Towards Fault Detection in Industrial Equipment through Energy Consumption Analysis: Integrating Machine Learning and Statistical Methods

Nada BADDOU [1*], *Afaf DADDA* [2], *Bouchra* RZINE [1] and *Hala HMAMED* [2]

[1]LGM, Faculty of Science and Technology, Sidi Mohamed Ben Abdellah University Fes, Morocco
[2]Mathematical and Computer Modeling Laboratory, University moulay Ismail, ENSAM, Meknes, Morocco

Abstract. Accurately forecasting the energy consumption of industrial equipment and linking these forecasts to equipment health has become essential in modern manufacturing. This capability is crucial for advancing predictive maintenance strategies to reduce energy consumption and greenhouse gas emissions. In this study, we propose a hybrid model that combines Long Short-Term Memory (LSTM) for energy consumption prediction with a statistical change-point detection algorithm to identify significant shifts in consumption patterns. These shifts are then correlated with the equipment's health status, providing a comprehensive overview of energy usage and potential failure points. In our case study, we began by evaluating the prediction model to confirm the performance of LSTM, comparing it with several machine learning models commonly used in the literature, such as Random Forest, Support Vector Machines (SVM), and GRU. After assessing different loss functions, the LSTM model achieved the strongest prediction accuracy, with an RMSE of 0.07, an MAE of 0.0188, and an R^2 of 92.7%. The second part of the model, which focuses on detecting change points in consumption patterns, was evaluated by testing several cost functions combined with binary segmentation and dynamic programming. Applied to a real-world case, it successfully detected a change point two months before equipment failure, offering the potential to reduce energy consumption by 27,052 kWh. This framework not only clarifies the relationship between equipment health and CO2 emissions but also provides actionable insights into emission reduction, contributing to both economic and environmental sustainability.

Keywords: Machine learning, energy saving, early failure prediction, Carbone footprint, predictive maintenance

1 Introduction

The carbon footprint of an industry is the total amount of greenhouse gases (GHG) emitted directly or indirectly by its activities, measured in kilograms of carbon dioxide equivalent (kg CO2e)[1]. These gases including carbon dioxide (CO_2), methane (CH_4), and nitrous oxide (N_2O) are significant contributors to climate change and global warming [2]. Greenhouse gas emissions can come from a variety of sources, such as the combustion of fossil fuels for energy, manufacturing processes, the transportation of raw materials and finished products, and waste management [3]. In this context, reducing energy consumption in the industrial sector, particularly for those relying on fossil fuels, is a crucial priority. This reduction not

* Corresponding author: nada.baddou@usmba.ac.ma

only helps factories cut costs but also mitigates environmental impact by lowering greenhouse gas emissions. As a result, in the global fight against climate change, industries must commit to energy-saving practices [4,5].

Machine learning in the context of efficient energy management offers an opportunity for more optimal development and management of factors influencing energy consumption [6]. By detecting deviations from current machine consumption, industries can anticipate imminent failures or abnormal changes [7], which are crucial for planning and executing maintenance activities proactively. In fact, predicting the evolution of energy consumption using machine learning algorithms allows for anticipating abnormal trends associated with machine malfunction, thereby providing an opportunity to prevent these issues in advance through timely maintenance. Consequently, this approach emphasizes the importance of predictive maintenance based on monitoring machine energy consumption to reduce maintenance costs, enhance operational reliability, optimize industrial system performance, and reduce carbon footprint.

Currently, there is a pressing need for energy management strategies in industrial manufacturing plants that can effectively monitor and predict energy consumption trends [8]. These strategies should link variations in energy patterns to changes in process conditions, raw materials, and equipment health while also considering their impact on the carbon footprint. However, there is limited research focused on predicting equipment failures and process anomalies while simultaneously linking these predictions to efficient energy management and carbon footprint reduction. Moreover, energy metering in factories often does not cover individual pieces of equipment, relying instead on general energy metering for the entire facility. This approach may not provide the necessary granularity to accurately identify the source of energy inefficiencies. This gap underscores the need for data-driven models capable of identifying changes in the energy consumption patterns of various industrial equipment and correlating them with the health of the equipment. Predicting equipment failures through such models can lead to significant energy savings and a reduced carbon footprint. To achieve this, it is essential to convert current signals from industrial equipment, which are typically collected and displayed in SCADA [9] (Supervisory Control and Data Acquisition) systems, into actionable energy consumption data. This conversion would enable more precise monitoring and prediction of energy trends, facilitating proactive maintenance and contributing to overall sustainability goals.

This paper aims to evaluate the energy and environmental benefits of predicting industrial equipment failures. To achieve this, we adopted a structured methodology, beginning with the identification of equipment on the production line that experienced sudden, run-to-failure downtime. Specifically, we conducted a case study on industrial equipment that encountered unexpected downtime due to bearing wear, analyzing its energy consumption behavior (in kWh) over the four months leading up to the mechanical failure. We then applied various statistical methods to detect the point at which the energy consumption trend began to shift. By calculating the difference in energy consumption before and after this change point, we determined the extent of energy overconsumption. Finally, we employed multiple machine learning techniques to predict energy consumption, demonstrating how such predictions could help mitigate unnecessary energy use and reduce greenhouse gas emissions.

The remainder of this paper is organized as follows: Section 2 offers a comprehensive review of the relevant background and existing literature, providing a critical analysis of prior work in the field, particularly focusing on machine learning models applied to energy consumption and predictive maintenance. This section also identifies gaps, notably the limited research that integrates energy management strategies with predictive maintenance. Our study addresses this by using a predictive model, analyzing energy consumption patterns with statistical methods, and linking significant changes to the potential early detection of equipment degradation. In Section 3, we present the methodology and the case study that underpins our research, detailing the equipment and energy consumption data used, as well as the advanced models and algorithms employed, including both change-point detection techniques and machine learning algorithm used in prediction including the construction of a prediction model in Python. We also provide a justification for the selection of these methods, grounded in their theoretical foundations and proven practical performance in similar studies. Subsequent sections are dedicated to presenting the results of our analysis, including the precision of the prediction and the detection of changes in energy consumption patterns that correlate with equipment health. We also engage in a thorough discussion of our findings, considering their implications for industrial applications, potential limitations, and comparisons with prior research. Finally, the paper concludes by summarizing the key insights from the study, highlighting the effectiveness of combining machine learning and statistical methods for predictive maintenance. We also offer recommendations for future research, particularly focusing on improving detection accuracy and scalability in complex industrial settings.

