Data-Driven Quality Assurance: Defect Prediction to Reduce Carbon Footprint in Manufacturing Logistics

Rokaya Douiri[1*], *Chaymae* Taraa[1], *Haj* Elmoussami[1] and *Latifa* Ezzine[1]

ENSAM school, Engineering of complex systems and structures, Meknes, Morocco.

Abstract. In modern manufacturing environments, minimizing carbon emissions while maintaining product quality has become a strategic imperative. One critical yet often overlooked contributor to elevated CO_2 emissions is the occurrence of unexpected quality defects, particularly those caused by human error for which no permanent poka-yoke solution exists. These defects frequently disrupt production flows, delay customer deliveries, and necessitate urgent corrective shipments leading to increased use of premium, carbon-intensive transportation. Traditional production planning systems typically neglect the variability introduced by non-conforming parts, assuming uniform quality across all outputs. To address this and to build on Industry 4.0 and Quality 4.0 frameworks, the research compares statistical and machine learning models for defect prediction using six months of real-world production data. An automatic training and testing procedure is applied to evaluate model performance, enabling an automatic selection of the optimal method per workstation. This allows for precise estimation of defect rates across the production line, supporting dynamic and environmentally-conscious planning decisions. By incorporating predicted quality deviations into daily planning routines, the model not only enhances on-time delivery performance but also reduces the reliance on expedited transport, contributing to significant carbon footprint reduction and more sustainable operations.

1 Introduction

This study aims to develop an automated system for predicting human-induced quality defects that lack preventive solutions, by comparing and selecting the most effective statistical and machine learning models for each workstation. Through a continuous training and testing process, the optimal model is identified per station and used to estimate the overall defect rate across the production line. By anticipating non-conformities in advance, the proposed approach supports decision-makers in avoiding last-minute deliveries, thereby reducing both CO_2 emissions and the operational costs associated with expedited logistics.

[*] Corresponding author: r.douiri@edu.umi.ac.ma

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

Quality issues continue to be a major source of disruption in production planning, particularly in high-volume, high-mix environments. Despite advancements in quality assurance techniques, many companies still adopt reactive strategies, where defects are only addressed after they have already affected the process. These disruptions may cause delays, require additional human or material resources, and ultimately harm customer satisfaction. While Statistical Process Control (SPC) and Six Sigma have improved defect detection, they do not necessarily offer predictive capabilities.

Over the past decade, predictive analytics has gained significant traction in the manufacturing sector. Recent studies have demonstrated that time series models such as SARIMA remain effective for forecasting trends in areas such as equipment reliability and production demand (Nguyen et al., 2023). Concurrently, machine learning algorithms particularly Multi-Layer Perceptrons (MLP), Long Short-Term Memory networks (LSTM), and Gradient Boosting—have shown improved accuracy in identifying patterns and anomalies, especially in defect detection and predictive quality control (Ali et al., 2024; Singh & Mehta, 2024). Moreover, explainable AI approaches have recently been proposed to enhance the interpretability of predictive models in industrial quality monitoring (Chen et al., 2024). Despite these advancements, the application of both statistical and machine learning models to forecast defect rates at the level of individual workstations or product categories remains limited, particularly in operational environments where variability and data granularity challenge model robustness.

To address this gap, the present study leverages six months of historical defect data collected from a real production line in the automotive sector. Each predictive model is automatically trained and validated on this dataset, and their performance is automatically evaluated using standard metrics such as RMSE and MAE. The study further discusses the operational implications of integrating defect forecasts into production decision-making including improved planning robustness and a potential reduction in last-minute corrective actions.

This paper is structured as follows: the first section reviews existing literature on defect prediction and quality analytics in manufacturing. The second outlines the methodology and data pre-processing steps for this case study. The third presents and compares the results obtained from each model. The paper concludes by highlighting practical recommendations for industrial implementation and directions for future research.

2 Literature review

This literature review synthesizes findings from a wide range of studies on defect prediction, quality management, and AI-driven manufacturing, highlighting key methods, applications, limitations, and future directions (Chinenye, Anyachebelu, et Abdullahi 2023) (Joseph Nnaemeka Chukwunweike et al. 2024). Ensemble and tree-based machine learning models, particularly Random Forest, AdaBoost, and XGBoost, consistently demonstrate strong predictive performance across various NASA datasets, reinforcing their utility in software quality assurance (Chinenye et al. 2023) (Alsaeedi et Khan 2019) (Ihsan Aquil 2020) (Sankhye et Hu 2020) (Iqbal et al. 2019) (Matloob et al. 2021). To enhance accuracy and efficiency, novel feature selection approaches such as MOFES and Information Gain have been proposed, effectively reducing dimensionality and computational load while maintaining model performance (Ni et al. 2019) (Balogun et al. 2019). In addition to algorithmic advancements, several studies emphasize real-world applications, including quality control in low-volume manufacturing, predictive monitoring using DPU-charts, and customer complaint handling demonstrating the operational relevance of these models (Verna et al. 2022) (Silva, Machado, et Sampaio 2024). Edge computing and AI integration in Industry 5.0 have also enabled significant data reduction while preserving analytical value, supporting real-time decision-making for quality and process management (Bajic et al. 2023) (Ejjami et Boussalham 2024) Elżbieta Skrzypek et al. 2025). Deep learning techniques and data augmentation (e.g., CVAE) further improve defect detection, particularly for imbalanced datasets in industrial scenarios (Yun et al. 2020) (Joseph Nnaemeka Chukwunweike et al. 2024). Unsupervised learning methods such as Fuzzy CMeans and Fuzzy SOMs show promise in cross-project defect prediction, performing comparably to supervised models in some cases (Li, Shepperd, et Guo 2020). Moreover, studies on Zero-Defect Manufacturing (ZDM) and evolving quality paradigms (Quality 4.0 and 5.0) frame a transition toward holistic, sustainable, and data-driven approaches, though current frameworks lack standardization, long-term empirical validation, and implementation metrics (Thota, Shajin, et Rajesh s. d.) (Psarommatis et al. 2020) (Elżbieta Skrzypek et al. 2025) (Psarommatis et al. 2022). Despite these advances, several research gaps persist. Many works rely on limited or imbalanced datasets, lack hyperparameter tuning, or narrowly apply algorithms and metrics, which constrains generalizability and robustness (Ni et al. 2019) (Iqbal et al. 2019) (Thota et al. s. d.) (Pachouly et al. 2022). Internal and external validity concerns are frequently noted, along with insufficient attention to dataset validation, interpretability, and real-world adaptability. Further research is needed to address class imbalance, explore filter-based and deep learning methods, improve cross-project prediction, and integrate tools that offer richer outputs such as defect severity or resource allocation (Ni et al. 2019) (Matloob et al. 2021). Practical deployment is also hindered by data silos, cybersecurity risks, workforce skill gaps, and high implementation costs, underscoring the need for secure, scalable, and standardized solutions (Elżbieta Skrzypek et al. 2025) (Joseph Nnaemeka Chukwunweike et al. 2024).

