Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-9 (2025) 217–222

# AI-enabled Early Faults and Anomalies Detection in Electric Inverters

**Armin Mazinani, Luca Davoli, Laura Belli, Gianluigi Ferrari**

*Internet of Things (IoT) Lab, Department of Engineering and Architecture, University of Parma, Parma, IT 43124 Italy (e-mail: {armin.mazinani, luca.davoli, laura.belli, gianluigi.ferrari}@unipr.it).*

**Abstract:** Early fault detection plays an important role in reducing maintenance costs and preventing unexpected and costly downtimes of industrial machines. To this end, Artificial Intelligence (AI)-based mechanisms offer efficient approaches to enhance fault detection accuracy while enabling *real-time* responses. In this paper, we evaluate different supervised and unsupervised AI-based fault detection models (namely: $k$-Nearest Neighbors, $k$-NN; Adaptive Boosting, AdaBoost; XGBoost; Random Forest, RF; Multi-Layer Perceptron, MLP; Long Short-Term Memory, LSTM) for electric inverters, comparing them in terms of prediction accuracy and computational complexity. The experimental results show that, among the considered supervised models, XGBoost and MLP achieve the highest accuracy—approximately 99%—while maintaining the lowest computational complexity, thus positioning them as highly effective in terms of fault detection. In contrast, the considered unsupervised models exhibit lower accuracy and reliability for fault detection.



*Keywords:* Machine Learning, Deep Learning, Fault Prediction, Electric Inverter

## 1. INTRODUCTION

The fourth industrial revolution, also driven by Artificial Intelligence (AI) and Internet of Things (IoT) concepts, is reshaping companies and industries by enhancing productivity, efficiency, and sustainability. In fact, AI and IoT constitute the foundation of industrial automation, thus enabling real-time machinery monitoring, data-driven decision making, and predictive analytics to optimize industrial operations and build heterogeneous decision support systems (Soori et al. (2024)).

Moreover, among the key components of actual industrial systems, electric motors (e.g., pumps, compressors, and Heating, Ventilation and Air Conditioning (HVAC) systems) play a critical role. In particular, *inverters* are essential for precise motor speed and torque control, converting Direct Current (DC) into Alternating Current (AC) to ensure energy efficiency and optimal performance (Tiwari et al. (2021)). As a matter of fact, inverter failures can lead to costly downtime, expensive repairs, and safety risks. This further motivates the need for effective fault detection mechanisms to extend the lifetimes of inverters, thus reducing maintenance costs and preventing unexpected failures. In this context, AI-driven anomaly detection techniques enable industries to proactively identify and address faults before they escalate, enhancing system reliability in the era of Industry 4.0 (Achouch et al. (2022)).

In this study, we propose an AI-based early fault detection framework for inverters, in particular leveraging advanced Machine Learning (ML) and Deep Learning (DL) techniques and targeting the development of a robust mechanism to detect anomalies, classify fault types, and provide early warnings, in the end ensuring a seamless operation in industrial environments. On the experimental side, we evaluate different ML models—namely: $k$-Nearest Neighbors ($k$-NN), Adaptive Boosting (AdaBoost), XGBoost, Random Forest (RF)—as well as DL models—namely, Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM). Finally, to further enhance the performance evaluation, we explore AutoEncoders (AEs) based on MLP and LSTM for unsupervised learning.

The rest of the paper is organized as follow. In Section 2 we discuss on relevant literature works. Section 3 presents the ML and DL models considered for performance evaluation. In Section 4, the experimental dataset, as well as the different processing tasks applied on it, are detailed, while Section 5 is devoted to a discussion on the performance results returned by the considered AI models. Finally, in Section 6 we draw our conclusions.

## 2. RELATED WORKS

With regard to the adoption of AI mechanisms for early fault detection in electric motors, in Egaji et al. (2020) the authors apply different ML models—including $k$-NN, Support Vector Regression (SVR), and RF—on a dataset consisting of $20,480$ vibration data sampled at a 20 kHz sampling rate and collected from sensors mounted on four bearings. The obtained results show that RF outperforms the others models in detecting faults on the basis of vibration data.

