. In this view, the authors in [6] explained the implementation of Industry 4.0 in the belt conveyor transport industry, whereas it discussed about the usage of advanced technologies like the Internet of Things and Internet of Service in order to enhance the conveyor operations and the entire transport system. This concept is not only limited to specific industries, but can be beneficial to various sectors including material transport. The technology’s goal is to make work easier and more efficient by creating fully autonomous conveyors that communicate with other devices and systems. As a result, it ultimately improves the enterprise logistics efficiency. Furthermore, an advanced system for monitoring conveyor belt health using RFID sensors and machine learning is presented in [7, 8], and stated that it accurately detects cracks, details and location, especially in coal mines. In addition to that, the system achieved 100% crack detection and IOT potential, therefore, they concluded that this system can significantly improves the conveyor belt safety in Australian coal mines.

[1,2,3] Department OF Instrument Technology, Andhra University, Visakhapatnam 530003, India
*Corresponding Author Email: anushapvsphd@gmail.com[1], dr.pswapna@andhrauniversity.edu.in[2]

Furthermore, many other authors introduced an intelligent monitoring system for coal conveyor belts, where it effectively addressed the common operational challenges, such as idling, deviation, foreign objects and proximity to personnel. It also addressed the leveraging computer vision, image processing, statistical techniques, and ADCN-based methods, which can offer an advantage of real-time monitoring, and alert through broadcasts. The system ensures the safe and efficient conveyor operation with faster detection speeds and higher accuracy [9, 10]. Similarly, the machine learning methods are presented for detecting faults in belt conveyor idlers, which is essential for system reliability in material transport. It also emphasized the vibration and acoustic techniques that cover the ML models like ANN, KNN, SVM, decision tree, deep learning and CNN. Additionally, they mentioned several challenges such as limited idler datasets and the potential use of drones for automated detection [11]. The earlier study also introduces a lightweight neural network for conveyor belt damage detection, enhancing the detection speed while maintaining accuracy. By combining MobileNet and Yolov4, it improves an accuracy boost of 3.5% and a speed increase of 188% while compared to Yolov4 alone. Moreover, it explored the various lightweight models by adjusting channel numbers, providing insights for high-speed conveyor monitoring. As a result, it provides a way to efficient conveyor damage detection, demonstrating the potential of lightweight neural networks in the field.

Moreover, many other researchers have been used the IoT and Machine Learning to monitor the health of conveyor belts, where Machine Learning is more often executed in the cloud. However, this approach may result in latency and privacy concerns. Therefore, a promising alternative is embedded to the machine learning or Edge AI, which integrates the machine learning capabilities directly into embedded systems situated at the network edge rather than relying on centralized cloud processing. This approach offers several advantages including reduced latency, enhanced privacy and decreased data transmission costs. In this view, the authors in [12] presents the edge impulse as a cloud-based Machine Learning Operations (MLOps) for developing the Tiny Machine Learning (TinyML) systems. It simplifies the TinyML design process by addressing the challenges related to fragmented software and diverse hardware, making it easier to optimize and deploy ML models across a wide range of embedded systems. Since October 2022, Edge Impulse has hosted over 118,000 projects by nearly 51,000 developers. It aims to democratize AI by reducing the expertise and resources required for building machine learning systems on resource-constrained devices. This framework has found the applications in industry, research and education, offering valuable insights for future research in this field. Additionally, Edge Impulse uses TinyML to create and deploy the machine learning models on small devices like microcontrollers, whereas TinyML optimizes these models to work well on devices with limited resources. Edge Impulse simplifies the process of building and running these models on edge devices. Further, a Vehicle Logo Recognition (VMR) system was introduced based on Convolutional Neural Networks (CNN) that doesn't require precise logo detection. It uses an efficient pre training strategy, ensuring robustness to various imaging conditions. With an accuracy of 99.07% on a dataset of 11,500 logo images, this CNN method proves effective in real-world applications. Its structure supports parallel implementation in GPUs, enabling real-time recognition. This research showcases Edge Impulse's efficiency in accurate and practical vehicle logo recognition systems [13].

