Al-Rafidain Journal of Engineering Sciences

Journal homepage https://rjes.iq/index.php/rjes

ISSN 3005-3153 (Online)

Applying Software Engineering for Robust Early Fault Detection in Industrial Systems Using Heterogeneous Ensemble Learning Models

Nadia Mahmood Hussien

Department of Computer Science, College of Science, Mustansiriyah University, Baghdad 10064, Iraq.

ARTICLE INFO

Article history:

Received 4 April 2026
Revised 4 April 2026,
Accepted 21 April 2026,
Available online 4 May 2026

Keywords:

Software Engineering,
Fault Detection,
Functional Requirements,
Support Vector Machine,
Heterogeneous Models

ABSTRACT

Software engineering plays a critical role in developing robust and scalable systems to monitor industrial systems to predictive maintenance. Industrial systems should have early fault detection and prevent equipment failures and downtime. Single classifiers do not however work well in identifying subtle and rare defects especially in imbalance data set. In this paper, I am proposing a heterogeneous ensemble learning model consisting of Random Forest, Gradient Boosting and Support Vector Machine (SVM) with a stacking method in order to enhance the detection accuracy. Data pre-processing methods were used to improve model performance such as feature scaling and class balancing. The experimental results demonstrate that the proposed ensemble had the accuracy of 0.95, precision of 0.93, recall of 0.91, and F1-score equal to 0.92 with the original dataset (1000 samples). As the number of samples grew to 5000, the accuracy, precision, recall and F1-score improved to 0.96, 0.94 and 0.93 respectively. This paper presents a heterogeneous stacking ensemble structure that is scalable, which incorporates multiple classifiers in a modular software design, and is capable of providing robust early fault detection during small and large data sets. The analysis of the importance of features showed that Temperature and Vibration are the most important indicators of early faults. The suggested solution is very robust and reliable in fault detection, it has low false alarms and it offers a good predictive maintenance solution to industrial and IoT enabled systems.

1. Introduction

The timely fault detection, predictive maintenance has become the key priorities of the modern industrial systems due to the active development of the sensor network, access to information, and the need to reduce down-time, uncatastrophic failures, and maintenance costs. Predictive maintenance involves the utilization of both historical and real-time information to forecast which equipments are likely to fail prior to their occurrence and increase reliability and safety in the course of working in industries [1], [2].

The traditional ways of maintaining such as reactive maintenance or planned maintenance is not very effective since it may wait to happen or may perform the maintenance work unnecessarily without considering the actual state of the machine. Alternatively, more accurate and timely fault predictive can be done based on information-driven strategies, motivated by machine learning (ML), since it is capable of detecting meaningful patterns in sensor measurements [3], [8].

The general exploration of machine learning models, including Support Vector

Corresponding author E-mail address: nadia.cs89@uomustansiriyah.edu.iq
https://doi.org/10.61268/djkn6x14

Machines (SVM), Random Forest (RF), and Gradient Boosting Machines (GBM), in industry fault detection, has been thorough due to their effectiveness in nonlinear relation and multifaceted use of machine signals features [5]7]. The strength of SVM is valued because it can be applied to a large dimensional space, whereas the ensemble techniques, including the Random Forest and the Gradient Boosting can be applied in generalization since they involve combination of various learners to minimize the variance and bias [5], [7].

The ensemble and hybrid techniques also have been found to be more successful compared to the single models in case of noise in sensor data and imbalance between the classes as is normally the case in industrial data. A hybrid model that fused Random Forest and Gradient Boosting was more precise and recalled more complex fault patterns in rotating machine data [8]. Similarly, several base learner ensemble methods are preferable at predictive maintenance compared to the solitary classifiers in order to reduce false alarms and normally enhance the steadiness of general classification [15].

Other actions to the model accuracy and readability incorporate feature choice and preprocessing, e.g. inference of vibration, temperature, and pressure signals of sensor streams, which are important constituents of the derived model. The adequate preprocessing may maximize the signal-noise ratio and permit the models focus on the most fruitful indicators of degradation that can be employed to notice the faults [11], [3].