Recent studies have explored machine learning approaches for predicting CO2 emissions from vehicles to address environmental concerns. Multiple Linear Regression (MLR), Support Vector Machine (SVM), and Random Forest (RF) algorithms have been investigated for their effectiveness in emission prediction (S.V et al., 2024; K. Bussaban et al., 2024). While S.V et al. (2024) found MLR to outperform SVM, K. Bussaban et al. (2024) reported SVM achieving the highest precision. These supervised learning techniques have shown promise in training models based on experience (Pooja S Kadam & Suhasini Vijayumar, 2018). However, traditional machine learning methods face limitations in capturing the complexity of emissions data. To overcome these challenges, Automated Machine Learning (AutoML) has emerged as a powerful tool, demonstrating improved prediction capabilities and efficiency compared to conventional approaches (Senthilselvi et al., 2024). These advancements in CO2 emission prediction contribute to global efforts in environmental protection and informed policy-making.

The literature review shows a GAP in linking the CO2 emission with urgent deliveries related to unexpected quality defects in production lines.

3 Case study

3.1 Problematic overview

To quantify the impact of quality defects on resources, an analysis of the reworked quantities was conducted over a six-month period for a database of 10 000 rows. This visualization highlights the variability of defects over time and emphasizes the need to consider them in capacity planning.

In addition to the line-level analysis, a more detailed study was conducted at the workstation level to identify those contributing most to the observed defects. A summary table shows the distribution of defects by workstation, allowing precise localization of risk areas in the production process.

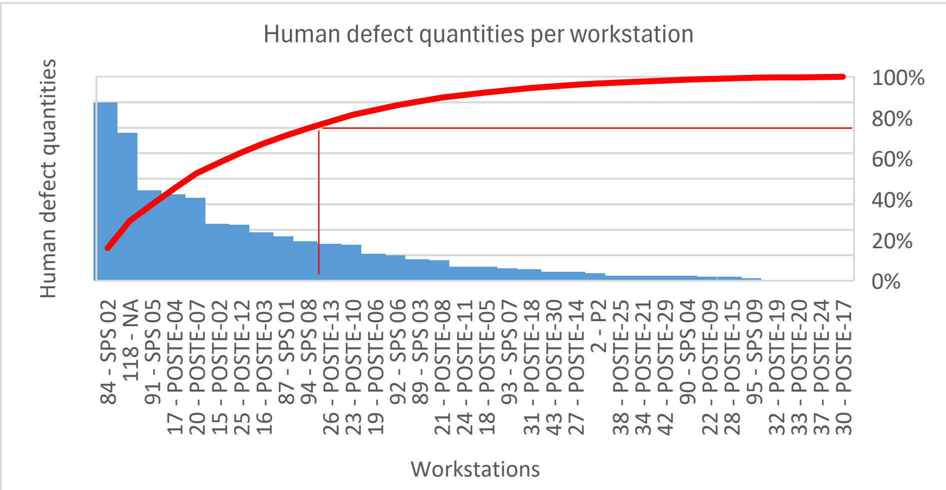


Figure 1: Human defect quantities per workstation

The Pareto chart illustrates the distribution of human defect quantities across different workstations. It shows that a small number of workstations are responsible for the majority of defects. Specifically, the first few workstations contribute to a large proportion of the total defects, while the remaining ones have relatively minor contributions.
To analyze the distribution of human defect quantities, the workstations can be grouped into two categories following the Pareto principle (80/20 rule). According to the chart, a small number of workstations account for the majority of human defects.

Table 1: Workstation's categories

Category	Description	Example of Workstations (from the chart)
Category 1 Critical Workstations (High defect rate)	Workstations responsible for approximately 80% of total human defects	From 84 – SPS 01 to 94 – SPS 08
Category 2 Non-critical Workstations (Low defect rate)	Workstations contributing to the remaining 20% of defects	From 26 – POSTE -13 to 30 – POSTE -17

3.2 Principal Component Analysis (PCA)

Within the scope of this research, a comprehensive investigation of the interrelationships between production line workstations and the types of defects produced was performed through Principal Component Analysis (PCA), The database used for this analysis consists of 5,831 rows and 13 columns. Each row represents a workstation from the most critical production line under study, while each column corresponds to a specific type of defect

generated. The values in the table indicate the quantity of each defect type produced by each workstation. This dataset served as the basis for conducting the Principal Component Analysis (PCA) to explore the relationships between workstations and the types of defects they generate
The following figure illustrates the principal components ranked according to their importance in XLSTAT.

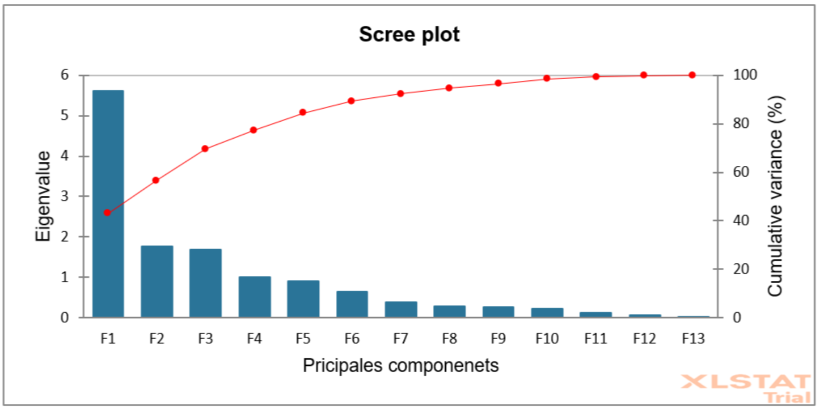


Figure 2: The principal components ranked according to their importance in XLSTAT

F1 is the most important principal component, explaining 43.128% of the total variability in the data. This means that this single component captures a large portion of the information contained in the original dataset.
F2 and F3 explain 13.584% and 12.998% of the variability, respectively. Together, the first three components account for more than 69% of the total variability, which suggests that the dimensionality of the data can be reduced by retaining these three components and discarding the others with lower variability.
After identifying the most significant principal components, it is essential to visualize how the variables and observations are positioned in the factorial space. The following graph provides a joint representation of both elements to facilitate the interpretation of relationships within the dataset. The workstations are represented by points, while the defects are shown as vectors (arrows) in the same factorial plane. This visualization helps to identify correlations between workstations and the types of defects they are most associated with.

