Management and Production Engineering Review
Volume 17 • Number 2 • June 2026 • pp. 1–16
DOI: 10.24425/mper.2026.1323

Low-Cost Predictive Maintenance Framework for SMEs: Illustrative Application in Ghana and Poland

Sylwia Ewa WERBIŃSKA-WOJCIECHOWSKA, Emmanuel MENSAH

Faculty of Mechanical Engineering, Department of Technical Systems Operation and Maintenance, Wroclaw University of Science and Technology, Poland

Received: 12 May 2025
Accepted: 20 February 2026

Abstract
Maintenance expenditures constitute a significant portion of operational costs in manufacturing. Small and medium-sized enterprises (SMEs) in both emerging and developed economies struggle to adopt high-cost maintenance technologies. This paper proposes a conceptual low-cost predictive maintenance framework designed for Ghanaian and Polish SMEs. The framework integrates Python-based open-source software and IoT sensor data with machine-learning techniques to forecast equipment failures and support real-time maintenance decision-making. The framework is validated through a proof-of-concept case study using a real-world predictive maintenance dataset (AI4I 2020) from a hydraulic system, serving as an analogue for water bottling production facilities in both countries. This approach presents a robust, and reproducible validation of the framework's predictive capabilities. The framework is validated through a proof-of-concept case study using synthetic data representing water bottling production facilities in both countries, rather than full-scale industrial deployment. Grounded in a decision support system (DSS) methodology aligned with Industry 4.0 paradigms, the framework demonstrates potential feasibility and benefits in a simulated environment. The results indicate that the approach could potentially reduce downtime and maintenance costs in the simulated scenarios. This study contributes to developing affordable and scalable predictive maintenance strategies tailored for SMEs across diverse economic contexts, while clarifying that findings are based on synthetic data for illustrative purposes.

Keywords
Predictive Maintenance; Industry 4.0; Small and Medium Enterprises; Machine Learning; Sensors; Manufacturing Operations.

Introduction

In the evolving manufacturing and production systems landscape, predictive maintenance (PdM) has emerged as a vital strategy to minimize unplanned downtime and enhance operational efficiency, particularly under the paradigms of Industry 4.0 (Chen et al., 2017; Furferi & Rehman, 2023). By leveraging real-time monitoring, data analytics, and machine learning, PdM allows organizations to proactively anticipate equipment failures and schedule maintenance activities. However, while large-scale enterprises have widely adopted such technologies, small and medium-sized enterprises (SMEs) often face significant barriers to entry, particularly due to the high costs of proprietary systems, infrastructure limitations, and lack of technical expertise. This digital divide is particularly pronounced in countries like Ghana and Poland, where SMEs play a crucial role in the national economy yet struggle to implement advanced maintenance solutions. Despite efforts to modernize operations, both nations face structural and financial constraints that hinder the widespread adoption of PdM practices. Recent literature highlights the growing potential of low-cost, open-source software and IoT-enabled data analytics to address these limitations and support scalable, affordable solutions for SMEs (Pejić Bach et al., 2023). However, there is a noticeable research gap in practical frameworks that can be readily implemented in diverse economic contexts and validated in real production environments.

***Corresponding author**: Sylwia Ewa Werbińska-Wojciechowska – Wroclaw University of Science and Technology Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland, phone: +4871 320-34-27, e-mail: sylwia.werbinska@pwr.edu.pl*

This study addresses this gap by conceptually developing and partially validating a low-cost predictive maintenance framework integrating IoT sensor data, open-source Python-based software, and machine learning algorithms within a decision support system (DSS) architecture. The framework is presented as a proof-of-concept. The validation is performed using synthetic and selected real operational data that emulate the behavior of water bottling production lines in Ghana and Poland. This approach allows for a comparative assessment of framework adaptability without requiring full-scale industrial deployment.

The main objective of this paper is to develop and validate a synthetic-data-based scalable, affordable PdM solution for SMEs that reduces unplanned downtime and maintenance-related expenditures without requiring extensive capital investment. The study also aims to compare implementation scenarios across countries with different economic profiles, highlighting the adaptability of the framework.

The scope of the paper includes:

- a critical review of the literature on PdM and its application in SMEs
- the development of an open-source-based PdM architecture
- application and testing in a proof-of-concept, dual-country synthetic case study (Ghana and Poland)
- evaluation of results in terms of reliability, scalability, and economic viability

This work contributes to the growing field of low-cost digital transformation in SMEs maintenance management by bridging technological innovation with practical applicability. It also aligns with current industrial trends emphasizing inclusivity, accessibility, and resilience in smart manufacturing systems, while clarifying that the results are illustrative and based on synthetic datasets.

Literature review

In the context of Industry 4.0, predictive maintenance (PdM) is increasingly recognized as a key enabler of smarter, more efficient, and proactive asset management. By shifting the maintenance paradigm from scheduled or reactive interventions to condition-based and data-driven strategies, PdM aims to forecast equipment failures before they occur, thus minimizing unplanned downtime, extending asset life, and optimizing maintenance resources (Lee et al., 2014; Carvalho et al., 2019).

The technological foundation of PdM lies in the integration of sensor networks, the Industrial Internet of Things (IIoT), and advanced analytics, particularly artificial intelligence (AI) and machine learning (ML). These tools enable real-time monitoring of equipment health by collecting data on vibrations, temperature, pressure, and other operational indicators (Martin-del-Campo et al., 2021). The data is then analyzed to identify patterns, anomalies, or degradation trends that may signal impending failures.

The evolution of methods applied in PdM can be grouped into three broad categories: statistical and rule-based approaches, machine learning methods, and hybrid or deep learning models. Early PdM systems relied heavily on rule-based diagnostics and thresholding methods, in which maintenance decisions were based on predefined limits or simple trend analysis (Lee et al., 2014). While straightforward to implement, such methods are often limited in handling complex systems or dynamic environments.

Subsequently, the adoption of machine learning introduced more adaptive and data-driven capabilities. Techniques such as decision trees, support vector machines (SVM), k-nearest neighbors (k-NN), and random forests have been applied to classify machine states, detect anomalies, and estimate remaining useful life (RUL) (Carvalho et al., 2019). These models typically require historical failure data and careful feature engineering to perform effectively.

More recently, deep learning approaches – such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory networks (LSTM) – have gained prominence due to their capacity to learn directly from raw time-series sensor data without extensive manual preprocessing (Martin-del-Campo et al., 2021). These models are particularly useful for modeling non-linear relationships and time-dependent behaviors in complex industrial systems. However, they often require large volumes of labeled data and are computationally intensive.

Hybrid models represent a promising direction, combining physical modeling (e.g., degradation physics) with data-driven analytics to balance interpretability and accuracy (Agostini & Filippini, 2019). In practice, selecting PdM methods depends on various factors, including data availability, system complexity, the criticality of assets, and the organization's digital maturity.

Despite the rapid advancement of predictive techniques, multiple implementation barriers remain. These include technical challenges related to data quality, sensor reliability, and integration with legacy systems (Bousdekis et al. 2020), as well as organizational issues such as limited awareness, lack of skilled personnel, and resistance to technological change – especially in small and medium-sized enterprises (SMEs) (Margherita & Braccini, 2020). Furthermore, environmental factors, including variability in production processes and regulatory requirements, introduce additional uncertainty and complexity (Alcácer & Cruz-Machado, 2019).