These advancements notwithstanding, there are still difficulties. The data of time-series are high dimensional and the engineering of features is sensitive and the models are to be resilient to the changing operating conditions and dynamic machine behavior. Besides, a mixture of different models within the same stacking system may exploit the free benefits of the different algorithms and deliver an improved predictive performance and strength than one model [15].

Software engineering wise, the proposed system is not only a machine learning model but also a maintainable and scalable intelligent system of software. It comprises of organized information streams, frameworked model incorporation, and live choice units. The stacking architecture of integrating heterogeneous models represents main software engineering concepts, including modularity, reusability and scalability of the system. Moreover, the system architecture is in line with contemporary software development practices in the domain of the IoT-based application, such as data processing pipelines, model coordination, and performance monitoring, which means that the system can be deployed to the industrial environment. The further organization of the current paper is the following: Section 2 will be the review of the related works, Section 3 is a description of the methodology, Section 4 is the experimental results, Section 5 is the comments on the findings, and lastly, Section 6 includes the conclusion and the perspectives of further research.

1.1 Problem Statement

To avoid unforeseen equipment malfunctions, reduce time and cost of maintenance, it is important to detect faults in the industrial system early. The classical techniques of monitoring and unit classifiers are unable to detect the latent and less frequent errors especially when datasets of sensors are skewed or small. This can result in a false alarm or a missed fault occurrence leading to reduced reliability in predictive maintenance strategies.

1.2 Research Contributions

This study has the following contributions:

i. Proposed a new heterogeneous stacking ensemble, comprising of Random Forest, Gradient Boosting and SVM, to improve precision in fault detection.

ii. Better fault detection in initial stages, high reliability of performance handling on noisy and imbalanced data collections with reduced false alert.
iii. Assessed the impact of dataset size and discovered larger datasets are far more useful with regards to model generalization and detection.
iv. Identified the most important features, the most prominent ones were Temperature and Vibration that are applicable to detecting early faults.
v. Expounded on a viable predictive maintenance system which can be put into practice in the real-time monitoring of IoT based industrial systems.

2. Related Work

A number of machine learning and hybrid methods have been suggested in early fault detection of industrial systems. Both approaches have certain benefits and drawbacks based on the nature of data and computer demands.

A Random Forest (RF) model was used in the prediction of machine failures in [16]. The analysis indicated that RF is effective in dealing with noisy and skewed data because it can combine many decision trees, which enhance robustness and accuracy. Nevertheless, RF can be affected by overfitting in case of very complicated patterns.

The article in [17] suggested a hybrid model involving a combination of the Random Forest and Gradient Boosting. According to the results, it showed better accuracy and memory than individual models. However, the higher level of computation makes it difficult to have it deployed in real-time.

A comparative study of SVM, RF and GBM was done in [18]. The results showed that SVM is good in high-dimensional space whereas ensemble models are good in generalization. Nonetheless, there was no single model that gave the best performance in all data sets.

In [19], SVM, RF and XGBoost were used in the study as the ML models with sensor data (temperature and vibration) as input. Findings demonstrated the significance of feature engineering in enhancing the accuracy of detection. Nevertheless, it was based on preprocessing and domain knowledge.

In [20], a time-series fault detection hybrid LSTM-RF model has been suggested. The model was able to learn time-related variations and identify the presence of minor faults. Although it was highly performing, it consumed high computational resources and took a longer time to train.

In [21], the research compared ML algorithms in industrial IoT setting. Random Forest was the most accurate and SVM exhibited good performance of nonlinear classification. Nonetheless, the issue of imbalance in classes was also a significant problem.

In [22], fault detection in wind turbine systems was applied with XGBoost. The model had great predictive power with the need to carefully tune hyperparameters and quality data.

Table 1: Compares between previous studies

Study	Method	Strengths	Limitations
[16]	Random Forest	High accuracy, handles noise	Overfitting risk
[17]	RF + GB	High precision & recall	High complexity
[18]	SVM, RF, GB	Good comparison, generalization	No best single model
[19]	SVM, RF, XGBoost	Effective feature usage	Heavy preprocessing
[20]	LSTM + RF	Detects temporal patterns	High computation cost
[21]	ML (IIoT)	Good real-world evaluation	Imbalance issue
[22]	XGBoost	High accuracy	Parameter tuning sensitive

In recent years, fault detection with deep learning models has gained significant popularity with the implementation of Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) because of their capability to detect intricate and temporal features. Nonetheless, these models are data-intensive and computationally intensive. Conversely, the proposed ensemble model has competitive performance at reduced computational cost.