2 Related works and Background

As we delve into the core of this study, it is essential to first understand and review prior research that has shaped the current landscape of energy consumption prediction and predictive maintenance. The following section provides a comprehensive review of the background and related works, exploring the advancements in machine learning and statistical methods that have been applied to this field. This analysis will highlight the strengths and limitations of existing approaches, setting the stage for the novel contributions of our study.

2.1 Related works

Several studies have explored degradation detection through various methods, while others have focused on monitoring changes in energy consumption patterns and efficient energy management.

In the study [10] the authors propose a data-driven approach for energy modeling using Subspace Identification (Sub-ID). This method models the dynamic energy consumption of industrial machines and incorporates an adaptive mechanism to adjust the model in response to concept drift, allowing the system to detect degradation or abnormal behavior over time. The focus is on maintaining energy efficiency and reducing maintenance costs through predictive modeling and real-time adjustments. In a similar context, the authors in [11] apply statistical methods to detect changes in the energy consumption patterns of buildings. This work focuses on identifying significant variations in energy consumption over time through statistical regression models and fitting techniques, allowing for the detection of anomalies in energy usage. A framework that leverages a dual Long Short-Term Memory (LSTM) architecture for change point detection and Remaining Useful Life (RUL) prediction is presented in [12]. The framework detects uncertain change points in the operational data of machinery and subsequently uses this information to predict the health index of the system, which is then used to estimate the remaining useful life. The dual LSTM design enables the model to capture both long-term and short-term dependencies in sensor data, providing high-precision, real-time predictions of machine degradation.

The originality of our work lies in the innovative combination of energy consumption tracking and predictive maintenance, an approach not fully explored in previous studies. While other works like [10] focus on adaptive models for energy efficiency or [12] combine LSTM with RUL prediction, our method uniquely integrates LSTM-based energy prediction with change point detection to monitor equipment degradation in real-time. This allows us to not only forecast energy usage but also to identify patterns that signal potential wear or failure, providing a direct link between energy behavior and equipment health. Unlike traditional statistical models such as those in [11], our LSTM approach captures nonlinear, long-term dependencies, making it more adaptable to complex, evolving industrial environments. By linking energy consumption patterns with predictive maintenance strategies, our work offers a proactive and efficient solution for managing equipment health and optimizing operational performance, a novel contribution to the field.

2.2 Background

2.2.1 Machine learning using in prediction

Several machine learning algorithms are frequently used in the literature for prediction tasks. The most commonly used according to the paper [13] are: Random Forest (RF) at 33%, neural network-based methods (e.g., ANN, CNN, and LSTM) at 27%, Support Vector Machine (SVM) at 25%, and k-means at 13%.

In this section, we briefly define a key machine learning methods that are relevant to our study:

- Random Forest (RF)

Introduced by Leo Breiman in 2001 [14] , RF is an ensemble method combining multiple decision trees. Each tree is built from random samples of the training data and features, enhancing robustness and handling high-dimensional data. RFs are widely used in various fields, including medicine and document retrieval, due to their ability to work with weak classifiers and provide internal estimates of variable importance and error.

- Support Vector Machines (SVM)

SVMs are effective for classification and regression tasks, aiming to find an optimal hyperplane that separates classes with maximum margin. They handle non-linear data using kernel functions, making them suitable for complex problems such as image classification and fraud detection [15].

- Artificial Neural Networks (ANN)

ANNs, inspired by biological neurons, process information through interconnected nodes. They include various architectures like GANs and Autoencoders (AE) for generating synthetic data and reducing data dimensionality, respectively. ANNs are widely applied in tasks requiring contextual understanding, such as text generation and sequence modelling [16].

- Recurrent Neural Networks (RNN)

RNNs process sequential data using recurrent connections to retain past information. Variants like LSTM and GRU address issues like gradient vanishing, with applications in machine translation, speech recognition, and time series prediction. LSTM networks, with their gating mechanisms, excel in handling long-term dependencies [17].
They have recurrent connections that enable them to memorize past information, making them suitable for tasks such as machine translation, speech recognition, and time series prediction.

- Long short term memory (LSTM)

LSTMs [18] are a variant of RNNs designed to better handle gradient vanishing issues with recurrent neural networks [17]. LSTMs use 3 gates to regulate the flow of information: input gate, forget gate and output gate, allowing the network to memorize information over longer periods.

- Gated Recurrent Unit (GRU)

GRUs [19] are another variant of RNNs aiming to simplify the architecture of LSTMs while retaining similar performance, the GRU architecture does not incorporate the memory cell characteristic of LSTM. Furthermore, because of the GRU's one-gate deficiency compared to LSTM, it boasts fewer network parameters. They use gates to control information flow, but with a lighter structure, making them more efficient in certain situations.

In this study, we decided to predict energy consumption using LSTM networks, which are well-suited for making predictions based on time series data. We will compare their performance with Random Forest, SVM, and GRU. The quality of the predictions was evaluated using commonly used error metrics in the literature, including R-squared (R^2), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Logarithmic Error (MSLE).

2.2.2 Change point detection

Change point detection involves identifying where the statistical properties of a time series change significantly. For a given time series $S_{1:T} = (x_1 \dots x_T)$, a change point exists at a time step δ (where $\delta \in (1, T-1)$) if the statistical properties of the segments $S_{1:\delta} = (x_1 \dots x_\delta)$, and $S_{\delta+1:T} = (x_{\delta+1} \dots x_T)$ are different. If there are n change points, the time series is divided into n+1 distinct segments.

The challenge is to determine both the number and locations of change points in the time series. This is generally achieved by defining a cost function for each segment and minimizing the overall segmentation cost. Formula (1) represents this method, where the cost function includes a penalty term βn to prevent overfitting. In other words, we aim to divide the data into segments where the cost function (i.e., the mean squared error) is as low as possible, meaning that each segment is as homogeneous as possible.

$$\sum_{i=1}^{n+1} C(S_{\delta_{i-1}+1:\delta_i}) + \beta n \quad (1)$$

Here, C ($S_{\delta_{i-1}+1:\delta_i}$) is the cost function for a segment ($\delta_{i-1} \dots \delta_i$) and βn is the penalty term.

Several criteria for selecting the penalty constant β are used in practice[20][20], such as Akaike's Information Criterion (AIC), Schwartz's Information Criterion (SIC), Bayesian Information Criterion (BIC), and modified BIC (mBIC), as it showed improved accuracy in test results according to study[11] we choose to use mBIC in the context of our work.