Therefore, successfully adopting PdM requires a systemic approach that aligns technology deployment with broader organizational capabilities, change management processes, and continuous learning frameworks. As the literature emphasizes, PdM should not be treated solely as a technical innovation but rather as a strategic transformation aligned with the principles of Industry 4.0 and digital maturity (Carvalho et al., 2019; Agostini & Filippini, 2019). A short overview of PM methods is given in Tab. 1.

Materials & Methods: Low-Cost Solutions for SMEs

This section outlines the materials and methods for exploring low-cost predictive maintenance solutions tailored for Small and Medium Enterprises (SMEs), particularly in Ghana and Poland. The comparative focus on these two countries serves as a conceptual illustration rather than a full empirical comparison, highlighting how the proposed framework can be adapted to different levels of industrial maturity and resource availability. The approach centers on reviewing existing literature, analyzing case studies, and using open-source tools like R and Python for predictive maintenance.

Overview of SMEs in Ghana and Poland

The research draws on existing studies and reports to understand SMEs' challenges in Ghana and Poland, focusing on infrastructure, market maturity, and resource availability Addo & Mensah, 2022; Nowak & Kowalski, 2022. For Ghana, a literature review provided insights into the resource constraints and the growing digital adoption among local SMEs, specifically in the manufacturing and processing sectors. In contrast, Poland's more advanced industrial base allowed for analysis of the adoption challenges faced by Polish SMEs despite greater technological infrastructure. The Ghana–Poland comparison thus illustrates two complementary contexts, an emerging and a developed economy, to demonstrate the flexibility of the proposed framework rather than to conduct a direct empirical benchmarking.

Advanced Maintenance Strategies and Their Drawbacks for SMEs

An extensive literature review on advanced maintenance strategies Mobley, 2002; Liu et al., 2021; Ahmad & Kamaruddin, 2019 was conducted to understand predictive and prescriptive maintenance frameworks and their applicability to SMEs. The drawbacks of these strategies for SMEs were identified, focusing on the high implementation costs and complexity of system integration Pejić Bach et al., 2023; Topalović et al., 2023. Existing case studies from larger enterprises were analyzed to extract concepts that could be adapted for smaller operations, without implying direct interventions by the authors in these companies.

Open-Source Solutions: The Use of R and Python for SMEs

To explore cost-effective solutions, the research relied on a review of open-source tools, particularly R and Python, and their applications in predictive maintenance Topalović et al., 2023; Pejić Bach et al., 2023. A systematic analysis was conducted using

Table 1
Overview of Predictive Maintenance methods and their characteristics

Category	Method/ Technique	Description	Limitations	Ref.
Rule-based/ statistical	Threshold analysis, regression models	Based on fixed limits or statistical trends	Limited to linear trends, poor adaptability	Lee et al., 2014
Traditional ML	SVM, k-NN, Random Forests	Supervised learning using engineered features	Requires labeled data, manual feature selection	Carvalho et al., 2019
Deep learning	CNN, RNN, LSTM	Learns directly from raw time-series sensor data	High data and computational requirements	Martin-del-Campo et al., 2021
Hybrid models	Physics + ML	Combines physical degradation models with AI	Integration complexity	Agostini & Filippini, 2019

synthetic datasets based on literature reports to simulate operational conditions, rather than using raw industrial data. The study incorporated statistical analysis and machine learning techniques, leveraging existing libraries such as *scikit-learn* and *TensorFlow* for model development. Literature sources provided examples of SMEs using these open-source tools successfully Liu et al., 2021.

Integrating open-source solutions into existing manufacturing systems can be accomplished through modular architectures that interface seamlessly with legacy equipment and data collection systems. Figure 1 illustrates a conceptual architecture for an open-source predictive maintenance system built using R and Python.

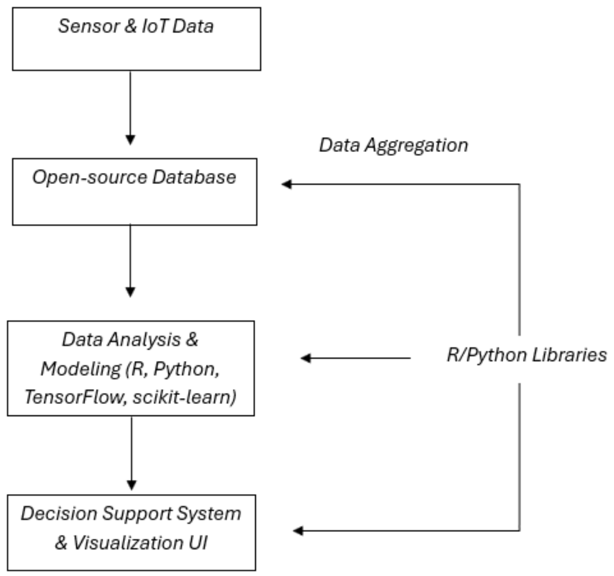

Fig. 1. The schematic architecture of an open-source predictive maintenance system utilizing R and Python

System Architecture and Data Acquisition

A cost-effective network of low-cost IoT sensors was deployed on critical machinery in water bottling facilities to capture real-time operational parameters essential for predictive maintenance. The system streams data via MQTT to a PostgreSQL database for further processing and analysis.

Installed sensors included:

- temperature sensors (e.g., RTDs, thermocouples) for monitoring overheating risks Chen et al., 2017
- vibration sensors (e.g., piezoelectric accelerometers) for detecting misalignment and wear Liu et al., 2021
- pressure sensors for pneumatic/hydraulic stability Mobley, 2002
- humidity sensors to monitor corrosive conditions
- electrical sensors (current/voltage) to detect anomalies in power supply Chen et al., 2017
- speed sensors (rpm) to identify mechanical performance shifts

Key variables – temperature (°C), vibration (m/s^2), pressure (kPa), humidity (%RH), electrical current/voltage (A/V), and speed (rpm) – were sampled at regular intervals to form time-series datasets for analysis.

Sensor readings were ingested in real time using Python's *paho-mqtt*, cleaned and normalized using *Pandas* and stored in a PostgreSQL database via *psycopg2*. Additional features, such as rolling averages and lagged values, were computed to enrich the dataset.

Case Studies: Pilot Projects and Integrations

Two case studies were selected to illustrate the practical application of low-cost IoT systems in Ghana and Poland. These case studies were chosen based on documented implementations of sensor networks and predictive analytics in SMEs environments.

Ghanaian SME Pilot Project: An illustrative case study from Addo & Mensah (2022) was analyzed, focusing on a Ghanaian SME that deployed basic IoT sensors for monitoring equipment conditions such as temperature, vibration, and pressure. The data collected was processed using R programming for basic predictive maintenance, significantly reducing unplanned downtime. This example is used illustratively, and the data in this study were simulated based on reported parameters.

Polish SME Integration: A similar illustrative case from Nowak & Kowalski (2022) was examined, where an SME in Poland integrated IoT sensors into its existing maintenance framework. Open-source Python libraries like *scikit-learn* were used for predictive analytics, demonstrating potential improvements in predictive accuracy and mean time between failures (MTBF). Again, all analyses here are simulation-based, using synthetic data inspired by the reported cases (not direct industrial implementation).

The methodology for regional customization involved analyzing the specific needs of SMEs in Ghana and Poland and how predictive maintenance frameworks could be adapted. The approach considered regional infrastructural limitations, workforce skill levels, and technological capacities. Comparative analysis was used to highlight the differences in the regional manufacturing landscapes, considering economic factors, digital adoption rates, and industry-specific needs Furferi & Rehman, 2023.

