Improving Automotive Production Forecasts via Explainable AI-Driven Feature Pruning

James AmRhine Ferreira

James.Ferreira@bmwmc.com

BMW Group, University of Rochester https://orcid.org/0009-0001-5550-0702

Research Article

Keywords: Predictive forecasting, Explainable artificial intelligence (XAI), Feature pruning, Automotive manufacturing, Sensor/IoT

Posted Date: September 5th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7522021/v2

Additional Declarations: The authors declare no competing interests.

Improving Automotive Production Forecasts via Explainable AI–Driven Feature Pruning

James Am Rhine Ferreira[1,2*]

[1]BMW Group, USA.
[2]University of Rochester, USA.

Corresponding author(s). E-mail(s): James.Ferreira@bmwmc.com ; jferrei5@simon.rochester.edu;

Abstract

This research uses explainable AI (XAI) to improve machine learning (ML) models in forecasting production by identifying and pruning low-value sensor-derived features in production forecasting tasks. The methodology entails the initial training of ML models, followed by a fine-tuning phase in which irrelevant features identified through explainability methods are eliminated. It builds on the throughput prediction architecture of BMW by performing XAI-driven refinement on top of an existing Extra Trees ensemble framework. Using Shapley values and permutation importance, redundant, unstable, or low-signal inputs are removed. This reduces overfitting, improves model interpretability, and compresses inference time without sacrificing accuracy. The result is both performance improvements and simplified model logic, enabling potential reductions in manufacturing costs and improved model transparency. Attribution monitoring also enables early detection of deteriorating predictive logic—supporting more transparent, auditable, and adaptable decision-making in automotive manufacturing environments.

Keywords: Predictive forecasting, Explainable artificial intelligence (XAI), Feature pruning, Automotive manufacturing, Sensor/IoT

1. Introduction

Accurate short-term forecasting of unit throughput in automotive assembly lines is both a critical KPI and a volatile signal—sensitive to bottlenecks, upstream delays, labor scheduling, and anomalies that cascade across checkpoints. Modern manufacturing telemetry is often

driven by IoT-generated sensor data, which is real-time, high-volume, and unstructured. As described by Syafrudin et al. (2018), such data is handled using a big data processing platform that employs Apache Kafka as a message queue, Apache Storm as a real-time processing engine, and MongoDB to store the sensor data from the manufacturing process.

Modern manufacturing environments generate granular telemetry from hundreds of sensors and checkpoints. Predictive models can use this data to anticipate slowdowns and optimize scheduling, but they are often opaque and overfit to redundant signals. The forecasting challenge is to build models that are interpretable, resilient to operational drift, and small enough to deploy across hundreds of localized end points in near real-time.

For example, in the BMW production pipeline (Jeereddy et al. 2019), each check- point's prediction model was an Extra Trees regressor (Geurts et al. 2006) trained on throughput counts—i.e., the number of units that had passed through that checkpoint or others—aggregated over fixed intervals of 5, 10, 15, 30, 45, 60, 120, 180, and 240 minutes. These lagged features captured recent activity levels over the past four hours and were used as primary inputs for short-term forecasting.

This paper introduces a method for refining ensemble regressors—specifically Extra Trees—by applying explainability-driven feature pruning and monitoring. To my knowledge, this is the first application of attribution-based pruning in short-term throughput forecasting for real-time automotive manufacturing. The technique applies post hoc attribution tools—Shapley values (Sundararajan & Najmi 2020) and permutation importance—to evaluate how much each input feature actually contributes to predictions in practice. Features that consistently offer low or unstable contributions across prediction intervals are removed. This reduces noise, lowers model complexity, reduces computational costs, improves inference time, and improves generalization.

Importantly, the attribution values themselves are monitored over time. Under normal conditions, a model's important features remain stable—e.g., lagged counts at a neighboring checkpoint may consistently drive predictions. But when conditions shift unexpectedly—such as a supplier delay, machine breakdown, or unmodeled bottleneck—those same features may lose predictive power. I track this degradation directly by analyzing attribution volatility across moving windows.

