Check for updates

An Intelligent Real-Time Edge Processing Maintenance System for Industrial Manufacturing, Control, and Diagnostic

Ovidiu Vermesan[1]*, Marcello Coppola[2], Roy Bahr[1], Ronnie Otto Bellmann[3], Jøran Edell Martinsen[3], Anders Kristoffersen[3], Torgeir Hjertaker[3], John Breiland[4], Karl Andersen[4], Hans Erik Sand[4] and David Lindberg[5]

[1]SINTEF Digital AS, Oslo, Norway, [2]STMicroelectronics, Grenoble, France, [3]Denofa AS, Fredrikstad, Norway, [4]NxTech AS, Fredrikstad, Norway, [5]Intellectual Labs AS, Oslo, Norway

This paper presents an artificial intelligence (AI) based edge processing real-time maintenance system for the purposes of industrial manufacturing control and diagnostics. The system is evaluated in a soybean processing manufacturing facility to identify abnormalities and possible breakdown situations, prevent damage, reduce maintenance costs, and increase production productivity. The system can be used in any other manufacturing or chemical processing facility that make use of motors rotating equipment in different process phases. The system combines condition monitoring, fault detection, and diagnosis using machine learning (ML) and deep learning (DL) algorithms. These algorithms are used with data resulting from the continuous monitoring of relevant production equipment and motor parameters, such as temperature, vibration, sound/noise, and current/voltage. The condition monitoring integrates intelligent Industrial Internet of Things (IIoT) devices with multiple sensors combined with AI-based techniques and edge processing. This is done to identify the parameter modifications and distinctive patterns that occur before a failure and predict forthcoming failure modes before they arise. The data from production equipment/motors is collected wirelessly using different communication protocols - such as Bluetooth low energy (BLE), Long range wide area network (LoRaWAN), and Wi-Fi - and aggregated into an edge computing processing unit via several gateways. The AI-based algorithms are embedded in the processing unit at the edge, allowing the prediction and intelligent control of the production equipment/motor parameters. IIoT devices for environmental sensing, vibration, temperature monitoring, and sound/ultrasound detection are used with embedded signal processing that runs on an ARM Cortex-M4 microcontroller. These devices are connected through either wired or wireless protocols. The system described addresses the components necessary for implementing the predictive maintenance (PdM) strategy in soybean industrial processing manufacturing environments. Additionally, it includes new elements that broaden the possibilities for prescriptive maintenance (PsM) developments to be made. The type of ML or DL techniques and algorithms used in maintenance modeling is dictated by the application and available data. The approach presented combines multiple data sources that improve the accuracy of condition monitoring and prediction. DL methods further increase the accuracy and

require interpretable and efficient methods as well as the availability of significant amounts of (labeled) data.

1 INTRODUCTION

Soybean is a nutrient-dense legume containing the essential amino acid profile and providing high protein content that makes soybean used as a food and feed ingredient. Typical soybean parameters are protein (46–48%), fat (max. 1%), moisture (max. 13%), sand/silica (max. 1%), crude fiber (max. 6%), etc. (Van Eys, 2012). Soybean seed chemical composition makes it possible through specific chemical processes to separate from soybean production three main products, crude vegetable oil, lecithin, and defatted meal as illustrated in **Figure 1**.

The soybean chemical processing involves equipment/motors used in different production phases. One of the most critical equipment to monitor is the motor running the hammer mill located in the crushing area of the production line. Two parallel hammer mills are used for crushing the lumps in the soybean meal. When hammers and shafts are worn, the vibration and sound parameters are changing that can indicate an imminent accident. Measuring the parameters of motors/equipment (e.g., vibration, temperature, sound, current/voltage, etc.) in soybean production lines requires monitoring and acquisition devices as data sources for analysis and maintenance.

Today, two essential maintenance strategies are applied in the food manufacturing processes. Corrective maintenance considers all maintenance actions that are allocated and completed after a failure (e.g., reacting to a system failure and correcting it). Proactive maintenance addresses the routine functionality checks to identify upcoming faults and predict failures before their occurrence.

Predictive and prescriptive maintenance approaches are part of the proactive maintenance strategies that take advantage of the latest technological developments in edge computing, AI, and IIoT. PdM (Hashemian, 2011) focuses on extending degradation mechanism knowledge and expanding the degradation propagation into the future to forecast system failures. This approach utilizes the knowledge and pattern discovery methodology. It also combines insights from degradation in past events with foreseen operating conditions in the future, and this supports a maintenance decision-making process. PsM applies degradation prediction information and extends the scope of the maintenance decision-making process beyond the motor/equipment itself.

One data source for maintenance operations is represented by IIoT devices that monitors the vibration of the motors and equipment. The vibration analysis techniques include the vibration measurement and its interpretation. Vibration signals are measured using 3-axis microelectromechanical systems (MEMS) based accelerometer sensors in the time domain.
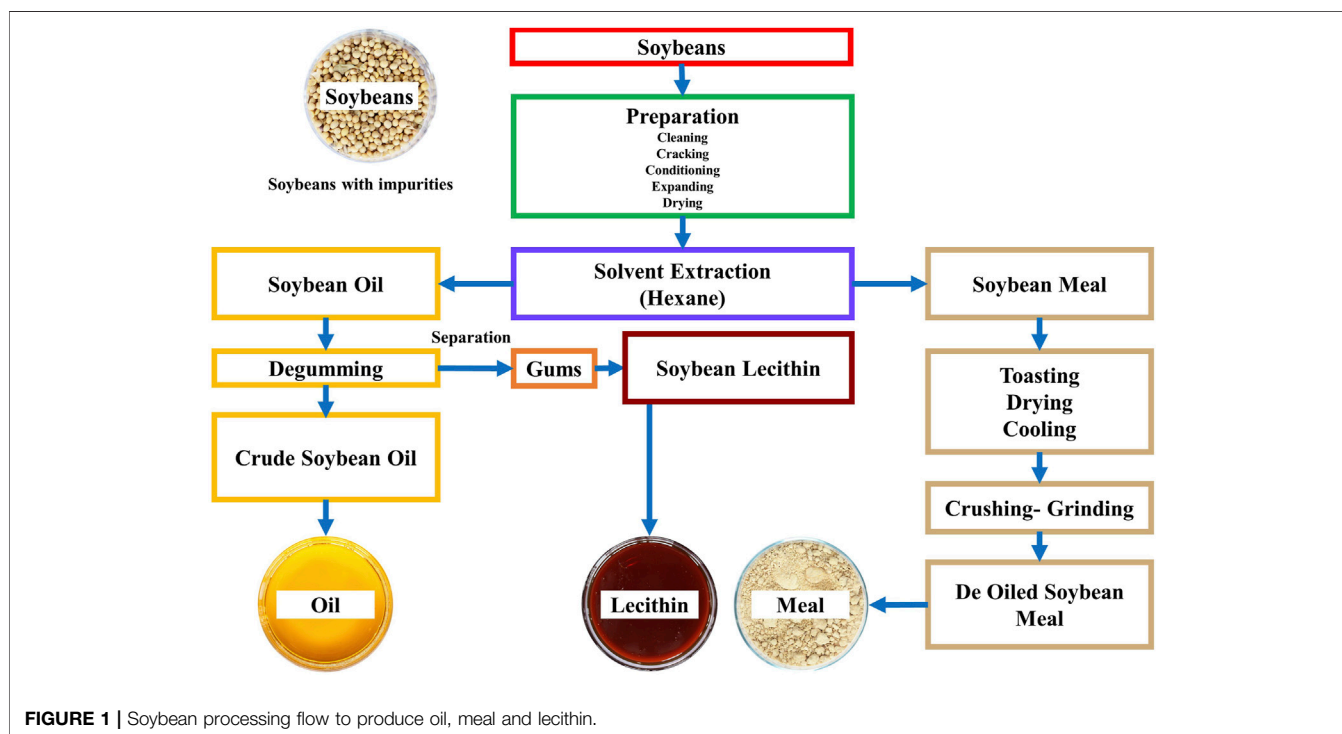

FIGURE 1 | Soybean processing flow to produce oil, meal and lecithin.

These signals are subsequently converted into the frequency domain through fast Fourier transform (FFT) processing, and the information is passed to the AI-based algorithms. When the motors and equipment are operating correctly, vibration is constant with a relatively small amplitude. When faults develop and the dynamic processes in the motors and equipment change, the vibration spectrum also changes, triggering the AI-based model to detect the deviations.

The tasks of PdM are closely related to modeling a system's normal behavior and detecting deviations, also known as anomalies, which may point to present or evolving failures. Anomalies translate to significant information regarding the health status of the motors and equipment, and anomaly detection can be achieved, for example, through classification or clustering. Fault detection, diagnosis and condition monitoring (Avenas et al., 2015) are critical for early diagnosis, and an anomaly detection is a common approach for fault detection (Vishwakarma et al., 2017). Considering the taxonomy of error, fault, and failure, an anomaly can be regarded as a potential error. An error, however, is caused by a fault, which may cause a failure. Consequently, anomaly detection may point to a fault and can be used for condition-based PdM. An overview of data-, model- and knowledge-driven predictive methods and a data-driven approach based on industrial data acquisition, edge processing, and decision-making is presented by (Ran et al., 2019). The data-driven methods are using signal processing techniques to analyze the signal for data modeling, including time-domain, frequency-domain, and time-frequency domain approaches. The frequency-domain approach, such as the Fourier-based analysis is used as an additional method for analyzing the vibration signals.

The system development includes a learning process (i.e., model training) that is based on historical raw sensor signals as well as an inference process where the trained AI-based model is exposed to real-time data to predict targets and make decisions. These phases include sub-processes, such as IIoT data acquisition and pre-processing feature engineering (e.g., feature extraction, concatenation, and selection) and AI-based model training and prediction. The data-driven PdM system is applied to industrial soybean processing manufacturing facilities. Different algorithms (Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN)) have been considered.

This paper is organized into six main sections. The introduction is included in **Section 1**. The evolution of the PdM and PsM and the description of data-, model- and knowledge-driven predictive methods are presented in **Section 2**. **Section 3** describes the intelligent real-time edge processing maintenance system architecture and design. The integration at the edge of the different ML and DL algorithms is described in **Section 4**. **Section 5** addresses the experimental setup, the specific experiments performed, and the results that were achieved. **Section 6** discusses future research challenges and presents the key open issues related to AI techniques and methods in implementing industrial PdM and PsM systems.

2 EVOLUTION OF THE PREDICTIVE AND PRESCRIPTIVE MAINTENANCE IN INDUSTRY 5.0 APPLICATIONS

The goal of maintenance for soybean production is to improve the performance, safety, and reliability of equipment. Motor failure may occur for a range of reasons such as loose bearings, misalignment, corrosion etc.

Traditional approaches to maintenance, such as reactive maintenance (RM) and preventive maintenance (PvM), are gradually being replaced by PdM solutions, which is based on continuous monitoring of the condition of equipment using IIoT devices to collect real-time signals and to identify trends in motor/equipment behavior. It is then possible to predict failures using prediction tools, models, and algorithms. This paper presents an AI-based edge processing real-time maintenance system for controlling and diagnosing industrial manufacturing equipment. This approach reduces unscheduled downtime of motors in the production line via early detection of potential motor faults. The system can be further tuned to improve energy efficiency by monitoring and controlling loads and operating conditions.

There are several approaches to PdM solutions for industrial processes (Jimenez et al., 2020). These approaches are described below.

First, knowledge-based models are built on experience and reflect in rules, facts, or cases that have been collected during previous operation and maintenance of the motors or equipment. This approach is mainly used in diagnostic systems that use rule-, case-, and fuzzy knowledge-based models for prognosis.