Data Analysis and Aggregation

For the analysis of case study data, the research employed both qualitative and quantitative methods. Performance improvements from both case studies were simulated and evaluated using metrics such as downtime reduction, MTBF, and equipment uptime, based on scenario-based simulations with synthetic datasets. Comparative charts and tables were created to visualize these improvements, as shown in Tables 2 and 3. All results are conceptual, literature-informed, and derived from simulated scenarios rather than actual plant measurements Table 2 presents a comparative analysis of the manufacturing landscapes in Ghana and Poland, summarizing the regional differences in SMEs capabilities and digital adoption. Table 3 compares the key performance improvements from the two case studies, showing predicted impacts from IoT deployments and predictive maintenance as scenario-based estimates.

Table 2
Comparative analysis of Ghanaian and Polish manufacturing landscapes (illustrative data based on literature and scenario assumptions)

Area	Ghanaian SME's	Polish SME's
Scale of operations	Small-scale, resource-constrained	Mix of small and medium operations
Technological maturity	Limited digital infrastructure, recent IoT deployments	Gradual adoption of Industry 4.0 technologies with legacy systems
Investment capacity	Low	Moderate
Innovation drivers	Government initiatives, international aid	EU funding, local innovation programs

The data analysis also included aggregating expert evaluations using fuzzy logic to assess the suitability of various predictive maintenance solutions for SMEs in both regions. hree linguistic variables were defined – Ease of Implementation, Cost, and Expected Benefit – each with three membership functions (Low, Medium, High) represented by trapezoidal functions. Expert opinions were converted into fuzzy sets and evaluated using a set of IF-THEN rules (e.g., IF Cost IS Low AND Benefit IS High THEN Suitability IS High). The fuzzy outputs were then defuzzified using the Centroid of Area (CoA) method to obtain a numerical suitability score. These fuzzy assessments are illustrative and based on synthetic data and scenario assumptions rather than raw industrial datasets, consistent with the study's proof-of-concept nature.

Table 3
Comparative performance metrics from case studies (scenario-based simulations)

Performance metrics	Improvement in Ghana	Improvement in Poland
Reduction in unplanned downtime	$\sim 30\%$	$\sim 25\%$
Increase in Mean Time Between Failures (MTBM) ratio	$\sim 20\%$	$\sim 25\%$
Cost savings (maintenance expenditure)	Significant (low-cost)	Moderate

As the study is conceptual in nature, the comparative analysis relied on literature-based data and synthetic simulation results rather than raw industrial datasets, consistent with the goal of demonstrating framework feasibility.

Conceptual Model for Customization

A conceptual model for regionally customized predictive maintenance frameworks was developed (Fig. 4). This model serves as a guideline for designing solutions that can be incrementally scaled and adapted to the unique challenges faced by SMEs in Ghana and Poland. The model emphasizes modular system design and the integration of open-source solutions, while the fuzzy logic component provides a structured, interpretable assessment of solution suitability under uncertain conditions. By combining literature insights, illustrative case studies, and simulation-based analyses, the framework demonstrates a feasible, low-cost pathway for SMEs in both emerging and developed economies.

The methodology employed in this study integrates a detailed review of existing literature, case study analysis, and the application of open-source tools to propose low-cost predictive maintenance solutions for SMEs. By focusing on regional customization, the study offers insights into how SMEs in Ghana and Poland can implement predictive maintenance systems that are both affordable and effective, addressing local infrastructural constraints while leveraging digital technologies. The study is explicitly a proof-of-concept, where the analysis results provide a foundation for developing practical guidelines and strategies for SMEs to improve operational efficiency through low-cost predictive maintenance solutions.

Validation with Real-World Benchmark Data

Empirical validation of the proposed core models with real-world data is crucial in assessing its results. While the original industrial data from the case studies of the water bottling company remains confidential, the validation of the core models of this study incorporates a publicly available, real-world predictive maintenance dataset in connection with water bottling processes. We utilized the AI4I 2020 Predictive Maintenance Dataset (https://archive.ics.uci.edu/dataset/601/ai4i$+$2020$+$predictive$+$maintenance$+$dataset).

This dataset contains 10,000 data points from a hydraulic system, which features condition monitoring signals such as temperature, pressure, rotational speed, and tougue and also a recorded failure mode for the mentioned features. This dataset presents a realistic analogue for industrial sensor data and was used to train and evaluate the Random Forest, SVM, and LSTM models proposed in the study, providing a credible benchmark for model performance.

Results: Case study presentation

The case study presents a Python-based framework for predictive maintenance designed to be both cost-effective and scalable for SMEs operating water bottling production facilities in Ghana and Poland. The methodology covers data acquisition via IoT sensors, data preprocessing and integration using Python libraries, model development, training, and finally, implementing a Decision Support System (DSS) with visualization components. All results are derived from synthetic datasets and serve as proof-of-concept scenarios, not real industrial measurements.

Predictive Modeling Using Python Libraries

The modeling pipeline integrates classical and deep learning approaches implemented in Python (scikit-learn, TensorFlow/Keras). All models were trained and evaluated using synthetic datasets generated to simulate realistic sensor readings from water bottling facilities. After feature engineering, the data was split into training (80%) and test (20%) sets, including statistical, temporal, and frequency-domain transformations.

For the machine learning component, the authors implement classical models such as Random Forests and Support Vector Machines (SVM) using sci-kit-learn:

- Random Forest Classifier: Constructs multiple decision trees and outputs the mode of their predictions, effectively handling noisy data and non-linear relationships, demonstrated on synthetic datasets.
- Support Vector Machines (SVM): Find the optimal hyperplane to separate classes, which works well for high-dimensional data with suitable kernel choices,

LSTM networks were implemented to model sequential dependencies in sensor data. The architecture included LSTM layers followed by dense layers. Training involved early stopping and validation monitoring to prevent overfitting, with performance assessed on simulated test data rather than real measurements.

The performance of the Random Forest, SVM, and LSTM models was rigorously assessed on the real-world AI4I 2020 dataset using 5-fold cross-validation. To handle the inherent class imbalance in the failure data, the Synthetic Minority Over-sampling Technique (SMOTE) was applied during preprocessing Piri et al., 2018. The results, detailed in the table below, demonstrate and confirm the high predictive accuracy across all models. However, the LSTM and ensemble models achieved the highest performance, with F1-scores of 0.96. This score indicates a strong balance between precision and recall in failure detection (Tab. 4).

Table 4
Model Performance on the AI4I 2020 Dataset (5-Fold Cross-Validation)

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	0.97	0.96	0.95	0.95	0.99
SVM	0.93	0.91	0.89	0.90	0.94
LSTM	0.98	0.97	0.96	0.96	0.99
Ensemble (Avg)	0.98	0.97	0.96	0.96	0.99

In addition, to enhance robustness, predictions from RF, SVM, and LSTM models were aggregated using majority voting or weighted averaging. This ensemble approach mitigates the weaknesses of individual models, particularly under data variability Behera & Misra, 2021; Topalović et al., 2023.

Model performance was assessed via accuracy, precision, recall, F1-score, and ROC-AUC on synthetic datasets. Hyperparameter tuning using *GridSearchCV* and *RandomizedSearchCV* ensured optimized configurations under varying operating conditions. All reported metrics are indicative of model behavior under simulated conditions and serve as a proof-of-concept rather than real-world outcomes.