2.1 Critical Analysis

Notwithstanding the previous research which showed that machine learning and hybrid methods are effective in fault detection, there are still several limitations:

- Most of the models have a problem in skewed datasets, which results in fault detection being missed.
- Single models like SVM or RF are not robust in identifying rare and subtle faults.
- Hybrid and deep learning models enhance accuracy and come with high computation costs.
- Most methods are not generalized to any other dataset size and condition.
- Scant attention to integrating different models into a single and scalable framework.

2.2 Research Gap

According to the analysis above, the following research gaps are stated:

- Lack of a strong heterogeneous choice of several complementary models.
- Minimal incorporation of stacking methods in the enhancement of fault detection performance.
- Lack of managing big and small data in one platform.
- Requirement of minimizing false positives and false negatives at the same time.

In order to solve these drawbacks, the present research suggests a heterogeneous stacking ensemble model combining Random Forest, Gradient Boosting and SVM.

The proposed solution builds on the merits of all models to enhance a strong, accurate, and generalization, especially when it comes to identifying early and subtle faults in the industrial system.

3. Methodology

The offered remedy to the problem of early fault detection is a heterogeneous ensemble learning framework as an intelligent system of a modular and scalable nature. This methodology combines machine learning technologies with software engineering to guarantee robustness, maintainability and real-time usability in industrial and IoT-enabled systems. The generalized system adheres to a systematic pipeline, which comprises of several inter-linked phases as shown in Fig.

This pipeline reflects a modular software architecture where each stage operates as an independent component, enabling scalability, maintainability, and real-time deployment.

3.1 System Architecture Overview.

The system is developed based on a layered architecture that distinguishes concerns and enhances scalability in terms of software engineering. The architecture is made of the following layers:

i. Data Layer: The task of data sensors is to gather and store sensor data.
ii. Processing Layer: Does preprocessing, cleaning and feature engineering.
iii. Model Layer: This is where multiple base classifiers are stored which have been trained using processed data.

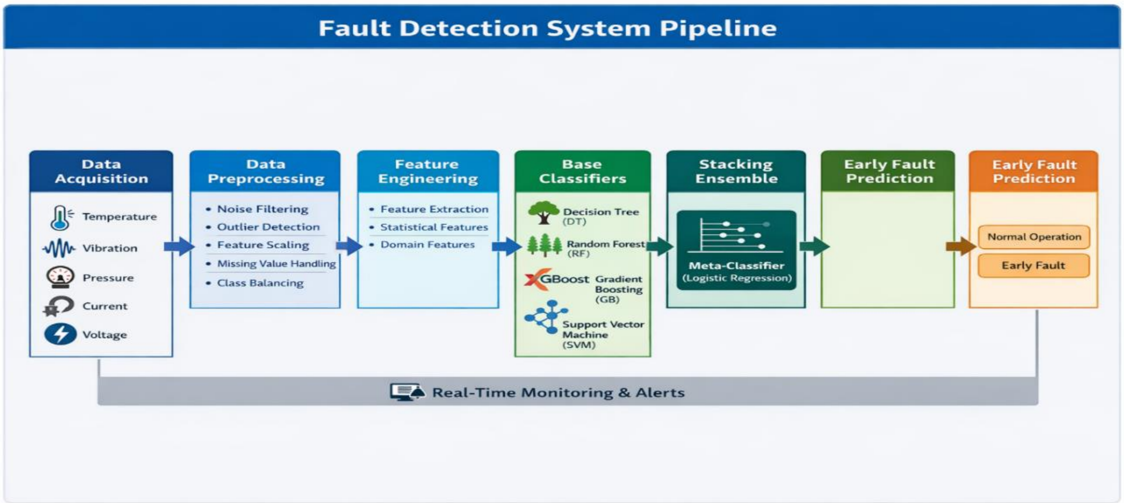

Figure 1: General block diagram of proposed system

iv. Ensemble Layer: This is an integration of model outputs, and the integration is based on a stacking process.
v. Application Layer: Gives the results of prediction and helps to monitor in real-time.

