Concept drift in predictive quality control: An explainable machine learning approach

José F. M. Martins[1][0009-0009-8459-2026]
André L. Ferreira[1,2][0000-0002-7028-426X]
João M. Fernandes[1,3][0000-0003-1174-1966]

[1] Dept. Informática, Universidade do Minho, Braga, Portugal
[2] Bosch Car Multimedia Portugal S.A., Braga, Portugal
[3] ALGORITMI and CCG/ZGDV Institute, Braga, Portugal

Abstract. In industrial manufacturing, Machine Learning (ML) models enable predictive quality control but often struggle with concept drift. This occurs when temporal shifts in data undermine model reliability. This study presents a drift-handling framework that combines automated monitoring, statistical drift detection and adaptive retraining to maintain the model performance in dynamic production environments. Incoming data are organized in a three-tier storage system, enabling selective incremental updates or full retraining based on the magnitude of drift. Drift is detected using statistical testsand model updates are guided by hold-out performance evaluation. Empirical results indicate that tree-based models outperform linear approaches in capturing non-linear patterns critical for quality control. Additionally, the integration of explainability methods, including SHAP-based feature attribution and natural language explanations, enhances transparency and supports root-cause analysis. The framework mitigates distributional shifts, optimizes the efficiency of retraining and provides robust interpretable ML solutions.

Keywords: Concept drift · Predictive quality control · Industrial machine learning · Drift detection · Adaptive retraining · Explainable AI · Manufacturing process optimization.

1 Introduction

Advanced driver-assistance features and in-vehicle systems, such as Driver Monitoring Systems and Heads-up Display, depend on precise manufacturing processes to ensure safety and reliability. Even minor deviations in production can have significant consequences, ranging from functional degradation to complete product rejection. Thus, manufacturers are under strong pressure to integrate quality assurance mechanisms that can detect potential defects early, reduce costs, and maintain the high standards required by the industry.

ML has emerged as a key enabler of predictive quality control in this context. By analyzing high-dimensional sensor and process data, ML models are capable of uncovering hidden patterns and correlations that traditional statistical

techniques may overlook. Previous studies have demonstrated promising results, showing that algorithms such as Random Forest, Support Vector Machines, and Gradient Boosting can predict defects, optimize workflows, and improve resource efficiency. These advances underscore the potential of ML to transform industrial quality control into a more proactive and data-driven discipline.

However, deploying ML in real-world manufacturing comes with persistent challenges. A critical issue is **concept drift**, that is, the gradual or sudden change in data distributions over time [1]. Drifts may result from variations in material properties, wear of production equipment, environmental fluctuations, or sensor recalibration. Left unaddressed, these shifts degrade model performance and compromise the reliability of predictive systems, ultimately leading to costly production interruptions or undetected defects. Unlike controlled laboratory settings, industrial environments are dynamic and unpredictable, making robust drift management a prerequisite for sustainable ML adoption.

Methods like online learning and adaptive ensembles address concept drift but often lack the efficiency or transparency needed for manufacturing, which limit their applicability in manufacturing. In particular, engineers often require explanations to interpret why model predictions change and to trace the underlying causes of quality problems. Without this level of explainability, the trust and the practical usability of predictive systems are reduced.

To address these challenges, we propose a drift detection and retraining system tailored to the needs of industrial manufacturing. The approach integrates three key components: (1) automated drift monitoring to detect changes in data distributions, (2) retraining mechanisms that adapt models efficiently without excessive computational overhead, and (3) explainability methods that provide engineers with actionable insights into model behavior. Combining these components ensures that predictive models maintain their accuracy, dependability, and transparency under different conditions.

The contributions of this manuscript are threefold. First, we design a drift detection framework that continuously monitors production data streams and signals when retraining is required. Second, we evaluate retraining strategies that balance model performance with computational efficiency, making them feasible for industrial deployment. Finally, we demonstrate how explainability techniques can be integrated into predictive quality control, supporting root-cause analysis and enhancing trust among domain experts.

By proactively addressing concept drift, the proposed system reduces downtime, supports shop-floor decisions, and enhances process resilience. It bridges the gap between research and industrial ML adoption.

2 Related work

ML has been applied in manufacturing for predictive quality, defect detection, and process optimization. Some studies [2, 3] demonstrate the effectiveness of ensemble models, such as Random Forest and XGBoost, for classification tasks

in production lines, achieving a level of accuracy above 90%. These works show how feature engineering and imbalance handling improve model robustness.