To perform optimal segmentation of a time series, several algorithms are cited in the literature, including binary segmentation (binseg) and dynamic programming (Dynp). Binary segmentation is fast but may be less accurate, while dynamic programming offers exact minimization of the objective function, although its computational cost is high for large time series. To simulate these methods, we chose Python as a tool. Python has a library specialized in offline change point detection called 'Ruptures', providing tools for analysing and segmenting non-stationary signals. It features a variety of algorithms for both exact and approximate detection across various parametric and non-parametric models. The library is designed for ease of use, offering a well-documented and consistent interface. Additionally, its modular structure allows for easy integration and extension of different algorithms and models.

In the 'Ruptures' library, we need to choose the cost function and segmentation algorithms to use, these cost functions include L1, L2, Rbf, and Normal (Table. 1). The choice of the cost function mainly depends on the type of change to be detected [21] and segmentation algorithms include: dynamic programming ("Dynp"), which finds the optimal segmentation by exactly minimizing the cost function, and binary segmentation ("Binseg")[22] where the data is iteratively divided into segments using a test statistic to determine the change point while approximately minimizing the cost function. Table 1 summarizes the equations of the various cost functions that will be tested in this study. We tested different cost functions to evaluate the evolution of data points across different segments relative to the average; we combined these functions with segmentation algorithms: dynamic programming ("Dynp"), and binary segmentation ("Binseg"). In the context of this study, we are looking for changes in the average energy consumption since the operation of the equipment under study, the pulper, naturally experiences fluctuations depending on the nature of the raw material being processed. Therefore, a significant change in the average will indicate a change in the operating conditions of the equipment, for this purpose we tested different cost functions to evaluate the evolution of data points across different segments relative to the average. We combined these functions with segmentation algorithms: dynamic programming ("Dynp"), and binary segmentation ("Binseg"), so we will have 8 models to explore.

Table. 1. Cost function in python

Cost function in python	Equation
L1 [23]	$C(x) = \sum_{i=0}^{n} \lvert xi - \mu \rvert$ where $xi \in S_{1:T}$ where μ is the mean of the values xi over the considered interval
L2	$C(x) = \sum_{i=0}^{n} (xi - \mu)^2$ where μ is the mean of the values xi over the considered interval
çRBF	$C(x) = \sum_{i=0}^{n} \frac{exp(-\lvert\lvert xi-\mu \rvert\rvert^2)}{2\sigma^2}$ where μ is the mean of the values xi over the considered interval
Normal [24]	$\mathrm{C(x)} = -\sum_{k=i}^{j} \log(\mathrm{Prob}(x_k \ \backslash \mu, \sigma) = \frac{1}{2\sigma^2}\sum_{k=i}^{j}(x_k - \mu)^2 + \frac{n}{2}\log(2\pi\boldsymbol{\sigma}^2)$ where μ and $\boldsymbol{\sigma}$ are the parameters of normal distribution ,d

3 Case study and methodology

3.1 Case study

This study is based on real-world data collected from a paper recycling plant. The focus of the investigation is on a critical piece of equipment known as a pulper, which is responsible for breaking down old paper with water to create paper pulp. This pulp then undergoes a preparation process to remove impurities before being used in the production of new paper as illustrated in Fig. 1. The pulper is integral to the operation of the paper machine; any downtime in the pulper leads to a complete halt in the entire production process, underscoring its critical importance to the plant's operations.

From a preliminary analysis of the available data and the general conditions in the plant, we identified four key points to consider. First, the limited historical amperage data. There is a lack of historical amperage data from the period preceding the failure of the pulper bearing, with only a maximum of 3 months of data recorded on the Distributed Control System (DCS). Second, the daily energy consumption data from the equipment, obtained from a separate energy meter outside the supervisory system, were utilized for change-point detection and calculating energy overconsumption. However, these data were not used for prediction purposes due to the limited number of samples (120 samples over 4 months). Third, the energy prediction model construction, where the construction of the energy prediction model was based on the available recent amperage data, with a sample size of 8,624.

 https://doi.org/10.1051/e3sconf/202560100079

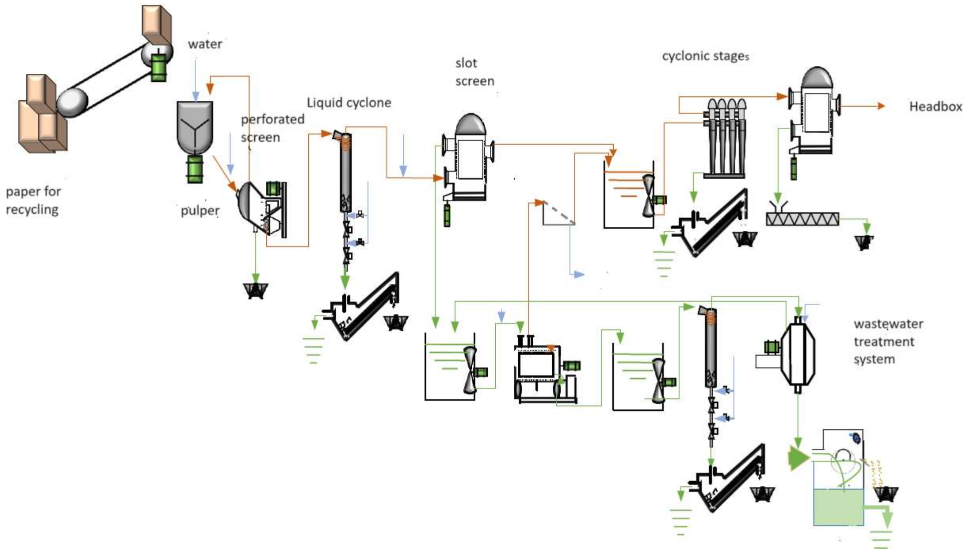


Fig. 1. Pulp preparation flowchart

3.2 Data collection and preparation

Given its proven effectiveness [25,26,27], we opted for the Long Short-Term Memory (LSTM) algorithm to perform predictions in our work. The following outlines the step-by-step process of data preparation and model development using LSTM. In this step, as previously explained, we use the recent current data to calculate the corresponding energy consumption using formula (6). Afterward, we load the data into Python from an Excel file and prepare it for training the LSTM model.

$$E=\sqrt{3}\times A\times U*\cos(\Phi)*DT \tag{6}$$

Where:
E is the energy consumption.
A is the current.
U is the voltage.
Cos (Φ) is the power factor (the cosine of the phase angle between the current and voltage).
DT is time period over which the energy is consumed.

The data is loaded from the specified Excel file using pd.read_excel(), after that, the energy consumption data is normalized using MinMaxScaler represented by formula (7) to scale the values between 0 and 1. This helps the model converge faster during training.

$$x(scaled) = \frac{\mathrm{x-min(x)}}{\mathrm{max(x)-min(x)}} \tag{7}$$

Where:
x is the original value.
min(x) is the minimum value in the dataset.
max(x) is the maximum value in the dataset.
x(scaled) is the scaled value.