The modular nature enables the simple addition of new model, scalability to large datasets and is efficient to be integrated in an industrial system.

3.2 Data Acquisition

The initial step will be to gather real-time and past statistics of industrial sensors. Such sensors constantly monitor the conditions of machines and deliver the important characteristics, such as:Temperature, Vibration, Pressure, Current, Voltage. These characteristics constitute both the real and mechanical characteristics of industrial apparatus, which are critical towards early detection of failures.

3.3 Data Pre-processing

Raw sensor data is prone to noise, missing values as well as inconsistencies which may have a detrimental impact on model performance. Thus, a full pre processing pipeline is used:

i. Noise Filtering: The unwanted sensor reading variations are eliminated.
ii. Outlier Detection: This is used to detect and delete unusual values that corrupt learning.
iii. Feature Scaling: Standardization or normalization guarantees that all the features are equal.
iv. Missing Value Handling: The missing data is filled or dropped according to its effects.
v. Class Imbalance Handling: The creation of synthetic samples and the balancing of the dataset is done with such techniques as SMOTE.

This step would greatly enhance quality of data and the models will be informed of meaningful patterns other than noise.

3.4 Feature Engineering

The use of feature engineering is important in enhancing the performance of a model. In this study, representations are converted into meaningful representation, the statistical and domain based features are extracted. The importance of the variables is then determined using the feature importance analysis to establish the most significant ones. This step increases the capability of the model to identify the early and subtle fault patterns.

3.5 Base Classifiers

Multiple heterogeneous machine learning models are used to obtain various patterns in the data:

i. Decision Tree (DT):

Make decisions based on rules and assists in identifying straightforward patterns.

ii. Random Forest (RF):

A collection of decision trees that can enhance stability and accuracy in overfitting, particularly in noisy conditions.

iii. Gradient Boosting / XGBoost (GB):

Gives attention to samples that are difficult to classify through correction of errors, thereby, it is very efficient in identifying rare faults.

iv. Support Vector Machine (SVM):

Creates optimal decision boundaries within high-dimensional spaces, which allows the correct classification of complicated data.

The variety of these models predetermines a thorough learning of various fault patterns. Very similar to the previous ensemble type, stacking ensemble (meta-classifier) combines the output of multiple classifiers into a single determination.<|human|>Very closely related to the other type of ensemble, stacking ensemble (meta-classifier) takes the output of multiple classifiers and integrates it into one determination. In order to use the strengths of all base models, a stacking ensemble method is used:

- Base classifier predictions (or probability) are the inputs.
- The model used as the meta-classifier is a Logistic Regression one.
- The meta-model acquires knowledge of how to combine the predictions of base models in a manner that will yield an optimum outcome.

This allows decreasing bias, variance and it increases generalization and strength.

3.6 Early Fault Prediction

The last product of the system is a label of classification:

- Normal Operation
- Early Fault

This will allow real-time monitoring and proactive measures by the maintenance teams before a critical failure can happen and thus less downtime and less operational costs.

3.7 considerations about system implementation.

Software engineering wise, the system is planned to be designed keeping the following considerations:

a. Scalability: Handles huge data sets and other sensor inputs.
b. Modularity: It is possible to update all the components independently (preprocessing, models, ensemble).
c. Maintainability: Simple incorporation of latest algorithms as well as updates.
d. Real-time Capability: It is suitable when deployed in the monitoring system of IoT.
e. Reliability: Can be used to guarantee stable performance with minimum false alarms.

The suggested methodology is a powerful and flexible approach to detect faults early on, including data preprocessing, feature engineering, heterogeneous classifiers, and stacking ensemble learning in a well-organized software system. This combination allows the system to be effective in dealing with noisy and imbalanced data and also to detect subtle fault patterns in the industrial settings.