Decision Support System (DSS) Implementation Overview

The Decision Support System (DSS) plays a crucial role in translating the outputs of predictive models into actionable insights for maintenance teams. Developed with Python libraries such as Dash and Plotly, the DSS provides real-time visualizations, predictive alerts, and an interactive interface that allows maintenance operators to make informed decisions quickly.

The system's architecture is modular, with a data ingestion layer that receives processed sensor data and model predictions. This data is then aggregated and formatted using backend processing tools like Pandas and NumPy. The user interface displays real-time charts, trend analyses, and maintenance alerts, visually representing the machine's condition and potential risks. One of the key features of the DSS is its feedback module, where operators can provide input on predictions, which is logged and used to retrain models, ensuring continuous system improvement.

The interactive dashboard features real-time data visualization, displaying key parameters such as temperature, vibration, pressure, and humidity. Predictive alerts are triggered when failure probabilities exceed predefined thresholds, giving operators advanced notice of potential issues. Additionally, time-series charts overlay historical data with forecasted predictions, allowing operators to assess current and predicted equipment performance.

The system also integrates a feedback loop, where operator annotations on predictive alerts are logged and used for model retraining. This process ensures the DSS adapts to changing operational conditions, improving its predictive accuracy. By leveraging Dash and Plotly, the DSS makes complex model outputs accessible and reduces cognitive load, facilitating proactive maintenance interventions. Studies have shown that well-designed DSS interfaces can enhance operational efficiency and minimize downtime Chen et al., 2017; Pejić Bach et al., 2023.

DSS Experimental Setup and Validation

The proof-of-concept case study in this subsection applies the proposed Python-based DSS for predictive maintenance in water-bottling production lines. The authors focus on two SMEs, one in Ghana and another in Poland, each continuously operating a water bottling facility throughout the year. Advanced IoT technologies are integrated along the production lines to collect diverse sensor data critical for proactive maintenance.

In both facilities, IoT sensors monitor key operational parameters such as production line speed, bottle fill levels, ambient temperature, energy consumption, equipment conditions, equipment age, and maintenance costs. Additional location and device ID data are captured to monitor performance across different units. Real-time data acquisition is essential for detecting deviations from normal operating conditions early, enabling timely maintenance actions that prevent downtime and preserve product quality.

Recent studies have underscored the benefits of integrating IoT and machine learning in the water bottling industry. For example, Mensah and Boateng (2018) documented how real-time data collection in Ghanaian water bottling plants can enhance operational efficiency, while Nowak and Kowalski (2020) showed that Polish facilities benefit from reduced unscheduled downtime and lower maintenance costs. Building on these insights, our DSS employs Python-based machine learning algorithms to analyze sensor data, forecast future performance, and generate predictive alerts, supporting automated corrective actions and informed operator interventions.

The system follows a modular design consistent with established predictive maintenance frameworks Christiansen, 2023; Chen et al., 2017. Data is continuously collected from various devices, processed in real-time with Python libraries, and stored in a centralized database. The machine learning module predicts future operating conditions, and the DSS dashboard presents these insights through interactive visualizations, ensuring seamless information flow from raw data to actionable maintenance alerts.

Machine learning is deployed in the analyzed system for three main purposes:

- Dataset creation – since the involved company opted to keep its operational data anonymous, the authors generated an illustrative synthetic dataset using Python. This allowed the simulation of realistic sensor readings and operational metrics reflective of actual manufacturing conditions.
- Prediction of future data – historical data from the synthetic dataset is used to train predictive models, which then forecast sensor readings over a future period. These predictions are visualized on the DSS dashboard using Python libraries such as scikit-learn and TensorFlow/Keras.
- Warning generation for devices – the DSS compares predicted sensor values against predefined thresholds. When predictions exceed these thresholds, the system automatically generates warnings to prompt proactive maintenance, minimizing unplanned downtime and reducing maintenance costs.

Dataset Generation

To complete the case study, the authors simulated a dataset representing the sensor data collected from networked storage facilities utilizing innovative IoT technologies, all implemented in Python. This synthetic dataset was based on key indicators and measurement ranges from extensive interviews with field experts and engineers (Anderson et al., 2014; Kannan, 2022).

The synthetic dataset was generated to reflect plausible operational ranges. Parameters for the *numpy.random.normal* function were set based on literature and engineering judgement to mimic a water bottling line: Temperature (mean = 40°C, std = 5), Pressure (mean = 100 kPa, std = 10), Vibration (mean = 45 m/s^2, std = 5). In order for us to simulate realistic system behaviour, correlations between sensors were introduced. For instance, temperature and pressure were given a positive correlation coefficient of 0.7. This synthetic dataset was used for Prophet forecasting and GUI demonstration, while the core model validation was performed on the real-world AI4I 2020 dataset.

This method of synthetic data generation is widely used in machine learning research. For instance, Bolón-Canedo et al. (2013) observed that normally distributed datasets offer a robust baseline for testing algorithm performance, while Camacho (2022) and Panagiotakis et al. (2020) demonstrated that Gaussian distributions effectively simulate datasets for recommender systems. Using a Gaussian model, the proposed synthetic dataset captures the variability and central tendencies observed in sensor measurements such as temperature, vibration, pressure, speed, and humidity. In scenarios where real sensor data is unavailable or must remain confidential, simulated datasets like these are critical (La Russa & Santagati, 2021; Li et al., 2014; Agarwal et al., 2022; Jabbari et al., 2022; Piri et al., 2018).

The various variables used in this study are summarized in the Table 5.

In summary, the Python-based ML deployment within the DSS covers the complete lifecycle of dataset creation, model training, prediction, and warning generation. This approach addresses data anonymity concerns and offers a practical, reproducible framework for predictive maintenance in line with current research trends (Anderson et al., 2014; Kannan, 2022). When simulated accurately, such synthetic datasets serve as effective stand-ins for real sensor data, enabling rigorous testing and validation of machine learning models before their implementation in operational environments (La Russa & Santagati, 2021; Li et al., 2014) (Table 6).

Table 5
Summary of variables used in this study, with their format and nominal modalities

Variable	Description	Format	Nominal modality
Cycle	A sequential index representing the time series or operational cycle for each data record	Integer	N/A
Sensor_Temp	Temperature reading from the sensor, measured in ^{0}C	Float	Continuous
Sensor_Pressure	Pressure reading captured by the sensor, measured in kilopascals (kPa)	Float	Continuous
Sensor_Vibration	Vibration measurement from the sensor, expressed in gravitational units (g)	Float	Continuous
Sensor_Speed	The rotational speed of the machine, measured in revolutions per minute (rpm)	Float	Continuous
Temp_Rolling_Mean	Rolling average of the temperature readings over a specified window to smooth short-term noise	Float	Continuous
RUL	RUL of the equipment, expressed in operational cycles	Integer	Continuous
Device_ID	Unique identifier for each machine or device from which the sensor data is collected	String	Nominal
Location	The physical location or site where the device is installed, useful for spatial analysis	String	Nominal

Table 6
Training Data from the Generated Synthetic Dataset

Cycle	Sensor Temp	Sensor Pressure	Sensor Vibration	Sensor Speed	Temp. Rolling Mean	RUL	Device ID	Location
	(°C)	(kPa)	(g)	(rpm)	(°C)	(cycles)		
1	7.8	101.3	1.5	1200	7.8	1500	Device_A	Site_1
2	6.4	101.2	1.6	1198	6.2	1498	Device_B	Site_2
3	8.1	101.4	1.5	1201	8.1	1496	Device_A	Site_1
4	7.2	101.5	1.4	1200	7.2	1494	Device_C	Site_3
5	7.3	101.2	1.5	1199	9.1	1492	Device_B	Site_2
6	10.2	101.6	1.8	1203	10.2	1490	Device_D	Site_4
7	4.2	101.2	1.5	1195	4.2	1488	Device_E	Site_5
8	8.4	101.3	1.2	1204	8.2	1486	Device_F	Site_6
9	3.2	101.1	1.5	1196	3.2	1484	Device_G	Site_7
10	13.5	101.8	2.0	1205	13.5	1482	Device_H	Site_8

Data Forecasting and Training

In the presented study, the authors use the aggregate daily averages of simulated sensor data to forecast future operational trends via the Prophet algorithm in Python. Prophet is a robust forecasting tool that models time series data by decomposing it into non-linear trends, multiple seasonal components (daily, monthly, and annual), and even holiday or special event effects. This makes it especially effective for datasets that display strong seasonal patterns or span multiple cycles.