The review by Tercan and Meisen [4] shows widespread use of ML for predictive quality, improving rejection rates and defect detection, while noting challenges in high-dimensional data, class imbalance, and production integration.

Other studies report industrial case studies, such as clustering for waste reduction [5] and convolutional neural network with long short-term memory (CNN-LSTM) fault detection with 92.5% accuracy, limited by data quality [6]. Borré et al. [7] present a hybrid CNN-LSTM architecture. Their approach aims to manage the uncertainties present in the data and yielded superior results compared to traditional reference models.

Mangal and Kumar [8] investigated predictive quality large-scale production datasets at Bosch, applying ensemble and big data techniques to achieve strong predictive performance (AUC = 0.718), despite challenges of high dimensionality and sparsity. Kausik et al. [9] provide a systematic review that analyzes over 300 studies over the last two decades to evaluate the effectiveness of widely used ML algorithms in manufacturing quality assurance applications.

Concept drift affects all these works, as changing processes and human factors require frequent retraining and are rarely addressed in deployed solutions. Recent industrial studies, such as Zink et al. [11], show that concept drift remains a pressing issue in several industrial domains. Explainability remains secondary, with most studies focusing on accuracy, despite transparency being critical for manufacturing decisions. Yet, in manufacturing contexts, transparency is critical for trust and decision-making among engineers and quality managers.

This work advances the state of the art by combining:

- Automated drift detection and retraining mechanisms validated in realistic production-like scenarios.
- Tree-based ML models that outperform linear alternatives for both classification and regression tasks.
- An explainability layer using SHAP and natural language explanations to bridge the gap between model outputs and actionable insights.
- An optimization module to suggest process adjustments, supporting quality improvement and defect reduction.

This method addresses research gaps by providing an adaptable and interpretable solution with high predictive performance for production environments.

3 Approach

In industrial manufacturing environments, predictive quality control heavily relies on ML models to detect defects, forecast failures, and optimize processes. However, the practical deployment of these models is challenged by **concept drift**. In this section, a drift-aware framework is designed to ensure model robustness, efficiency, and interpretability in real-world production settings.

3.1 Motivation and design principles

Concept drift can cause models with excellent initial performance to degrade within days or weeks whenever the evolving nature of the process is unconsidered.

Continuous full retraining is often impractical in industrial settings. Retraining demands substantial computational resources, can interfere with ongoing production, and requires extensive validation to ensure that model updates do not inadvertently degrade performance. Thus, a drift-aware approach is essential, in order to balance adaptability with efficiency and to preserve historical knowledge while remaining responsive to recent changes in production data.

This approach is guided by four key principles:

1. **Robustness to drift:** The system must identify both gradual and sudden changes in the data distribution to avoid predictive performance degradation.
2. **Computational efficiency:** Updates should be performed without disrupting production processes, prioritizing incremental adjustments over full retraining whenever possible.
3. **Data retention and continuity:** Historical data is valuable, and systems must balance past knowledge with new information.
4. **Interpretability:** Industrial applications require transparent and explainable predictions to support decision-making and ensure trust in the system.

The model selected to address these requirements was Extreme Gradient Boosting (XGBoost), a scalable gradient boosting framework introduced by Chen and Guestrin [10]. XGBoost has become a standard choice for predictive tasks on structured industrial data, offering a balance between accuracy, efficiency, and robustness. Its sparsity-aware optimization, regularization mechanisms, and parallelized implementation enable efficient retraining while maintaining high predictive power. Moreover, its compatibility with explainability tools such as SHAP facilitates the interpretability required in manufacturing contexts. These properties made XGBoost the most obvious candidate for sustaining long-term deployment in dynamic production environments.

3.2 Data flow and storage architecture

To achieve these objectives, a multi-tiered data storage architecture was designed to distinguish the "state" of the data. The data flow is organized as follows:

- **DB_Buffer:** Temporary storage for the most recent production batch, typically containing at least 1000 samples. This database serves as the first checkpoint for incoming data before it is validated and merged.
- **DB_Recent:** A curated repository of validated recent samples used for incremental updates of the predictive models. This dataset provides a snapshot of the most relevant production conditions.
- **DB_Older:** A historical archive that contains older samples. It provides contextual information during full retraining, ensuring that the model does not forget important patterns that may recur in production.

This architecture supports flexible retraining, allowing the system to dynamically switch between incremental updates and full retraining based on the severity of the detected drift.