3.3 Predictive model elaboration

We choose Python to build LSTM model, the process begins with generating data sequences for training the LSTM model using the create_sequences function. This function constructs sequences of consecutive amperage values, where each sequence consists of a defined length (in this case, 10 values) used to predict the next value. Once the sequences are prepared, the data is split into training and testing sets, with 70% allocated for training and the remaining 30% for testing. The LSTM model is then built, comprising two LSTM layers, each containing 50 units, followed by a dense layer with a single output unit. The Adam optimizer is used for training, and the loss function is set to mean squared error, which is appropriate for predicting continuous values such as current. The model is trained for 50 epochs with a batch size of 32, and its performance is evaluated after each epoch using the validation set. After training, the model's predictions on the test set are generated and scaled back to the original range using scaler.inverse_transform. Finally, the actual and predicted results are plotted to visualize and assess the model's performance.

To confirm the performance of the LSTM model, and as previously described, we compared it with some of the most commonly used machine learning algorithms for prediction tasks. For the construction of these algorithms in Python, GRU follows the same logic as LSTM. It has a similar architecture with two GRU layers, each containing 50 units, followed by a dense layer with one output unit. The model is also compiled with the Adam optimizer and mean squared error as the loss function, using 50 epochs and a batch size of 32.For Random Forest, we implemented the RandomForestRegressor from the sklearn.ensemble library. The model was trained using the same training and testing split, with the number of trees set to 100. Random Forest, being an ensemble method, aggregates the predictions of multiple decision trees to enhance accuracy and robustness. Lastly, for SVM, we utilized the SVR (Support Vector Regressor) from the sklearn.svm module. A radial basis function (RBF) kernel was applied to capture non-linear patterns in the data. The hyperparameters such as the regularization parameter C and the kernel coefficient γ were fine-tuned through cross-validation.

All models were compared based on their performance metrics, including R-squared (R^2), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Logarithmic Error (MSLE), to assess which algorithm provides the most accurate predictions.

3.4 Method Description

The method proposed in this study offers a versatile approach applicable to a wide range of industrial settings. The process begins with the collection of energy consumption data from the equipment or installation being monitored. This data is then meticulously prepared to enable effective model training. An LSTM-based prediction model is subsequently developed, which is then used in tandem with the statistical change-point detection method detailed in this study.

Upon detecting at least one change point, the system triggers an alarm that includes calculating the energy overconsumption and its equivalent in kilograms of CO_2e. This alert serves as a prompt for decision-makers in the plant to investigate potential issues with the equipment. If no change points are identified, the monitoring process is repeated, ensuring continuous oversight of the equipment's energy consumption patterns as illustrated in Fig.2.

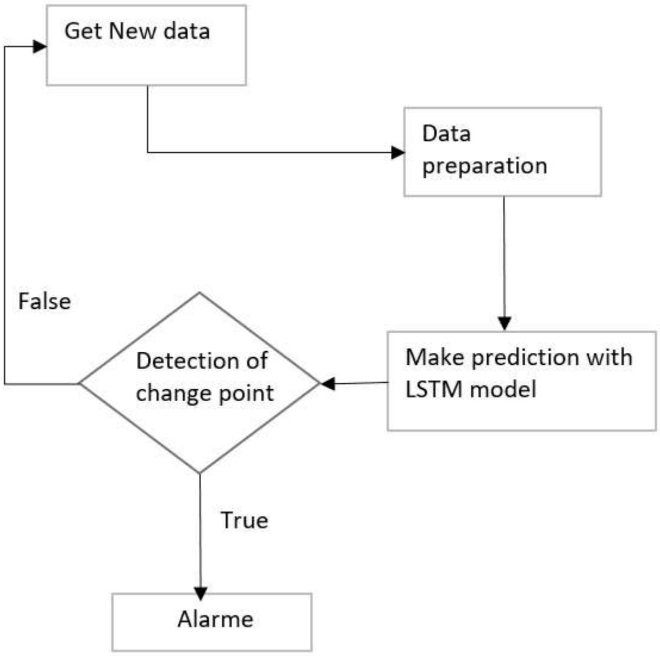


Fig. 2. Flowchart of the proposed method.

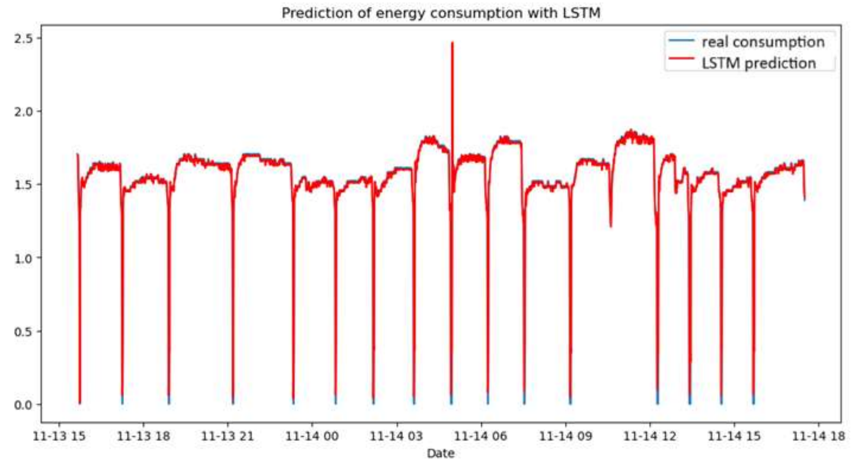
4 Results and discussion

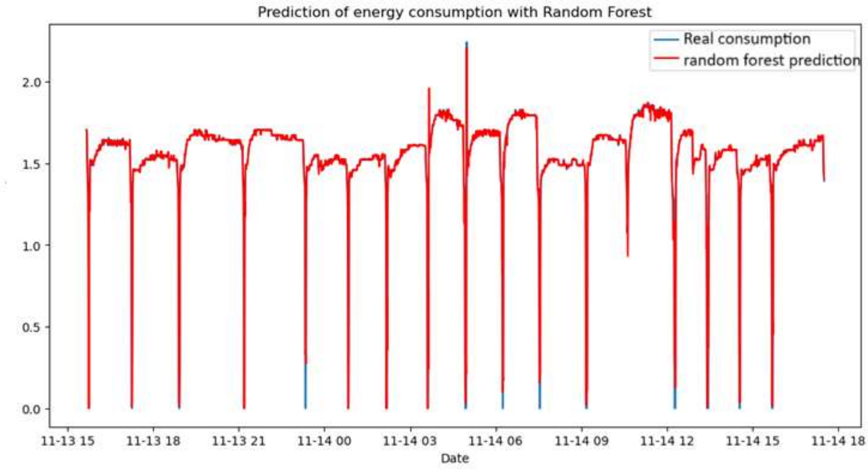
4.1 Prediction of the pulper's energy consumption over time