Data-driven models use historical data and real-time data collected by sensors, IIoT devices, etc., to develop a model of normal system behavior. Data-driven approaches implement statistical (Carden and Fanning, 2004), stochastic, and ML or DL. Managing the uncertainty inherent in these collected data is a challenge for data-driven (Dai and Gao, 2013) models; however, this uncertainty can be mitigated by probability calculations.

Physics or model-based approaches uses the laws of physics to assess component degradation. This approach incorporates a physical understanding of the target equipment and develops models to represent the system's behavior. This approach models the physical behavior of the equipment and relies on the accuracy of simulations to measure degradation in specific components, systems, equipment, or motors.

Hybrid models combine different approaches in a multi-model solution. These models address complex PdM tasks by optimizing the type and amount of information and/or data sources and the number of targeted diagnostics and predictive elements.

Electric motors and equipment are sources of noise and vibration. This noise and vibration can be used as a diagnostic tool in industrial maintenance. Noise and vibration associated with electrical issues are caused primarily by the action of unequal electromagnetic forces on the motor's stator or rotor. Common causes of unequal magnetic forces include open or shorted rotor

and stator windings and other inter-turn winding faults, broken rotor bars, static and/or dynamic air gap eccentricity, uneven air gap flux distribution, unbalanced current phases, torque oscillations, or pulses and magnetostriction.

Vibration and noise in electric motors may occur due to mechanical, electromagnetic, or aerodynamic causes. Mechanical problems typically involve defective bearings, imbalance, looseness, misalignment, or end winding damage due to mechanical shock, impact, or wear. Aerodynamic problems typically involve ventilation fans and may include discrete blade passing frequencies (BPF), resonant motor housings, or broadband turbulence.

The electromagnetic content of a noise or a vibration signal can be determined if the signal disappears immediately when the machine is switched off. A low-frequency narrowband analysis of the bearing vibration signal (Jin et al., 2014) can identify the shaft rotational frequency. The 50 Hz discrete frequency corresponds to the frequency of the electrical supply and the harmonic of the shaft rotational frequency. The discrete low-frequency component (around 3 Hz) is associated with base motion, specifically the induction motor unit and a centrifugal pump that is mounted on a steel mezzanine floor over a cooling pond.

PdM solutions can monitor vibrations, noise or sound, ultrasonic signals, temperature and thermography, and electric parameters (e.g., current, voltage, load) to collect data from industrial equipment/motors. Three types of accelerometers are used to measure vibrations: piezoelectric, piezoresistive, and MEMS.

Piezoelectric accelerometers usually use lead zirconate titanate (PZT) to sense vibrations and produce an electric charge that is converted to an electrical output under acceleration. Piezoelectric accelerometers have very low noise and can be used on a three-axis or single axis measurement setup. Piezoelectric accelerometers are AC coupled and cannot measure the gravity vector or sustained acceleration. A three-axis piezoelectric sensor that measures vibration in the x, y, and z directions can be used to measure tangential motion or vibration in the axis of rotation. Several mechanical forces in rotating machines (e.g., soft footing) produce tangential motions in the casing that need to be measured and analyzed. Piezoelectric accelerometers are used extensively in applications that require broad temperature tolerance or that vibrate above 50 g. Piezoelectric accelerometers require external analog-to-digital converters (ADCs) and pre-processing hardware, which increases the cost and the size of these sensors.

Piezoresistive accelerometers use a strain gauge to measure changes to resistance under acceleration. The piezoresistive strain gauge requires amplification and temperature compensation. The sensors are DC coupled and have an extensive bandwidth (0 Hz to several thousand hertz). The signal output can be integrated to calculate velocity and displacement during shock events.

MEMS accelerometers are micromachined components that measure capacitive changes under acceleration. The MEMS based sensors are DC coupled and have low-power consumption.

Advances in the performance of capacitive MEMS accelerometer sensors make them comparable to piezoelectric accelerometer sensors while retaining their advantages of lower

costs, higher integration, and industrial tolerance. These advances include the integration of ADCs, filters, and embedded processing building blocks for ML.

Vibration can be used to monitor and diagnose the condition of industrial equipment for PdM, and detect problems such as load imbalance, misalignment, and ball-bearing failures. New approaches can use the amplitude and frequency of a vibration to determine the type of failure.

Different standards have been developed for the sensor systems used to monitor vibrations; these standards provide guidelines for the use of vibration to monitor industrial equipment and motors. The ISO 2954:2012 standard (ISO 2954, 2012) specifies the minimum vibration intensity that a vibration measurement instrument must be able to detect. The ISO 20816 standard series (ISO 20816-1, 2016) that follows the ISO 10816 (Gomez, 2020) establishes conditions and procedures for measuring and evaluating vibrations from motors/equipment. It defines a vibration severity standard that uses the Root Mean Square (RMS) velocity in Hz of the installed motor's housing as a condition indicator. The measured vibration from the machine is classified based on machine size, mounting strategy, and machine class.

The draft standard ISO/DIS 20816-3 (ISO/DIS 20816-3, 2022) document defines the general requirements for evaluating the vibration of various coupled industrial motor classes with a power above 15 kW and operating speeds (revolutions per minute (rpm) between 120 rpm and 30,000 rpm when measurements are made stationary. The non-rotating and rotating parts of industrial motors are measured under normal operating conditions, and the standard provides guidelines for evaluating these measurements. The guidelines call for steady vibration and define acceptable changes in vibration amplitude during use.

Different motor-specific statistical deviations of measured vibrations and compare the results to that of acceptable limit-values provided by ISO/DIS 20816-3, and is applied to vibration measurement of medium-sized industrial electric motors in group 2 powered above 15 kW up to 300 kW and speed range from 120 to 15,000 rpm with rigid motor supports. The vibration zone boundaries are defined by the standard. The limits apply to the RMS values of vibration velocity and displacement in the frequency range from 10 Hz to 1,000 Hz. The velocity limit for medium-sized motor of 45 kW with rigid support is up to 2.8 mm/s for unrestricted operation as presented in **Figure 2**.

The PdM system presented in this paper uses different approaches that include statistical calculations, time/frequency domain analysis, ML and DL techniques for vibration signals processing.

The vibrations indicate motors unbalance, eccentricity, turbulence, resonance problems, misalignment of couplings and bearings, worn gears or bearings, rubbing, bent/cracked shafts, fan blade defects, defective/misadjusted drive belts/chains, sleeve-bearing problems, and mechanical looseness/weakness (Selcuk, 2016).

Vibration analysis can be used for detecting and identifying problems by having vibration pattern of the normal operating motor/equipment used as reference and then comparing and sorting subsequent patterns.

ISI/DIS 20816-3	Medium-size Machines		Large Machines	
Advisor	Group 2		Group 1	
Velocity	Rated Power			
mm/s RMS	15 kW - 300 kW		300 kW - up	
— 11.0 —	DAMAGE OCCURS			
— 7.1 —		RESTRICTED OPERATION		
— 4.5 —				
— 3.5 —		UNRESTRICTED OPERATION		
— 2.8 —				
— 2.3 —				
— 1.4 —				
— 0.7 —	NEWLY COMMISSIONED MACHINERY			
— 0.0 — Foundation	Rigid	Flexible	Rigid	Flexible

FIGURE 2 | ISO/DIS 20816–3 Guidelines for vibration monitoring.

Statistical time domain parameters such as peak, velocity and acceleration RMS, standard deviation, skewness, kurtosis, crest-, factor are used for describing the vibration signals (Wang et al., 2020).

Frequency-domain parameters offer additional information for fault diagnosis (Javed et al., 2015). The vibration signals are transformed by applying the FFT, decomposing the signal into its different frequencies. Approaches such as time dependent frequency analysis, short-time Fourier transform, continuous and discrete wavelet transform are used for analyzing the vibration signals.

AI systems use data, knowledge, and algorithms to simulate cognitive functions and learn and solve problems independently. Different clasifications and definitions of AI and machine intelligence are presented in Vermesan (2020).

Edge ML for IIoT applications can be applied across edge at micro-, deep- and meta-edge devices by adapting the ML model for inference capabilities (e.g., microcontroller, embedded processing unit, PLC, gateway, micro-server, on-premises server, etc.). These devices have different process capabilities, and the ML model must be scaled to fit the processing capabilities of the edge device. Model compression techniques are used to satisfy memory and runtime requirements. Tools like TensorFlow Lite (TensorFlow), PyTorch Mobile (PyTorch), and ONNX Runtime (ONNX) are deployed to optimize the model's memory footprint and runtime using techniques such as layer fusion, quantization and pruning. However, fine-tuning the ML model on industrial edge devices is challenging. This process requires several iterations and checks to ensure that the edge devices have the computing and local storage capabilities to run the model training process in a reasonable amount of time.

DL uses a complex structure of algorithms modelled on neural networks (NNs). DL algorithms are seen as the mathematically complex evolution of ML algorithms. DL describes algorithms that analyze data with a logic structure through supervised and unsupervised learning using a layered topology of artificial NNs. DL requires large amounts of data and computing power. The development of transfer learning techniques—for example, using

pre-trained models—can reduce the amount of data that are required.

Feature extraction and classification in the DL algorithm are done in the same phase because the features are extracted automatically, and the algorithm learns from the errors.

3 INTELLIGENT REAL-TIME EDGE PROCESSING MAINTENANCE SYSTEM ARCHITECTURE AND DESIGN

The intelligent real-time edge-processing maintenance system is implemented using a layered architecture composed of several layers, which are described in the following paragraphs. The layers include IIoT devices, communication networks, intelligent edge-computing units, analytics, and prediction and decision-making components and functions. The approach supports the development of AI-driven IoT autonomous capabilities and increases the decision abilities of IoT edge devices, creating new business scenarios that incorporate the use of intelligent things into traditional manual and semiautomated tasks such as PdM (Vermesan and Bacquet, 2020; AI4DI).

The communication networks provide data transfer using different wireless protocols such as Bluetooth, Wi-Fi and LoRaWAN. The wireless protocol stack IEEE 802.15.1 (IEEE 802.15.1, 2005) is used for Bluetooth communication by deploying IIoT sensors connected to a gateway via Bluetooth. The raw condition data from the IIoT sensors is transmitted to a secure server to be processed and analyzed.

The Bluetooth IIoT devices operate within the 2.4 GHz radio range and use frequency hopping to enable the sensor devices to transmit RF signals in industrial environments where motors under monitoring are located. The IIoT devices integrate the sensors (vibration, temperature, etc.) and a system-on-chip processing unit that combines an Arm® processor with a floating-point unit (FPU), with a 2.4 GHz multiprotocol radio (supporting Bluetooth software). The Bluetooth IIoT devices are connected to an industrial-grade Bluetooth gateway with communication backhaul options for Wi-Fi and Ethernet.

The wireless protocol stack IEEE 802.11a/b/g/n/ac/ax, point-to-point/star/cluster-tree (IEEE 802.11ax, 2021), in the ISM 2.4 GHz band is used for the Wi-Fi IIoT sensors and gateway. The Wi-Fi IIoT sensors are optimized to boost the radio link quality in industrial environments subject to random and diverse disturbances using two omnidirectional antennas that can assure a line-of-sight (LOS) maximum radio range of tens of meters. As the IIoT sensor nodes are operated in industrial environments (soybean production facility), the radio range is extended by adding Wi-Fi bridges/repeaters.

The LoRaWAN, Low Power, Wide Area (LPWA) networking protocol is used to enable several battery-operated IIoT sensors to connect wirelessly, forming a network architecture deployed in a star-of-stars topology where the gateways dispatch messages between IIoT devices and a central network server.