In the same way, Edge Artificial Intelligence (EDGE-AI) is explored for real-time recognition of DC motor operational states based on sound data. It involves the training CNN models and deploys them to the microcontroller units (MCUs) for efficient on-device processing. It demonstrates a high accuracy and a low latency in classifying the motor states, and making it valuable for predictive maintenance [14]. The authors in [15] discussed about the Edge Artificial Intelligence (Edge AI) in monitoring and reporting water usage to combat waste. The method in [15] combines the IoT, Edge Computing and Machine Learning to create an efficient detection system. The approach minimizes energy use and enhances privacy by reducing data transmission to the cloud. Another study utilized the TinyML and deep learning to create an IoT system for indoor asset tracking [15]. Traditional positioning systems like GPS fall short for indoor use, so a system uses RF signal data and a neural network is introduced to classify the location of objects in an indoor environment. Edge Impulse was also employed for data processing, training, evaluation and deployment. The system achieved a classification accuracy of 88%, which can be improved to 94% with post-processing. Additionally, it encompasses the hardware and firmware development, system integration and cyber security measures, offering a flexible solution for indoor asset tracking and continued model improvement [16]. In another study, a TinyML-based gas leakage detection system was introduced, and it can identify and alert users about dangerous gas leaks. The system can be configured to detect various gases, including LPG, smoke, alcohol, propane, hydrogen, methane, carbon monoxide, ammonia, sulphur, benzene and carbon dioxide. The experiments focused on detecting ammonia and smoke, achieving good results [17].

Furthermore, another study was proposed by the authors in [18] using TinyML-compatible sensors to estimate the shelf life of fresh date fruits, which is crucial for producers and suppliers. During storage, they observed that the attributes of fruit by employing the low-cost spectral sensors along with regression models for estimation. Vacuum and modified atmosphere packaging are extended the shelf life when it stored at 5°C.

However, it reduced at room temperature. The TinyML sensors can provide real-time predictions of shelf life by combining with the edge computing, benefiting various stakeholders in the supply chain. This low-cost approach ensures that the fresh food availability year-round and has been experimentally validated for its accuracy was exceeding of 93% [18]. Similarly, the authors in [19] presented the TinyML approach to monitor real-time running states of rail vehicles. Using MEMS sensors and neural networks on IoT edge devices, it achieves over 99% accuracy in identifying complex states on subways. Its cost-effectiveness and adaptability extend beyond rail to cars, ships and aircraft, which offer the promising applications in real-time monitoring systems [19]. Moreover, an ultralow-power IoT device was exhibited for monitoring construction tool usage. Using Bluetooth Low Energy (BLE) and Near-Field Communication (NFC), it tracks the activities, and detects the misuse while monitoring at storage conditions. The author in [20] presents an ultralow-power IoT device designed for monitoring construction tool usage. Employing BLE and NFC, the device tracks activities, detects potential misuse, and monitors storage conditions. Integrated with TinyML, the device achieves a high classification accuracy of 90.6% across four tool usage categories, ensuring prolonged battery life. This research showcases the feasibility of edge-based fine monitoring, paving the way for more sophisticated utility analysis. Subsequent efforts could explore the adaptability of TinyML to diverse tool positions, contributing to a more comprehensive analytical framework. Therefore, being inspired by the benefits of embedded machine learning, this paper presents conveyor belt prediction system based on TinyML for real-time conveyor belt fault using Edge Impulse. The proposed system analyses vibration sensory data at the edge using TinyML technology. The outline of this article is presented as follows: chapter 2- describes the methodology, whereas chapter 3 discusses about the results of the proposed method. Finally, the conclusions of this work mentioned in chapter 4.

II. METHODOLOGY

The vibration data acquisition and processing were conducted on the experimental setup depicted in Fig.1. This setup comprises essential components including a computer, a belt conveyor system, a LIS2DW12 accelerometer sensor, and an ESP32 microcontroller. ESP32 is a versatile and cost-effective platform designed by Espressif Systems, offering a powerful set of features for IoT applications.

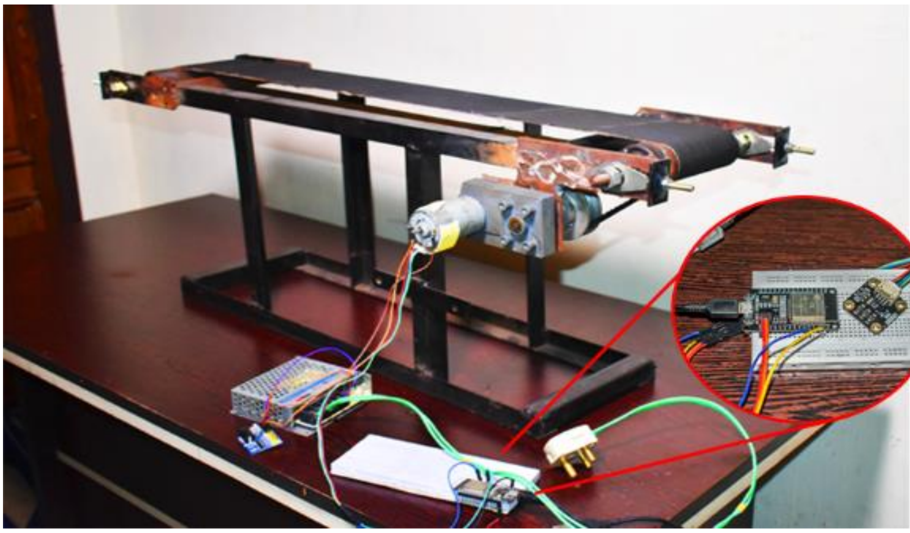

Figure 1. Experimental setup for the proposed work.