Attribution drift occurs when the model's internal prioritization of features changes significantly over time. This is detected by comparing the current feature importance rankings to those from W predictions earlier. If the rank correlation between these attribution vectors falls below a predefined threshold δ, drift is flagged:

$$\text{rank-corr}(\mathbf{A}_t, \mathbf{A}_{t-W}) < \delta$$

The threshold δ defines what constitutes "too much change" and is selected empirically based on attribution stability across known operating conditions.

This form of "explainability drift" acts as an early warning signal: even if the model's output has not yet diverged dramatically, its internal logic may be shifting. In live systems, this allows for soft failure detection, triage, and targeted retraining. The technique is model-agnostic and requires no changes to the infrastructure.

By compressing the feature set and integrating real-time interpretability signals, the results can lead to a leaner, more robust, and more auditable forecasting system. These goals align with those articulated in Gross et al. (2024), who demonstrated that explainability-based pruning yields smaller, faster models with tighter error margins and higher operator trust in manufacturing quality control workflows.

2. Background

2.1. Related Work

Explainable AI (XAI) methods have been proposed as promising tools for manufacturing processes (Sofianidis et al. 2021; Senoner et al. 2022; Yoo & Kang 2021). The European XMANAI project (Lampathaki et al. 2021) applied XAI across several manufacturing domains to demonstrate sector-specific value. Several prior works have examined feature selection (Venkatesh & Anuradha 2019) without integrating the ML model directly into the selection process (Bins & Draper 2001). While improving ML models through explainability methods has been established in the XAI literature (Bento et al. 2021; Sun et al. 2022; Sofianidis et al. 2021; Gross et al. 2024), to the best of my knowledge, no prior work has used XAI to identify and remove non-contributive features for short-term throughput forecasting in automotive manufacturing.

2.2. Unit Production Forecasting in Automotive Plants

Short-term production forecasts help factory-floor managers anticipate parts shortages, downtime, and shifts in labor needs. Regression models such as Random Forest, XGBoost, and Extra Trees are widely used due to their flexibility and performance on structured sensor data.

Previous work by Jeereddy et al. (2019) introduced a checkpoint-level forecasting system spanning 120 different checkpoints across two assembly halls. For each checkpoint, models were trained to predict unit counts across three forward intervals (current hour, +1 hour, +2 hours), producing 360 models in total. The input features included rolling histories of throughput at 5- to 240-minute intervals, time-of-day encoding, and working-time availability indicators.

Their system operated without interpretability or attribution tools—limiting the ability to audit forecasts, compress redundant input space, or detect failures driven by anomalous external events (e.g., a supplier failure not present in training data). Adding explainability methods does not directly "predict the unprecedented," but it lets the model diagnose deviations in its own logic—a form of model self-awareness that is absent in raw Extra Trees inference.

2.3. Explainable Machine Learning

Post hoc explainability techniques such as SHAP (SHapley Additive exPlanations) (Lundberg & Lee 2017) and permutation feature importance quantify the influence of each feature on a model's output. In the context of ensemble models such as Extra Trees, these methods compute marginal contributions by measuring prediction changes when features are perturbed or excluded.

SHAP values provide a theoretically grounded framework rooted in cooperative game theory, assigning each feature a weighted contribution to the final prediction. In contrast, permutation importance evaluates the decrease in performance when feature values are randomly shuffled. Both approaches reveal which features consistently drive predictions and which contribute to noise or instability.

The benefits extend beyond model accuracy. As shown in Gross et al. (2024), explainability-based pruning enhances interpretability, lowers computational cost, and builds trust in automated forecasting systems. In high-throughput environments such as automotive manufacturing, where hundreds of model instances run in parallel, this interpretability translates directly into operational auditability and maintainability.

3. Dataset and Problem Formulation

The dataset used in this work is a reconstruction of the BMW manufacturing dataset from Jeereddy et al. (2019), which was collected from a real-world automotive production environment consisting of two assembly halls and 120 monitored checkpoints. The production of each checkpoint log occurs at five-minute intervals, resulting in high-resolution time-series data suitable for short-horizon forecasting.