The LoRaWAN platform has a layered topology comprising five layers: an IIoT device layer, a gateway layer, a network layer, an application layer, and a processing/analysis layer. The IIoT device layer is part of the micro-edge implementation and includes the sensors (three-axis vibration, temperature, humidity, etc.), the microprocessor, LoRaWAN wireless communication modules, and the embedded software and/or operating system.

The gateway layer is part of the deep-edge implementation and includes the LoRa concentrator and the host microcontroller unit (MCU). The LoRa concentrator entitles the gateway to receive and transmit LoRa messages and contains a LoRa baseband processor and two front-end chips. The LoRaWAN frequency band used is EU868 (863–870 Mhz). Different options were evaluated for LoRaWAN system communication backhaul, including Wi-Fi, Ethernet, and cellular.

The evaluation process has included LoRaWAN gateways connected online using a network server deployed by a third party or integrated into the gateway. An integrated LoRaWAN gateway solution stacked on a single-board computer is under evaluation.

As part of the LoRa architecture, the network layer ensures communication between devices and gateways, and between gateways and network servers and application servers. The application layer includes the application server, which provides the information related to the gateways/devices profiles, categorizes the types of data based on properties of sensors, and stores data in a database. The processing/analysis layer can be used for applying AI/ML techniques for situation analysis. This layer utilizes all the information provided by the network and information layers by synthesizing information from multiple sources over time and identifying the events, the context, and the situation. ML and other AI techniques can learn and identify abnormal situations using the data collected.

The PdM architecture deployed considers that the IIoT sub-systems are connected to different edge gateways. The information collected from the various IIoT devices is aggregated to an on-premises edge server, as presented in **Figure 3**. The proposed edge computing solution improves the performance, security, operating cost, and reliability of IIoT and AI-based platform, applications, and services.

The system design is based on a heterogeneous wireless sensor network that consists of sensor nodes and IIoT devices with different communication interfaces, computing processing, sensing range and AI-based capabilities.

In industrial environments using real-time communication, the coverage area of wireless devices, their radio channel characteristics, and the reliability of the data may suffer from noise, co-channel interferences, strong vibrations, and interferences from other devices using ISM bands, among other limitations. The signal strength may be severely affected by the reflections from the walls (multi-path propagation), interferences from other devices using ISM bands (863–870 MHz or 2.4–2.5 GHz), and the noise generated from the equipment or heavy machinery. It is essential to maintain data integrity under these conditions.

The implementation solution presented in this paper addresses the wireless network performance requirements, including latency and reliability, and design as part of an edge processing concept to leverage the computing and storage
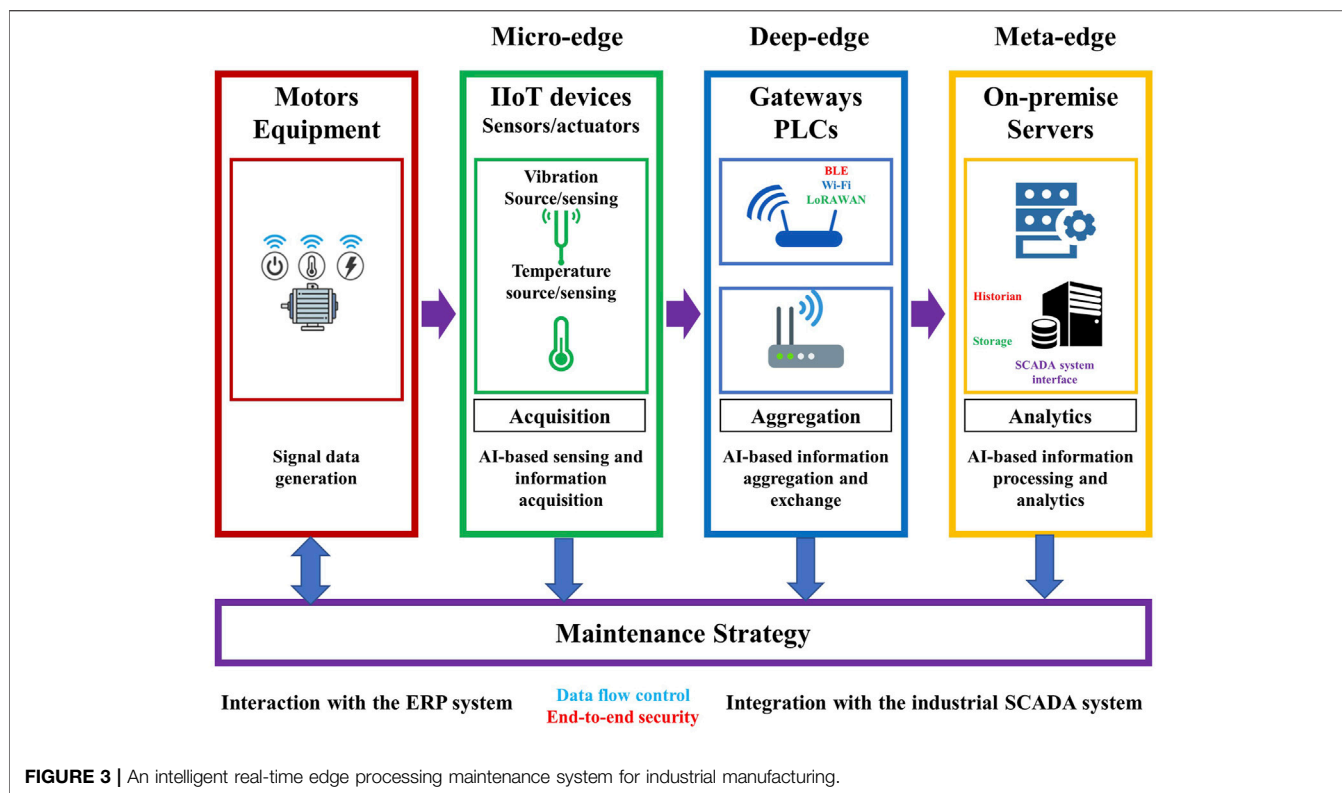
FIGURE 3 | An intelligent real-time edge processing maintenance system for industrial manufacturing.

capabilities of edge IIoT wireless devices. IIoT edge devices are optimally deployed to locations across the factory floor and placed on the motors/equipment, with several gateways and wireless repeaters implemented to address network overload, latency, and interference.

The gateway placement for protocols such as Bluetooth, Wi-Fi and LoRaWAN considers different characteristics of industrial systems, such as the wireless channel, fading, multi-path, and performance requirements. The traffic for monitoring IIoT devices and gateways in this implementation is characterized by small packets and relatively small communication distances that ensure the latency, bandwidth, and throughput to reliably collect and monitor the parameters for the motors/equipment. For example, three gateways collect data from 32 IIoT nodes spread across the production facility for Bluetooth communication.

Adopting edge processing using several IIoT devices associated with maintenance tasks that may have various latency and reliability requirements was part of the design. The optimal assignment of wireless gateways was adapted to the dynamic environment of intelligent soybean manufacturing systems to reduce interference, improve the monitoring and control of IIoT devices, and reduce time delay, data dropout, or transmission congestion. In this context, a preliminary inspection of the site was performed for Wi-Fi connectivity, and an analysis of the load on different channels was conducted. For the deployment, the channel with a lower load was selected for each IIoT device.

The PdM implementation performs maintenance based on the motors/equipment health status indicators. The edge computing

approach is integrated and interfaced with industrial supervisory control and data acquisition (SCADA) infrastructure and linked through the historian component. The processing at the edge and the AI deployment use the data stored in the historian unit linked to the industrial SCADA system.

4 EDGE INTELLIGENCE INTEGRATION

The implementation and integration of data-driven PdM and edge intelligence follow the phases of the design methodology framework presented in **Figure 4**:

- Operational analysis and assessment.
- Data collection and acquisition.
- Feature engineering
- AI modeling
- Edge processing

The operational analysis and evaluation include the operating conditions of the motors/equipment, environmental restrictions on data acquisition, time cost (e.g., PdM system design and development, debugging, deployment), economic costs (e.g., research and development, SW/HW implementation, testing).

The data collection and acquisition include the conversion and processing the signals from the IIoT sensors into data and information to be modeled by AI-based techniques. The PdM system performance depends on the efficient design and
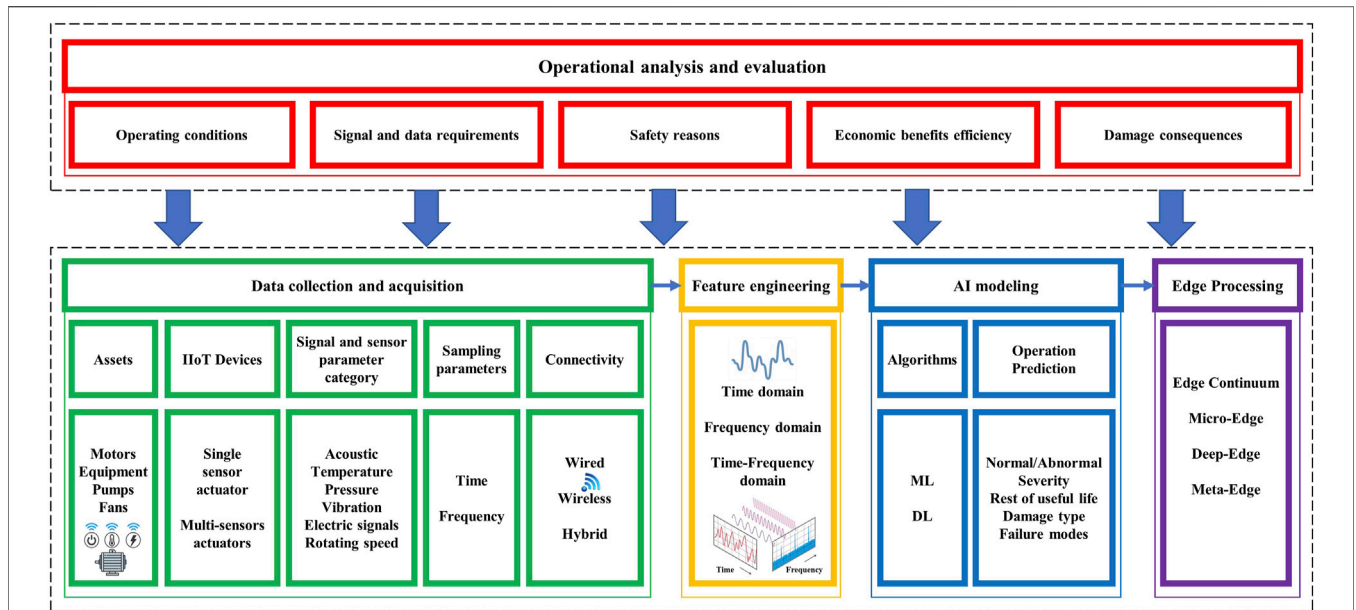
FIGURE 4 | Design methodology framework for PdM edge intelligence integration.

implementation of the components covered by the functions integrated in this phase.

The feature engineering phase implements the signal functions by implementing functional module for time-, frequency- and time-frequency domain analysis.

The AI modelling phase includes the AI model selection (based on the model complexity, target processing unit, computing resources), model training (supervised, unsupervised, labeled/unlabeled data) and the model prediction (with requirements such as time-efficiency, accuracy).

An essential feature of the proposed solution is that the model training is carried out at the edge (micro-, deep-, meta-), and the computing power and energy requirements are aligned with the computing power and storage capacity of the edge devices. For the model to be deployed on edge devices, such as an intelligent IIoT node, edge gateway, or edge server, the model is developed to optimize computing resources.

Activating AI processing loads at the micro-edge (sensors/actuators and chips/microcontrollers) level makes various data collection processes faster and more data rich. The transition to the AI-embedded circuit technologies allows for offloading of network-data-carrying loads while providing edge real-time information, processing, and analytics.