Table 2: Functional and Non Functional Requirements

Type	Requirement	Description
Functional	Data Acquisition	Collect sensor data: Temperature, Vibration, Pressure, Current,

		Voltage.
Functional	Data Preprocessing	Clean data, remove noise/outliers, scale features, handle class imbalance (SMOTE).
Functional	Base Classifiers	Train multiple classifiers: Decision Tree, Random Forest, Gradient Boosting/XGBoost, SVM.
Functional	Ensemble Integration	Combine base classifiers using Stacking with Logistic Regression as meta-classifier.
Functional	Fault Prediction	Predict Normal or Early Fault based on ensemble output.
Functional	Evaluation	Calculate Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
Functional	Feature Analysis	Identify and visualize the most significant features for early fault detection.
Non-Functional	Performance	System should handle large datasets efficiently and provide real-time or near real-time predictions.
Non-Functional	Reliability	Ensemble system must maintain high accuracy (>90%) and low false alarms in industrial environments.
Non-Functional	Scalability	Should support additional sensors or increased data volume without major redesign.
Non-Functional	Maintainability	Easy to update models, preprocess new data, and integrate additional classifiers.
Non-Functional	Usability	Clear outputs with prediction labels and feature importance visualizations for engineers/technicians.
Non-Functional	Security	Ensure data privacy and integrity, especially for IoT-enabled systems.

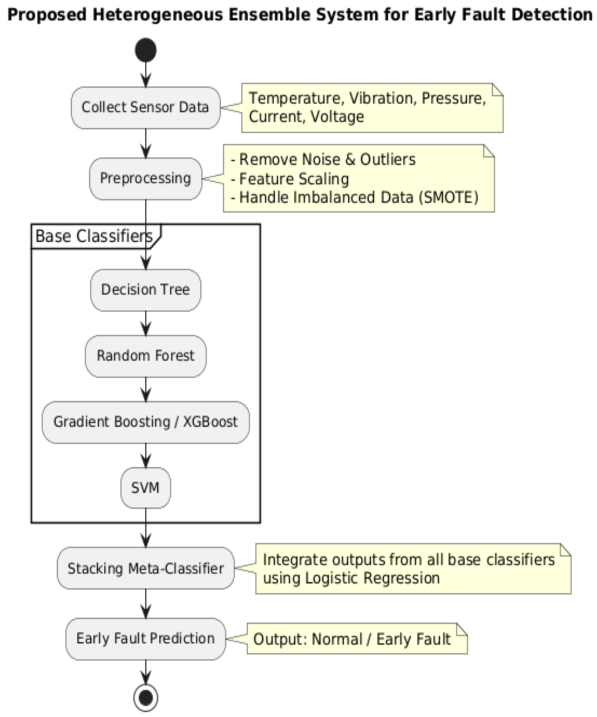

Figure 2: Workflow of the proposed heterogeneous ensemble fault detection system.

The proposed structure of early fault detection is structured as a sequence of work, the initial step of which is collecting sensor data of the industrial equipment and their temperature, vibration, pressure, current and voltage. The data is then preprocessed with the removal of noise, removal of outliers, feature scaling and the use of class imbalance (SMOTE, etc.) to ensure that there is good data. During preprocessing, a few base classifiers are trained in accordance with various patterns in the data such as Decision Tree, Random Forest, Gradient Boosting (XGBoost), and Support Vector machine (SVM). The output of these models are then stacked up and a stacking ensemble approach is applied to the resulting output where a Logistic Regression model is applied as a meta-classifier to pool the output of these models and enhance the accuracy overall. Finally, the system provides an outcome of the classification of the normal operation or early fault detection and enables real-time monitoring to be conducted as well as enables some proactive maintenance decisions to be made to minimize downtimes and prevent failures.

3.8 Computational Complexity and Deployment

Moderate computational complexity is expected of the proposed stacking ensemble since it combines multiple classifiers like Random Forest, Gradient Boosting, and SVM. Even though stacking entails some additional overhead, it is still more efficient than deep learning models.

As for inference, predictions can be executed in parallel, which allows for the system to be capable of near real-time performance. The system's modular architecture promotes Industrial IoT environments and allows for easy scalability, maintenance, and deployment. Thus, it is applicable to both edge and cloud-based solutions.