Similarly, in Mari et al. (2023) an AI-based fault detection for induction motors is explored using vibration data. The results show that the proposed model achieves 100% accuracy, in contrast to an Artificial Neural Network (ANN)-based mechanism which obtains a 97% accuracy.


10.1016/j.ifacol.2025.08.139

A hybrid model, combining LSTM and One-Class Support Vector Machine (OCSVM), is proposed for fault detection in helicopter vibration data in Vos et al. (2022). In detail, two versions of the LSTM-OCSVM model are evaluated, a *first* version with a single LSTM layer and a *second* version with two LSTM layers, both then followed by an SVM classifier. Moreover, a standalone SVM model is also considered for comparison, finally obtaining that OCSVM outperforms the other models; achieving a 2% improvement in fault detection accuracy.

Monitoring and fault detection in bearings using vibration signals are investigated in Pham et al. (2024), where the authors propose an AI-based monitoring model built on a ResNet-50. Then, bearing vibration data (collected at a 12 kHz sampling frequency) are classified into four categories (namely: normal state, misalignment damage, inner raceway damage, outer raceway damage), achieving a 96.527% accuracy on the test set.

In Cong et al. (2024), real-time anomalies detection for fan activity is investigated, classifying fan anomalies into four distinct categories and proposing a hybrid approach combining Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs) for anomaly detection. According to their results, the proposed model achieves an accuracy equal to 98.3% and a F1-Score equal to 0.98.

Finally, audio-based anomaly detection for electric motor-driven fan systems is explored in Chuphal et al. (2024) using Mel-Frequency Cepstral Coefficients (MFCCs). This study simulates various fault scenarios and exploits audio recordings to extract relevant features. Considering an anomaly detection model based on ANN, the proposed system achieves a 97.15% accuracy and a 97.29% precision.

## 3. BACKGROUND

As detailed in Section 1, in this study we aim to evaluate different AI-based solutions for fault detection in electric inverters. In particular, the models we consider fall into two categories, namely ML and DL, but we also evaluate AEs models as unsupervised learning approaches. For the sake of completeness and clarity, in the following, a brief overview on the considered AI models is carried out.

### *3.1 Supervised Learning Models*

An AI "supervised learning" mechanism uses labeled data to train the model, for classification and regression: each input sample corresponds to a predetermined output—in the proposed approach, two classes will be considered, namely *normal data* and *noisy data*. Hence, the goal of this approach is to unveil the relationship between input sequences and their corresponding labels, finally attempting to correctly classify new unknown input sequences.

*$k$-Nearest Neighbors ($k$-NN)* This supervised ML model, introduced by Guo et al. (2003), classifies new data points based on the class of the majority of their $k$-nearest neighbors in the feature space. In fact, during the classification process, this algorithm identifies the $k$ data points closest to the new instance on the basis of a chosen distance metric—e.g., the Euclidean distance. As a consequence, this non-parametric method is particularly effective for pattern recognition tasks; however,it can be computationally expensive for large datasets.

*Adaptive Boosting (AdaBoost)* This ensemble method, defined by Schapire (2013), combines the predictions of different weak classifiers—performing better than random guessing, but still performing poorly in complex classification tasks—to create a stronger classifier. Thus, the use of multiple classifiers (as enabled by AdaBoost) leads to superior predictive performance with respect to individual weak learners. Moreover, the use many iterations in AdaBoost decreases the chance of overfitting.

*Random Forest (RF)* RF, defined by Cutler et al. (2012), is a supervised ML technique fundamentally built from multiple Decision Trees (DTs) to improve the prediction accuracy and reduce overfitting. However, RF is a computationally expensive algorithm, leading to slow performance when applied to large datasets for training and prediction.