Activating AI processing loads at the micro-edge level system on a chip (SoC) allows implementation of intelligent distributed IIoT device architectures that distribute, and offload data-processing loads to different layers of application architecture across the edge continuum, improving data management and processing and optimizing communication bandwidth. Manufacturing processes powered by AI-enabled decisions at the edge, use edge AI circuits as part of the more extensive industrial system integrated into upstream and downstream operations.

Integrating AI software and hardware stacks into industrial processes has many challenges, including AI model-building, programming AI-based SoCs and deploying these systems to various industrial applications. Edge IIoT architecture integrates AI methods/algorithms to pre-process information in real-time at the edge and transfer relevant and valuable data to industrial processes.

The PdM design methodology framework is used as the basis for the AI-based edge processing real-time maintenance system for industrial manufacturing control and diagnostics, which will be integrated into several layers that map to the functional domains of the high-level reference hybrid system's architecture. The AI-based edge processing real-time maintenance system layers are:

- Physical layer
- Control layer
- Signal analysis layer
- Prediction layer
- Decision-making layer

Several parameters (e.g., vibration, temperature, sound, current, rotating speed) are monitored as input signals to represent the performance of the motor/equipment. By using the vibration signal to perceive the status of the motor/equipment, vibration monitoring is considered appropriate for identifying failure. When correlated with electric signals, the vibration information can precisely identify defects in the motor or equipment operations.

The accelerometer signals are pre-processed, and depending on the application, different analysis techniques in time, frequency, and time-frequency domains are applied to acceleration and velocity signals to extract valuable intelligence about the motor condition.

Signal data analysis and engineering processing includes data processing in the time, frequency, and wavelet domains as illustrated in **Figure 5**.

Time series forecasting refers to the estimation of parameters that change over time. These parameters are measured until instant t and the value that will be predicted is instant t + dt. Typically, variable of interest measurements can be obtained at regular intervals, making it possible to predict future values. In a time-series, it is common to identify the following:

- Trends or long-term increase/decrease in values. Seasonal phenomena, or phenomena that determine changes in values over time, that constantly repeat the same duration.
- Cyclical phenomena, which push fluctuating increase/decrease in values, that do not always have the same duration (i.e., they are not periodic).

One of the most important things to understand when analyzing data is the relationship between the measured quantities. The time-domain analysis of the signals measured by the IIoT devices on motors/equipment makes use of several statistical time-domain features, such as:

- Acceleration peak (g)
- Velocity RMS (mm/s)
- Acceleration RMS (g)
- Displacement (μm)
- Standard deviation
- Kurtosis
- Crest factor
- Skewness

The time domain statistical parameters, equations and description is presented in **Table 1**.

The vibration RMS velocity technique determines the vibration signal's trends over time. As machines wear, their vibration velocity increases, and monitoring RMS velocity trends provides an indicator of wear that is compared to pre-determined thresholds, signaling the need for maintenance. Acceleration is compared to pre-determined thresholds to detect bending or breakage in mechanisms for motors. Many defects typically display periodic "spikes" in the signal, and identifying trends, such as increasing acceleration or instability in the acceleration profile over time, are indicators of wear and damage.

Because these features examine the probability density function (PDF) of the signal, which changes when the condition of the motor/equipment changes, the skewness and kurtosis can be affected accordingly. Skewness measures whether the signal is negatively or positively skewed, and kurtosis measures the peak value of the PDF and indicates whether the signal is an impulse in nature (e.g., a shock signal).

Normal operation motor vibration signals have a normal distribution and a skewness value of zero. While kurtosis is calculated from the peak of the PDF of the vibration signal, skewness is obtained from the mean value of the PDF of the vibration signal. The kurtosis value of the normal motor's vibration signal is around three, and the skewness value is approximately zero. When the PDF of the vibration signal changes due to faults/failures, the kurtosis increases to a value greater than three, and the skewness shifts to a negative or positive value.

The entropy of the vibration signal can be calculated as the histogram of the PDF, and it measures the degree of randomness in the vibration signal. Based on the features mentioned above, several of the following non-dimensional features can be calculated (Caesarendra and Tjahjowidodo, 2017):
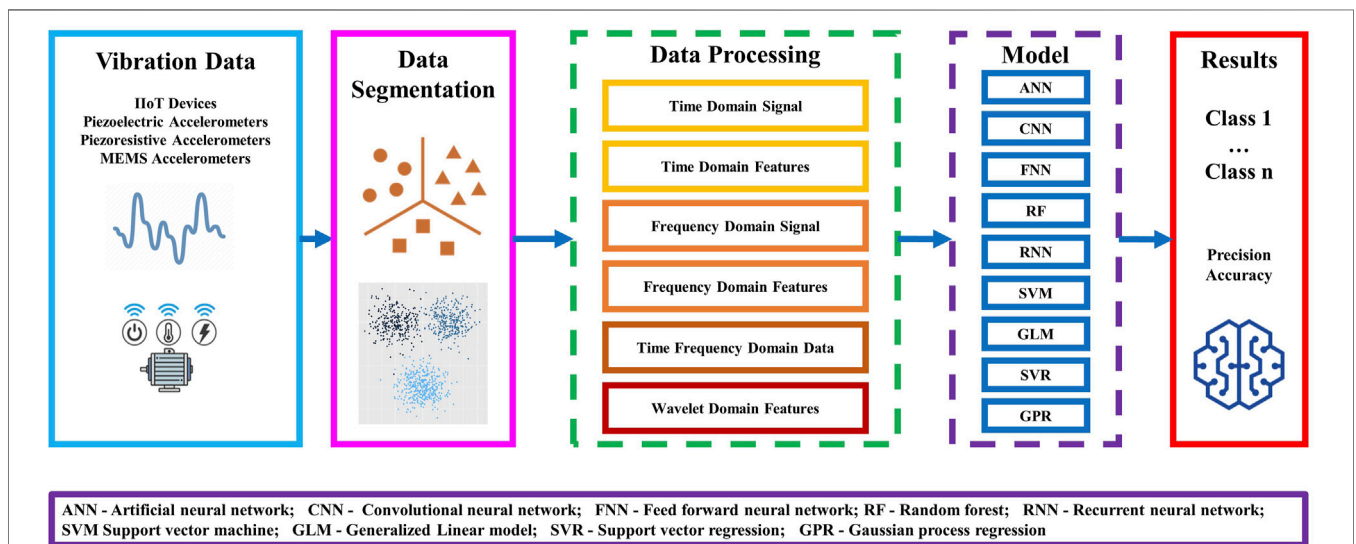

FIGURE 5 | Signal data analysis, engineering processing and AI-based model building flow.

- The shape factor, defined as the RMS divided by the mean, represents a value affected by an object's shape but independent of its dimensions.
- The crest factor, defined as the standard deviation divided by the RMS, calculates how much impact happens during the rolling element and tread contact.

Using a FFT to convert the acceleration and velocity data from the time domain to the frequency domain offers more detailed insights into motor conditions. For instance, a strong signal at a single frequency related to the rotation rate can indicate an imbalance or bent shaft in motors. General looseness or a broken gear tooth can be impacted by introducing different harmonic content. A strong signal amplitude modulated by a lower frequency is a diagnostic tool for gear mesh analysis and can be identified by examining the frequency domain.

The Fourier transform is defined as:

$$F(\omega) = F\left(f(t)\right) = \int_{-\infty}^{+\infty} f(t)e^{i\omega t}dt$$

Where ω frequency, and F (ω) the frequency-dependent Fourier transform and t representing time, f(t) the time-dependent function.

The FFT is an algorithm that computes the discrete Fourier transform (Heckbert, 1998; Seryasat et al., 2010). In order to deal with large amount of sampling vibration data, short time Fourier transform (STFT) analysis that combines time domain with frequency domain to analyze the vibration characteristics of the target points in a long period and wavelet transform (WT) is used (Zhu et al., 2014).

Depending on the motor's design, the sensor bandwidth of the vibration accelerometers ranges from 0.1 Hz to 5–10 kHz for synchronous AC motors with speeds up to 3,600 rpm and DC motors with speeds from 10 to 7,000 rpm or more. As changes in the acceleration reading or temperature accumulate throughout the velocity data integration, vibration monitoring sensors must have good stability over time and across temperatures. The vibration sensor should be as small as possible to maximize the options for placement on the monitored motors/equipment. The vibration sensor's mass should be low to prevent it from affecting the motors' vibration characteristics.

TABLE 1 | Time domain statistical parameters.

| Parameter	Equation	Description
Peak	$x_p = \max|x_i|$	Represents the maximum value of a signal
Mean	$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$	Represents the average value of a signal
Standard deviation	$x_{std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$	Statistical metric defining the amount of variation in the signal, that is independent of DC bias. A low value indicates that the values tend to be more concentrated and closer to the mean, while a high value indicates that the values tend to spread out over a wider range
RMS	$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	Root Mean Square (RMS) or the quadratic mean is the square root of the mean square (the arithmetic mean of the squares of a set of values). RMS can be calculated from the power spectral density of a signal (Seryasat et al., 2010)
Kurtosis	$x_{kur} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^4}{x_{std}^4}$	Quantifies the peak value of the probability density function. Statistical parameter that allows to analyze the distribution of the vibratory amplitudes contained in a time domain signal. It corresponds to the moment of fourth order norm. For a Gaussian distribution, its value is 3. Kurtosis indicates heavy-tailed or light-tailed datasets in comparison to a normal distribution. Datasets with high Kurtosis tend to have outliers (Dron et al., 2004)
Crest factor	$CF = \frac{x_p}{x_{rms}}$	Corresponds to the ratio between the crest value (maximum absolute value reached by the function representative of the signal during the considered period of time) and the RMS value (efficient value) of the signal (Dron et al., 2004)
Skewness	$x_{ske} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^3}{x_{std}^3}$	Quantifies the asymmetry behavior of a vibration signal through its probability density function. The skewness is a metric of symmetry in a dataset. A dataset is symmetric if it looks the same to the left and right of the center point. A normal distribution has a skewness of 0. Negative skew value illustrates that the distribution concentrates at the right. Positive skew value illustrates that the distribution concentrates at the left (Singh, 2021)

Where x_i is a digital signal series, i = 1, 2, ..., N; N is the number of elements of the digital signal.

5 EXPERIMENTAL SETUP

The experimental set-up comprises the physical/field part, including the equipment and motors, and the IIoT devices mounted on the motors. The system is intended to monitor electric motors such as such as AEG AM 225 MX4 (hammer mill/crushing) with the following characteristics: 45 kW power, 220/380 V, 150/87A, 50 Hz, and ~1,500 rpm; and ABB HXR 315 4 B3/HXUR 638G2 B3 (preparation/conditioning) with the following characteristics: 200 kW power, 380/220 V, 360/624 A, 50 Hz, and ~1,500 rpm. In the set-up for the proof-of-concept, smaller motors with lower vibration frequency characteristics have been installed on the lab bench.

The PdM demonstrator performs maintenance based on the health status indicators of the motors and equipment. The IIoT-based sensors are used to measure unusual patterns of the motors' parameters, such as vibration level, temperature, and current consumption. Based on experience, such distinctive patterns can precede potential failures.