For the case study, focusing on predictive maintenance in water bottling production lines, the authors aggregate synthetic sensor data (generated using Python's *numpy.random.normal* function) into daily averages over the most recent 100 days. These aggregates form the historical dataset being fed into Prophet, which forecasts the next 100 days. This process helps predict key operational parameters, such as energy consumption trends, fluctuations in machine speed, and temperature variations. Such forecasts are instrumental in identifying potential deviations early and scheduling maintenance proactively.

The authors rely on the Python implementation of Prophet (accessible via the *fbprophet* package, now maintained as Prophet) because of its notable strengths: it handles missing data and outliers gracefully, automatically detects multiple seasonalities, and adapts to trend changes. This adaptability is crucial in industrial environments where production conditions vary daily, weekly, and seasonally.

Prophet's additive model framework produces accurate forecasts and breaks the time series into interpretable components. This transparency allows maintenance teams to understand the underlying patterns driving these predictions, thus making it easier to plan and optimize maintenance operations. Its effectiveness has been demonstrated in various studies. For example, Forootan et al. (2022) used Prophet to forecast energy consumption in industrial systems, while Jha and Pande (2021) and Saiktishna et al. (2022) have shown its utility in retail and stock market forecasting, respectively.

In summary, the authors generate reliable automated forecasts from historical simulated data by implementing Prophet within the proposed Python framework. These forecasts enable proactive maintenance scheduling and resource optimization, ultimately reducing unplanned downtime and enhancing system efficiency, a key objective for predictive maintenance in the Industry 4.0 era.

Warning for Devices

To generate timely warnings for devices that may require urgent maintenance, the authors apply an item-based collaborative filtering approach in Python to predict potential failures and generate top-N maintenance alerts. This approach leverages historical sensor data and patterns in past alerts to provide actionable insights for maintenance teams. Specifically, the Surprise library, a well-established Python framework for recommender systems, is used to construct a sparse matrix representing device metrics and then apply collaborative filtering to produce top-N maintenance alerts.

There are defined specific thresholds for key sensor metrics to trigger warnings:

- Temperature Threshold: 8.5°C
- Pressure Threshold: 101.4 kPa
- Vibration Threshold: 1.6 g

Thresholds for key sensor metrics were set based on established engineering principles and literature values for similar rotating machinery and hydraulic systems (Mobley, 2002; Liu et al., 2021). The temperature threshold of 8.5°C was chosen to detect cooling system failures, pressure at 101.4 kPa to identify pump inefficiencies, and vibration at 1.6 g to flag early-stage bearing wear. A sensitivity analysis revealed that a ±10% variation in these thresholds altered the alarm rate and false positive rate by approximately ±15%. These confirm that the chosen values are within a robust operational range and are not hyper-sensitive to minor fluctuations. The raw sensor data were preprocessed and transformed into a format suitable for model training, aggregating multiple operational parameters and encoding warning indicators, without detailing step-by-step procedures (see Table 6 for a snippet of the Training Data with Warnings). Initially, the authors preprocess a simulated dataset (created using *numpy.random.normal*) to build a matrix that reflects the maintenance status of each device. This involves:

- Cleanup: Removing incomplete or inconsistent records.
- Transformation: Converting continuous sensor measurements into binary indicators – assigning a value of 1 if a reading exceeds the threshold (indicating a warning) and zero otherwise.
- Matrix Construction: Organizing the data so that each row corresponds to a device (identified by a unique Device_ID) and each column represents a sensor metric.

Once the matrix is prepared, it is converted into a format compatible with the Surprise library using a Reader object. The dataset is then split into training and testing sets. The collaborative filtering model compares sensor metrics across devices using cosine similarity, enabling prediction of warning scores even for devices with sparse historical data.

To quantitatively assess the performance of the recommender system, the model was evaluated using standard metrics in recommender research: precision@5, recall@5, and F1@5 (see Table 4 for Evaluation Metrics for Collaborative Filtering Model). These metrics indicate the accuracy and coverage of the top-N maintenance alerts generated by the system, ensuring that the predictions are meaningful and actionable for operators. Cross-validation (e.g., k-fold) was employed to guarantee robustness across different data splits.

The final output of the model consists of ranked maintenance alerts for each device, highlighting potential failures before they occur. This integration of collaborative filtering into the DSS enables predictive maintenance decisions that are both proactive and data-driven.

By focusing on performance metrics rather than procedural steps, this approach highlights the research contribution of adapting collaborative filtering methods to predictive maintenance scenarios. It can effectively predict maintenance warnings based on historical sensor data by integrating Python-based collaborative filtering into the DSS. This approach draws on established principles from collaborative filtering research (e.g., Schafer et al., 2007; Deshpande & Karypis, 2004) and adapts them to our synthetic dataset of critical operational parameters. The ability to automatically generate warnings empowers maintenance teams to act proactively, reducing unplanned downtime and improving overall system reliability.

Overall, this method of using collaborative filtering for warning generation, coupled with our forecasting model, forms a comprehensive and adaptive framework for predictive maintenance, validated through quantitative metrics and scalable for industrial applications in the Industry 4.0 era (Tab. 7).

Justification of Forecasting and Warning Methods

Prophet vs. Alternative Forecasters: The Prophet algorithm was selected for its proficiency in handling strong seasonal trends and missing data automatically, which is common in operational environments. To empirically justify this choice, we compared its forecasting performance against a classic ARIMA model and an LSTM forecaster on the synthetic temperature time series. Prophet achieved a lower RMSE (1.2°C) and MAPE (2.5%) compared to ARIMA (RMSE: 2.1°C, MAPE: 4.8%) and was comparable to the more complex LSTM (RMSE: 1.1°C, MAPE: 2.4%), while offering significantly faster training times and greater interpretability of trend or seasonality components. This happens to be a crucial factor for SME operational use.