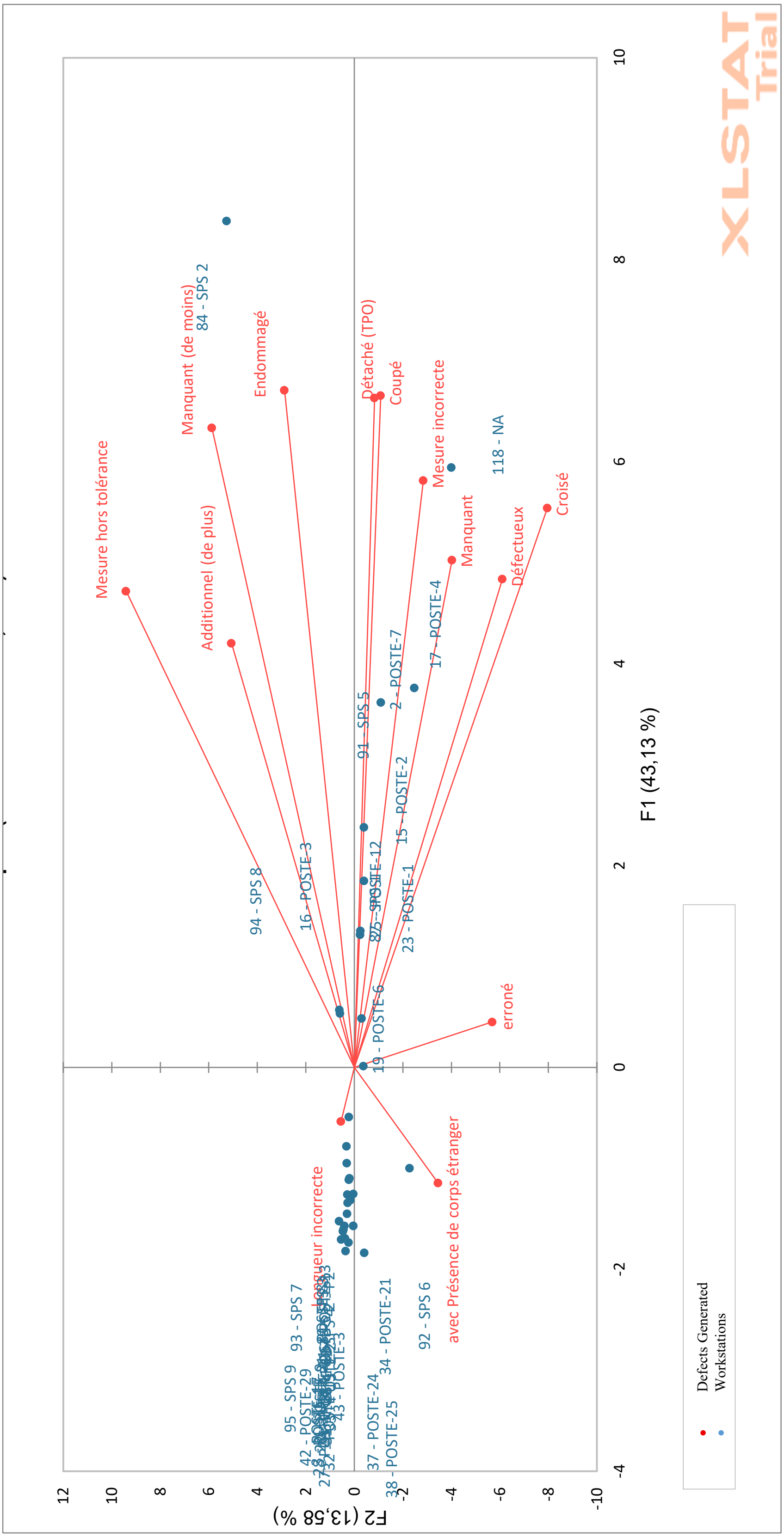


Figure 3: Correlation analysis between different workstations and type of defect generated

This graph serves as a key tool to target the workstations that contribute the most to specific defects. The results of the Principal Component Analysis revealed significant links between certain stations on the line and specific types of defects.
The analysis of correlations between stations showed weak interdependence in defect rates, indicating that the behaviour of one workstation is not directly representative of others. As a result, it is not relevant to model the entire line globally.
Prediction was therefore carried out station by station, in order to more accurately capture the specificities of each workstation and improve the model's precision.

3.3 Analysis of manpower impact

The analysis of the Table below revealed a significant gap between the quantities of defects generated by different operators, indicating a direct impact of the operators on quality, especially at stations where operations are performed manually.
This variation suggests that human factors such as training level, attentiveness or fatigue, may contribute to the discrepancies observed. Although these elements are difficult to quantify and thus cannot be directly integrated into predictive models, they highlight areas for targeted improvement actions, such as:

- Implementing tailored training programs.
- Conducting operator-specific quality audits.

Table 2: Number of produced defects per operator

Operator ID	Sum of produced defects
6085	347
7347	140
6006	77
5591	67
5753	57
5776	46
6052	40
7175	29
6157	17
4155	15
6669	12
6799	7
8141	4
5778	4
13580	3
8312	2
6229	2
7302	2
13590	1
236	1

3.4 Impact of production volume on the number of defects generated

This graph highlights a growing correlation between production volume and the number of recorded defects.