It boasts a dual-core processor for multitasking, integrated Wi-Fi and Bluetooth for seamless connectivity, a plethora of GPIO pins for device interfacing, and a focus on low power consumption for energy-efficient operations. The ESP32 supports various programming languages, with C++ being the most common, and can be programmed using popular development frameworks like Arduino IDE and ESP-IDF. This microcontroller is widely used across IoT projects due to its adaptability and robustness. It comes with essential features such as a dual-core ESP32 processor, 4 MB of flash memory, 8 MB of PSRAM, and built-in Wi-Fi, Bluetooth, and USB to serial converter capabilities. The Arduino IDE serves as an excellent development environment for programming ESP32 boards, providing a user-friendly interface, library and board management tools, a serial monitor for debugging, and the ability to compile and upload code efficiently. This combination of ESP32 microcontrollers and Arduino IDE simplifies the development of IoT solutions and benefits from a thriving open-source community offering extensive documentation and support [21].

This author in [22] examined the performance of the Xtensa LX6 microprocessor, housed in devices like the Espressif ESP32 and ESP32 Cam, renowned for their user-friendly IoT capabilities. Our goal is to assess the processor's speed when running neural network applications with different input sizes and varying neural network architectures. This exploration delves into the potential of tiny Machine Learning (tinyML) on these microcontrollers, contributing to fields like mobile robotics, computer science, and electrical engineering. This research utilized an ESP32 chip to forecast wind speed, employing TinyML-based models like Artificial Neural Network (ANN) and Linear Regression. The system predicted maximum wind speed to manage sudden gust-related risks. By analysing temperature and humidity data from a DHT22 sensor, displayed on LCD screens, the ANN model outperforms Linear Regression, achieving 75.866% accuracy compared to 71.029%. This showcases TinyML's potential in real-time wind speed forecasting [23]. This research article serves as a valuable guide for those looking to harness the potential of ESP32 microcontrollers and Edge Impulse platform for real-world edge computing applications.

Our conveyor belt system, fundamental to this research, was meticulously designed with precise specifications at its core using the components shown in the table 1. Critical factors such as motor RPM, power requirements, bearing selection, and belt mechanisms were scrupulously considered.

The focal point of the setup is the motor, carefully chosen for its 8 RPM rating, ensuring precise and synchronized conveyor movement. This selection aligns seamlessly with the project's demand for accuracy and consistency. The motor's power characteristics were meticulously matched to the conveyor's load requirements to guarantee efficient operation. Furthermore, our bearing selection process involved the meticulous choice of 6201 bearings, all identical in size. This selection significantly enhances the system's reliability and longevity while minimizing friction and optimizing overall system efficiency.

Table 1. Components of conveyor belt monitoring system

No.	**Component**
1	ESP32 Development Board
2	Micro USB Cable
3	LIS2DW12 triple axis accelerometer
4	12V 8RPM DC motor with rectangular Gear Box
5	Power Supply: 220V AC to 12V DC converter
6	6201ZZ bearings
7	Pulley with diameter 6.2 cm
8	78 cm*18 cm * 31 cm (l*b*h) Conveyor stand
9	Conveyor Belt

Central to the functionality of our conveyor is the integration of a high-efficiency belt mechanism. This component not only facilitates smooth motion transfer but also amplifies the conveyor's overall operational effectiveness. Its design allows for the seamless transmission of power from the motor to the conveyor belt, perfectly in line with the system's core objective of efficiency.

One study explores Ultra-Wide Band (UWB) technology in industrial settings, focusing on machine vibration monitoring using the LIS2DH12 motion sensor. It indicates potential for real-time anomaly detection in machinery lacking integrated sensors, allowing timely notifications. Further experiments are required to fully understand UWB's capabilities and its impact in industrial environments. This adaptable UWB technology, when combined with appropriate sensor-equipped UWB modules, could be beneficial in various high-frequency monitoring scenarios [24]. Another study introduces a wireless monitoring system for geotechnical restraint systems. It utilizes Wireless Accelerometer Nodes and Load Anchor Cell Sensors powered by the LIS2DW12 motion sensor, enabling remote monitoring, hazard alerts, and predictive maintenance. Real-world case studies highlight its practical effectiveness. The LIS2DW12 sensor ensures accurate three-axis data capture essential for wireless transmission via the TR-76 IQRFR transceiver [25]. In our paper, we are embracing this advanced sensor interfacing. A LIS2DW12 triple-axis accelerometer sensor, strategically placed near the motor and bearing arrangement, is employed to capture precise vibration data.