*Multi-Layer Perceptron (MLP)* MLP is a type of NN consisting of multiple layers of neurons using non-linear activation functions to capture complex patterns. Moreover, a MLP typically comprises three main components: (i) the *input layer*, receiving the input data and with a number of neurons equal to the number of features in the considered dataset; (ii) the *hidden layers*, consisting of one or more layers of neurons enabling the NN to learn and extract complex patterns from the input data; and (iii) the *output layer*, producing the final predictions and containing a specific number of neurons for each specific task (e.g., one neuron for binary classification, multiple neurons for multi-class classification). The computation performed by a single neuron can be mathematically expressed as:

$$y = \phi\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

where: $x_i$ represents the $i$-th input; $w_i$ is the $i$-th corresponding weight; $b$ is the $i$-th bias term; $\phi(\cdot)$ denotes the activation function.

*Long Short-Term Memory (LSTM)* LSTM is a particular type of Recurrent Neural Network (RNN) defined to effectively process sequential data. In detail, LSTM features an integrated *memory cell* and *gating mechanism* allowing it to keep short- and long-term dependencies of complicated patterns. This allows LSTM to be useful for time-component issues, including Natural Language Processing (NLP) and time-series forecasting.

### *3.2 Unsupervised Learning*

Unsupervised learning is an alternative AI-based approach for classification, where the model is trained without labeled data. Thus, instead of relying on predefined class labels, the model autonomously identifies patterns, structures, and dependencies within the data.

An illustrative unsupervised learning model is represented by AE, a Neural Network (NN) developed to efficiently learn input data representations. More precisely after compressing the input into a lower-dimensional latent

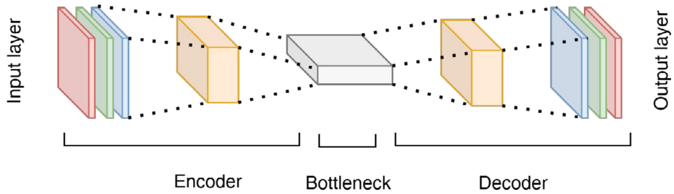


Fig. 1. General architecture of an AE.

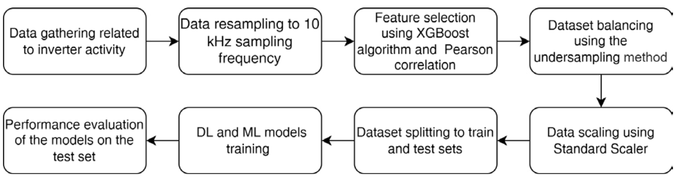


Fig. 2. Block representation of the proposed tasks chain for detecting faults and anomalies in electric inverters.

space, AE performs an effort to accurately recreate the original data. Then, to ensure that the reconstructed data closely approximates the input data, AE tries to minimize the reconstruction error.

As shown in Fig. 1, an AE model mainly includes, as main components, an *encoder*, a *bottleneck*, and a *decoder*.

To this end, in this study two types of AEs are implemented: the first one based on MLP and the second one based on LSTM.

## 4. PROPOSED MODEL

For the sake of clarity and completeness, the considered dataset is described in Subsection 4.1, while the proposed tasks chain for detecting faults and anomalies in electric inverters (whose block representation is shown in Fig. 2), comprising data preprocessing and model training, are detailed in Subsection 4.2 and Subsection 4.3, respectively. Finally, the metrics considered for the experimental evaluations are defined in Subsection 4.4.

### *4.1 Dataset*

The dataset, containing both *normal* and *noisy* inverter activities, consists of data referring to three channels, namely vibration, temperature, and power, collected at 12.8 kHz, 50 Hz, and 10 kHz, respectively. The data related to normal and noisy inverter activity have been provided by the C.O.B.O. S.p.A., Leno branch company. [1] Then, for completeness, the specific features provided by each considered channel—resulting in 19 features in total, with 3,599,992 samples for normal data and 1,749,995 samples for noisy data—are detailed in Table 1, while an illustrative comparison between the differences among some normal and noisy vibration features is shown in Fig. 3.

### *4.2 Preprocessing*

The processing tasks to be carried out for preparing the *raw* data to be used for the faults detection, comprising

[1] The public disclosure of the dataset is under consideration by the company.