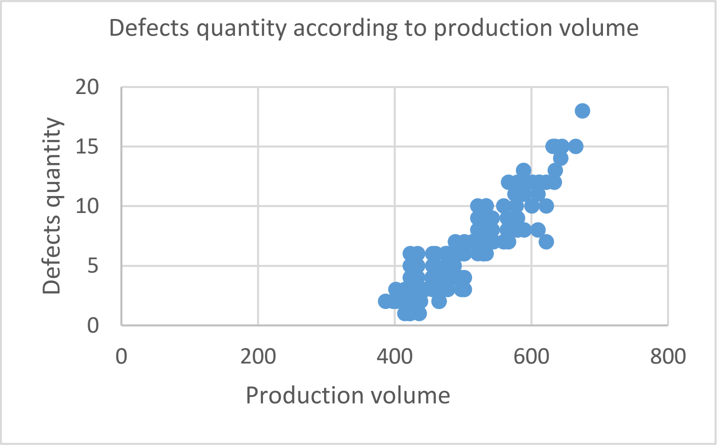


Figure 4: Defects quantity according to production volume

The trend shows that as production volume increases, the number of defects rises as well, reflecting a direct effect of production load on quality.
This observation reinforces the relevance of including production volume as an explanatory variable in predictive models, as it is a key factor in defect occurrence. Incorporating this variable into forecasting approaches will allow for better anticipation of rework needs and optimization of actual capacity planning.

3.5 Analysis of the Impact of the Temporal Factor on Defects

In order to detect potential recurring patterns related to the organization of production, an analysis was conducted on the distribution of defects according to the day of the week. This approach aims to identify whether certain days are more prone to defect occurrence.

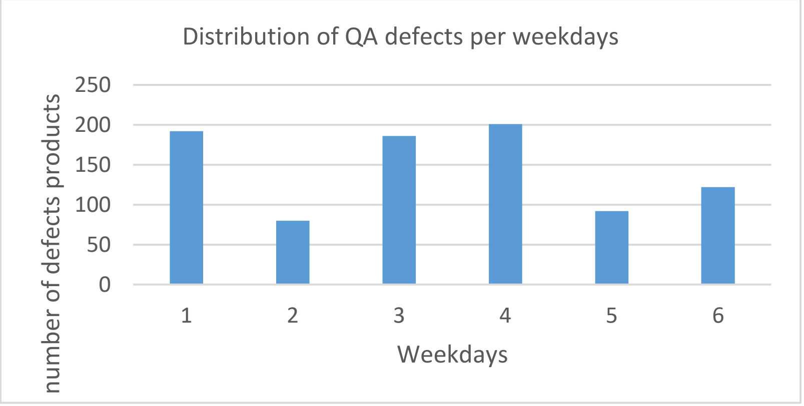


Figure 5: Distribution of quality defects per weekdays

The interpretation of the chart presented in that figure reveals a significant variation in the number of defects depending on the production day. Specifically, Days 1, 3, and 4 show considerably higher defect levels compared to Days 2, 5, and 6. This disparity may be attributed to various factors, such as:
Human factors: accumulated fatigue, reduced alertness, or fluctuating concentration levels.
Organizational factors: team changes, beginning or end of the workweek.
The influence of the temporal factor highlights the relevance of incorporating the "production day" variable into predictive models. Doing so can enhance the accuracy of forecasts, improve the anticipation of non-conformity peaks, and help guide corrective actions toward specific days or conditions.

3.6 Prediction Using the SARIMA Model

To anticipate defect quantities and better integrate their impact into production planning, a forecasting approach was undertaken. This step aims to provide a decision support tool upstream, allowing the detection of potential non-conformity peaks and guiding proactive corrective actions.
Initially, a statistical model based on SARIMA was applied to evaluate forecasting capability using historical data.

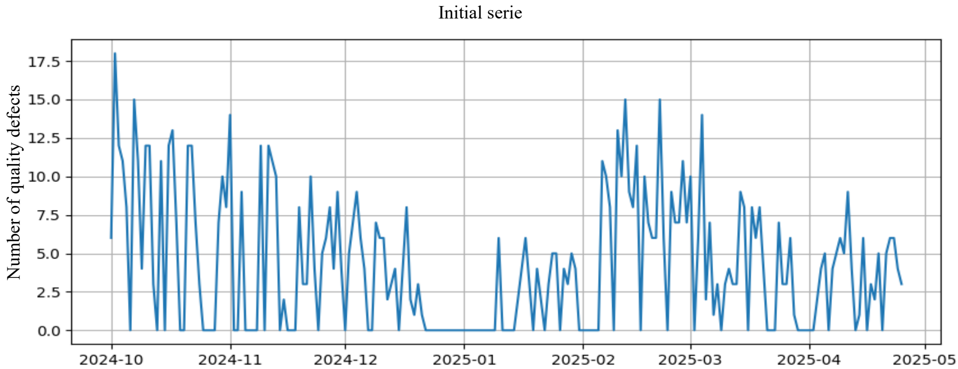


Figure 6: Ititial SARIMA series generated by Python

Before constructing a SARIMA forecasting model, it is essential to determine the optimal values of the parameters p, d, q, P, D, and Q, which represent respectively the autoregressive components, differencing, moving averages, and their seasonal counterparts.
To achieve this, the autocorrelation function (ACF) and partial autocorrelation function (PACF) graphs are used:

- The ACF graph identifies overall dependencies in the data, notably the presence of seasonal structures or significant autocorrelations.
- The PACF graph helps detect direct relationships between observed values and their lags by filtering out the influence of other lags.

These two tools are particularly useful for selecting the orders p and q and their seasonal equivalents P and Q based on the decay of correlations and their statistical significance. After analyzing these graphs, the initial parameters for the SARIMA model can be defined for fitting and validation.

For example, the ACF graph used to determine parameter q examines how many bars significantly exceed the confidence bounds (blue lines), from which q = 3 was selected. The figure below shows the resulting graph generated by the Python program.

ACF - q parameter

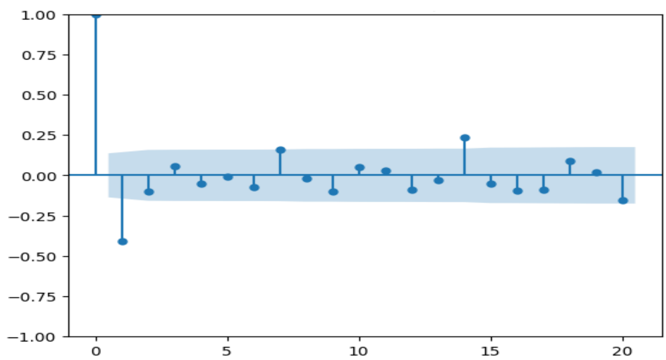


Figure 7: q parameter

The PACF graph is used to determine the parameter P: similarly, we observe how many bars significantly exceed the confidence bounds (blue lines), from which P = 5 was identified, as illustrated in the following figure.