This data acquisition complements the conveyor's fine-tuned control and monitoring capabilities. The captured data undergoes adept processing, both offline and in real-time, leveraging Python and Edge Impulse.

In Fig 2, the structural model of the conveyor system is delineated, comprising four primary components: the roller shaft, bolt, bearing chock, and adjusting bracket. The adjusting bracket is securely fastened to the conveyor frame, while the roller shaft is mounted within the bearing chock. A bolt, connected to the bearing chock via a nut, permits the movement of the bearing chock within the adjusting bracket. This setup enables controlled adjustments to the roller shaft's position, ensuring precision in the conveyor's operation.

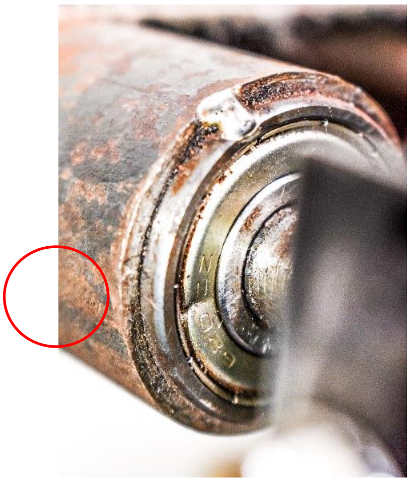

Figure 2. Fault created in bearing

Additionally, the research incorporates code-based data acquisition via an Arduino program running on an ESP32 microcontroller. This code orchestrates the data collection process from a LIS2DW12 accelerometer sensor, which is interfaced to ESP32 using the I2C communication protocol. The ESP32 interacts seamlessly with the sensor, capturing acceleration data. Within the loop function of Arduino program, the ESP32 continuously retrieves acceleration data from the sensor's X, Y, and Z axes. Raw data is diligently extracted and subsequently converted into acceleration values, accommodating a range of +/- 2g. These processed values are then sent to the serial monitor. This data which is send to the serial monitor can be loaded to data acquisition tab of Edge impulse cloud using edge Impulse CLI. The edge Impulse cloud enables real-time monitoring and offers opportunities for further analysis.

This multifaceted data acquisition methodology serves as a foundational element for vibration analysis of our research paper. The synergy of precision mechanical specifications, advanced sensor technology, and meticulous programming underscores our commitment to excellence across all facets of this research. In the initial phase of our experimentation, we commenced by conducting trials with well-functioning bearings. Subsequently, artificial faults were deliberately induced through the utilization of precision machining tools, resulting in the creation of dents and cracks on the bearing surfaces, as visually depicted in Fig.2. Following the creation of these artificial faults, the faulty bearings were then meticulously reinstalled within the test-bed configuration. Vibration data, crucial for our analysis, were meticulously recorded under identical motor loads. The resulting dataset encompasses vibration data collected from both the faulty bearings and the pristine, non-faulty bearings. Throughout this study, variable-length vibration acceleration signals were meticulously recorded at a sampling rate of 338 samples per second (Hz) and we have 40 seconds data in each sample, ensuring a comprehensive and high-resolution dataset for our analysis and subsequent investigations.

III. RESULTS AND DISCUSSION

From sensor node, we collected vibration sensor data in the edge impulse. The fig.3 shows the procedure from data acquisition to Model deployment at the edge. The Edge Impulse data acquisition tab was employed for collecting data in our experiment. The data acquisition tab was configured with parameters such as labels for samples and sample lengths in milliseconds. Edge Impulse automatically detected the sample frequency.

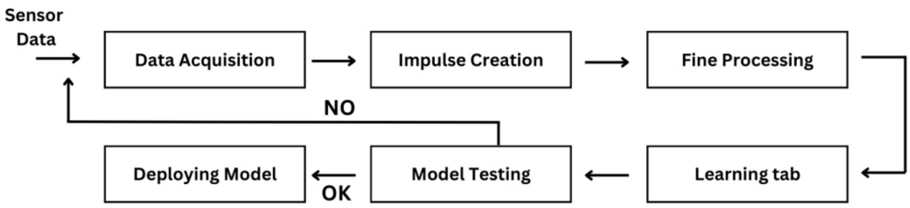

Figure 3. Work flow of architecture

Following the data collection process in this tab, we visualized a plot representing the measured samples. As shown in fig.4, the plot of the collected samples displayed three-line plots, each representing accelerometer values along the x, y, and z axes, respectively. Moving on to the Impulse Design tab, we constructed a complete Impulse with three primary blocks: Input, Processing, and Learning. In the Input block, parameters like window size, window increase, and frequency were specified. In the processing block, we defined parameter types and selections. For model construction, we opted for the Edge Impulse classification in the Learning block, which employed an ANN based algorithm. The final block displayed two output features: 'With_fault' and 'Without_fault.'