Table 1. Feature description of the dataset.

| **Feature Name** | **UM*** | **Description** |
|---|---|---|
| **Vibration** (VIB) **– Sampling rate: 12.8 kHz** | | |
| VIB_Y_banco_8620 | g | Monoaxial ICP 100 mV/g |
| VIB_Z_banco_8621 | g | Monoaxial ICP 100 mV/g |
| VIB_Torque | Nm | Measured by the Torque Meter |
| VIB_Y_CUS_8619 | g | Monoaxial ICP 10 mV/g |
| VIB_Z_CUS_8618 | g | Monoaxial ICP 10 mV/g |
| VIB_Speed | V | Square wave from 0 to 5 V, 60 pulses per engine revolution |
| VIB_XNDE_8623 | g | Tri-axial ICP 100 mV/g |
| VIB_YNDE_8623 | g | Tri-axial ICP 100 mV/g |
| VIB_ZNDE_8623 | g | Tri-axial ICP 100 mV/g |
| VIB_XDE_8622 | g | Tri-axial ICP 10 mV/g |
| VIB_YDE_8622 | g | Tri-axial ICP 10 mV/g |
| VIB_ZDE_8622 | g | Tri-axial ICP 10 mV/g |
| **Temperature** (TEMP) **– Sampling rate: 50 Hz** | | |
| TEMP_Ext_motor_Temp | °C | External motor temperature |
| **Power** (POWER) **– Sampling rate: 10 kHz** | | |
| POWER_W_Voltage | V | Voltage from the W-phase of the motor |
| POWER_V_Voltage | V | Voltage from the V-phase of the motor |
| POWER_U_Voltage | V | Voltage from the U-phase of the motor |
| POWER_W_Current | A | Current from the W-phase of the motor |
| POWER_V_Current | A | Current from the V-phase of the motor |
| POWER_U_Current | A | Current from the U-phase of the motor |
| *UM: Unit of Measurement. | | |

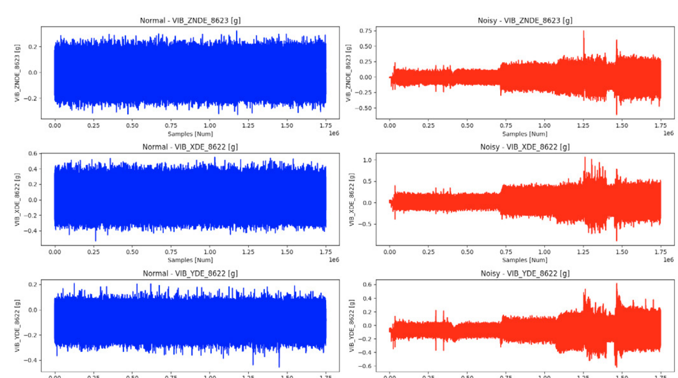


Fig. 3. Illustrative comparison between some normal and noisy vibration data.

data resampling, feature selection, data balancing, and data scaling and splitting, are detailed in the following.

*Data Resampling* Given that the considered dataset is composed of channels being sampled at different sampling rates, in order to address this "sampling inhomogeneity" a Fourier transform (as defined in SciPy (2025)) has been applied to resample both normal and noisy data to a uniform 10 kHz sampling rate.

*Feature Selection* Feature selection plays a crucial role in enhancing the model performance, reducing the computational cost and preventing the overfitting problem. Thus, by selecting the most relevant features from a given dataset, this helps to improve the generalizability of the

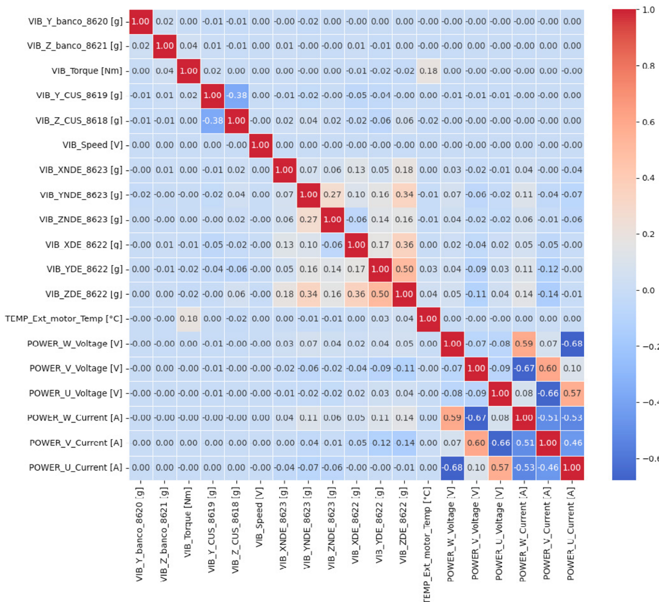


Fig. 4. Pearson correlation matrix of the features composing the considered electric inverter dataset.