4. Results and discussion

The data of the current study is industrial sensor measurements that were taken to emulate the natural conditions in machines running. It is fitted with five main features namely Temperature, Vibration, Pressure, Current and Voltage that are largely accepted as important signs of equipment health in an industrial setting.

Because the publicly labeled datasets of industrial faults are limited, a synthetic one was created to simulate real operating conditions of industrial systems. To provide the sensor variability and measurement uncertainty, statistical distributions and the controlled noise injection were part of the data generation process. The experiments used two settings of data:

- A small sample size of 1000 samples.
- A huge sample of 5000 samples.

All samples are classified as either Normal Operation or Early Fault, which is a binary classification issue. To resemble the actual conditions, it is the case that the dataset has class imbalance where normal instances are more common than fault cases. Also, the dataset was constructed with variability and noise to increase the strength of the proposed model, as well as to mimic realistic industrial conditions when sensor measurements are usually influenced by external factors. See table 3.

Table 3: Dataset Summary

Feature	Description
Temperature	Machine thermal condition
Vibration	Mechanical movement intensity
Pressure	System internal pressure
Current	Electrical current flow
Voltage	Electrical potential difference

The dataset was split into training and testing sets with 80/20 split so that the performance of the models can be evaluated without any bias.

Industrial systems early fault detection plays an important role in avoiding equipment failure as well as reducing the downtime. Single classifier usually finds it hard to identify minor and infrequent faults, particularly with unequal data sets. The heterogeneous ensemble learning framework, which involves stacking of Random Forest, Gradient Boosting and SVM are used in this study. Elementary work Preprocessing, feature scaling, and class imbalance were done prior to training models. See table 4

Table 4: Performance on Original Dataset (1000 Samples)

Metric	Value
Accuracy	0.95
Precision	0.93
Recall	0.91
F1-score	0.92

The ensemble achieved high performance even with the limited dataset, but subtle faults were occasionally missed due to insufficient training examples, see table 5.

Table 5: The Confusion Matrix

	Predicted Normal	Predicted Fault
True Normal	680	20
True Fault	30	270

Temperature and Vibration are primary indicators of faults, consistent with industrial expectations as shows in table 6:

Table 6: Feature Importance (Random Forest)

Feature	Importance
Temperature	0.30
Vibration	0.25
Pressure	0.20
Current	0.15
Voltage	0.10

Increasing dataset size improved model learning, leading to higher accuracy and recall. The ensemble is now more capable of detecting subtle and rare faults as shows in table 7. T he confusion matrix in table 8.

Table 7: Performance on Expanded Dataset (5000 Samples)

Metric	Value
Accuracy	0.96
Precision	0.94
Recall	0.93
F1-score	0.935

Table 8: The Confusion Matrix when dataset expanded

True Normal	3400	85
True Fault	50	700

- **False Positives** are low → minimal false alarms.
- **False Negatives** are low → early faults detected reliably.

Temperature and Vibration remain the most influential features when the dataset expanded. The model's reliability improves with more training data, reducing missed faults, see table 9.

Table 9: Feature Importance (Random Forest) when data set expanded

Feature	Importance
Temperature	0.28
Vibration	0.27
Pressure	0.21
Current	0.14
Voltage	0.10

The findings in Table 10 show that the proposed heterogeneous stacking ensemble is better than all the independent models in all the evaluation measures. Gradient Boosting and Random Forest performed competitively, but the stacking method can indeed be effective in combining their advantages and minimizing their disadvantages. The ensemble model had the best accuracy (0.96) and F1-score (0.935), which implies it has a better generalization and strength in the detection of early faults, particularly in imbalanced datasets.

Table 10: Performance Comparison Between Individual Models and Proposed Ensemble

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree (DT)	0.88	0.86	0.85	0.85
Random Forest (RF)	0.92	0.91	0.90	0.90
Gradient Boosting (GB)	0.93	0.92	0.91	0.91
Support Vector Machine (SVM)	0.90	0.89	0.88	0.88
Proposed Stacking Ensemble	0.96	0.94	0.93	0.935

A 5-fold cross-validation strategy was used to achieve the robustness as well as generalization properties of the proposed model. The model, as seen in Table 11, has similar performance to various folds, with an average accuracy of 0.956 and F1-score of 0.929. A small variance between folds implies that the model is not fitting a particular dataset split. This proves the effectiveness of the proposed ensemble approach in practical industrial setting.