PACF - p parameter

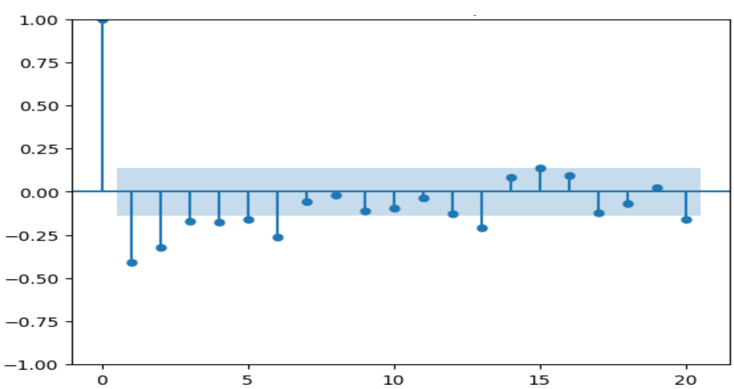


Figure 8: p parameter

Since the series does not exhibit a pronounced trend but shows slight non-stationarity, it is reasonable to assume that a first-order differencing (d = 1) will be sufficient to render the series stationary. Indeed, a weak trend can often be eliminated by a single differencing, allowing for the effective application of a SARIMA model.
Similarly, to determine the seasonal parameters P, Q, and D, Figure below presents the ACF and PACF graphs.

P = 5 and Q = 5 based on ACF and PACF graphs.
The data are daily, and the behavior exhibits a recurring monthly pattern; therefore, it is logical to set:
s = 30, since approximately 30 days ≈ 1 month.

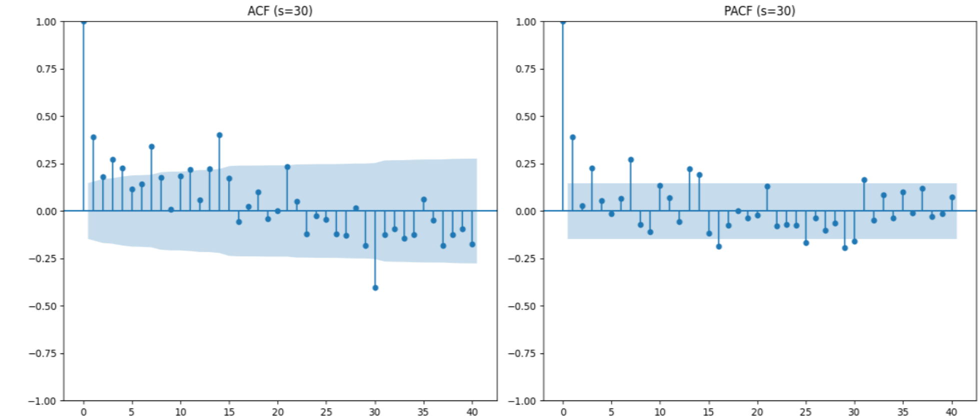
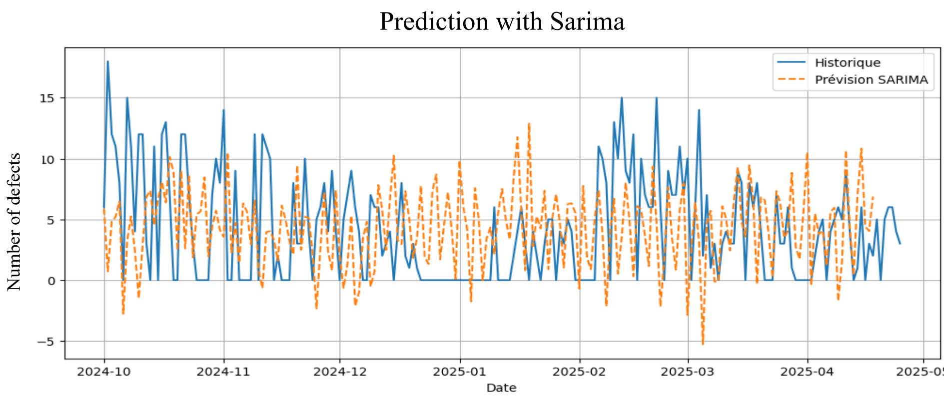


Figure 9: ACF & PACF graphs

The SARIMA model was applied to predict defect quantities based on the collected time series data. After selecting the optimal parameters, the time series was modeled, and the predictions were compared to the actual values



.

Figure 10: Prediction with SARIMA

The prediction graph shows that the model generally follows the trend of defects well, capturing both the overall tendency and potential seasonal components. However, a more detailed analysis was performed by plotting the prediction errors (the difference between actual and predicted values), as well as by conducting a normality test.

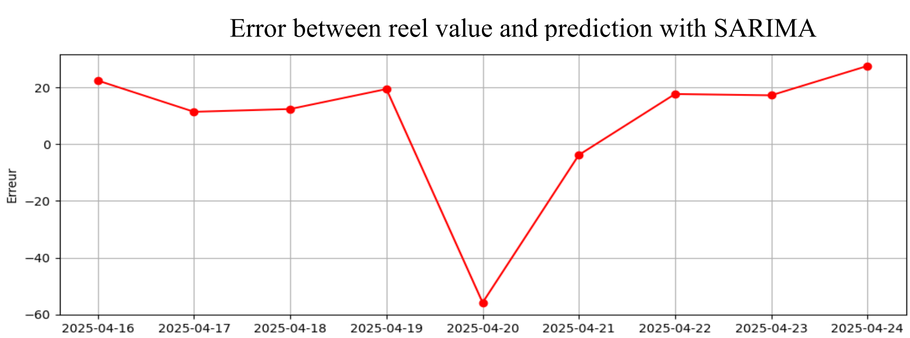


Figure 11: Error plot

The analysis of the forecast error normality, shown in Figure 38, is a crucial step in evaluating the adequacy of the applied model. Indeed, many statistical models, such as SARIMA models, rely on the assumption that the residuals follow a standard normal distribution. If this assumption is not met, it may indicate that the model does not adequately capture the underlying structure of the data, particularly in the presence of nonlinearities or complex interaction effects.