The sensor integrated into the IIoT device (Sensor Tile Wireless Industrial Node, 2022) used in the experiments is a three-axis MEMS accelerometer that contains three ultrawide bandwidth (DC to 6 kHz) acceleration sensors, a 12-bit analog-

FIGURE 6 | STWIN sensor tile wireless industrial node.
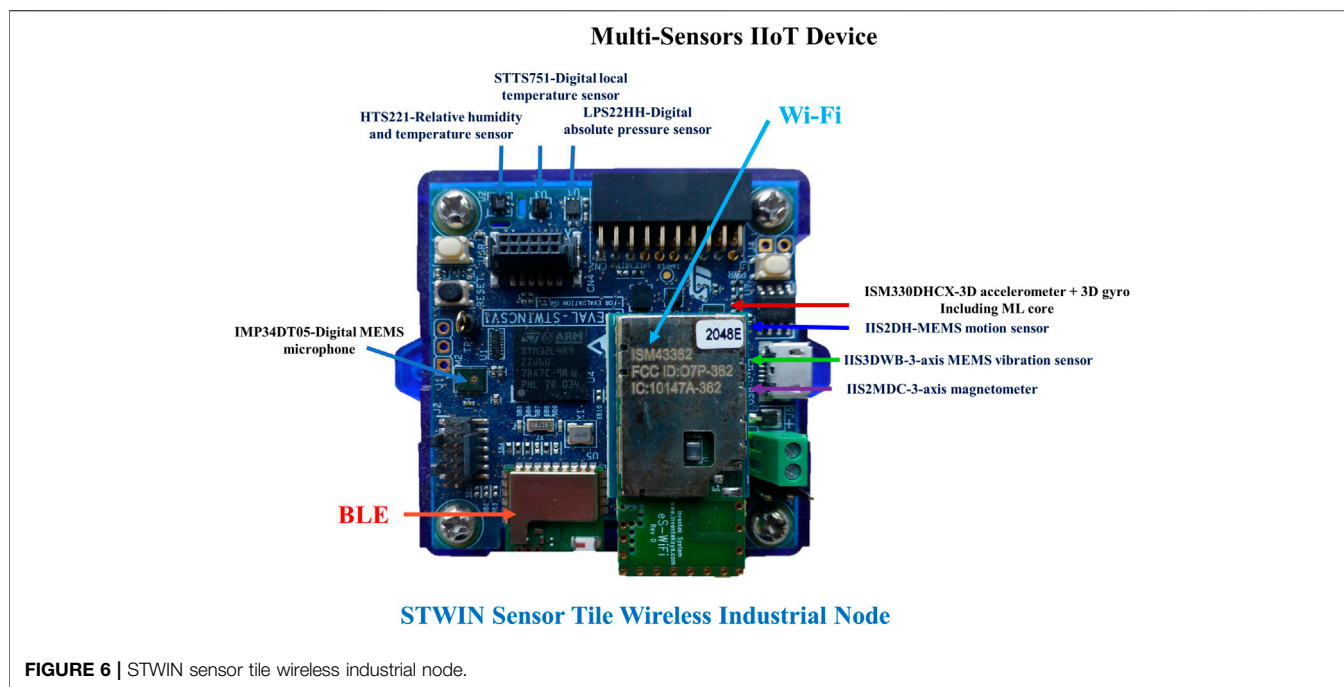
to-digital converter, a user-configurable digital filter chain, a temperature sensor, and a serial peripheral interface. The MEMS accelerometer sensor operates from 2.1 to 3.6 V with a low current consumption of 1.1 mA. The operating temperature ranges from −40°C to +105°C, and the MEMS vibration sensor has a selectable sensitivity (±2, ±4, ±8, or ±16 g) that can be tailored to a range of application requirements.

There is another three-axis MEMS accelerometer in the same IIoT device and a three-axis gyrometer for six degrees of motion sensing, including an inter-integrated circuit interface, sensor hub capability, a 9 Kbyte first in, first out algorithm, a programmable finite state machine for data processing, and the core blocks for ML.

The processing capabilities of the IIoT device are ensured by an Arm® Cortex®-M4 processor with environmental sensing, vibration monitoring and sound/ultrasound detection. It also features a debugger, a floating-point unit to handle the processing, embedded signal processing libraries running on an STM32 ARM Cortex-M4 microcontroller. The module can be powered externally or by an internal lithium-ion battery. The module has BLE and Wi-Fi connectivity. The multi-sensor IIoT device is illustrated in **Figure 6**.

STM32 ARM Cortex M-based boards allow AI/ML to integrate easily and cost effectively. They can run autonomously at the edge with no cloud connection, which provides privacy data security and optimizes latency.

5.1 Fusing the Predictive Maintenance Application with Machine/Deep Learning

In the previous sections, the PdM application was described using statistical and predictive modeling techniques to identify patterns in historical data and capture the relationships among the

relevant parameters and factors. This process allowed to assess the capabilities and risks associated with PdM.

In this section, the process is extended by applying ML to real-time measurements of selected key parameters to assess the current and future functioning. The fusing of the PdM application with ML/DL is shown in **Figure 7**. The fusing encompasses all the steps in the pipeline generation based on the sensor data, including feature extraction and selection, model creation, training, and classification.

Two AI solutions for fusing were designed, one with ML and the other with DL.

The ML solution creates untrained ML models to be embedded in MCUs, running unsupervised learning algorithms directly on the target microcontroller. Given that the training process is integrated within the device, restrictions may apply. Several optimizing and configuration features are therefore available, such as limiting random access memory (RAM) and flash memory, frequency filtering, flash memory optimization, generation of alternative candidates and selection of the best model candidate. A microcontroller emulator can be used to test and debug the model before deployment on the device.

The DL solution creates pretrained NN models with frameworks such as Tensorflow, Keras, or ONNX and converts them into optimized code for the MCUs. It can also optimize memory usage during runtime and run large NNs by storing weights and activation buffers in external flash memory and RAM, respectively.

Both solutions are standalone, end-to-end solutions for anomaly detection and classification tasks, but they are complementary rather than mutually exclusive. The ML solution is based on unsupervised algorithms and is completely automated. As it skips the complexity of full

FIGURE 7 | Fusing the PdM application with ML/DL.
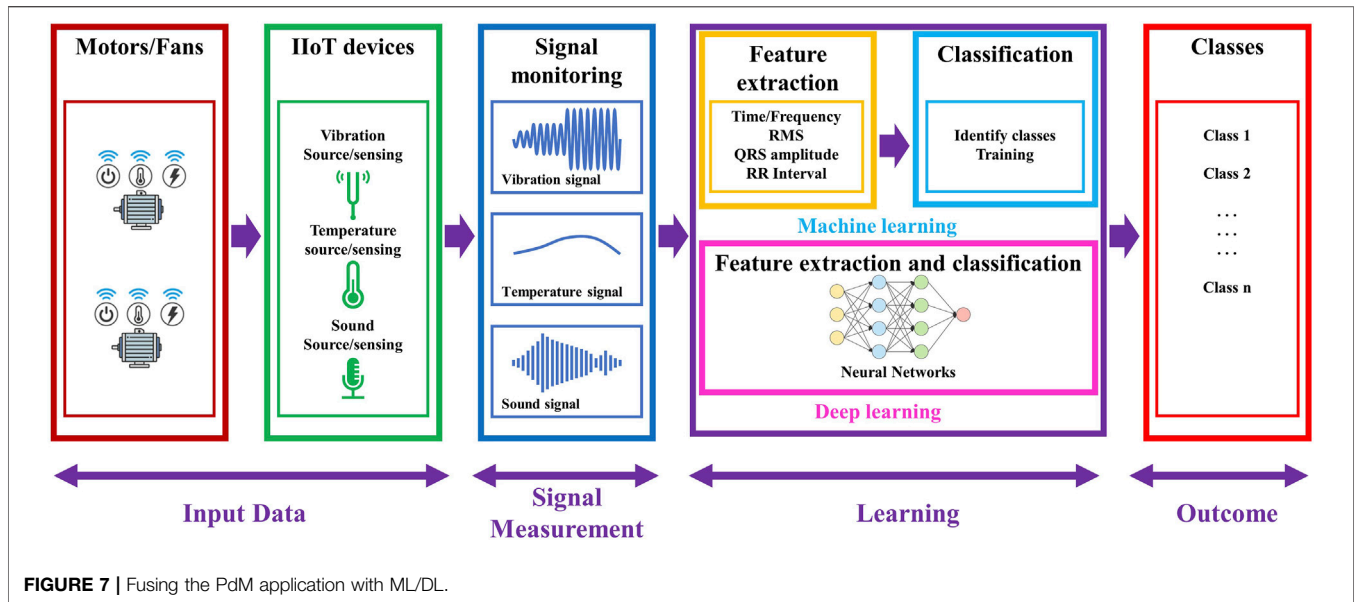
pipeline generation based on the data, it is intended as a rapid prototyping system. The DL solution, on the contrary, follows all the step, including building and training the NN. Both solutions generate optimized code to be included in the microcontroller application, and both run on-device inference and display output.

5.2 Machine Learning Solution for Predictive Maintenance

In short, anomaly detection with ML solution provides a dynamic model to detect anomalies in the PdM system. Based on the sensor data representing both normal and abnormal behavior, a set of model candidates is generated using various ML algorithms, and the best one regarding accuracy, confidence and memory usage is selected. Accuracy reflects the candidate's ability to correctly identify normal signals as normal behavior and abnormal signals as abnormal behavior, whereas confidence reflects the candidate's ability to separate the two.

The generation process for the best candidates can be stopped any time after achieving satisfactory results; however, it is recommended that the process runs as long as the accuracy and confidence are lower than 80%. For example, the selected candidate for the PdM application has an accuracy of 98.95%, confidence 80.1%, RAM 5.7kB + Buffer 6.2kB and Flash 13.2 kB.

The selected model can be installed on multiple systems mounted on different motors. Each model will learn its motor's behavior and create a distinct trained model, as motors have different behaviors. During the inference step, the models take the vibrations coming from the motors as input and are able to identify and report anomalies correctly.

The models will function based on a state machine, as shown in **Figure 8**.

The model functions for the "init," "learning" and "detection" states are called in the main loop. After initialization, the model runs a predefined number of training cycles, after which it

switches to inference mode and is ready to detect anomalies as required by the PdM application. The model can be programmed to retrain, either periodically or with a user interrupt callback.

The ML model was implemented with NanoEdge AI Studio (https://www.st.com/en/development-tools/nanoedgeaistudio. html), following the flowchart as shown in **Figure 9**.

The flow steps in the chart are detailed in the following text:. At the start, if any memory restrictions exist, the maximum amount of RAM and flash can be set.

For the logging of sensor data, a simple datalogger code was generated that reads and logs the raw sensor data directly on the serial port of the processing device, so that logs can be retrieved from a computer using serial tools such as Tera Term or from the console of the integrated development environment (IDE).

The signals for normal and abnormal behavior can be imported from different sources, and all settings can be configured in the graphical user interface (GUI); no code writing is required at this stage. Different configurations exist in terms of data format, sensor inputs and using a serial interface or secure digital (SD) card to store the data. The signals can be visualized as shown in **Figure 10**, in both temporal and frequency plots for the accelerometer Z-axis.

Depending on project constraints, the buffer size, signal lengths, and sampling frequencies will vary. For the 3-axis sensor, a collection of 100 signals was acquired for each normal and abnormal behavior, at a baud rate at 115,200 bps and with a buffer size of 512 samples on each axis, in total 1,536 values per signal. With a sampling frequency of 1,667 Hz, each buffer represents a temporal signal of ~300 ms, which is sufficient to capture the essence of motor vibration patterns.