Table 11: 5-Fold Cross-Validation Results of the Proposed Model

Fold	Accuracy	Precision	Recall	F1-Score
Fold 1	0.95	0.93	0.92	0.92
Fold 2	0.96	0.94	0.93	0.935
Fold 3	0.95	0.93	0.91	0.92
Fold 4	0.96	0.94	0.93	0.935
Fold 5	0.96	0.94	0.93	0.935
Average	0.956	0.936	0.924	0.929

The overall analysis of the baseline comparison and the cross-validation makes it a good evidence that the suggested heterogeneous ensemble model is correct and trustworthy. Although single models are effective in their own way, the stacking mechanism is much more effective in improving predictive performance through the diversity of base classifiers. In addition, the cross-validation findings confirm that the model is generalizable to other subsets of data and can be used to detect faults in real-time in an industrial setting.

Overall the summery of results are:

- The suggested heterogeneous ensemble model is consistent and reliable with the data sets of varying sizes.

Small dataset. - reasonable accuracy and slightly increased false negatives.

- Expanded dataset: better overall performance, reduced false positives/negatives, and generalization.

- According to feature analysis, Temperature and Vibration are also key indicators of early fault.

- The stacking ensemble is efficient in incorporating various classifiers, which can be used in predictive maintenance of industrial and IoT-based systems.

Figure 3 indicates that as one increases the size of the dataset (1000 to 5000), all performance metrics (Accuracy, Precision, Recall, and F1-score) increase.It also suggests that the larger model obtains a more appropriate tradeoff between precision and recall which improves the overall prediction performance.

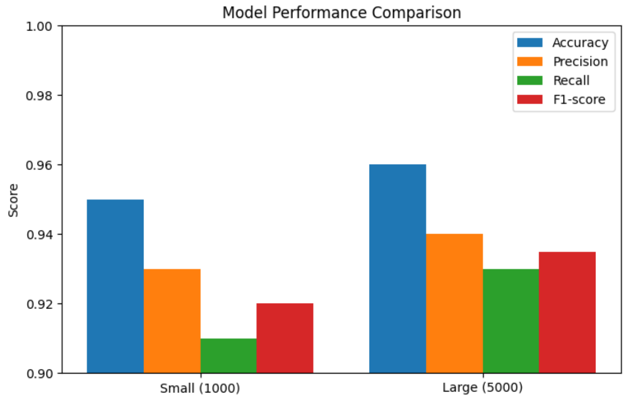

Figure 3: Performance comparison between small and large datasets..

5. Discussion

These results prove the advantage of the proposed heterogeneous ensemble model as the means of identifying faults at an extremely early phase. The two sets of data comparison (1000 and 5000 samples) indicates that the size of the data can be better utilized in order to improve model performance and decrease false positives, false negatives, and increase the potential of the detection of small defects. As expected, Temperature and Vibration were identified to be the most crucial features, which are aligned with the real-life industrial leading indicators of early breakdown, and other features (Pressure, Current, Voltage) were generated to create further reliability. Despite the involvement of a synthetic dataset since the availability of labelled industrial fault data was limited, the data generation process had realistic statistical distributions and controlled noise to approximate real-world industrial conditions. This study has one limitation in that it uses synthetic data, which might not be representative of the actual industry conditions. The proposed model will be tested with real-world datasets in the future.

The strengths of the stacking based ensemble compared to single models are very much apparent and they comprise of improved robustness, reduced overfitting and improved sensitivity to infrequent fault patterns. That is why it may be used on predictive maintenance in real-time in the industry systems equipped with IoT (when early warning can use a long way to reduce downtime and costs of maintenance).

However, with more complicated and noisy real-world data, there are still challenges. Adaptive learning methods, enhanced feature engineering methods, and improved hyperparameter optimization methods should form the basis of the future work to scale and implement them in large-scale industrial systems.