3.3 Drift detection

Two techniques are used. The **Kolmogorov–Smirnov (KS) test** provides a non-parametric measure of distributional change between buffered and recent datasets. Following standard statistical practice [13], a significance level of $\alpha = 0.05$ is adopted: a p-value below this threshold indicates a statistically significant shift in feature distributions. The **Wasserstein distance** complements the KS test by quantifying the magnitude of distributional change in a geometrically interpretable way [14]. A threshold of 0.2 was determined empirically through pilot experiments on production data, consistent with values reported in related industrial drift monitoring studies [11].

Regarding batch size, a minimum of 1000 samples is required to ensure that the statistical tests operate with adequate power. This value was calibrated empirically based on production throughput at the target facility at the time, ensuring that drift detection runs are both statistically reliable and operationally feasible within a two-hour monitoring window. This value can and must be changed whenever the production phase demands so.

If no drift is detected, the new data are integrated into DB_Recent, and an incremental and warm-start retraining is triggered. Thus, the model continuously adapts to small changes in production without unnecessary full retraining.

3.4 Strategies for model update

Warm retraining for minor drift When minor drift is detected, the system performs *incremental learning*. The existing XGBoost model is updated using the new batch of data, retaining prior knowledge through the boosting mechanism. Incremental updates allow the model to incorporate the latest trends in production while avoiding computational overhead and downtime. Warm retraining also ensures smoother model transitions, preventing abrupt changes in predictions that could disrupt quality monitoring.

Full retraining for significant drift In cases of significant drift, two candidate models are trained: one on the DB_Recent dataset and one on DB_Buffer. Performance is evaluated on a hold-out validation set using metrics appropriate for the task—F_β score for classification and mean absolute error (Mean Absolute Error (MAE)) for regression. This comparison determines whether the model should be fully retrained or incrementally updated.

The system follows these two rules: (1) If the candidate model demonstrates at least 10% improvement over the current model, DB_Recent is appended to DB_Older, DB_Buffer is promoted to DB_Recent, and full retraining is performed. (2) If performance gains are marginal, the datasets are merged, and the

current model is updated incrementally. This strategy ensures that computational resources are used efficiently and that retraining occurs only when it is truly beneficial for predictive performance.

3.5 Explainability layer

One of the most important requirements for industrial deployment is interpretability. Using SHAP (SHapley Additive exPlanations) values, which measure each feature contribution to a particular prediction, a post-hoc explainability module was put into place. SHAP provides:

- **Global feature importance:** Identifying which production parameters have the most influence on model predictions across all samples.
- **Local explanations:** Understanding the reasoning behind individual predictions, which is crucial for troubleshooting defects and performing root-cause analysis.

By integrating SHAP into the workflow, engineers gain actionable insights, allowing them to adjust process parameters proactively rather than reacting solely to model outputs. This enhances both trust in the model and the overall quality management process.

3.6 Optimization and feedback loop

The system also includes an optimization module that leverages model predictions and SHAP insights to suggest process adjustments. This module operates at the intersection of predictive quality and process engineering:

- **Feature prioritization:** SHAP values identify the critical variables that affect defect probability, guiding engineers to focus on impactful adjustments.
- **Process tuning suggestions:** Based on historical trends and model outputs, the module recommends parameter adjustments, reducing defect rates and stabilizing production.
- **Feedback incorporation:** Changes implemented in production are recorded and used in subsequent retraining cycles, creating a closed-loop system for continuous improvement.

This feedback mechanism helps the system predict and prevent defects, improving efficiency and reducing waste.

3.7 Implementation considerations

Some practical considerations guided the implementation:

- **Batch size and update frequency:** Thresholds for retraining were tuned based on production throughput and computational constraints. A minimum batch of 1000 samples ensures statistical validity for drift detection.

- **Data validation:** Automated checks for missing values, outliers, and inconsistent formats are performed before integration into DB_Recent, ensuring high-quality training data.
- **Computational resource allocation:** Incremental updates are prioritized on edge devices or local servers to minimize latency, while full retraining is scheduled during low-production periods to reduce disruption.

Together, these implementation choices ensure that the framework remains computationally feasible in production environments, while maintaining the adaptability, interpretability, and knowledge retention required for robust deployment.