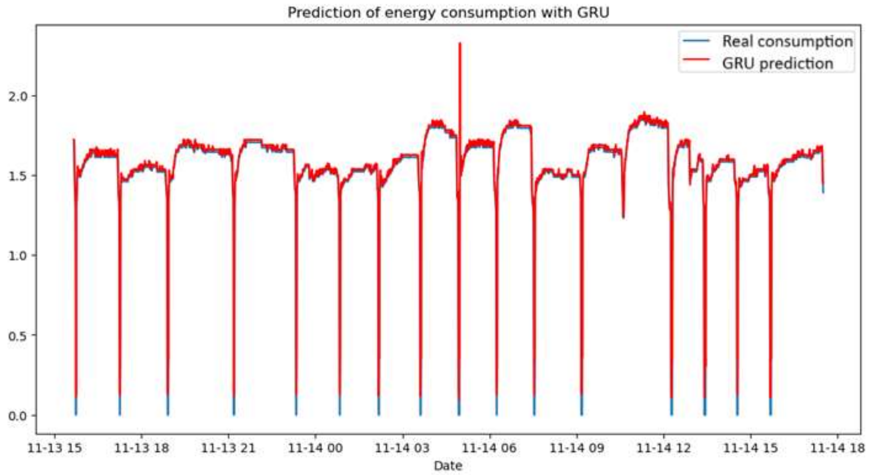
4.1.1 Prediction Plotting: Comparison of Predicted Values vs. Test Data

The objective of this part of the study is to evaluate the prediction quality of the LSTM-based model in comparison with other machine learning models. Initially, this is done by plotting the predictions and the actual validation data (test data) for each model. Fig. 3 illustrates the various curves for each model, providing a visual comparison of their predictive accuracy. The models under comparison include Random Forest, GRU, SVM, and LSTM, the latter being the focus of this study. Note that the test data period spans from November 13, 2023, at 16:00 to November 14, 2023, at 18:00, covering a total of 26 hours of predictions.

Visually, and as shown in Fig. 3, we observe that the predictions from the LSTM, GRU, and Random Forest models closely align with the actual data. LSTM performs better in predicting both drops and peaks compared to the other algorithms, while Random Forest provides good overall predictions but struggles to accurately capture consumption drops. GRU is less accurate, despite having a generally correct trend. Unlike SVM, which is significantly less accurate in capturing major fluctuations.


Prediction of energy consumption with LSTM
real consumption
LSTM prediction
2.5
2.0
1.5
1.0
0.5
0.0
11-13 15
11-13 18
11-13 21
11-14 00
11-14 03
11-14 06
11-14 09
11-14 12
11-14 15
11-14 18
Date


Prediction of energy consumption with Random Forest
Real consumption
random forest prediction
2.0
1.5
1.0
0.5
0.0
11-13 15
11-13 18
11-13 21
11-14 00
11-14 03
11-14 06
11-14 09
11-14 12
11-14 15
11-14 18
Date



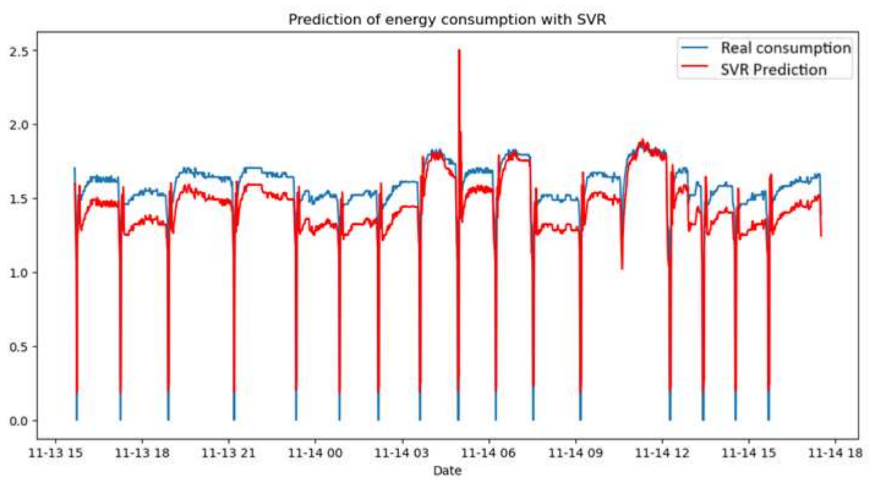


Fig. 3. Prediction of energy consumption using various ML algorithm (LSTM, GRU, Random forest, SVM)

4.1.2 Comparison of Error Metrics across Different Machine Learning Algorithms

In this section, to further detail the comparison, we have included the error metrics for each model. we compare the results of errors metrics across LSTM, Random Forest, GRU, and SVM. The metrics considered include R^2 (Coefficient of Determination), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and MSLE (Mean Squared Logarithmic Error). These metrics allow for a comprehensive assessment of the accuracy and reliability of the predictions made by LSTM, Random Forest, GRU, and SVM models, as summarized in Table. 2.

Table. 2. Results on comparison between differents statistical models used in detection of change point

	LSTM	GRU	SVM	Random forest

R2	0.927	0.922	0.583	0.928
RMSE	0.070	0.072	0.169	0.070
MAE	0.0188	0.0265	0.154	0.0189
MSLE	0.0018	0.0019	0.0058	0.0020

The comparison across models revealed that LSTM achieved an R^2 score of 0.927, indicating that it explains 92.7% of the variance in the energy consumption data, closely matching Random Forest (0.928). Both models performed significantly better than GRU (0.922) and SVM (0.583). In terms of RMSE, LSTM and Random Forest both had a value of 0.070, outperforming GRU (0.072) and SVM (0.169). LSTM also had the lowest MAE at 0.0188, followed closely by Random Forest (0.0189), with GRU at 0.0265 and SVM at 0.154. Finally, regarding MSLE, LSTM achieved a value of 0.0018, slightly better than GRU (0.0019) and Random Forest (0.0020), while SVM had the highest MSLE at 0.0058.