Figure 12: Error distribution

Shapiro-Wilk Test: This test is specifically designed to detect deviations from normality in small samples. It compares the distribution of observed data to a theoretical normal distribution.

- Null hypothesis (H_0): The data follow a normal distribution.
- If the p-value ≤ 0.05, H_0 is rejected: the data are not normally distributed.

→ The residuals do not follow a normal distribution ($p \leq 0.05$).

D'Agostino's Test: This test combines measures of skewness and kurtosis to assess whether the data deviate from a normal distribution.

- H_0: The data follow a normal distribution.
- A p-value ≤ 0.05 indicates non-normality.

→ Non-normal distribution according to D'Agostino's test ($p \leq 0.05$).

Kolmogorov-Smirnov (K-S) Test: This test measures the maximum distance between the empirical distribution function of the data and that of a normal distribution.

- A p-value ≤ 0.05 also indicates deviation from normality.

→ Non-normal distribution according to the K-S test ($p \leq 0.05$).

The obtained results show that the residuals from the SARIMA model do not follow a normal distribution, which limits the model's reliability in capturing the actual behavior of the process. This observation justifies the use of machine learning techniques, which are better suited for data with nonlinear or complex relationships. These methods, being less dependent on classical statistical assumptions, offer greater generalization capabilities and can improve prediction accuracy in an industrial context characterized by multiple sources of variability.

3.7 Prediction Using Machine Learning Models

3.7.1 Prediction per workstation

As a continuation of the forecasting approach, this section focuses on the application of machine learning models to estimate defect quantities. Models such as neural networks and decision trees enable the effective exploitation of multidimensional data and the capture of complex patterns among explanatory variables. Their implementation aims to enhance prediction accuracy by overcoming the limitations observed with traditional statistical models.
Before implementing machine learning models, an essential phase of data understanding and preparation was carried out.
The dataset used for our analysis consists mainly of three variables:
Production volume
Day of the week (represented by an integer from 1 to 7), covering 133 days equivalent to 6 months
The workstations concerned by the prediction: 34 in total
The number of defects observed per workstation
Encoding of Variables
The day of the week was kept as a simple numeric variable, without using cyclic encoding. This decision was justified by the absence of strong seasonality in the data, as demonstrated during the exploratory analysis.
Handling Missing Values
The dataset contains a large number of zeros in the defect's column. After inspection, these values were found not to represent missing data, but rather actual situations where no defects were detected. They were therefore retained, as they are crucial for training a realistic model capable of distinguishing between good and defective days.
Decision Not to Normalize
The production volume was used as-is, without normalization, in order to preserve its scale and concrete interpretability in the prediction results.
The goal of this step is to predict the daily number of defects per workstation based on two explanatory variables: day of the week and production volume. These predictions are then used to estimate the defect rate across the entire production line.
For each workstation, four supervised regression models were tested:
Decision Tree Regressor
Random Forest Regressor
Gradient Boosting Regressor
Multi-Layer Perceptron Regressor (Neural Network)

These models were chosen for their ability to capture non-linear relationships and potential interactions between variables.

For each workstation, an automated training and validation process was performed. Using TimeSeriesSplit cross-validation, the code evaluates each model's performance based on the Mean Squared Error (MSE). The model with the lowest MSE is selected.
The code uses a scientific and automated approach to select the best model for each production workstation. Here is the full process:
Data → Train/Test Split → For Each Model → Hyperparameter Tuning → Evaluation → Best Model Selection

Step 1: Data Splitting

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- 80% for training (hyperparameter tuning)
- 20% for final testing (objective, unbiased evaluation)
- random_state=42 ensures reproducibility of results

Step 2: Pipeline Preparation
Each model is integrated into a standardized pipeline:

```
models = {
   'DecisionTree': Pipeline([
      ('preprocessor', preprocessor),
      ('model', DecisionTreeRegressor(random_state=42))
   ]),
   'RandomForest': Pipeline([
      ('preprocessor', preprocessor),
      ('model', RandomForestRegressor(random_state=42))
   ]),
   # ... other models
}
```

Automatic standardization of data
Avoids data leakage
Ensures consistent preprocessing across models

Step 3: Hyperparameter Tuning
Machine learning models have:

- Learnable parameters (e.g., weights, coefficients)
- Hyperparameters (e.g., max tree depth, number of trees, learning rate)

To identify the best hyperparameter configuration, we use Grid Search, which exhaustively tests all defined combinations.
TimeSeriesSplit cross-validation is applied to respect the temporal nature of industrial data and avoid biases from shuffling time-ordered data.

Step 4: Evaluation and Comparison
Main Metric: Mean Squared Error (MSE)
$MSE=(1/n)\times\Sigma(yreal-ypred)^2$

Step 5: Final Evaluation
Each selected model is evaluated on the test set using Root Mean Squared Error (RMSE). The model with the lowest RMSE is selected as the optimal model for the corresponding workstation.

Step 6: Feature Analysis
The selected model is further evaluated using additional metrics such as MSE, MAE, and R^2. For tree-based models, feature importance is analysed to understand the influence of production volume and day of the week on predicted defects.

3.7.2 Global Prediction Construction

Once individual predictions per workstation are made, four aggregation methods are proposed to estimate the overall defect rate for the production line:

- Maximum across all Workstation
- Simple average
- Weighted average based on production history (production quantity × observed defects)
- Total sum of defects

Weighted average is especially relevant as it considers each station's contribution and criticality regarding non-conformity.

Weight Calculation Formula:
$W_i=(Q_i \times D_i)/\Sigma(Q_j \times D_j)$
Where:
Q_i = Total production volume for workstation i
D_i = Historical total defects for workstation i
W_i = Weight for workstation i

4 Results and discussion

4.1 Results of the case study

The performance of the models varies significantly depending on the workstations, with very satisfactory results (R^2 scores close to 1 and low errors).

Other models show more limited performance, with low or even negative R^2 scores, indicating a poor ability to explain the variance of defects at these stations.

This disparity can be explained by the complex and multifactorial nature of defects.

Production volume and day of the week alone are not sufficient to accurately predict the number of defects.

Several non-quantifiable factors come into play, such as:
- Shift or workstation changes,
- The arrival of new operators,
- Human errors,
- Physical or mental workload,
- The work area itself, since zone P3 is entirely manual.

These elements, often random or human-related, create a level of variability that is difficult for machine learning models to capture, even with advanced techniques.