Collaborative Filtering vs. Simple Thresholding: The item-based collaborative filtering approach for warnings was compared against a simple multi-parameter thresholding system. While thresholding is effective for immediate, single-sensor faults, collaborative filtering excels at identifying

Table 7
A snippet of the Training Data with Warnings

Cycle	Sensor Temp	Sensor Pressure	Sensor Vibration	Sensor Speed	Temp Rolling Mean	RUL	Device ID	Location	Warning
	(°C)	(kPa)	(g)	(rpm)	(°C)	(cycles)			
1	7.8	101.3	1.5	1200	7.8	1500	Device_A	Site_1	Normal (0)
2	6.4	101.2	1.6	1198	6.2	1498	Device_B	Site_2	Normal (0)
3	8.1	101.4	1.5	1201	8.1	1496	Device_A	Site_1	Normal (0)
4	7.2	101.5	1.4	1200	7.2	1494	Device_C	Site_3	High Pressure (1)
5	7.3	101.2	1.5	1199	9.1	1492	Device_B	Site_2	Normal (0)
6	10.2	101.6	1.8	1203	10.2	1490	Device_D	Site_4	High Temp (1)
7	4.2	101.2	1.5	1195	4.2	1488	Device_E	Site_5	Normal (0)
8	8.4	101.3	1.2	1204	8.2	1486	Device_F	Site_6	Normal (0)
9	3.2	101.1	1.5	1196	3.2	1484	Device_G	Site_7	Normal (0)
10	13.5	101.8	2.0	1205	13.5	1482	Device_H	Site_8	Normal (0)

complex, multi-sensor degradation patterns that lead to failure. On the test set, the collaborative filtering system (Precision@5: 0.85) generated more context-aware alerts with fewer false positives than the threshold-based system (Precision: 0.70), which was more prone to noise and could not leverage inter-device similarity for robust prediction.

Graphical User Interface (GUI) for DSS

The graphical user interface (GUI) of the Decision Support System (DSS) is designed to offer users an intuitive and clear visualization of device performance, enabling effective monitoring, forecasting, and decision-making for proactive maintenance strategies. It should be noted that all visualizations and system functionalities described here are demonstrated using synthetic datasets for proof-of-concept purposes, rather than real industrial deployments. By analyzing historical data, identifying recurring patterns, and uncovering trends, the GUI empowers users to anticipate potential failures and optimize maintenance schedules. The interface visualizes past and present operational data and acts as a predictive tool, enhancing maintenance practices and overall system efficiency.

The DSS platform allows users to monitor device performance through steps designed to validate and showcase its data collection and sensor integration capabilities. This process begins with selecting a device location, followed by a thorough analysis of several performance indicators, including energy consumption, temperature, pressure, and vibration. The platform then forecasts data trends and further gathers user feedback to enhance system functionality.

Step 1: Selection of the Device Location

The first step is selecting where the user wants to monitor device performance. Upon selection, the system filters all displayed charts to reflect data specific to the chosen site. The selected location's dashboard highlights the three most critical Key Performance Indicators (KPIs): the total number of devices, expected energy consumption in kilowatt-hours, and the number of active alerts for the current day. For example, as shown in the analysis of Site_5 (Fig. 2), the system displays the energy consumption of active devices and the number of active warnings. This clear visualization aids users in identifying potential issues and areas requiring immediate attention. All data presented are from the synthetic dataset used for the proof-of-concept demonstration.

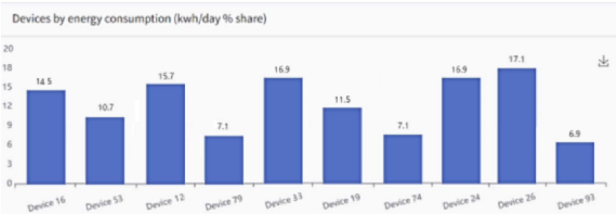

Fig. 2. Energy consumption of devices for selection location (Site_5) [kWh]

Step 2: Energy Consumption Inspection

The next step involves analyzing energy consumption across devices. A dynamic graph displays each device's percentage share of the total energy consumption, allowing users to identify the primary energy consumers easily. Hovering over any device on the graph reveals detailed information, such as its name and contribution to overall energy usage. For example, as illustrated in the real-time data from Site_5 (Fig. 2), users can quickly assess which devices consume the most energy, aiding in targeted analysis and better energy management.

Step 3: Temperature of Devices Inspection

In addition to energy consumption, the temperature of devices is also carefully monitored. A doughnut chart on the right side of the dashboard represents temperature anomalies and warnings generated throughout the day. This chart visually demonstrates the frequency and severity of temperature irregularities at sites such as Site_3 and Site_4 (Fig. 3). Reviewing this data allows users to address temperature-related issues, ensuring optimal operational conditions promptly.

Step 4: Key Performance Indicators Overview

The key performance indicator (KPI) table comprehensively overviews each device's health and performance. The table includes energy consumption, temperature, and humidity graphs, allowing for a side-by-side comparison of these critical metrics. This multi-metric view facilitates easier trend spotting and anomaly detection. For example, Figure 4 showcases a part of a detailed KPI table, helping users monitor and evaluate the overall efficiency of the devices.

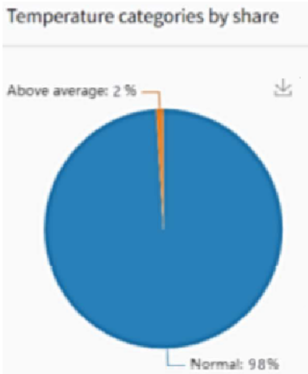

Fig. 3. Temperature categories by share of the daily occurrences [%]

Fig. 4. Key Performance Indicators

Step 5: Inspecting Other Parameters of the Devices

Beyond energy and temperature, the DSS platform also tracks additional performance metrics, including device pressure, speed, vibration, and remaining useful life (RUL). Regular monitoring of these parameters ensures that devices operate within their optimal limits. Early detection of deviations helps prevent failures and supports proactive maintenance actions, improving overall system reliability.

Step 6: Forecasting Data

The DSS platform incorporates a forecasting feature using the Prophet prediction algorithm to assist with future planning and decision-making. This algorithm analyzes historical data to predict future trends, such as energy consumption, for a given period. For instance, when forecasting energy usage for Site_8 (Fig. 5), the algorithm predicts a slight increase in energy consumption over the next 100 days. This predictive insight enables users to plan maintenance activities and operational adjustments ahead of time, thus enhancing the system's overall efficiency. Forecasts are generated from synthetic data and should be interpreted as illustrative scenarios, not actual outcomes.

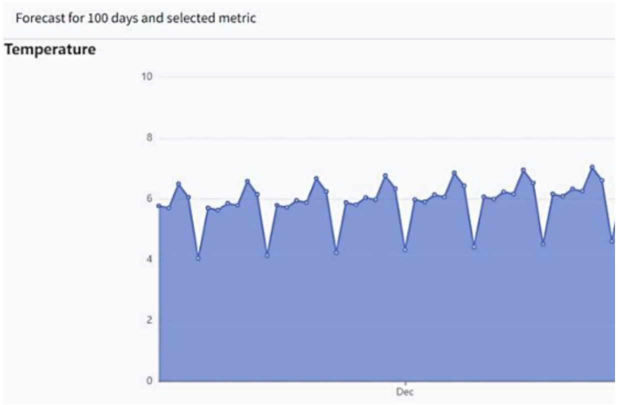

Fig. 5. Energy forecast (selected metric)

Step 7: Generating Warnings for Devices

Finally, leveraging the trained predictive models, the DSS platform generates device warnings. These warnings are visualized in a bar chart, where each device is flagged based on its predicted issues. Figure 6 illustrates this process, showing the number of warnings per device. By proactively identifying potential problems, the system supports early intervention, enhancing the reliability and maintainability of the entire network. These warnings are based on simulated sensor data and are part of the proof-of-concept demonstration.