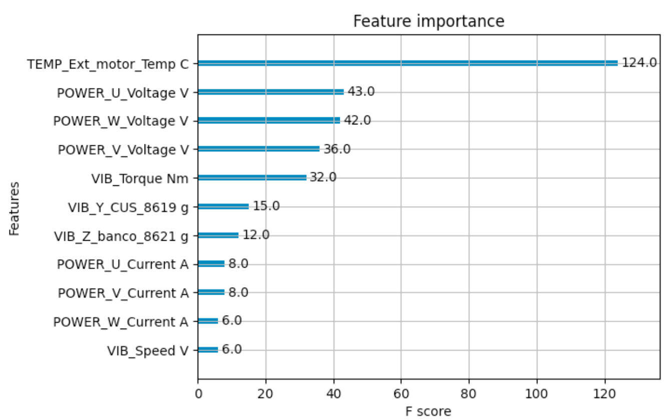


Fig. 5. Evaluation of the importance of the features composing the considered dataset using XGBoost.

model. To this end, we exploit the Pearson correlation and the XGBoost algorithm to evaluate the features' importance, then selecting the most relevant features for the considered experimental analysis. For clarity and completeness, the results of the evaluation through Pearson and XGBoost are shown in Fig. 4 and Fig. 5, respectively.

*Data Balancing* Since in the dataset, as described in Subsection 4.1, the number of normal samples is twice with respect to that of noisy samples, in order to prevent overfitting and to create a more balanced training set, an under-sampling mechanism (as defined in Lemaître et al. (2017)) has been applied to the normal data. As a result, the balanced dataset consists of 1,749,995 normal samples and of an equal number of noisy samples.

*Data Scaling and Splitting* Finally, in order to ensure a consistent features distribution, we apply a standard zero-mean, unit-variance scaler (defined in scikit-learn (2025b)) to normalize the data. Then, the dataset is split

Table 2. Architecture of the evaluated supervised and unsupervised DL models.

| **Supervised** | |
|---|---|
| **MLP** | **LSTM** |
| Dense(128, activation=relu) | LSTM(128, activation=relu) |
| Dropout(0.3) | Dropout(0.3) |
| Dense(64, activation=relu) | Dense(64, activation=relu) |
| Dropout(0.3) | Dropout(0.3) |
| Dense(32, activation=relu) | Dense(32, activation=relu) |
| Dropout(0.3) | Dropout(0.3) |
| Dense(1, activation=sigmoid) | Dense(1, activation=sigmoid) |
| **Unsupervised** | |
| **MLP-based AE** | **LSTM-based AE** |
| InputLayer(10) | InputLayer(10) |
| Dense(128, activation=relu) | LSTM(128, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(64, activation=relu) | LSTM(64, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(32, activation=relu) | LSTM(32, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(16, activation=relu) | LSTM(16, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(32, activation=relu) | LSTM(32, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(64, activation=relu) | LSTM(64, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(128, activation=relu) | LSTM(128, activation=relu) |
| BatchNormalization() | BatchNormalization() |
| Dense(10, activation=linear) | Dense(10, activation=linear) |

as 80%/20% training/testing, used to evaluate the models performance.

### 4.3 Model Training

In the following, a detailed discussion of the considered DL models is provided. In detail, in order to prevent overfitting, an early stopping mechanism was employed during training. Then, during the experimental evaluations, we observed that the temperature feature changes significantly between normal and noisy conditions, with the models tending to overfit this feature and, thus, leading to biased predictions. To ensure more reliable and generalizable results, we excluded the temperature feature from the training process.