To confirm the performance of the LSTM model and validate its learning progression across epochs, we chose to monitor the Mean Squared Error (MSE) as the error metric to track the model's evolution. Fig. 4 illustrates the temporal evolution of the MSE across both the training and validation datasets over the epochs, thereby providing valuable insights into the model's learning process. The consistent decline in MSE, as shown in Fig. 4, demonstrates that the LSTM model learns effectively while maintaining generalization, thus avoiding overfitting.

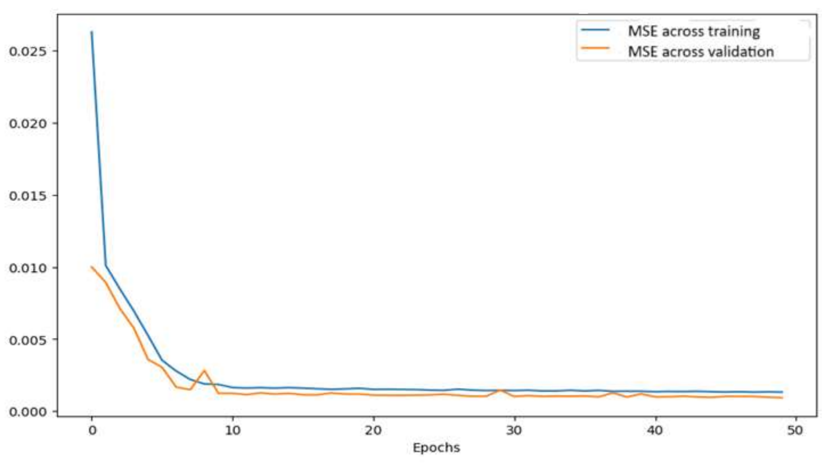


Fig. 4. Mean squared Error across training and validation over the epochs

Given these results, LSTM was chosen as the preferred model due to its ability to consistently deliver strong performance across all error metrics. Its ability to capture short-term and long-term dependencies in time series data makes it particularly well-suited for all task of prediction especially for predicting energy consumption.

4.2 Detection of trend changes in energy consumption

We evaluated eight different models of change point detection in Python as described in section 2.2, to analyse daily energy consumption data from the pulper over a four-month period preceding its mechanical failure.

4.2.1 Analysis of Models 1 and 2: L2 Cost Function with Binary Segmentation and Dynamic Programming

For models 1 and 2, we will use the same cost function, L2 and both binary segmentation and dynamic programming will be employed for segmentation. Table. 3 provides details of the two models, including the results obtained for the change point position, expressed by the change point index and the date of the change point, as well as the difference in average energy consumption before and after the change point. Fig. 5 illustrates the evolution of energy consumption over the four-month period, showing the detected change point and the average consumption before and after the change point for both models.

Table. 3. Result of Dynp and Binseg used in segmentation for the same cost function L2

Model	Segmentation algorithm	change point index	change date	difference (KWh)
1	Dynp	115	24-02-2024	458
2	Binseg	115	24-02-2024	458

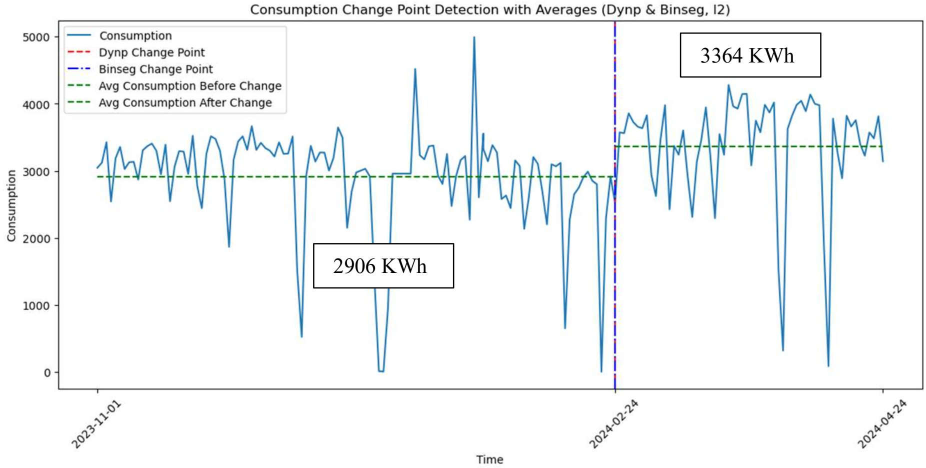


Fig. 5. Evolution of Energy Consumption over 4 Months with change point detection with L2 cost function and Dynp and Binseng

As shown in Fig. 5, both the Dynp and Binseg segmentation algorithms detected the same change point on February 24, 2024, corresponding to index 115. The difference in average energy consumption before and after the change point is approximately 458 kWh, indicating a 15.6% increase in energy overconsumption.

4.2.2 Analysis of Models 3 and 4: L1 Cost Function with Binary Segmentation and Dynamic Programming

For models 3 and 4, the same segmentation approaches—binary segmentation and dynamic programming—are used; however, the cost function applied is L1 instead of L2. Table. 4 presents the details of these models, including the results for the change point position, expressed by the change point index and the date of occurrence, as well as the difference in average energy consumption before and after the change point. Fig. 6 illustrates the evolution of energy consumption over the four-month period, indicating the detected change point and the average consumption before and after this point for both models.

Table. 4. Result of Dynp and Binseg used in segmentation for the same cost function L1

Model	Segmentation algorithm	change point	change date	difference (KWh)
3	Dynp	140	21-03-2024	450
4	Binseg	140	21-03-2024	450

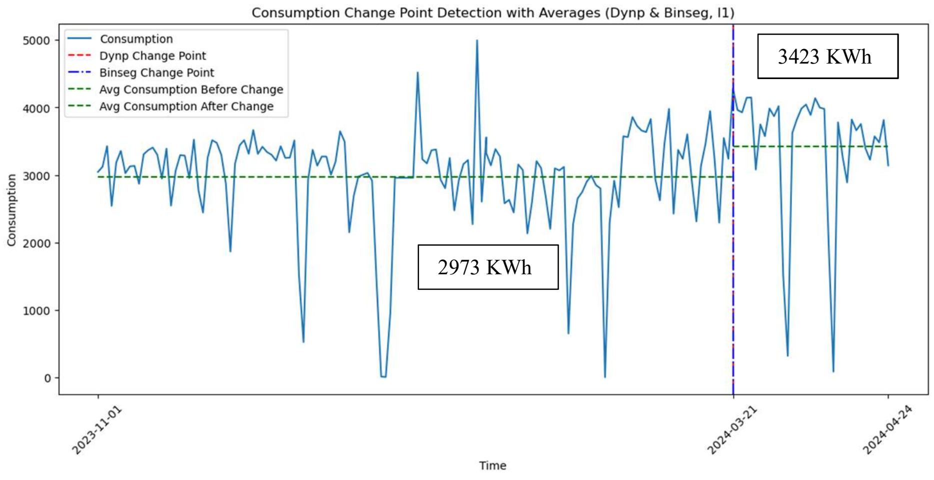


Fig. 6. Evolution of Energy Consumption over 4 Months with change point detection with l1 cost function and Dynp and Binseg

As shown in Fig. 6 and Table 4, both the Dynp and Binseg segmentation algorithms detected the same change point on March 21, 2024, corresponding to index 140. The difference in average energy consumption before and after the change point is approximately 450 kWh, indicating a 15.1% increase in energy overconsumption.