3.8 Summary of advantages

Our system bridges the gap between state-of-the-art predictive models and practical industrial deployment, ensuring adaptability, interpretability, and tangible impact on manufacturing efficiency and product quality. It provides a comprehensive solution to predictive quality control in manufacturing:

- **Resilience to drift:** Automatic detection and adaptive retraining maintain high predictive performance even in dynamic production environments.
- **Operational efficiency:** Incremental updates reduce computational load and avoid unnecessary retraining.
- **Knowledge retention:** Historical data are preserved and used strategically to prevent catastrophic forgetting.
- **Interpretability and actionable insights:** SHAP explanations allow engineers to understand model decisions and optimize processes proactively.
- **Closed-loop optimization:** Recommendations based on model outputs and feature importance are fed back into production, driving continuous quality improvements.

4 Results

4.1 Classification performance

The classification experiments show a clear distinction between linear and tree-based approaches in modeling the production data. Linear models consistently underperform, as the relationships between features and target variables exhibit strong non-linear characteristics. In contrast, XGBoost-based tree models achieved substantially higher predictive performance across all evaluated metrics, highlighting their suitability for complex manufacturing environments.

Fig. 1 presents the performance of the best-performing tree-based classifier, illustrating the trade-offs involved in threshold selection, where the F_β score is maximized while monitoring additional metrics such as precision, recall, and specificity. The F_β metric was deliberately configured to emphasize the reduction of false positives, reflecting the critical operational requirement to avoid unnecessary interventions in the production line. Each point along the threshold curve

provides insight into how model sensitivity can be balanced against operational constraints, offering engineers practical guidance for real-world deployment.

The classification results reveal structured patterns in feature influence and error distribution. Some features exhibited consistently high importance across multiple validation splits, suggesting that these variables encode stable physical relationships between production conditions and defect occurrence. This consistency was observed even when minor batch-to-batch variations introduced distributional noise.

The non-linear decision boundaries learned by XGBoost allow it to partition the feature space in ways that linear models cannot, particularly in regions where multiple features interact jointly to determine defect probability. Linear models systematically underestimate defect risk in these interaction-dominated regions, which explains their inferior recall and F_β scores. This structural limitation of linear approaches is not merely a matter of hyperparameter tuning: it reflects a fundamental mismatch between the model family and the geometry of the decision boundary in the production data.

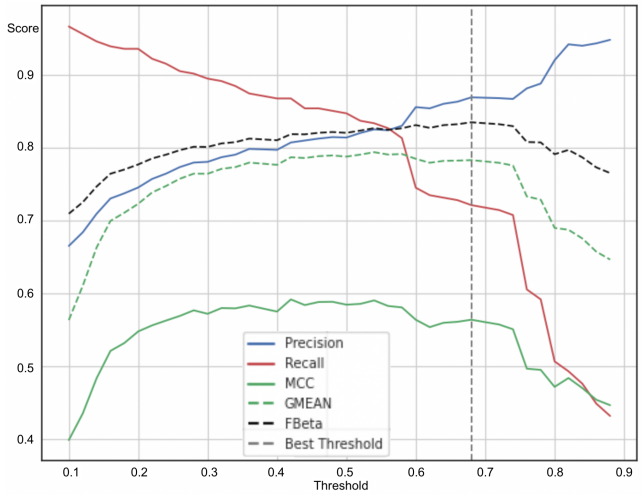


Fig. 1. Classification Result

4.2 Regression performance

In regression tasks, linear models showed moderate effectiveness, suggesting that a subset of the relationships between input features and target variables can be approximated through linear dependencies. Nevertheless, XGBoost models demonstrated superior performance, achieving lower error rates and exhibiting more robust generalization across production batches. Fig. 2 depicts the regression performance of the best-performing model, illustrating predicted versus actual outputs and the residual distribution.

The regression analysis confirms that, while some linear patterns exist in the dataset, tree-based models effectively capture complex interactions among fea-

tures that linear models cannot. Residual analysis indicates that prediction errors are distributed relatively evenly across the operational range, with no systematic bias toward high or low predicted values. This uniform error distribution suggests that the model generalizes well across different operating regimes, avoiding the common pitfall of overfitting to the most frequent operating condition.

In contrast, linear models exhibited a characteristic pattern of underestimation at high output values and overestimation at low values, consistent with their inability to model the curvature present in the target function. The superiority of XGBoost from its ability to iteratively correct residual errors through gradient boosting, effectively approximating complex non-linear response surfaces that arise from the interaction of multiple process variables. This robustness is valuable in cases, where ensuring consistent prediction accuracy across all process conditions is critical for maintaining product quality and reducing defects.