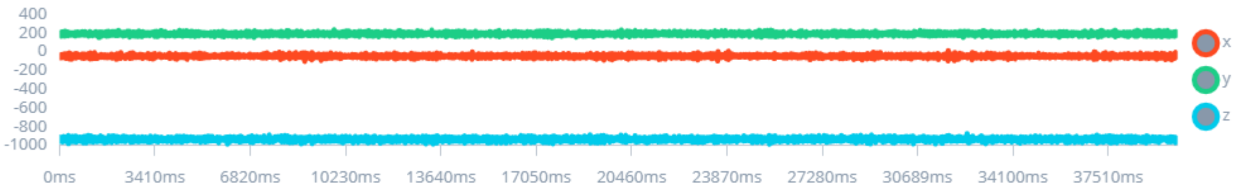

Figure 4: Sample recording in edge impulse data acquisition tab

Fig.5 showcases the results from the processing tab. We applied wavelet transformation for preprocessing, and detailed parameters for the wavelet transformation were observable. In the same processing tab, we could navigate to the generate features tab to generate features and assess clusters in the plot to confirm the presence of useful data for our model.

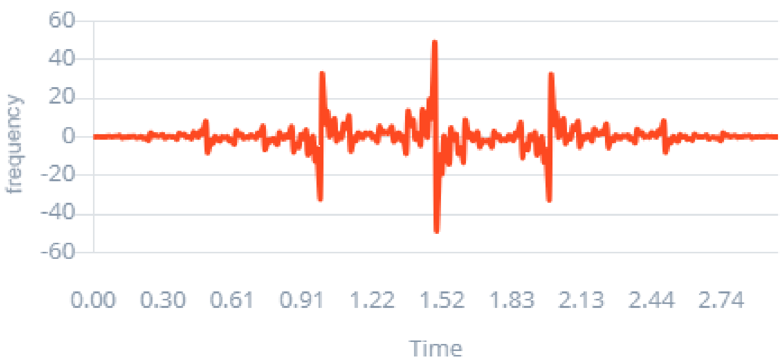

Figure 5. Wavelet Transformation

Developed a TensorFlow script tailored for training an ANN using dense layers for classification. The script began by importing essential libraries, notably TensorFlow and its submodules, to facilitate the construction and training of neural networks. It then proceeded to define critical hyper parameters such as epochs, learning rate, and batch size, with flexibility in parameter configuration. Data batching was implemented to enhance training efficiency. The neural network model was defined using the Sequential API, comprising three dense layers with specific configurations. To optimize model performance, an Adam optimizer was employed, categorical crossentropy as the loss function and a custom callback was included for monitoring training progress. The Adam optimization algorithm extends the principles of stochastic gradient descent by incorporating adaptive learning rates and momentum, enhancing its efficiency in optimization tasks .Here's the mathematical formulation

$$\theta_{new} - \theta_{old} - \frac{Learning\ rate\ \times m}{\sqrt{v}{+}\in}$$

In each iteration, the algorithm updates parameters (θ_{new}) based on the current parameter values (θ_{old}), the gradient (g), and the moving averages (m and v). The learning_rate hyperparameter plays a crucial role, determining the step size during each iteration and influencing convergence and stability. The first-moment estimate (m) represents the exponentially decaying average of past gradients, while the second-moment estimate (v) captures the exponentially decaying average of past squared gradients. To prevent division by zero and improve numerical stability, a small constant (∈ or epsilon) is added to the denominator. The update rule involves the square root of the sum of the second moment estimate and epsilon ($\sqrt{v}{+}\in$), contributing to the denominator of the parameter update formula. This comprehensive approach enhances the adaptability and performance of the optimization process.

Categorical Cross entropy is commonly used for multi-class classification problems. For a single training example, if ytrue represents the true class distribution and ypred represents the predicted distribution, the loss is calculated as follows:

$$\text{Loss} = -\sum_{i} y_{true,i} \cdot \log(y_{pred,i})$$

In the context of a single training example, the loss calculation involves comparing the true class distribution (represented by ytrue) with the predicted distribution (represented by ypred). The computation, conducted for each class iteration (i), evaluates the discrepancy between the true probability of class i (y_(true,i)) and the corresponding predicted probability (y_(pred,i)). This iterative process ensures a comprehensive assessment of the model's performance across all classes, allowing for an effective measure of the classification loss.