This highlights the importance of integrating more contextual or qualitative data in the future to improve prediction accuracy and better understand the root causes of nonconformities.
The interactive prediction feature allows for real-time estimation of the number of defects and their overall rate across the line, based on two variables: the day and the production volume.

This mechanism demonstrates the model's ability to generalize its forecasts across all stations while providing immediate visibility into expected quality levels. The automatic calculation of the defect rate facilitates the interpretation of results and enhances the model's usefulness as a rapid quality performance assessment tool.

The predictions obtained allow for upstream estimation of the defect rate likely to appear based on the production day and volume.

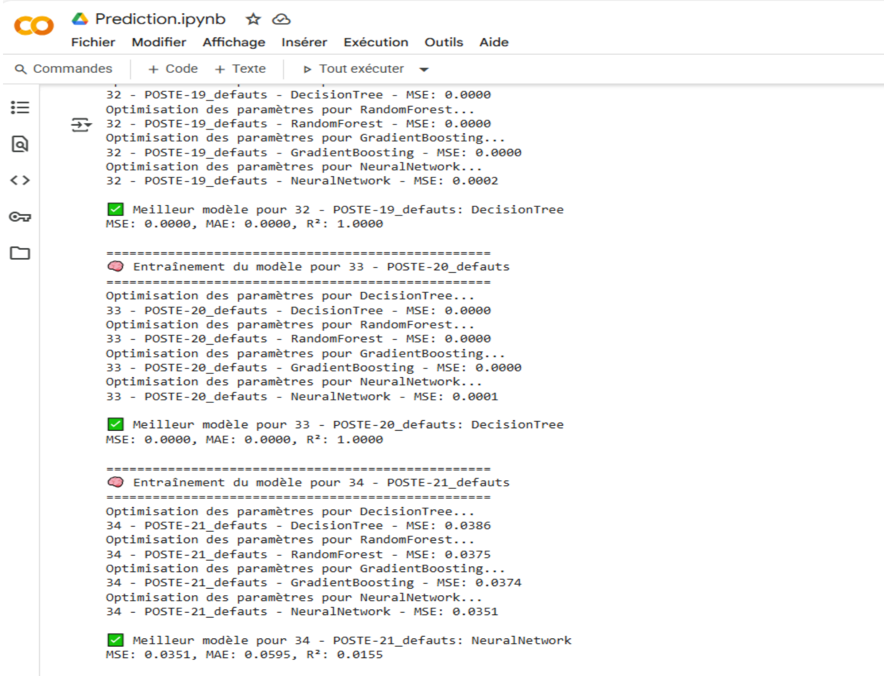


Figure 13: Process of the automatic selection of the optimal predicting model

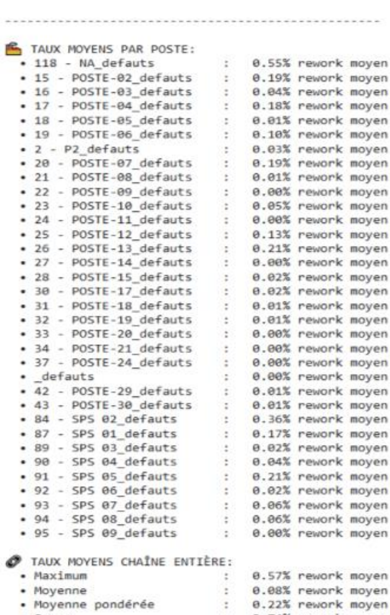


Figure 14: Process of the automatic selection of the optimal predicting model and aggregation

4.2 Environmental impact of quality defect's prediction

The prediction of quality defects can have a significant impact on the urgent deliveries' carbon footprint.

First, quality defects consume part of the production line's capacity, often leading to delays in meeting delivery deadlines. To compensate for these delays, companies may need to split shipments into multiple deliveries or rely on emergency transportation modes, both of which increase carbon emissions. Second, the occurrence of defects generates scraps that are considered industrial waste and may be harmful to the environment. Even when recycled, their treatment requires additional energy consumption. Finally, some defective products may become obsolete and require external subcontracting for further processing, which once again contributes to the carbon footprint through additional transportation.
Moreover, defect prediction plays a preventive role by allowing early identification of potential quality issues before they occur. This proactive approach helps the decision makers to find solutions for reducing waste generation, optimizing production capacity, and ultimately minimizing the overall environmental impact of the manufacturing process.

5 Conclusion and perspectives

This research has demonstrated the effectiveness of integrating predictive analytics into quality control for fully manual manufacturing environments. By evaluating both statistical and machine learning models across multiple workstations, the study has shown that it is possible to estimate defect rates with reasonable accuracy, even in the absence of automation or sensor-based monitoring. The automatic model selection framework provides a practical and scalable solution that adapts to the specific behaviour of each station, offering actionable insights for production planning teams. More importantly, the integration of defect forecasts into operational decision-making can significantly reduce the need for urgent logistics interventions, thereby lowering CO_2 emissions and contributing to more sustainable production processes.
As this study focused exclusively on human-induced defects in manual workstations, future research could explore the integration of additional human-related variables, such as shift patterns, training levels, fatigue indicators, or operator-specific historical performance, to enhance the accuracy of defect predictions. Furthermore, embedding the prediction engine into a stochastic planning model would allow planners to dynamically adjust production schedules and buffer stocks in real time, accounting for anticipated quality variability. This would enhance the resilience of manual production systems to disruption, reduce overproduction and rework, and optimize resource allocation.
In addition, future work will aim to develop **defect-specific prediction models**, enabling more targeted and effective preventive actions for each defect category. This next phase will rely on a dedicated database quantifying the number of defects generated **before and after prediction implementation**, allowing for the establishment of the **resulting carbon footprint balance**. Such an approach would further strengthen the link between predictive quality control and sustainable manufacturing.