4.2.3 Analysis of Models 5 and 6: rbf Cost Function with Binary Segmentation and Dynamic Programming

For models 5 and 6, we will use the same cost function, rbf. Both binary segmentation and dynamic programming will be employed for segmentation. We will present a graph comparing the results of the two models to examine the impact of changing the segmentation algorithm. The results are shown below in table. 5 and fig. 7.

Table. 5. Result of Dynp and Binseg used in segmentation for the same cost function rbf

Model	Segmentation algorithm	change point	change date	difference (KWh)
5	DYNP	115	24-02-2024	458

6	BINSEG	115	24-02-2024	458

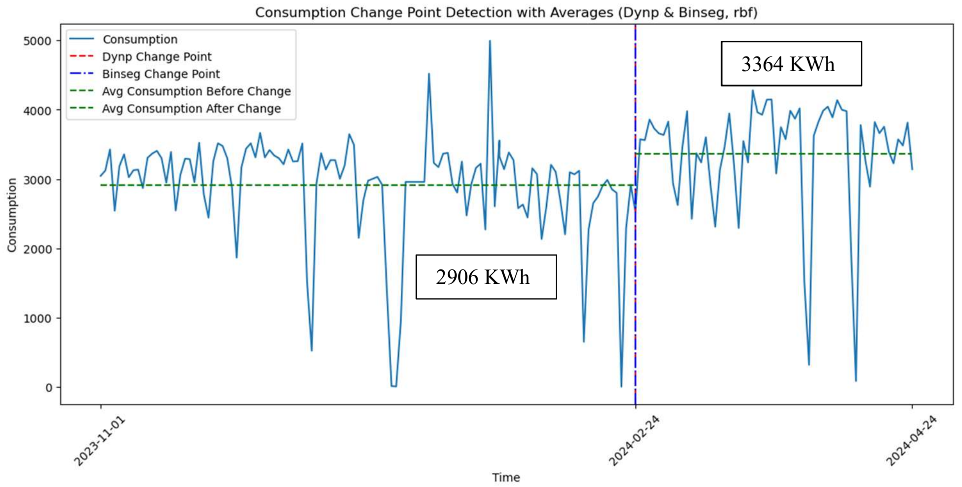


Fig. 7. Evolution of Energy Consumption over 4 Months with change point detection with rbf cost function and Dynp and Binseg

As shown in Fig. 7 and Table 5, both the Dynp and Binseg segmentation algorithms detected the same change point on February 24, 2024, corresponding to index 115. The difference in average energy consumption before and after the change point is approximately 458 kWh, indicating a 15.6% increase in energy overconsumption.

4.2.4 Analysis of Models 7 and 8: Normal Cost Function with Binary Segmentation and Dynamic Programming

As with the previous models, models 7 and 8 apply the same approach by using binary segmentation and dynamic programming for change point detection, but with a different cost function. In this case, the normal cost function is employed. Table. 6 summarizes the change point position, including the index and corresponding date, as well as the difference in average energy consumption before and after the change point. Fig. 8 illustrates the evolution of energy consumption over the four-month period, highlighting the detected change point and the average consumption before and after for both models.

Table. 6. Result of Dynp and Binseg used in segmentation for the same cost function normal

Model	Segmentation algorithm	change point	change date	difference (KWh)
7	DYNP	40	11-12-2023	-124
8	BINSEG	40	11-12-2023	-124

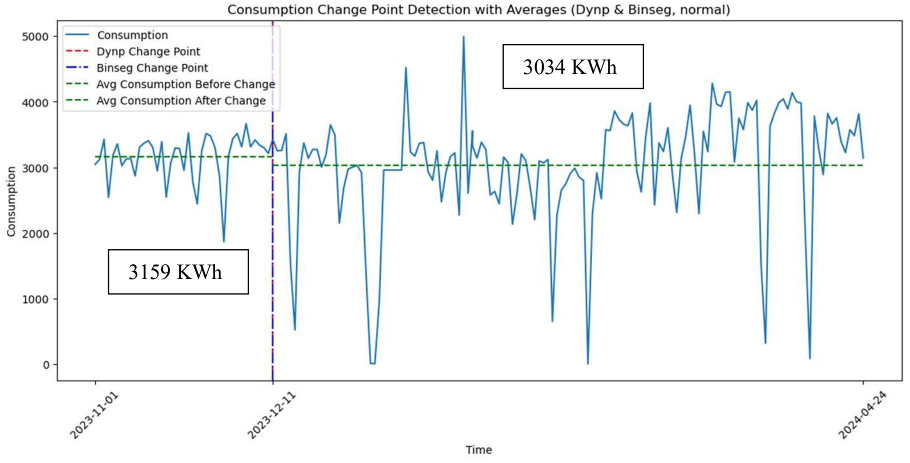


Fig. 8. Evolution of Energy Consumption over 4 Months with change point detection with normal cost function and Dynp and Binseg

As shown in Fig. 8 and Table 6, both the Dynp and Binseg segmentation algorithms detected the same change point on December 11, 2023, corresponding to index 40.

As a general result, changing the segmentation algorithm did not affect the outcomes when using the same cost function. However, changing the cost function produced different outcomes: models based on the L2 and RBF cost functions detected a change at index 115, models using the L1 cost function detected a change at index 140, and models based on the Normal distribution detected a change at index 40. The point at index 115 is more significant as it shows a larger difference in average energy consumption before and after this detected change point. Models based on L1 detected the change point almost a month later, while the Normal model detected a change point early in the time series variation, which is far from reality. The Normal model appears distinct from the others due to its mechanism based on the normal distribution, making it sensitive to small changes in the distribution. In contrast, models based on the L2 and RBF cost functions are more consistent with each other, detecting changes at similar points with significant differences in energy consumption before and after the change point.