Subsequently explored the learning interface designed to facilitate algorithm selection. Within this interface, a straightforward ANN architecture with four layers was configured. Specifically, this architecture featured an initial input layer with 336 features, corresponding to the number of samples used for model training. The network culminated in an output layer with two classes. Furthermore, incorporated two dense layers, each consisting of 20 and 10 neurons, respectively. In our experimental setup, we conducted training over 30 epochs, employing a learning rate of 0.0005. The fig.6 shows network architecture with the layers used in the model and beside it the results of training accuracy. It shows very good results with 97.5% accuracy proves that the ANN algorithm was effectively implemented as part of our research methodology.

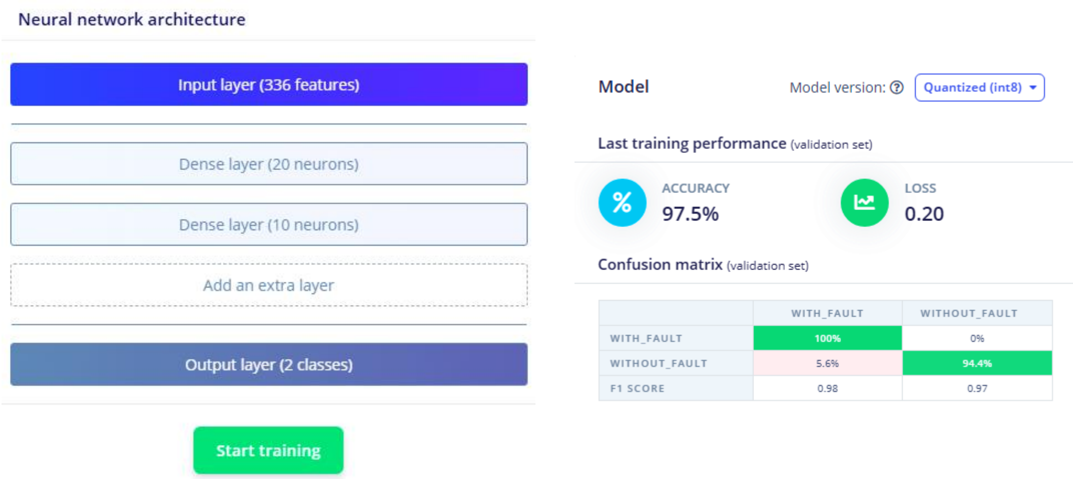

Figure 6. Network architecture and training performance

The below plot is the plot of the data points classification status in your training data here all the points that are in green colour are classified correctly and all that are in red are the misclassified values. here we can observe very few are misclassified samples out of all our points.

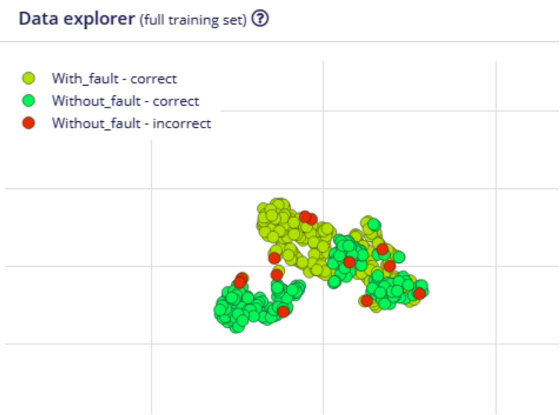

Figure 7. Data Visualisation of classification

The model was compiled using categorical cross-entropy loss and trained using the 'fit' method, with training and validation datasets. Additionally, optional per-channel quantization was explored to reduce memory requirements. Fig.8 illustrates on-device model performance, showcasing inference time, peak RAM, and flash usage. The data highlights the model's exceptional results, affirming its efficacy in practical scenarios.

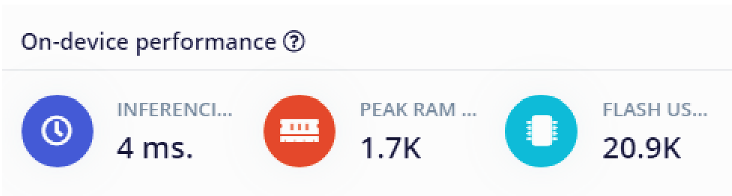

Figure 8. On device performance of the model

For assessing model accuracy with unknown data, the model testing tab was used. In the Model Testing tab, model testing results both the confusion matrix and testing outcomes for unknown data is presented. Furthermore, live classification performed through the Live Classification tab, enabling real-time sample analysis and result prediction.