Filter settings were activated in the signal pre-processing steps. By providing filtering, only the frequencies that represent the characteristics of the motor vibration are kept, and the rest are attenuated. Filtering techniques also help
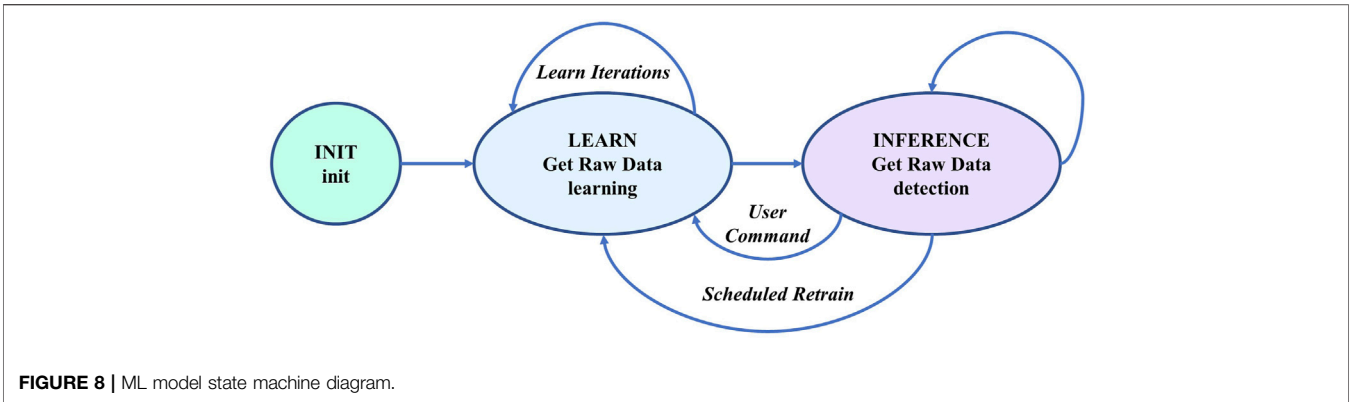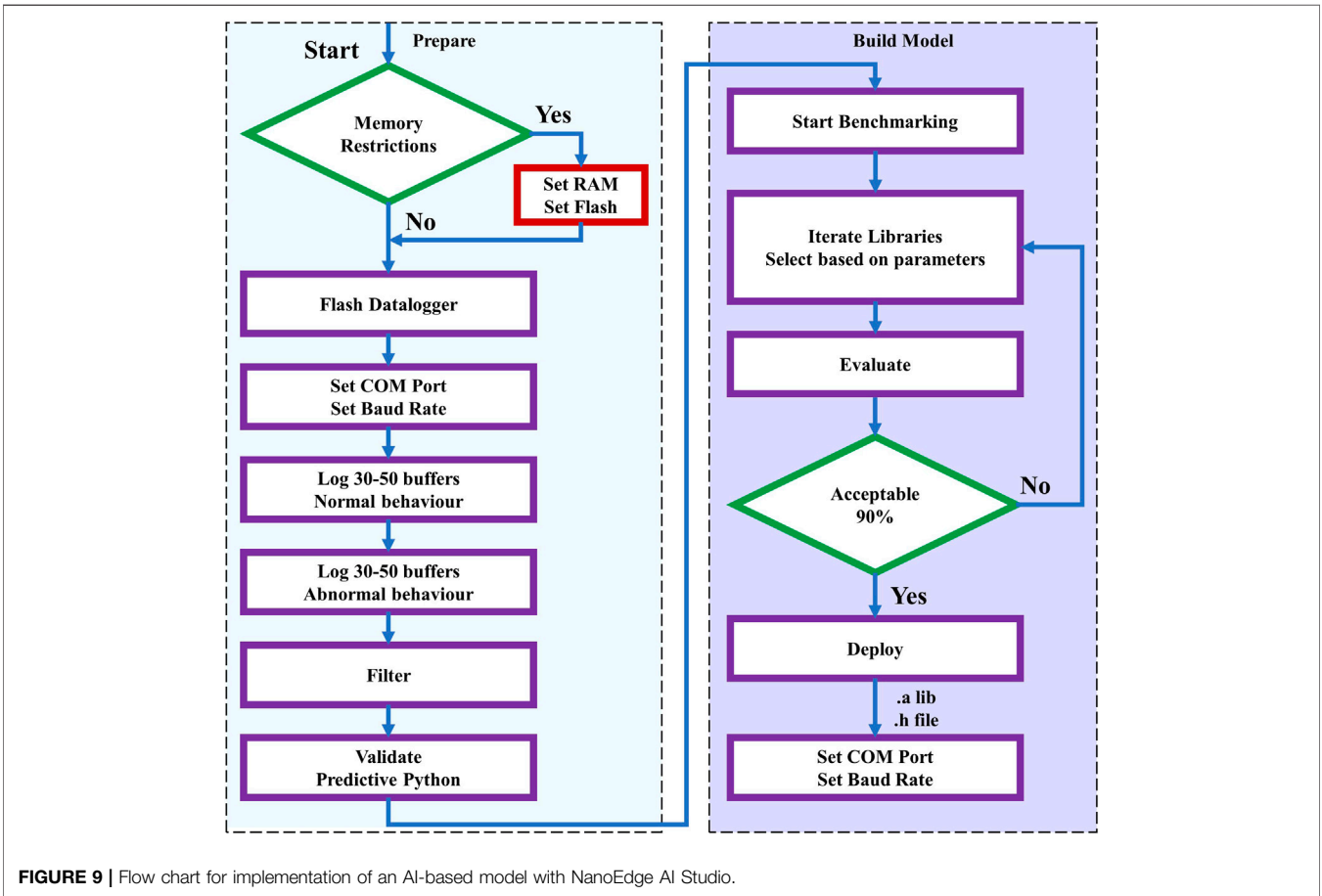
FIGURE 8 | ML model state machine diagram.

FIGURE 9 | Flow chart for implementation of an AI-based model with NanoEdge AI Studio.

eliminate high-frequency noise that interferes with the vibration signal.

As shown in **Figure 10**, the signals can be visualized in both the temporal and frequency plots for the accelerometer Z-axis.

The graph's x-axis corresponds to the number of samples (512), while the y-values contain an indication of the mean value of each sample across all 100 signals, their min-max values, and their standard deviation.

The abnormal signal was acquired by applying a variation to the load of the motor. During inference, it was recognized as an anomaly. Another variation, not seen before, has also been applied, leading to the similarity dropping instantly while an anomaly occurs as expected.

It is important to note that these signals mainly provide context for the automatic search for the most efficient algorithms and optimization for target AI-based embedded systems during benchmarking. The model is trained on the target directly, where only the "normal" behaviors are learned.
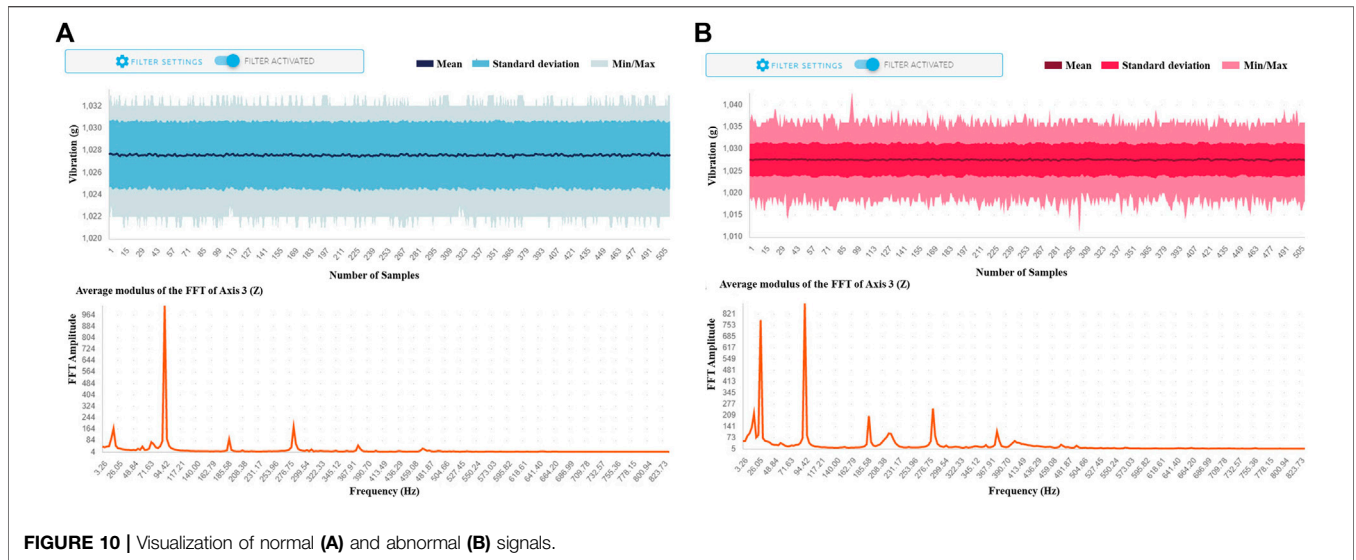
FIGURE 10 | Visualization of normal **(A)** and abnormal **(B)** signals.

Based on the collected data, the benchmarking process starts searching through a pool of ML algorithms and tests combinations of three elements: preprocessing, principal component analysis (PCA), model using algorithms such as K-Means, random forest (RF), support vector machines (SVM), and hyper-parameters for each model. Each combination is a library that is evaluated for balanced accuracy, confidence and memory usage, and the results provide a ranking of these libraries.

Each library has its own emulator, which allows for easy testing with real data to determine if it provides acceptable results, without the need to download, link or compile. If the library functions as expected, it can be deployed and downloaded in the form of a library and header file. The firmware wrapped with the generated files is then compiled and flashed onto the microcontroller target. A snapshot from the anomaly detection is shown in **Figure 11**.

5.3 Deep Learning Solution for Predictive Maintenance

In short, anomaly detection with a DL solution entails a model generated and trained using a ML framework being input into a model converter or code generator in order to deploy it on the embedded device. Various converters are available to reduce the size and other resource requirements so that models developed using high-end APIs can be deployed on boards.

For the implementation of the AI/DL solution, code generation was chosen in which the model is translated into a low-level programming language more suitable for embedded devices in terms of performance. Inferences can be made with high accuracy while also accounting for limited resources; the generated files and NN library are then merged with the firmware, compiled, and flashed onto the target microcontroller.