Each data point includes a comprehensive set of structured features:

- **Prior Shift Production:** Total number of units produced during the immediately preceding shift.
- **Current Shift Production:** Running total of units produced during the current shift.
- **Time-of-Day Indicators:** Encoded as hour, quarter-hour, and shift segment to capture temporal patterns and shift effects.
- **Lagged Throughput Features:** For each checkpoint, the number of units processed is aggregated across fixed lag intervals of 5, 10, 15, 30, 45, 60, 120, 180, and 240 minutes. These lag windows capture short- and medium-term dynamics of flow.
- **Working Minutes:** The number of scheduled working minutes remaining within the hour, accounting for breaks and interruptions.

The forecasting objective is to predict the number of units that will be produced at a given checkpoint within the current hour, the next hour, and two hours ahead. This is treated as a regression problem. For each of the 120 checkpoints, I trained three separate Extra Trees models per checkpoint—each corresponding to a specific prediction horizon (0h, +1h, and +2h). The end-of-process checkpoints for the halls were the checkpoints tested as they were the official points for the KPIs being tested.

The original work of Jeereddy et al. (2019) evaluated multiple regressors, including Random Forest, XGBoost, and LSTM, and found that Extra Trees yielded the best performance for hourly unit production forecasting across checkpoints. This work builds directly on that baseline by exclusively using Extra Trees as the predictive model and extending it through the integration of post hoc explainability tools—specifically Shapley values and permutation importance—for targeted feature pruning and attribution monitoring, which were not part of the original system.

Note that for confidentiality reasons, the exact throughput (y-axis) was left off of the charts in Jeereddy et al. (2019).

4. Methodology

This section outlines the complete forecasting pipeline used in this study, including model selection, attribution-based feature pruning, and the extension of attribution signals for risk-aware monitoring. The goal is to produce a forecasting system that is not only accurate and efficient but also introspective and robust to upstream volatility.

4.1. Pipeline Overview

I apply SHAP and permutation importance to every Extra Trees model trained for checkpoint-level throughput prediction. Feature importance is evaluated across rolling time windows to detect attribution drift. Features that demonstrate low, unstable, or redundant contributions are pruned. For example, overlapping lag intervals from upstream checkpoints often exhibit multicollinearity—introducing variance without improving signal. By removing these data, the model becomes leaner and more robust, with a reduced risk of overfitting and shorter retraining time.

This process is implemented via the SHAP Python library (Lundberg & Lee 2017) and the permutation importance tools in Scikit-learn (Pedregosa et al. 2011). Feature pruning is performed iteratively: at each stage, the least informative features are removed, and the model is retrained and re-evaluated. Performance metrics such as MAE and RMSE are tracked alongside attribution stability metrics.

4.2. Base Model: Extra Trees Regressor

The base model is an Extra Trees Regressor, an ensemble method composed of multiple decision trees, where both the feature splits and split thresholds are selected at random. This increased randomness reduces variance and improves generalization compared to standard decision trees or Random Forests.

The choice of Extra Trees is consistent with prior internal research. Among tested models—including XGBoost, Random Forest, and LSTM—Extra Trees offered the best accuracy-to-complexity ratio on the BMW throughput forecasting task (Jeereddy et al. 2019).

For a regression task, the output $y^{\wedge}$ for an input vector $\mathbf{x}$ is computed as the average of predictions from all T trees:

$$y = \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x})$$

where $f_t(\mathbf{x})$ is the output of the t-th tree.

In classification tasks, predictions are typically determined by majority vote or average class probability:

$$y = \text{mode}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_T(\mathbf{x}))$$
$$orP(y = c \mid \mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}(f_t(\mathbf{x}) = c)$$

The distinguishing factor of Extra Trees over Random Forests is the increased randomness: while both select a random subset of features at each node, Extra Trees also selects the split threshold randomly rather than choosing it based on optimal impurity reduction. This introduces greater decorrelation among trees, making the ensemble more robust to noise and overfitting—the tradeoff is improved generalization at the cost of slightly higher bias.