The unsupervised models were trained exclusively on normal data, while their performance was evaluated on both normal and noisy data. Then, since unsupervised learning does not rely on labeled data, the training dataset for these models contained no labels. In contrast, the supervised models were trained using both normal and noisy data, with labels being included in the input to fulfill the training process.

For all DL models, a binary cross-entropy was used as the loss function, and the Adam optimizer (defined in Keras (2025a)) was applied. Additionally, an adaptive learning rate (as defined in Keras (2025b)) was considered to enhance training's stability and convergence. For the sake of clarity and completeness, the architecture of the evaluated DL models is shown in Table 2.

Furthermore, in order to ensure a balanced evaluation and reliable results, $k$-fold Cross-Validation (as defined in scikit-learn (2025a)), with $k = 5$, was used for training the ML models, including $k$-NN, RF, AdaBoost, and XGBoost.

### *4.4 Evaluation Metrics*

In order to evaluate the performance of the considered ML and DL models, the following evaluation metrics are considered—assuming the following notation: TP = number of True Positives; FP = number of False Positives; TN = number of True Negatives; FN = number of False Negatives; TS = number of Total Samples.

- **Accuracy ($\mathcal{A}$):** corresponds to the overall percentage of correct predictions and can be expressed as
$$\mathcal{A} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TS}}.$$
- **Precision ($\mathcal{P}$):** identifies the ability of a model to correctly identify positive instances and can be calculated as
$$\mathcal{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$
- **Recall ($\mathcal{R}$):** returns the ability of a model to detect all relevant positive instances and can be calculated as
$$\mathcal{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$
- **F1-Score ($\mathcal{F}$):** corresponds to the harmonic mean of Precision and Recall, providing a balance between these two metrics, and can be expressed as
$$\mathcal{F} = 2 \times \frac{\mathcal{P} \times \mathcal{R}}{\mathcal{P} + \mathcal{R}}.$$
- **Confusion Matrix:** returns a prediction summary, in detail providing the numbers of correct and incorrect predictions for each class. With regard to to a binary classification, the confusion matrix can be expressed as
$$\begin{bmatrix} \mathrm{TN} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TP} \end{bmatrix}.$$

## 5. RESULTS AND DISCUSSION

The experimental performance results returned by the ML and DL models introduced in Section 3, assessed on the basis of the evaluation metrics detailed in Subsection 4.4—namely, precision, recall, F1-Score, accuracy—for fault detection, are summarized in Table 3. These results indicate that all models achieve a high accuracy, ranging from 98% to 99%. Then, the confusion matrices for each considered model are shown in Fig. 6. As a result, among the evaluated models, RF and XGBoost return the lowest prediction error. However, while they both exhibit similar classification performance, XGBoost significantly outperforms RF in terms of computational efficiency, reducing the inference time by 93% (0.1151 s for XGBoost *vs.* 1.1674 s for RF), making XGBoost the preferred choice for real-time applications.

Finally, the experimental evaluation results returned by both supervised and unsupervised DL models are presented in Table 4, where it can be noted that supervised models significantly outperform their unsupervised counterparts in terms of precision, recall, F1-score, and accuracy. In fact, both MLP and LSTM supervised models—whose confusion matrices as shown in Fig. 7 and Fig. 8, respectively—yield similar results, with MLP requiring 14.44% less memory and fewer parameters if compared to LSTM, thus making it a more efficient choice in terms of computational resources.

Table 3. Experimental performance results returned by the considered ML models.

| Model | Configuration | $\mathcal{P}$ | $\mathcal{F}$ | $\mathcal{R}$ | $\mathcal{A}$ |
|---|---|---|---|---|---|
| $k$-NN | Num. neighbors = 10 | 0.98 | 0.98 | 0.99 | 0.98 |
| Adaboost | Num. estimators = 50 | 0.98 | 0.99 | 0.98 | 0.98 |
| XGBoost | Eval. metric = “logloss” | 0.99 | 0.99 | 0.99 | 0.99 |
| RF | Num. estimators = 50 | 0.99 | 0.99 | 0.99 | 0.99 |