For the DL model, different architectures were considered, with the final choice being convolutional neural network (CNN)
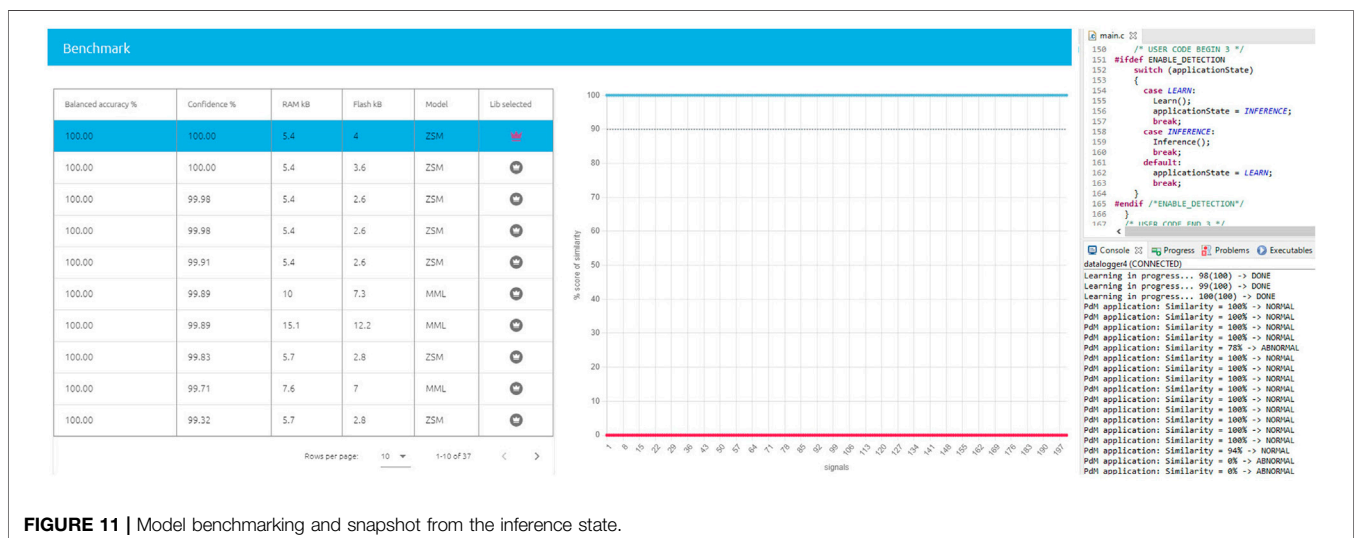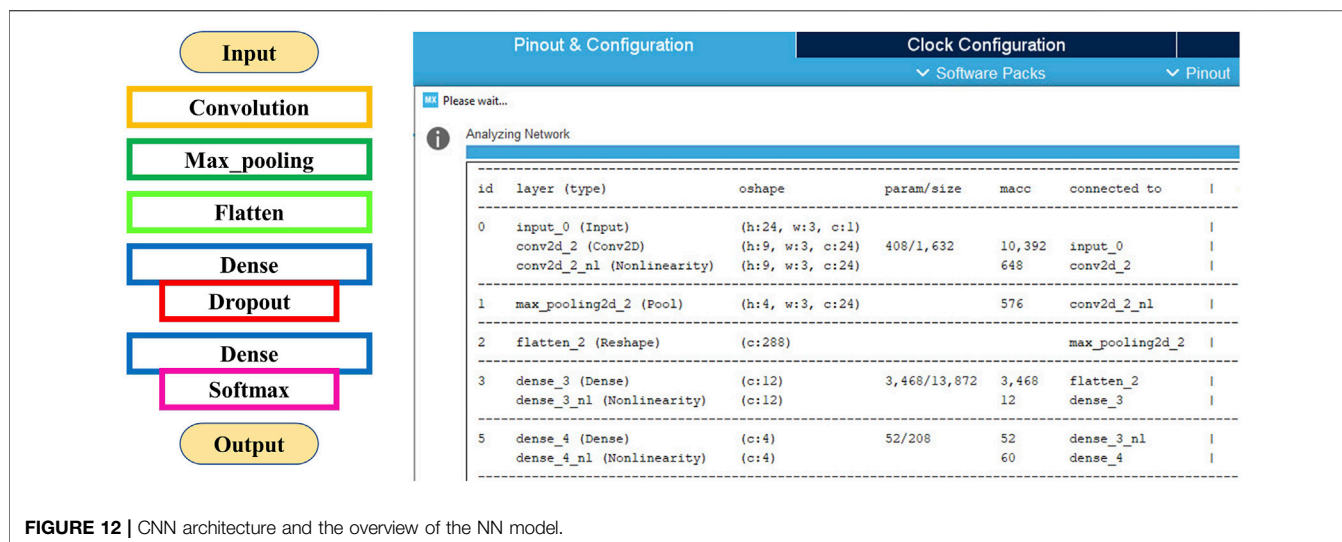

FIGURE 11 | Model benchmarking and snapshot from the inference state.

FIGURE 12 | CNN architecture and the overview of the NN model.

architecture (Jing et al., 2017; Kiangala and Wang, 2020; Silva and Capretz, 2019). CNNs combine the feature extraction process and the classification of the extracted features in the same algorithm, which was one of the decisive factors in its selection. This advantage emerges from the capability extracting knowledge from raw data. CNNs usually present a structured pattern of layers, combining successive convolutional layers with ReLU (rectified linear unity as activation function) and pooling layers. The convolution operation performed by the convolutional layer is the feature extractor, which is a filter (also known as a kernel) that slides over data, combining information (Jing et al., 2017).

The CNN architecture for the PdM proof-of-concept is a sequential model, with one input and output layer and four hidden layers: one convolution layer, one pooling layer, and two dense layers (**Figure 12**).

The input layer is a one-dimensional time series sensor data, comprising of motor vibrations along the X,Y,Z axis. In the convolution layer, the appropriate number and size of the convolution kernel performs one dimensional convolution operations. Using ReLU as the activation function, vanishing gradient problem is avoided and better computation performance is obtained. The max-pooling layer is appended to the convolutional layer and carries out a 2×1 max-pooling operation. In the flatten layer, the extracted features from output of the max-pooling convolution layer are extended to a one-dimensional vector. A dropout operation with a ratio set to 0.3 is executed after the first dense layer. It randomly selects and deletes neurons from the model in order to form a random subset of the neurons, solve the overfitting problem, and enhanced the generalization ability of the NN model. The output layer contains 4 neurons. Using softmax as the activation function, one normal behavior and three types of abnormal behavior are identified after training.

The model was implemented in Python using Keras, with the layers designed and added one-by-one. For the purpose of supervised learning, the sensor data was classified with one normal behavior and three types of abnormal behavior, for a total of four classes. The trained model was saved in an open-source HDF5/H5.h5 file (Hierarchical Data Format version 5).

The conversion into optimized C code was implemented with STM32 Cube AI, an extension of the CubeMX tool, which offers simple and efficient interoperability with ML frameworks. The state machine consists mainly of two states with two functions "init" and "process", respectively, with the former initializing the NN model and the latter being a continuously running function for collecting raw data from the sensors on board and making inferences in real-time.

The fusion of PdM application with AI/DL solution is shown in **Figure 13**.

An overview of the NN model is given in **Figure 12**, showing the required memory needed to store the model and how many multiply-and-accumulate (MAC) operations are required.

Various experiments were conducted with both TensorFlow Lite and Cube. AI on the same model as described above. The results show that the latter is faster and requires less flash and RAM than the former (**Figure 14**).

Summing up, for anomaly detection based on sensor measurements, such as vibrations, current consumption and temperature variations, ML algorithms (such as RF, SVM) are very efficient and easy to configure, and they can be trained on the target directly. Moreover, minimal amount of data input is needed to select the proper algorithm.

The results also showed that the combination of CNN with the optimization outperformed traditional ML techniques (RF, SVM). The model created by the AI/DL solution achieved accuracy rates as high as 98% for one of the datasets and 95% for the other. As with any other DL, the DL solution for PdM requires more investment than ML solutions.

The ML solution was implemented with NanoEdge AI Studio, whereas the DL solution was implemented with CUBE. AI. The former is a useful toolkit to test the ML solution and using proper input signal capturing and formatting and correctly selecting the generated libraries, it is possible to obtain useful results for the
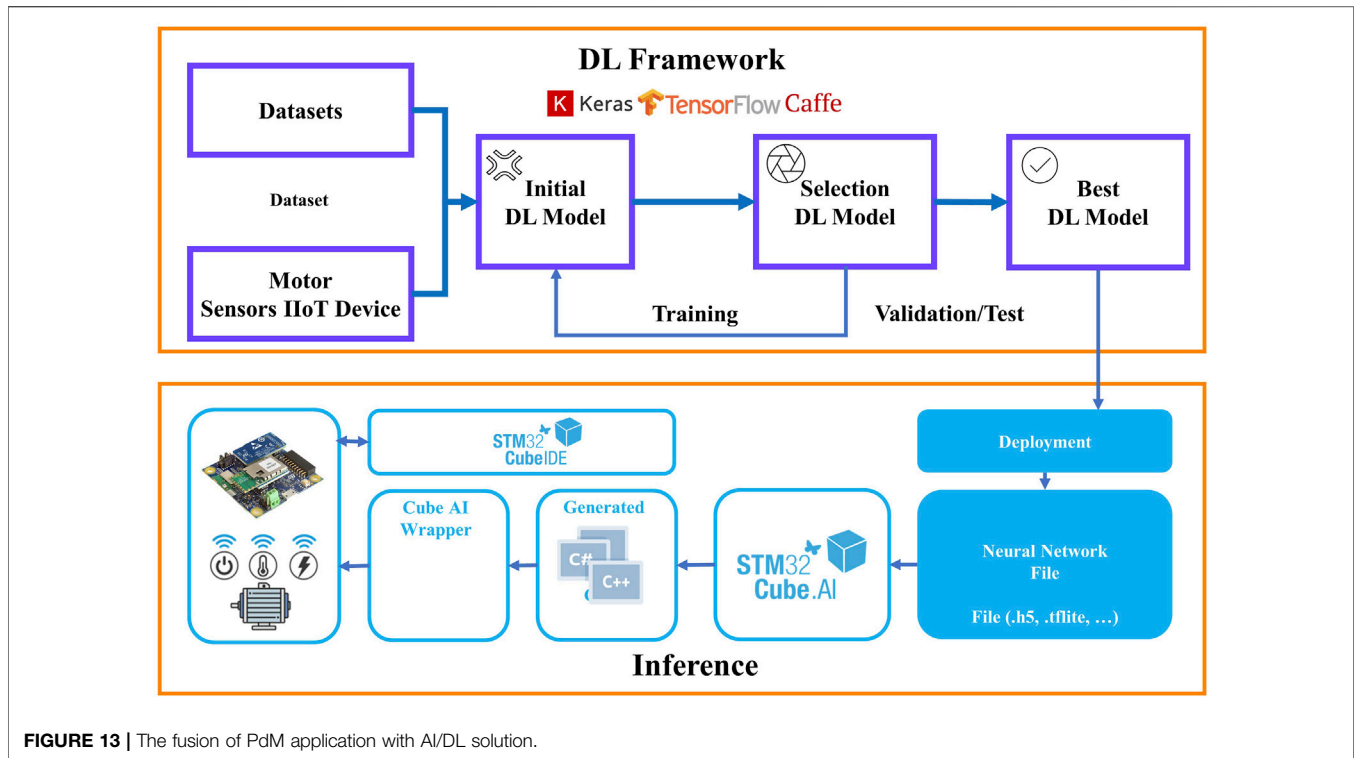
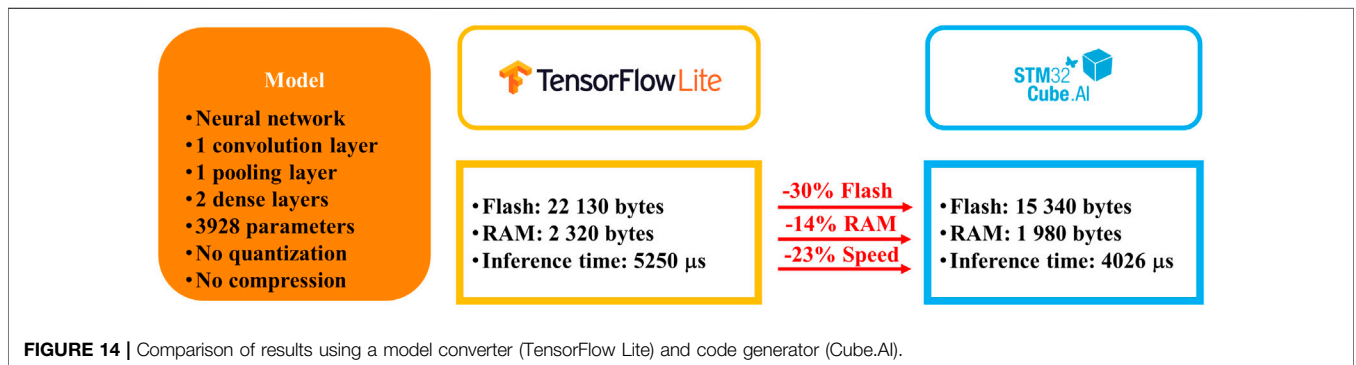**FIGURE 13 |** The fusion of PdM application with AI/DL solution.