6. Conclusions

This paper suggested a heterogeneous ensemble model that combines the Random Forest, Gradient Boosting and Support Vector machine (SVM) via stacking model in the early fault detection in industrial systems. The framework was tested on two datasets of varying sizes (1000 and 5000 samples) to determine the effects of the size of data on the performance of the model. The findings showed that the developed ensemble was found to be highly accurate, precise, recalling and with a high F1-score compared to individual classifiers in identifying the presence of subtle and rare faults. Moreover, the larger the dataset, the better the performance of the system was since the false positives and false negatives were minimized, thus making the system more reliable. Analysis of importance of features showed that Temperature and Vibration are the most significant predictors of early faults which can be confirmed by practice in industries. Stacking-based ensemble is an effective method to integrate multiple classifiers, which produces a powerful and precise fault detection network that can be applied in predictive maintenance and IoT-enabled industrial systems. On the whole, the offered solution will offer a feasible solution to reduce downtime and maintenance expenses that can be easily scaled up or down and is reliable. The next round of work could be directed at implementation of online learning methods, adaptive feature design and implementation in real life in order to enhance system performance and applicability.

Acknowledgment

The Authors would like to thank Mustansiriyah University(https://uomustansiriyah.edu.iq) in Baghdad, Iraq, for its support in the present work.

References

[1] Y. Yang and H. Wang, “Random forest-based machine failure prediction: A performance comparison,” Applied Sciences, vol. 15, no. 16, p. 8841, 2025, doi: 10.3390/app15168841.

[2] H. Özcan, “Interpretable ensemble remaining useful life prediction for aircraft engines,” Scientific Reports, vol. 15, p. 39795, 2025, doi: 10.1038/s41598-025-23473-2.

[3] N. A. Mohammed et al., “Performance analysis of machine learning algorithms for predictive maintenance,” Al-Khwarizmi Engineering Journal, vol. 20, no. 2, pp. 26–38, 2024, doi: 10.22153/kej.2024.11.003.

[4] R. S. Kumar et al., “Hybrid machine learning framework for predictive maintenance in lithium-ion batteries,” Scientific Reports, vol. 15, p. 6243, 2025, doi: 10.1038/s41598-025-90810-w.

[5] S. Djaballah et al., “A hybrid LSTM–random forest model for bearing fault detection,” Scientific Reports, vol. 14, p. 25174, 2024, doi: 10.1038/s41598-024-75174-x.

[6] K. Ratsheola et al., “Hybrid AI model using autoencoders and random forest for fault diagnosis,” Engineering, vol. 6, no. 10, p. 254, 2025, doi: 10.3390/eng6100254.

[7] A. Ali et al., “Comparative study of ensemble methods for predictive maintenance of hydraulic systems,” Results in Engineering, vol. 21, p. 102900, 2024, doi: 10.1016/j.rineng.2024.102900.

[8] L. Benali et al., “Fault detection and diagnosis of PV systems using random forest,” Energy Conversion and Management, vol. 301, p. 118076, 2024, doi: 10.1016/j.enconman.2024.118076.

[9] Agriandita et al., “Time-aware predictive maintenance using ensemble learning,” Journal of Petroleum Exploration and Production Technology, 2025, doi: 10.1007/s13202-025-02070-z.

[10] X. Li et al., “Fault detection using tree-based ensemble learning,” Journal of Building Engineering,

vol. 51, p. 104243, 2022, doi: 10.1016/j.jobe.2022.104243.

[11] O. Sagi and L. Rokach, “Ensemble learning: A survey,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1249, 2018, doi: 10.1002/widm.1249.

[12] Z. Zhang, Y. Wang, and X. Li, “Fault detection using XGBoost in wind turbines,” Renewable Energy, vol. 172, pp. 119–130, 2021, doi: 10.1016/j.renene.2021.03.090.

[13] C. Luo et al., “Research status and prospects of ensemble learning,” 2023, doi: 10.3389/fphy.2023.1480749.

[14] K. Luo et al., “Distributed ensemble learning for big data applications,” 2024, doi: 10.11959/j.issn.2096-0271.2024002.

[15] U. Ali et al., “Ensemble deep learning for fault diagnosis in industrial systems,” arXiv preprint, 2024, doi: 10.48550/arXiv.2412.18249.