The effectiveness of the statistical method for detecting changes in energy consumption trends is demonstrated by the case study's results, which are verified by calculating the excess energy consumption of 27052 KWh, equivalent to 19477 Kg CO2e that would result from two months of operating the pulper in degraded mode.

Conclusion

This study addresses a critical issue in industrial maintenance by focusing on a paper recycling plant and utilizing real-world data to enhance energy management and predictive maintenance. The dataset, comprising daily energy consumption records from industrial equipment and high-frequency measurements taken every 36 seconds, facilitated the development of a Machine Learning (ML)-based prediction model.

The study is divided into two key components. First, we developed a robust energy consumption prediction model designed to anticipate shifts in consumption patterns. This model enables decision-makers to schedule maintenance proactively, thereby preventing unnecessary energy overconsumption and reducing greenhouse gas emissions. Our data-driven methodology, validated by the factory's engineering team, offers a valuable approach that can be adapted to optimize similar processes in other industrial contexts. Overall, our strategy not only reduces operational costs but also contributes to minimizing environmental impact. By extending this approach to other industrial settings, it can serve as a practical tool for enhancing both energy efficiency and sustainability across diverse sectors.

Second, we employed statistical methods to accurately identify change points in the energy consumption time series, based on data collected over four months leading up to the equipment failure due to bearing issues. This change detection was effectively linked to the equipment's health status, allowing us to estimate energy overconsumption up to two months before the actual failure occurred.

Despite its limitations, such as the lack of extensive historical failure data on mechanical or electrical equipment failures, this study confirms the link between predictive maintenance and energy efficiency. To develop a comprehensive strategy, future research should focus on cataloguing various failure states and their impacts on energy consumption, thereby enhancing the predictive model's accuracy and applicability.

References

1. S. El Majaty, A. Touzani, Y. Kasseh, Circular Economy as an Important Lever to Reduce Greenhouse Gas Emissions: Case of an Electricity Distribution Company in Morocco, Nat. Environ. Pollut. Technol. 23, 819–828 (2024).
2. H. Dong, L. Zhang, Transition towards carbon neutrality: Forecasting Hong Kong's buildings carbon footprint by 2050 using a machine learning approach, Sustain. Prod. Consum. 35, 633–642 (2023).
3. Bains, K. Sridhar, S.B. Dhull, P. Chawla, M. Sharma, P.K. Sarangi, V.K. Gupta, Circular bioeconomy in carbon footprint components of nonthermal processing technologies towards sustainable food system: A review, Trends Food Sci. Technol. 149, 104520 (2024).
4. Beisheim, K. Rahimi-Adli, S. Krämer, S. Engell, Energy performance analysis of continuous processes using surrogate models, Energy 183, 776–787 (2019).
5. Z. Geng, R. Zeng, Y. Han, Y. Zhong, H. Fu, Energy efficiency evaluation and energy saving based on DEA integrated affinity propagation clustering: Case study of complex petrochemical industries, Energy 179, 863–875 (2019).
6. L. Ye, P. Du, S. Wang, Industrial carbon emission forecasting considering external factors based on linear and machine learning models, J. Clean. Prod. 434, 140010 (2024).
7. P.R.X. do Carmo, J.V.L. do Monte, A.T. Filho, E. Freitas, M.F.F.S.L. Tigre, D. Sadok, J. Kelner, A data-driven model for the optimization of energy consumption of an industrial production boiler in a fiber plant, Energy 284, 129239 (2023).
8. T. Qiao, Adaptive optimization of thermal energy and information management in intelligent green manufacturing process based on neural network, Therm. Sci. Eng. Prog. 55, 102953 (2024).
9. P. Bangalore, L.B. Tjernberg, An artificial neural network approach for early fault detection of gearbox bearings, IEEE Trans. Smart Grid 6, 980–987 (2015).
10. M.A. Bermeo-Ayerbe, C. Ocampo-Martinez, J. Diaz-Rozo, Data-driven energy prediction modeling for both energy efficiency and maintenance in smart manufacturing systems, Energy 238, 121691 (2022).

11. S. Touzani, B. Ravache, E. Crowe, J. Granderson, Statistical change detection of building energy consumption: Applications to savings estimation, Energy Build. 185, 123–136 (2019).
12. Z. Shi, A. Chehade, A dual-LSTM framework combining change point detection and remaining useful life prediction, Reliab. Eng. Syst. Saf. 205, 107257 (2021).
13. T.P. Carvalho, F.A.A.M.N. Soares, R. Vita, R.P. Francisco, J.P. Basto, S.G.S. Alcalá, A systematic literature review of machine learning methods applied to predictive maintenance, Comput. Ind. Eng. 137, 106024 (2019).
14. M. Sega, Y. Xiao, Multivariate random forests, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 1, 80–87 (2011).
15. A.K. Abbas, N.A. Al-haideri, A.A. Bashikh, Implementing artificial neural networks and support vector machines to predict lost circulation, Egypt. J. Pet. 28, 339–347 (2019).
16. G.P. Zhang, Neural Networks for Classification: A Survey (2013).
17. S.H. Noh, Analysis of gradient vanishing of RNNs and performance comparison, Inf. 12 (2021).
18. S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Comput. 9, 1735–1780 (1997).
19. J. Chung, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, arXiv:1412.3555v1 [cs.NE] (2014).
20. N.R. Zhang, D.O. Siegmund, A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data, Biometrics 63, 22–32 (2007).
21. Truong, L. Oudre, N. Vayatis, ruptures: change point detection in Python (2018).
22. P. Fryzlewicz, Wild Binary Segmentation for multiple change-point detection (2014).
23. K. Yano, l1 Trend Filtering-Based Detection of Short-Term Slow Slip Events: Application to a GNSS Array in Southwest Japan, J. Geophys. Res.: Solid Earth (2022).
24. R. Killick, P. Fearnhead, Optimal detection of changepoints with a linear computational cost (2014).
25. Pagano, A predictive maintenance model using Long Short-Term Memory Neural Networks and Bayesian inference, Decis. Anal. J. 6, 100174 (2023).
26. L. De Simone, E. Caputo, M. Cinque, A. Galli, V. Moscato, S. Russo, G. Cesaro, V. Criscuolo, G. Giannini, LSTM-based failure prediction for railway rolling stock equipment, Expert Syst. Appl. 222, 119767 (2023).
27. Y. Jiang, P. Dai, P. Fang, R.Y. Zhong, X. Zhao, X. Cao, A2-LSTM for predictive maintenance of industrial equipment based on machine learning, Comput. Ind. Eng. 172 (2022).