References

1. Alsaeedi, Abdullah, et Mohammad Zubair Khan. 2019. « Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study ». Journal of Software Engineering and Applications 12(05):85-100. doi:10.4236/jsea.2019.125007.
2. Bajic, Bojana, Nikola Suzic, Slobodan Moraca, Miladin Stefanović, Milos Jovicic, et Aleksandar Rikalovic. 2023. « Edge Computing Data Optimization for Smart Quality Management: Industry 5.0 Perspective ». Sustainability 15(7):6032. doi:10.3390/su15076032.
3. Balogun, Abdullateef Oluwagbemiga, Shuib Basri, Said Jadid Abdulkadir, et Ahmad Sobri Hashim. 2019. « Performance Analysis of Feature Selection Methods in Software Defect Prediction: A Search Method Approach ». Applied Sciences 9(13):2764. doi:10.3390/app9132764.
4. Chinenye, Obidike, Kene Tochukwu Anyachebelu, et Muhammad Umar Abdullahi. 2023. « Software Defect Prediction System Based on Decision Tree Algorithm ». Asian Journal of Research in Computer Science 16(4):32-48. doi:10.9734/ajrcos/2023/v16i4368.
5. Ejjami, Rachid, et Khaoula Boussalham. 2024. « Industry 5.0 in Manufacturing: Enhancing Resilience and Responsibility through AI-Driven Predictive Maintenance, Quality Control, and Supply Chain Optimization ». 6(4).
6. Ihsan Aquil, Mohammad Amimul. 2020. « Predicting Software Defects Using Machine Learning Techniques ». International Journal of Advanced Trends in Computer Science and Engineering 9(4):6609-16. doi:10.30534/ijatcse/2020/352942020.
7. Iqbal, Ahmed, Shabib Aftab, Umair Ali, Zahid Nawaz, Laraib Sana, Munir Ahmad, et Arif Husen. 2019. « Performance Analysis of Machine Learning Techniques on Software Defect Prediction Using NASA Datasets ». International Journal of Advanced Computer Science and Applications 10(5). doi:10.14569/ijacsa.2019.0100538.
8. Joseph Nnaemeka Chukwunweike, Andrew Nii Anang, Adewale Abayomi Adeniran, et Jude Dike. 2024. « Enhancing Manufacturing Efficiency and Quality through Automation and Deep Learning: Addressing Redundancy, Defects, Vibration Analysis, and Material Strength Optimization ». World Journal of Advanced Research and Reviews 23(3):1272-95. doi:10.30574/wjarr.2024.23.3.2800.
9. Li, Ning, Martin Shepperd, et Yuchen Guo. 2020. « A Systematic Review of Unsupervised Learning Techniques for Software Defect Prediction ». Information and Software Technology 122:106287. doi:10.1016/j.infsof.2020.106287.
10. Matloob, Faseeha, Taher M. Ghazal, Nasser Taleb, Shabib Aftab, Munir Ahmad, Muhammad Adnan Khan, Sagheer Abbas, et Tariq Rahim Soomro. 2021. « Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review ». IEEE Access 9:98754-71. doi:10.1109/access.2021.3095559.
11. Ni, Chao, Xiang Chen, Fangfang Wu, Yuxiang Shen, et Qing Gu. 2019. « An Empirical Study on Pareto Based Multi-Objective Feature Selection for Software Defect Prediction ». Journal of Systems and Software 152:215-38. doi:10.1016/j.jss.2019.03.012.
12. Pachouly, Jalaj, Swati Ahirrao, Ketan Kotecha, Ganeshsree Selvachandran, et Ajith Abraham. 2022. « A Systematic Literature Review on Software Defect Prediction

Using Artificial Intelligence: Datasets, Data Validation Methods, Approaches, and Tools ». Engineering Applications of Artificial Intelligence 111:104773. doi: 10.1016/j.engappai.2022.104773.

13. Psarommatis, Foivos, Gökan May, Paul-Arthur Dreyfus, et Dimitris Kiritsis. 2020. « Zero Defect Manufacturing: State-of-the-Art Review, Shortcomings and Future Directions in Research ». International Journal of Production Research 58(1):1-17. doi:10.1080/00207543.2019.1605228.
14. Psarommatis, Foivos, João Sousa, João Pedro Mendonça, et Dimitris Kiritsis. 2022. « Zero-Defect Manufacturing the Approach for Higher Manufacturing Sustainability in the Era of Industry 4.0: A Position Paper ». International Journal of Production Research 60(1):73-91. doi:10.1080/00207543.2021.1987551.
15. Sankhye, Sidharth, et Guiping Hu. 2020. « Machine Learning Methods for Quality Prediction in Production ». Logistics 4(4):35. doi:10.3390/logistics4040035.
16. Silva, Anabela Costa, José Machado, et Paulo Sampaio. 2024. « Predictive Quality Model for Customer Defects ». The TQM Journal 36(9):155-74. doi:10.1108/tqm-09-2023-0302.
17. Thota, Mahesh Kumar, Francis H. Shajin, et Paulthurai Rajesh. s. d. « Survey on Software Defect Prediction Techniques ». International Journal of Applied Science and Engineering.
18. Elżbieta Skrzypek, Piotr Grudowski, et Gdansk University of Technology of Kalisz,. 2025. « CONDITIONS AND CONSEQUENCES OF QUALITY DEVELOPMENT – FROM QUALITY 4.0 TO 5.0 ». Scientific Papers of Silesian University of Technology. Organization and Management Series 2025(221):529-46. doi:10.29119/1641-3466.2025.221.29.
19. Verna, Elisa, Gianfranco Genta, Maurizio Galetto, et Fiorenzo Franceschini. 2022. « Defects-per-Unit Control Chart for Assembled Products Based on Defect Prediction Models ». The International Journal of Advanced Manufacturing Technology 119(5-6):2835-46. doi:10.1007/s00170-021-08157-1.
20. Yun, Jong Pil, Woosang Crino Shin, Gyogwon Koo, Min Su Kim, Chungki Lee, et Sang Jun Lee. 2020. « Automated Defect Inspection System for Metal Surfaces Based on Deep Learning and Data Augmentation ». Journal of Manufacturing Systems 55:317-24. doi:10.1016/j.jmsy.2020.03.009.
21. Sathananthavathi, V., & Sushilkumar, M. (2024, November). Developing a Strategy for Vehicle Pollution Reduction. In 2024 International Conference on IoT, Communication and Automation Technology (ICICAT) (pp. 1027-1031). IEEE.
22. Varun, S., Prabhu, A., Vijeesh, P. K., & Chelliah, B. J. (2024, April). A Novel Approach to Protect Environment by CO 2 Emission Detection. In 2024 International Conference on Computing and Data Science (ICCDS) (pp. 1-6). IEEE.
23. Bussaban, K., Kularbphettong, K., Raksuntorn, N., & Boonseng, C. (2024). Prediction of CO2 emissions using machine learning. Edelweiss Applied Science and Technology, 8(4), 1-11.