FIGURE 14 | Comparison of results using a model converter (TensorFlow Lite) and code generator (Cube.AI).

PdM application. The latter has proven to be a useful toolkit for implementing the DL solution, due to interoperability with various ML frameworks.

In addition to anomaly detection, for the implementation of the DL solution, more features have been considered: *classification*, which enables automatic identification of a machine state among many different possible states and *extrapolation*, which uses mathematical regression models in order to estimate a target value using other known features. The classification library can be used on top of the anomaly detection library to determine the source of the problem after an anomaly has been detected. The extrapolation library can be used to predict future behavior. The insight and results will be valuable for the DL solution.

It is important to note that the above frameworks, although designed for the food processing industry, can be applied to other

manufacturing sectors, chemical engineering plants, or similar facilities involving motors and equipment that are in need of PdM.

6 FUTURE RESEARCH CHALLENGES IN INTELLIGENT MAINTENANCE SYSTEMS FOR INDUSTRIAL MANUFACTURING, CONTROL, AND DIAGNOSTICS

This section discusses various current open research and innovation topics and provides recommendations for future research directions on different issues related to intelligent maintenance systems for industrial manufacturing, control, and diagnostics.

Continued research on developing new algorithms for multi-sensors and multi-protocol techniques for data analytics using ML and DL is needed especially for PdM solutions that require considerable amounts of data to train extensive neural networks. This implies that the development of robust algorithms that use small-scale datasets that can provide similar accuracy as those that rely on more comprehensive datasets is necessary.

Developing techniques to enhance the scalability and model migration across devices in the edge continuum from micro-, deep-, and meta-edge is another important topic. These features are essential for a trained model to ensure enhanced applicability when applied to other devices at the edge, thus simplifying the PdM design process.

Another future research area is related to the dataset structure and the requirement to balance the feature of datasets used in PdM solutions using ML and DL algorithms. This issue is important because many challenges need to be addressed when handling mechanical and electrical data that are measured from the motors/equipment, such as the accuracy, availability, and quality. In this context, the balance between the observed data volume for normal operations of the motors/equipment that far exceeds the practical data volume for an anomalous operation is a critical issue. This imbalance between the two creates problems in model training, which requires more data under fault status scenarios to construct a more balanced complete dataset.

The significance of data to AI-based algorithm's performance is self-evident. In this context, further research is needed to implement less complex acquisition systems based on IIoT devices at the edge using datasets originating from the actual industrial operating motors/equipment in addition to the public datasets provided by a few open-source platforms. The use of *in situ* datasets improves the fine-tuning of the AI-based model and helps solve practical problems in industrial manufacturing more efficiently.

Unsupervised learning based on training the AI-based model with unlabeled data in PdM implementations is another significant research direction in the future. The existing AI solutions mainly focus on supervised learning that requires datasets to be uniquely labeled.

Further research is needed to study the efficiency of the AI-based models used in PdM applications, the benchmarking tools for evaluating these models, and the consequences of misdiagnosis.

The misdiagnosis and mitigation actions that address a certain degree of loss or breakdown of the system under maintenance are essential elements that must be evaluated to provide safe and reliable industrial PdM systems.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not readily available because policy restrictions. Requests related to datasets should be directed to OV (Ovidiu.Vermesan@sintef.no).

AUTHOR CONTRIBUTIONS

OV coordinated the structure and elaboration of the paper and the alignment and integration of individual contributions. All authors contributed ideas, content, perspectives, and references based on the work in different common projects and discussed the manuscript.

FUNDING

REFERENCES

AI4DI, Artificial Intelligence for Digitising Industry. Available online at: https://ai4di.eu/.

Avenas, Y., Dupont, L., Baker, N., Zara, H., and Barruel, F. (2015). Condition Monitoring: A Decade of Proposed Techniques. *IEEE Ind. Electron. Mag.* 9 (4), 22–36. doi:10.1109/mie.2015.2481564

Caesarendra, W., and Tjahjowidodo, T. (2017). A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and its Application for Degradation Trend Estimation of Low-Speed Slew Bearing. *Machines* 5 (4), 1–28. doi:10.3390/machines5040021

Carden, E. P., and Fanning, P. (2004). Vibration Based Condition Monitoring: A Review. *Struct. Health Monit.* 3 (4), 355–377. doi:10.1177/1475921704047500

Dai, X., and Gao, Z. (2013). From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis. *IEEE Trans. Ind. Inf.* 9 (4), 2226–2238. doi:10.1109/tii.2013.2243743

Dron, J. P., Bolaers, F., and Rasolofondraibe, I. (2004). Improvement of the Sensitivity of the Scalar Indicators (Crest Factor, Kurtosis) Using a De-noising Method by Spectral Subtraction: Application to the Detection of Defects in Ball Bearings. *J. Sound Vib.* 270 (1–2), 61–73. doi:10.1016/s0022-460x(03)00483-8

Gomez, A. R. (2020). Measurement and Analysis of Vibrations - Evaluation of the Criteria of Acceptance ISO Standard 10816-6. *Ship Sci. Technol. J.* 13 (26), 39–44. e-IIN: 2619-645X. doi:10.25043/19098642.196

Hashemian, H. M. (2011). State-of-the-Art Predictive Maintenance Techniques. *IEEE Trans. Instrum. Meas.* 60 (1), 226–236. doi:10.1109/tim.2010.2047662

Heckbert, P. S. (1998). Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. Available online at: http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf.

IEEE 802.11ax (2021). IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks-specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN. Available online at: https://standards.ieee.org/ieee/802.11ax/7180/.

IEEE 802.15.1, (2005). IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Specific Requirements - Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPAN). Available online at: https://www.en-standard.eu/ieee-802-15-1-2005-ieee-standard-for-information-technology-local-and-metropolitan-area-networks-specific-requirements-part-15-1a-wireless-medium-access-control-mac-and-physical-layer-phy-specifications-for-wireless-personal-area-networks-wpan-2/.

ISO 20816-1 (2016). Mechanical Vibration - Measurement and Evaluation of Machine Vibration - Part 1: General Guidelines. Available at: https://www.iso.org/standard/63180.html.

ISO/DIS 20816-3 (2022). Mechanical Vibration - Measurement and Evaluation of Machine Vibration - Part 3: Industrial Machinery with a Power Rating above 15

kW and Operating Speeds between 120 Min^{-1} and 30,000 Min^{-1}. Available online at: https://www.iso.org/standard/78311.html?browse=tc.

ISO 2954 (2012). Mechanical Vibration of Rotating and Reciprocating Machinery - Requirements for Instruments for Measuring Vibration Severity. Available online at: https://www.iso.org/standard/21835.html.

Javed, K., Gouriveau, R., Zerhouni, N., and Nectoux, P. (2015). Enabling Health Monitoring Approach Based on Vibration Data for Accurate Prognostics. *IEEE Trans. Industrial Electron.* 62 (1), 647–656. doi:10.1109/tie.2014.2327917

Jimenez, J. J. M., Schwartz, S., Vingerhoeds, R. A., Grabot, R. A. B., and Salaün, M. (2020). Towards Multi-Model Approaches to Predictive Maintenance: A Systematic Literature Survey on Diagnostics and Prognostics. *J. Manuf. Syst.* 56, 539–557. Elsevier. doi:10.1016/j.jmsy.2020.07.008

Jin, X., Zhao, M., Chow, T. W. S., and Pecht, M. (2014). Motor Bearing Fault Diagnosis Using Trace Ratio Linear Discriminant Analysis. *IEEE Trans. Industrial Electron.* 61 (5), 2441–2451. doi:10.1109/tie.2013.2273471

Jing, L., Zhao, M., Li, P., and Xu, X. (2017). A Convolutional Neural Network Based Feature Learning and Fault Diagnosis Method for the Condition Monitoring of Gearbox. *Measurement* 111, 1–10. doi:10.1016/j.measurement.2017.07.017

Kiangala, K. S., and Wang, Z. (2020). An Effective Predictive Maintenance Framework for Conveyor Motors Using Dual Time-Series Imaging and Convolutional Neural Network in an Industry 4.0 Environment. *IEEE Access* 8, 121033–121049. doi:10.1109/access.2020.3006788

ONNX. Available online at: https://onnxruntime.ai.

PyTorch. Available online at: https://pytorch.org/mobile/home/.

Ran, Y., Zhou, X., Lin, P., Wen, Y., and Deng, R. (2019). A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *IEEE Commun. Surv. Tutorials* 20, 1–36. doi:10.48550/arXiv.1912.07383

Selcuk, S. (2016). Predictive Maintenance, its Implementation and Latest Trends. *Proc. Institution Mech. Eng. Part B J. Eng. Manuf.* 231 (9), 1670–1679. doi:10.1177/0954405415601640

Sensor Tile Wireless Industrial Node (2022). STEVAL-STWINKT1 - SensorTile Wireless Industrial Node Development Kit and Reference Design for Industrial IoT Applications. Avalable online at: https://www.st.com/en/evaluation-tools/steval-stwinkt1.html.

Seryasat, O. R., Shoorehdeli, M. A., Honarvar, F., and Rahmani, A. (2010). "Multi-fault Diagnosis of Ball Bearing Using FFT, Wavelet Energy Entropy Mean and Root Mean Square (RMS)," in 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10-13 Oct. 2010, 4295–4299. doi:10.1109/ICSMC.2010.5642389

Silva, W., and Capretz, M. (2019). "Assets Predictive Maintenance Using Convolutional Neural Networks," in 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Toyama, Japan, 8-11 July 2019, 59–66. doi:10.1109/SNPD.2019.8935752

Singh, V. (2021). Skewness in Statistics - Overview, Concepts, Types, Measurements and Importance. Available online at: https://www.naukri.com/learning/articles/skewness-in-statistics-overview-concepts-types-measurements-importance/.

TensorFlow. Available online at: https://www.tensorflow.org/lite.

Van Eys, J. E. (2012). *Manual of Quality Analyses for Soybean Products in the Feed Industry.* 2nd edition. France: GANS Inc. Available online at: https://ussec.org/wp-content/uploads/2012/09/Manual-of-Quality-Analyses-2nd-edition.pdf.

Vermesan, O., and Bacquet, J. (2020). *Internet of Things – the Call of the Edge - Everything Intelligent Everywhere.* Gistrup: River Publishers. Available online at: https://riverpublishers.com/pdf/ebook/RP_E9788770221955.pdf. doi:10.13052/rp-9788770221955

Vermesan, O. (2020). *AI for Digitizing Industry Road-Mapping Requirements.* AI4DI ECSEL H2020 Project Report. Available online at: https://ai4di.eu/.

Vishwakarma, M., Purohit, R., Harshlata, V., and Rajput, P. (2017). "Vibration Analysis Condition Monitoring for Rotating Machines: A Review," *Mater. Today Proc.*, Vol. 4, N0. 2, Part A, 2659-2664, ISSN 2214-7853. doi:10.1016/j.matpr.2017.02.140

Wang, T., Lu, G., and Yan, P. (2020). A Novel Statistical Time-Frequency Analysis for Rotating Machine Condition Monitoring. *IEEE Trans. Industrial Electron.* 67 (1), 531–541. doi:10.1109/tie.2019.2896109

Zhu, C., Huang, H., Liu, H., Song, C., Ma, F., and Liu, Z. (2014). "On-line Vibration Monitoring and Diagnosing of a Multi-Megawatt Wind Turbine Gearbox," in *Philippe Velex, International Gear Conference 2014: 26th -28th August 2014* (Lyon: Chandos Publishing), 1089–1098. ISBN 9781782421948. doi:10.1533/9781782421955.1089