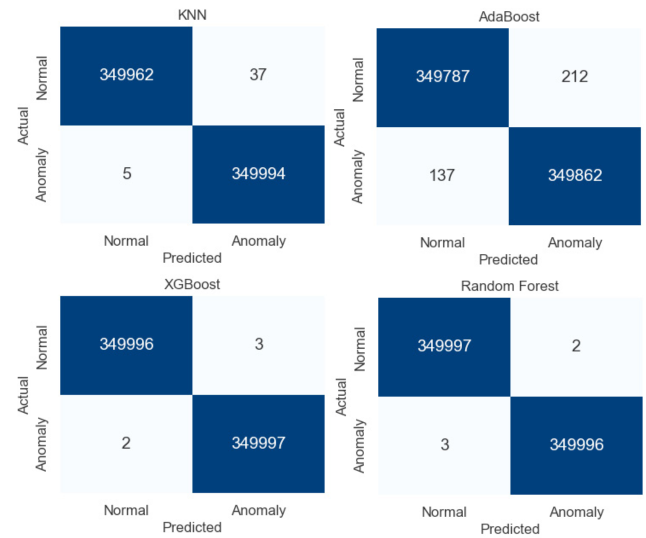


Fig. 6. Confusion matrices of the considered ML models.

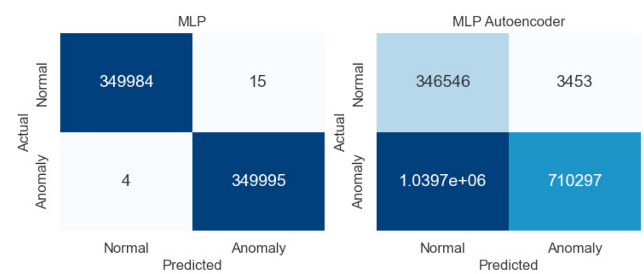


Fig. 7. Confusion matrices of MLP and MLP-based AE.

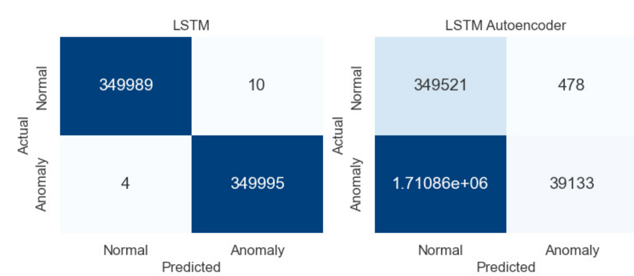


Fig. 8. Confusion matrices of LSTM and LSTM-based AE.

## 6. CONCLUSIONS

In this paper, we evaluate multiple supervised and unsupervised AI models for early fault detection of electric inverters. The performance of each model is assessed using evaluation metrics including precision, recall, F1-

Table 4. Experimental performance results and corresponding complexity returned by the considered supervised and unsupervised DL models.

| Model | $\mathcal{P}$ | $\mathcal{F}$ | $\mathcal{R}$ | $\mathcal{A}$ | Num. of Parameters | Memory Size [KB] |
|---|---|---|---|---|---|---|
| MLP | 0.99 | 0.99 | 0.99 | 0.99 | 35,333 | 138.02 |
| MLP-based AE | 0.50 | 0.99 | 0.40 | 0.58 | 77,200 | 301.57 |
| LSTM | 0.99 | 0.99 | 0.99 | 0.99 | 244,613 | 955.52 |
| LSTM-based AE | 0.98 | 0.4 | 0.02 | 0.18 | 796,496 | 3,110 |

Score, accuracy, and model complexity. The experimental results demonstrate that supervised DL models significantly outperform their unsupervised counterparts in terms of classification accuracy and complexity (namely, number of parameters and model size). Among the evaluated models, MLP and XGBoost provide the best *trade-off* between prediction accuracy and computational efficiency. These models not only achieve the highest classification performance but also maintain a relatively low complexity, making them appropriate for deployment on resource-constrained devices. Future research activities will focus on the deployment—and consequent experimental evaluation—of the considered AI models on edge hardware compliant with IIoT scenarios—e.g., in terms of processing speed, data accuracy, etc. Then, a comparison of the obtained results with publicly-available benchmarks, as well as exploring quantization techniques for optimized lightweight model deployment on resource-constrained devices, till adopting hyperparameter optimization mechanisms, will represent interesting research characterizations and extensions.