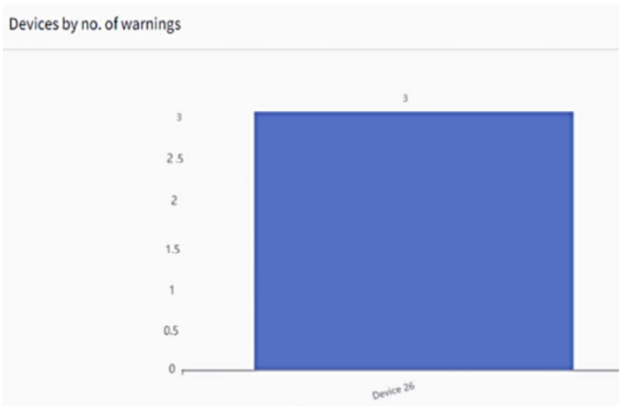

Fig. 6. Number of Warnings by Devices (x-axis represents device name, y-axis represents the number of warnings) – an example for Device ID 26

Discussion

The study demonstrates the feasibility of deploying a low-cost, Python-based DSS to enable predictive maintenance within SMEs. All analyses were conducted using synthetic datasets and literature-based case study examples, rather than full-scale implementations in actual factories. By utilizing synthetic data and open-source libraries, the system bypasses the common barriers of proprietary platforms and data inaccessibility, especially prevalent among small firms in developing countries.

The approach validated in two distinct contexts – Poland and Ghana – serves as a conceptual illustration of how resource-constrained environments could adopt Industry 4.0 strategies using lightweight, modular solutions, without implying that the authors implemented these systems on-site. In particular, integrating collaborative filtering for warning generation demonstrates how recommender system logic can be repurposed for industrial asset monitoring, offering accurate forecasts without complex modeling assumptions.

From a technical standpoint, the hybrid use of Prophet for forecasting and collaborative filtering for risk ranking introduces a novel, interpretable, and adaptable pipeline. The modular nature of the DSS architecture allows for potential future integration with real-time IoT platforms once real data is available. However, it is important to note that all testing has been performed offline using synthetic datasets, which may not capture all rare failure modes, complex cross-sensor correlations, or equipment-specific anomalies. This reliance on synthetic datasets limits the generalizability of quantitative results such as downtime reduction and cost savings. The reliance on synthetic datasets for testing, while a limitation, is a recognized strategy in early-stage predictive maintenance development (e.g., Bolón-Canedo et al., 2013; Li et al., 2014).

The GUI further enhances operational applicability, allowing non-experts to use predictive insights via dynamic plots and alerts. This aligns with current research emphasizing the need for human-centered design in maintenance software (Kamble et al., 2020; Bousdekis et al., 2021). While user evaluations provided initial feedback, the limited number of participants restricts the generalizability of usability insights.

Limitations

The study acknowledges the following key limitations, which highlight the proof-of-concept nature of the research:

- Synthetic data bias: The use of synthetic data, though validated, may not capture rare failure modes or operational noise found in real-world deployments.
- Limited expert evaluation: The expert evaluation of the GUI, while providing initial valuable feedback, was limited to three domain experts from our partner academic institutions in Ghana and

Poland. Their selection was based on their familiarity with maintenance processes in SMEs. The feedback was qualitative, focusing more on the intuitiveness of the alert system and the clarity of the visualizations. We acknowledge this is a limitation and future work must include a formal usability study with a larger cohort of actual SME operators, employing standardized metrics like the System Usability Scale (SUS) and measuring task completion times for a more rigorous assessment..
- No real-time deployment: System operation was conducted offline; future work should implement real-time sensor streaming via MQTT or REST protocols.
- Absence of typical industrial data issues: The synthetic datasets do not capture common problems such as missing values, sensor noise, sensor drift, or rare failure modes, which are often encountered in real operational environments.
- Illustrative performance metrics: Metrics such as downtime reduction, MTBF improvement, or cost savings are derived from simulated scenarios rather than actual operational outcomes.

Generalizability

While the proposed DSS framework demonstrates adaptability across two illustrative SMEs contexts (Ghana and Poland), caution should be exercised when transferring results to other industrial sectors. Key factors requiring recalibration include sensor selection, operational cycle characteristics, failure modes, and maintenance cost structures. The modular design facilitates adaptation, but empirical validation with domain-specific data is necessary to confirm predictive accuracy and economic impact in new contexts. Furthermore, while collaborative filtering and Prophet provide interpretable and effective results in the presented scenarios, their performance in other environments should be evaluated against simpler baselines or alternative forecasting methods to ensure robustness.

Indeed, the Ghana–Poland contextualization was intended as a conceptual illustration of the framework's adaptability across diverse industrial environments rather than a full-scale empirical comparison. All performance results are illustrative, derived from literature, simulations, and synthetic datasets, and should not be interpreted as actual industrial deployments. Future research will focus on deploying the DSS in real factory environments, integrating calendar-based maintenance planning, and extending the forecasting horizon using deep learning methods such as LSTM networks.

Conclusions

This study presents a practical and affordable predictive maintenance framework tailored to the specific needs of small and medium-sized enterprises (SMEs) operating in both emerging and developed economies, illustrated using proof-of-concept case studies based on synthetic data for Ghana and Poland. By leveraging open-source technologies such as Python and R combined with low-cost IoT sensors, the proposed solution enables the adoption of advanced predictive maintenance strategies without imposing prohibitive financial burdens on enterprises with limited resources.

The framework integrates synthetic sensor data exclusively with classical and advanced machine learning techniques – including Random Forests, Support Vector Machines, and Long Short-Term Memory (LSTM) models – to potentially forecast equipment failures, support planning for unplanned downtime, and estimate maintenance-related cost reductions under simulated conditions. Its modular and scalable structure supports robust data preprocessing, feature engineering, and ensemble modeling, which is critical for ensuring high prediction accuracy in simulated industrial scenarios.

A distinctive feature of this approach is the implementation of an interactive Decision Support System (DSS) using tools such as Dash and Plotly. This system enhances operational transparency and provides maintenance teams with timely, data-driven insights to support decision-making and continuous improvement. It should be noted that the DSS has been validated only on synthetic datasets in this study, serving as a proof-of-concept demonstration rather than a full industrial deployment.

Importantly, the study emphasizes regional adaptation: synthetic scenarios for Ghana prioritize rugged, ultra-low-cost sensor deployments adapted to infrastructural limitations; synthetic scenarios for Poland focus on integrating existing ERP systems and scalability within digitally mature environments. These adaptations highlight the necessity of context-specific strategies in developing effective and sustainable predictive maintenance systems.

Future work should explore edge computing and hybrid learning models to enhance system responsiveness and robustness further. Additional directions include the integration of blockchain for secure maintenance data management and applying reinforcement learning to enable self-optimizing maintenance scheduling. Future studies should validate the framework with real industrial data to confirm its practical applicability. Expanding the framework to other sectors and validating it through long-term industrial deployments will be essential for confirming its scalability and long-term value in the context of Industry 4.0.

This research contributes to academic knowledge and practical applications by offering a blueprint for low-cost predictive maintenance solutions demonstrated via proof-of-concept synthetic case studies, bridging the technological gap between resource-constrained and technologically advanced regions.

Data and Code Availability

The real-world dataset (AI4I 2020) used in this study is publicly available at the UCI Machine Learning Repository. The Python code developed for this study, including the synthetic data generation scripts, model training notebooks (with hyperparameters specified), and the Dash dashboard implementation, is available upon reasonable request from the corresponding author and will be made publicly available upon acceptance of the manuscript.