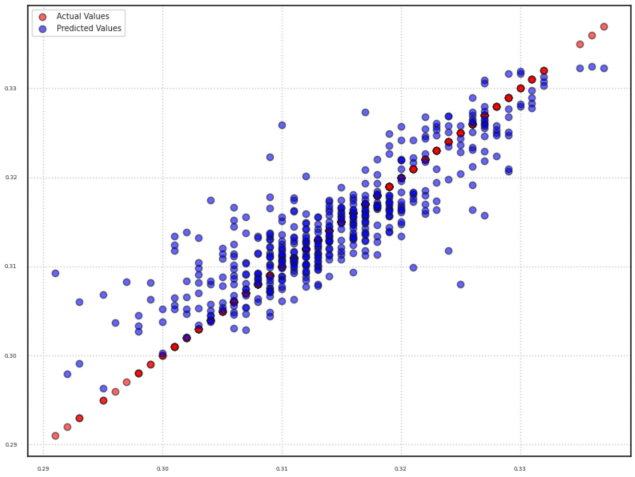


Fig. 2. Regression Result

4.3 Retraining and drift handling

The proposed drift detection and retraining framework was validated through controlled simulations and operational data monitoring. When only minor distributional shifts were observed, the system correctly triggered warm start retraining, updating the model incrementally while preserving previously learned knowledge. This approach avoided unnecessary computational overhead and minimized disruption to production workflows.

Conversely, when significant concept drift occurred, the system successfully identified the drift through statistical tests and initiated full retraining. In these cases, two candidate models were evaluated on a hold-out validation set, and the model with superior performance was deployed. The framework ensures that models remain accurate over time.

Beyond mere detection, the framework allows detailed analysis of drift magnitude and feature-level contributions. This capability provides engineers with

actionable insights, highlighting which process parameters are most sensitive to changes and may require additional monitoring or intervention. The adaptive retraining strategy thus not only maintains model performance but also enhances situational awareness within the production system.

4.4 Explainability and optimization

To improve transparency and ease actionable decision-making, SHAP-based feature attribution was applied to both classification and regression models. A SHAP analysis-derived feature importance ranking example is shown in Fig. 3. By visualizing the contribution of each variable to the model's predictions, engineers can identify process-critical measurements and understand the drivers of potential defects. This insight supports root-cause analysis and informs targeted process adjustments, increasing trust in automated predictive systems.

In addition, a large language model (LLM) was employed to generate natural language explanations of SHAP outputs and statistical metrics. This layer of interpretability allows non-specialist users to understand model behavior and rationale behind predictions. The natural language explanations ensure that predictive insights are accessible to quality managers and production engineers alike.

To illustrate the practical value of SHAP in this context, consider a scenario where the model predicts an elevated defect probability for a given production unit. The SHAP decomposition reveals that this prediction is driven primarily by an abnormal reading in a specific sensor channel, combined with a process temperature slightly above the nominal range. Without SHAP, an engineer would only observe the high defect probability with no actionable information. With SHAP local explanations, the root cause is immediately identifiable, enabling a targeted intervention rather than a broad and disruptive process halt. This capability is particularly valuable after drift-triggered retraining events, where SHAP can reveal which feature distributions have shifted most significantly, providing an interpretable bridge between statistical drift detection and process engineering decisions.

Finally, an optimization module was incorporated to recommend operational adjustments aimed at minimizing defect rates.

By simulating variations in input features and predicting their impact on production quality, the system provides actionable guidance for process improvement. This proactive approach reduces reliance on costly trial-and-error methods, supports continuous quality enhancement, and strengthens the overall resilience of production processes.

4.5 Discussion of results

The integrated evaluation shows that combining tree-based models, drift-aware retraining, SHAP-based explainability, and optimization feedback creates a cohesive framework for quality management. Classification and regression analyses confirm that XGBoost tree-based models effectively capture complex non-linear relationships in production data — relationships that arise from the interaction

of multiple process variables and that linear models are structurally incapable of representing. The performance gap between XGBoost and linear baselines was most pronounced in classification tasks, where the F_β score differential highlights the cost of false positives in production contexts. In regression, the advantage of XGBoost is reflected in lower MAE values and a more uniform residual distribution, indicating better generalization across operating regimes. Explainability tools and optimization modules translate predictive outputs into actionable operational guidance, enhancing practical value in industrial settings.