Model testing results

ACCURACY
97.65%

	WITH_FAULT	WITHOUT_FAULT	UNCERTAIN
WITH_FAULT	97.5%	0%	2.5%
WITHOUT_FAULT	0%	97.8%	2.2%
F1 SCORE	0.99	0.99	

Figure 9. Testing accuracy performance

In the preceding Fig. 6, it is evident that our resulting model achieved a loss of 0.20 percent, Model Training Accuracy of 97.5%, and a validation accuracy of 97.65%. Fig. 9 provides the validation set's confusion matrix. The confusion matrix, presented in the figure, offers a comprehensive view of the model's performance. In the top-left corner, you'll find the percentage of values that truly belong to the "With fault" class and were 97.5% correctly predicted as "With fault". Adjacent to it, in the top-right cell, the value '0' signifies instances where the true class was "With fault," but the model predicted them as "Without faults". In bottom you'll notice '0', which represents cases where the actual class was "With fault," but the model erroneously predicted "Without fault." Lastly, in the last cell of the matrix, we have 97.8%, indicating instances where the actual class was "Without fault," and model classified them as "Without fault."

Right most column shows the percentage of samples predicted as uncertain is clearly visible. The bottom row of the matrix displays the F1 Score for both classes, reflecting the individual accuracies of each class. This comprehensive analysis allows us to assess the model's performance in distinguishing between "With fault" and "Without fault" instances, providing valuable insights into its predictive capabilities. Upon successful model building and testing, we proceeded to deploy it onto our microcontroller using the Deployment tab. Within this tab, carefully selected the coding platform for programming the microcontroller and initiated the model conversion process with a simple click on 'Build.' After completing the construction, the automatic download of Arduino libraries was observed, allowing for easy loading of our model into the microcontroller. This deployment process, supported by Edge Impulse, made it smooth for us to integrate and deploy our model onto the microcontroller.

IV. CONCLUSION

In conclusion, this research endeavor has harnessed the rapid advancements in electronics and software, underscoring the practicality of a methodology aimed at delivering an innovative, compact, and cost-effective system capable of predicting faults in typical conveyor belt systems. Our approach leverages recently emerged, cost-efficient microcontrollers like ESP32, standard sensors, Arduino IDE and appropriate software tools to introduce a contemporary Edge AI technique, exemplified by Edge Impulse, in the form of on-device intelligence—an inherently efficient strategy. Within our deployment framework, we have incorporated mechanisms for simulating conveyor belt system faults, thereby facilitating the training and validation of neural network models tailored to this domain. Notably, we successfully trained a neural network model using vibration sensor data, meticulously optimized for minimal in-situ computing resources. Various model variants were evaluated, with the most effective iteration utilizing accelerometric data, achieving an impressive fault diagnosis accuracy rate of 97.65%.

Importantly, our system operates in a decentralized manner, without the need for high-bandwidth data transmission to the cloud or centralized processing. This result in a simplified infrastructure, reduced energy consumption, and heightened privacy. Moreover, the installation of IoT equipment, such as Arduino and ESP32, is non-invasive and straightforward. This article not only contributes to the current body of knowledge but also paves the way for future research. It offers insights into the strengths and limitations of different implementation alternatives for incorporating on-device intelligence into locally installed conveyor belt systems. Future work will focus on optimizing various aspects of the proposed system, including hardware selection, neural network model accuracy, setup standardization, networking solutions, power autonomy, and user-friendliness. Enhancing the commercial standards in this domain will remain a paramount priority, fostering innovation and advancing the field of predictive maintenance for conveyor belt systems.

ACKNOWLEDGMENT

There is no specific funding to support this research.

REFERENCES

[1] M. Zhang *et al.*, “Application of Lightweight Convolutional Neural Network for Damage Detection of Conveyor Belt,” *Applied Sciences*, vol. 11, no. 16, p. 7282, Aug. 2021, doi: 10.3390/app11167282.

[2] M. Zhang, H. Shi, Y. Zhang, Y. Yu, and M. Zhou, “Deep learning-based damage detection of mining conveyor belt,” *Measurement*, vol. 175, p. 109130, Apr. 2021, doi: 10.1016/j.measurement.2021.109130.

[3] Ms. Sayali Todkar and Prof. Milind Ramgir, “Design of Belt Conveyor System,” *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 7, no. 7, 2018.

[4] M. A. Alspaugh, “Latest Developments in Belt Conveyor Technology,” 2004.

[5] G. Singh, M. Singh, and A. Kumar, “APPLICATION OF CONVEYOR BELT: A REVIEW PAPER.” [Online]. Available: https://pramanaresearch.org/

[6] G. Fedorko, "Implementation of Industry 4.0 in the belt conveyor transport," *MATEC Web of Conferences*, vol. 263, p. 01001, 2019, doi: 10.1051/matecconf/201926301001.