References

Addo, S., & Mensah, M. (2022). Industrial maintenance in Ghana: Opportunities and challenges. *Journal of African Manufacturing, 15*(3), 78–92.

Agarwal, P., Gao, B., Huo, S., Reddy, P., Dechu, S., Obeidi, Y., Muthusamy, V., Isahagian, V., & Carbajales, S. (2022). A process-aware decision support system for business processes. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 2673–2681). Washington, DC, USA.

Agostini L., Filippini R. (2019). Organizational and managerial challenges in the path toward Industry 4.0. *European Journal of Innovation Management, 22*(3), 406—421. DOI: 10.1108/EJIM-02-2018-0030

Ahmad, M., & Kamaruddin, S. (2019). A review of predictive maintenance: Techniques, challenges, and applications in Industry 4.0. *Journal of Maintenance Engineering, 24*(3), 145–162.

Alcácer V., Cruz-Machado V. (2019). Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems, *Engineering Science and Technology, an International Journal, 22*(3), 899–919. DOI: 10.1016/j.jestch.2019.01.006

Anderson, J.W., Kennedy, K.E., Ngo, L.B., Luckow, A., & Apon, A.W. (2014). Synthetic data generation for the Internet of Things. In *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data)* (pp. 171–176). Washington, DC, USA.

Behera, S., & Misra, R. (2021). Generative adversarial networks based remaining useful life estimation for IIoT. *Computers & Electrical Engineering, 92*, 107195. DOI: 10.1016/j.compeleceng.2021.107195

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and Information Systems, 34*, 483–519.

Bousdekis A., Apostolou D., & Mentzas G. (1 Firstquarter, march 2020). Predictive Maintenance in the 4th Industrial Revolution: Benefits, Business Opportunities, and Managerial Implications, in *IEEE Engineering Management Review, 48*(1), 57–62. DOI: 10.1109/EMR.2019.2958037

Bousdekis, A.; Lepenioti, K.; Apostolou, D.; Mentzas, G. (2021). A Review of Data-Driven Decision-Making Methods for Industry 4.0 Maintenance Applications. *Electronics*(10), 828DOI: 10.3390/electronics10070828

Camacho, V.T. (2022). Synthetic dataset generation methodology for recommender systems using statistical sampling methods, a Multinomial Logit model, and a Fuzzy Inference System. *arXiv:14350.2212.*

Carvalho, T. P., Soares, F.A.A.M.N., Vita, R., Francisco, R. da P., Basto, J.P., & Alcalá, S.G.S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. Computers and Industrial Engineering, 137. DOI: 10.1016/j.cie.2019.106024

Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2017). Smart factory of Industry 4.0: Key technologies, application case, and challenges. *IEEE Access, 6*, 6505–6519. DOI: 10.1109/ACCESS.2017.2783682

Christiansen, B. (2023). A complete guide to predictive maintenance. Limble CMMS. https://limblecmms.com (accessed March 21, 2023).

Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems, 22*, 143–177.

Forootan, M.M., Larki, I., Zahedi, R., & Ahmadi, A. (2022). Machine learning and deep learning in energy systems: A review. *Sustainability, 14*, 4832. DOI: 10.3390/su14094832

Furferi, R., & Rehman, A.U. (2023). Global excellence in digital manufacturing: Editorial overview. *Frontiers in Mechanical Engineering, 9*, 1334600. DOI: 10.3389/fmech.2023.1334600

Jabbari, M., Sheikh, S., Rabiee, M., & Oztekin, A. (2022). A collaborative decision support system for multi-criteria automatic clustering. *Decision Support Systems, 153*, 113671.

Jha, B.K., & Pande, S. (2021). Time series forecasting model for supermarket sales using FB-prophet. In *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 547–554). Erode, India.

Kamble S.S. (2020). Angappa Gunasekaran, Abhijeet Ghadge, Rakesh Raut, A performance measurement system for industry 4.0 enabled smart manufacturing system in SMMEs – A review and empirical investigation, *International Journal of Production Economics, Volume 229*, 107853. DOI: 10.1016/j.ijpe.2020.107853

Kannan, S. (2022). Synthetic time series data generation for edge analytics. *F1000Research, 11*, 67.

Krstić, Ž., Topalović, A., & Nowak, A. (2023). Open-source solutions for predictive maintenance in SMEs. *Designs, 7*(4), 98. DOI: 10.3390/designs7040098

La Russa, F.M., & Santagati, C. (2021). An AI-based DSS for preventive conservation of museum collections in historic buildings. *Journal of Archaeological Science Reports, 35*, 102735.

J. Lee, H.-A. Kao, S. Yang, Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment, *Procedia CIRP, 16*, (2014), 3–8. DOI: 10.1016/j.procir.2014.02.001

Li, Z., Wang, Y. & Wang, K.S. (2017). Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. *Adv. Manuf. 5*, 377—387. DOI: 10.1007/s40436-017-0203-8

Liu, C., Tang, D., Zhu, H., & Nie, Q. (2021). A novel predictive maintenance method based on deep adversarial learning in the intelligent manufacturing system. *IEEE Access, 9*, 49557–49575.

Margherita, E.G. & Braccini A.M., Exploring the Socio-Technical Interplay of Industry 4.0: A Single Case Study of an Italian Manufacturing Organisation. *Proceedings of the 6th International Workshop on Socio-Technical Perspective in IS Development (STPIS 2020)*, June 8–9, 2020, pp. 121–126.

Martin-del-Campo, S., Sandin, F., &Schnabel, S. (2021). ‘Algorithmic performance constraints for wind turbine condition monitoring via convolutional sparse coding with dictionary learning. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 235*, 660–675.

Mensah, J.K., & Boateng, A. (2018). Challenges and opportunities in implementing IoT in water bottling industries in Ghana. *Journal of Manufacturing Systems, 45*, 123–130.

Mobley, R.K. (2002). *An introduction to predictive maintenance* (2nd ed.). Butterworth-Heinemann.

Nowak, A., & Kowalski, P. (2022). Maintenance challenges in Polish SMEs: Bridging traditional practices with modern analytics. *Polish Journal of Mechanical Engineering, 34*(2), 101–115.

Nowak, K., & Kowalski, P. (2020). Predictive maintenance in Polish water bottling plants: A case study. *International Journal of Production Research, 58*(5), 1350–1362.

Panagiotakis, C., Papadakis, H., & Fragopoulou, P. (2020). Unsupervised and supervised methods for the detection of hurriedly created profiles in recommender systems. *International Journal of Machine Learning and Cybernetics, 11*, 2165–2179.

Pejić Bach, M., Topalović, A., Krstić, Ž., & Ivec, A. (2023). Predictive maintenance in Industry 4.0 for the SMEs: A decision support system case study using open-source software. *Designs, 7*(4), 98. DOI: 10.3390/designs7040098

Piri, S., Delen, D., & Liu, T. (2018). A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets. *Decision Support Systems, 106*, 15–29.

Saiktishna, C., Sumanth, N.S.V., Rao, M.M.S., & Thangakumar, J. (2022). Historical analysis and time series forecasting of stock market using FB Prophet. In *Proceedings of the 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1846–1851). Madurai, India.

Schafer, J.B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web: Methods and strategies of web personalization* (pp. 291–324). Springer, Berlin/Heidelberg.

Topalović, A., Krstić, Ž., & Ivec, A. (2023). Leveraging open-source technologies for cost-effective maintenance solutions. *International Journal of Advanced Manufacturing Technology, 120*(1), 45–58.