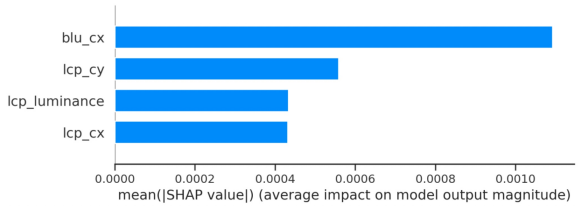


Fig. 3. Shap Prediction

Collectively, these results indicate that the proposed approach not only achieves high predictive performance but also provides a robust, interpretable, and operationally relevant system. By maintaining model currency and transparency, the framework supports informed decision-making and proactive interventions, ultimately contributing to reduced defect rates, improved production efficiency, and increased trust in automated quality control systems.

5 Conclusion

This paper presents an integrated ML pipeline for predictive quality in industrial manufacturing, tackling concept drift and model interpretability. In dynamic environments, shifts in data distributions from material changes, sensor recalibration, or process updates can degrade model performance. The proposed system uses drift detection, adaptive retraining, and tree-based models to maintain accuracy against both gradual and sudden changes.

Experimental results confirm that XGBoost-based tree models substantially outperform linear approaches in capturing the non-linear dependencies between process variables and quality outcomes. The retraining framework effectively balances computational efficiency and model adaptability, dynamically selecting between incremental and full retraining based on drift severity, thereby preserving historical knowledge with minimal production disruption.

Explainability is addressed through SHAP-based feature attribution and natural language explanations, enabling engineers and quality managers to identify the most influential process variables, support root-cause analysis, and make data-driven decisions. The integrated optimization module further enhances the framework by recommending realistic process adjustments that minimize defect rates without resorting to costly trial-and-error methods.

Overall, this work demonstrates that combining drift-aware retraining, interpretable ML, and process optimization yields a robust, transparent, and adaptable solution for predictive quality control in industrial manufacturing. Future research will explore hybrid architectures integrating tree-based and deep learning models, automated calibration of drift detection thresholds, and deployment across diverse manufacturing domains to assess generalizability and scalability.

Acknowledgments. This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Unit Project Scope UID/00319/2025 - Centro ALGORITMI (ALGORITMI/UM) DOI: 10.54499/UID/00319/2025

References

1. J. Monteiro, D. Ramos, D. Carneiro, F. Duarte, J. M. Fernandes, and P. Novais, "Meta-learning and the new challenges of machine learning," *Int. J. Intell. Syst.* 36(11):6240–6272, 2021. doi
2. R. S. Peres, J. Barata, P. Leitão, and G. Garcia, "Multistage quality control using machine learning in the automotive industry," *IEEE Access* 7:79908–79916, 2019. doi
3. H. Tercan and T. Meisen, "Machine learning and deep learning based predictive quality in manufacturing: A systematic review," *J. Intell. Manuf.* 33(7):1879–1905, 2022. doi
4. S. Sankhye and G. Hu, "Machine learning methods for quality prediction in production," *Logistics* 4(4):35, 2020. doi
5. S. Singh and S. Singhal, "Implementation and analysis of the clustering process in the enhancement of manufacturing productivity," *J. King Saud Univ. Eng. Sci.* 33(7):482–490, 2021. doi
6. P. Knab, *Applied machine learning for error detection in industrial winding processes*, Bachelor thesis, University of Mannheim, Germany, 2020.
7. A. Borré, L. O. Seman, E. Camponogara, S. F. Stefenon, V. C. Mariani, and L. D. S. Coelho, "Machine fault detection using a hybrid CNN-LSTM attention-based model", *Sensors* 23(9):4512, 2023. doi
8. A. Mangal and N. Kumar, "Using big data to enhance the Bosch production line performance: A Kaggle challenge," *IEEE Int. Conf. on Big Data (Big Data 2016)*, pp. 2029–2035, 2016. doi
9. A. K. Kausik, A. B. Rashid, R. F. Baki, and M. M. J. Maktum, "Machine learning algorithms for manufacturing quality assurance: A systematic review of performance metrics and applications", *Array* 26:100393, 2025. doi
10. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2016)*, pp. 785–794, 2016. doi.
11. R. Zink, B. Ioshchikhes, and M. Weigold, "Concept drift monitoring for industrial load forecasting with artificial neural networks", *Procedia CIRP* 130:120–125, 2024. doi
12. N. Jourdan, T. Bayer, T. Biegel, and J. Metternich, "Handling concept drift in deep learning applications for process monitoring", *Procedia CIRP* 120:33–38, 2023. doi
13. F. J. Massey Jr., "The Kolmogorov-Smirnov test for goodness of fit," *J. Am. Stat. Assoc.* 46(253):68–78, 1951. doi
14. C. Villani, *Optimal Transport: Old and New*. Springer, Berlin, 2009. doi