## ACKNOWLEDGEMENTS

The Authors would like to thank C.O.B.O. S.p.A., Leno branch, for its support in providing data and feedbacks. The work of the Authors was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP J33C22002880001, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”). The work reflects only the Authors’ views; the European Union is not responsible for any use that may be made of the information it contains.

## REFERENCES

Achouch, M. et al. (2022). On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Applied Sciences*, 12(16), 8081. doi:10.3390/app12168081.

Chuphal, M. et al. (2024). Anomaly Detection in Electro-Mechanical Devices Using MFCC. In *2024 2nd International Conference on Disruptive Technologies (ICDT)*, 986–991. Greater Noida, India. doi: 10.1109/ICDT61202.2024.10489646.

Cong, D.L. et al. (2024). Real-Time Fan Anomaly Detection Using Embedded Machine Learning. In *2024 9th International Conference on Applying New Technology in Green Buildings (ATiGB)*, 362–367. Danang, Vietnam. doi:10.1109/ATiGB63471.2024.10717767.

Cutler, A., Cutler, D.R., and Stevens, J.R. (2012). *Random Forests*, 157–175. Springer New York. doi: 10.1007/978-1-4419-9326-7_5.

Egaji, O.A., Ekwevugbe, T., and Griffiths, M. (2020). A Data Mining based Approach for Electric Motor Anomaly Detection Applied on Vibration Data. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 330–334. London, UK. doi:10.1109/WorldS450073.2020.9210318.

Guo, G. et al. (2003). *KNN Model-Based Approach in Classification*, 986–996. Springer Berlin Heidelberg. doi: 10.1007/978-3-540-39964-3_62.

Keras (2025a). Adam optimization. URL `https://keras.io/api/optimizers/adam/`. Accessed on May 5, 2025.

Keras (2025b). LearningRateSchedule. URL `https://tinyurl.com/keraslrs`. Accessed on May 5, 2025.

Lemaître, G., Nogueira, F., and Aridas, C.K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(1), 559–563. doi:10.5555/3122009.3122026.

Mari, S. et al. (2023). Machine Learning for Anomaly Detection in Induction Motors. In *2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, 1057–1062. Milan, Italy. doi: 10.1109/MetroXRAINE58569.2023.10405548.

Pham, V.N. et al. (2024). Using Artificial Intelligence (AI) for Monitoring and Diagnosing Electric Motor Faults Based on Vibration Signals. In *2024 International Conference on Information Networking (ICOIN)*, 720–725. Ho Chi Minh City, Vietnam. doi: 10.1109/ICOIN59985.2024.10572186.

Schapire, R.E. (2013). *Explaining AdaBoost*, 37–52. Springer Berlin Heidelberg. doi:10.1007/978-3-642-41136-6_5.

scikit-learn (2025a). Cross-validation: evaluating estimator performance. URL `https://tinyurl.com/scikitlcv`. Accessed on May 5, 2025.

scikit-learn (2025b). StandardScaler. URL `https://tinyurl.com/scikitlss`. Accessed on May 5, 2025.

SciPy (2025). Resample. URL `https://tinyurl.com/scipyresample`. Accessed on May 5, 2025.

Soori, M. et al. (2024). Intelligent Robotic Systems in Industry 4.0: A Review. *Journal of Advanced Manufacturing Science and Technology*, 4(3), 2024007–2024007. doi:10.51393/j.jamst.2024007.

Tiwari, D. et al. (2021). A Review of Circular Economy Research for Electric Motors and the Role of Industry 4.0 Technologies. *Sustainability*, 13(17), 9668. doi: 10.3390/su13179668.

Vos, K. et al. (2022). Vibration-based Anomaly Detection using LSTM/SVM Approaches. *Mechanical Systems and Signal Processing*, 169, 108752. doi: 10.1016/j.ymssp.2021.108752.