[7] S. Dey, O. Salim, H. Masoumi, and N. Karmakar, "Health monitoring of mining conveyor belts." [Online]. Available: https://ro.uow.edu.au/coalhttps://ro.uow.edu.au/coal/788

[8] F. T. Zohra, O. Salim, H. Masoumi, N. C. Karmakar, and S. Dey, "Health Monitoring of Conveyor Belt Using UHF RFID and Multi-Class Neural Networks," *Electronics (Switzerland)*, vol. 11, no. 22, Nov. 2022, doi: 10.3390/electronics11223737.

[9] Z. Li, X. Zhu, and J. Zhou, "Intelligent Monitoring System of Coal Conveyor Belt Based on Computer Vision Technology," in Proceedings - 2019 6th International Conference on Dependable Systems and Their Applications, DSA 2019, Institute of Electrical and Electronics Engineers Inc., Jan. 2020, pp. 359–364. doi: 10.1109/DSA.2019.00055.

[10] D. Qu, T. Qiao, Y. Pang, Y. Yang, and H. Zhang, "Research on ADCN Method for Damage Detection of Mining Conveyor Belt," IEEE Sens J, vol. 21, no. 6, pp. 8662–8669, Mar. 2021, doi: 10.1109/JSEN.2020.3048057.

[11] F. Alharbi *et al.*, "A Brief Review of Acoustic and Vibration Signal-Based Fault Detection for Belt Conveyor Idlers Using Machine Learning Models," *Sensors*, vol. 23, no. 4. MDPI, Feb. 01, 2023. doi: 10.3390/s23041902.

[12] S. Hymel *et al.*, "Edge Impulse: An MLOps Platform for Tiny Machine Learning," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2212.03332

[13] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, Aug. 2015, doi: 10.1109/TITS.2014.2387069.

[14] K. Strantzalis, F. Gioulekas, P. Katsaros, and A. Symeonidis, "Operational State Recognition of a DC Motor Using Edge Artificial Intelligence," *Sensors*, vol. 22, no. 24, Dec. 2022, doi: 10.3390/s22249658.

[15] D. Loukatos, K. A. Lygkoura, C. Maraveas, and K. G. Arvanitis, "Enriching IoT Modules with Edge AI Functionality to Detect Water Misuse Events in a Decentralized Manner," *Sensors*, vol. 22, no. 13, Jul. 2022, doi: 10.3390/s22134874.

[16] D. Avellaneda, D. Mendez, and G. Fortino, "A TinyML Deep Learning Approach for Indoor Tracking of Assets †," *Sensors*, vol. 23, no. 3, Feb. 2023, doi: 10.3390/s23031542.

[17] V. Tsoukas, A. Gkogkidis, E. Boumpa, S. Papafotikas, and A. Kakarountas, "A Gas Leakage Detection Device Based on the Technology of TinyML †," *Technologies (Basel)*, vol. 11, no. 2, Apr. 2023, doi: 10.3390/technologies11020045.

[18] R. Srinivasagan, M. Mohammed, and A. Alzahrani, "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits," *Sensors*, vol. 23, no. 16, Aug. 2023, doi: 10.3390/s23167081.

[19] S. Zhou, Y. Du, B. Chen, Y. Li, and X. Luan, "An Intelligent IoT Sensing System for Rail Vehicle Running States Based on TinyML," *IEEE Access*, vol. 10, pp. 98860–98871, 2022, doi: 10.1109/ACCESS.2022.3206954.

[20] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, and M. Magno, "Design and Performance Evaluation of an Ultralow-Power Smart IoT Device With Embedded TinyML for Asset Activity Monitoring," *IEEE Trans Instrum Meas*, vol. 71, 2022, doi: 10.1109/TIM.2022.3165816.

[21] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156739.

[22] M. Ziaul and H. Zim, "XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE TinyML: Analysis of Xtensa LX6 microprocessor for Neural Network Applications by ESP32 SoC."

[23] C. Kiang Hong, M. Azlan Abu, M. Ibrahim Shapiai, M. Fadzli Haniff, R. Sham Mohamad, and A. Abu, "Analysis of Wind Speed Prediction using Artificial Neural Network and Multiple Linear Regression Model

using Tinyml on Esp32," *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences Journal homepage*, vol. 107, pp. 29–44, 2023, doi: 10.37934/araset.107.1.2944.

[24] A. Bonci, E. Caizer, M. C. Giannini, F. Giuggioloni, and M. Prist, "Ultra Wide Band communication for condition-based monitoring, a bridge between edge and cloud computing," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 1670–1677. doi: 10.1016/j.procs.2022.12.367.

[25] M. Pieš, R. Hájovský, and J. Velička, "Wireless measuring system for monitoring the condition of devices designed to protect line structures," *Sensors (Switzerland)*, vol. 20, no. 9, May 2020, doi: 10.3390/s20092512.