4.3. Explainability-Driven Feature Pruning

Following Gross et al. (2024), I apply SHapley Additive Explanations (SHAP) values and permutation feature importance to quantify the marginal contribution of each input feature to model predictions. These tools allow:

- Identification and pruning of stale, redundant, or low-impact features (e.g., lag windows with high multicollinearity or low influence).
- Improved interpretability of feature contributions (e.g., attributing influence to "Checkpoint 12 at +30 min").
- Model bloat and overfitting risk are reduced by shrinking the input space.

This pruning process is performed iteratively: after training each model, features with unstable or weak attribution scores across rolling time windows are removed, the model is retrained, and performance metrics (MAE, RMSE) are tracked.

In addition to improving the forecast variance and retraining speed, attribution values are also monitored longitudinally to detect structural drift. For example, if a top- contributing checkpoint suddenly loses predictive influence—due to a supplier failure, machine halt, or misaligned upstream flow—this change surfaces as attribution volatility. While the system does not directly predict unprecedented disruptions, it enables soft failure detection by revealing when the model's logic begins to deteriorate. This pruning not only improves generalization and training efficiency, but also prepares the system for attribution-based introspection—as detailed in the following section.

To train and evaluate each model, I use 5-fold cross-validation. For each fold, 80% of the data were used for training and 20% for validation, and the results were averaged across all folds. Shapley values were computed using the SHAP Python library, and permutation importance was calculated using Scikit-learn.

Mathematically, Permutation Importance for a feature i is computed as:

$$I_i = P_{original} - P_{permuted(i)}$$

where $P_{original}$ is model performance with the intact dataset and $P_{permuted(i)}$ is the performance after random permutation of feature i.

The Shapley value $\phi_i(f)$ for a model f and feature i is computed as:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} [f(S \cup \{i\}) - f(S)]$$

where N is the full feature set, S is a subset excluding i, and $f(S)$ is the model prediction with only the features in S. Because exact Shapley values are computationally expensive for large n, I used approximation strategies by sampling. This approach transformed the original BMW forecasting architecture into a more auditable, self-aware, and data-efficient system suitable

for high-throughput automotive environments.

4.4.Attribution Drift for Risk-Sensitive Forecasting

While attribution-guided pruning enhances model efficiency and interpretability, the same SHAP-based signals can be repurposed to monitor prediction stability under volatile conditions. This section formalizes the use of attribution drift as a proxy for risk-aware forecasting—enabling soft alerts when the model operates under input regimes it was not trained to handle.

This is especially relevant in high-variance environments. For example, BMW experienced an unprecedented supplier-related outage that disrupted upstream flow. The original study noted that this type of anomalous event could not be addressed by standard throughput forecasting models, as no historical data existed to train on (Jeereddy et al. 2019). Although the current system does not implement explicit risk models, it enables soft risk signaling through SHAP-based attribution drift.

Specifically, when SHAP values exhibit increased variance or when the top contributing features shift abruptly in rank, the system treats this as a proxy for degraded model confidence. These volatility patterns act as quasi-risk indicators, alerting operators that predictions may be unreliable—without requiring changes to the base model or retraining logic. This endows the forecasting system with introspective capability absent from the raw Extra Trees inference. This approach also reduces data requirements since pruning removes non-contributive signals while retaining causal dependencies.

The system monitors SHAP variance and top feature rank shifts across rolling time windows as a form of internal diagnostic. When feature importance becomes unstable—either in magnitude or rank order—the model raises soft alerts that out- puts may be unreliable. These attribution-based diagnostics enable the model to make low-confidence predictions before hard errors propagate through the pipeline. It offers a lightweight early warning system for out-of-distribution conditions, enabling downstream safeguards such as human verification, triggering alternate forecasting protocols, or initiating retraining workflows.

Unlike external anomaly detection or ensemble-based uncertainty estimates, this method leverages the model's own attribution logic to flag out-of-distribution inputs or internal logic erosion in real time. By layering interpretability with self-monitoring, the forecasting system can transition from a static estimator to an introspective model—capable of producing outputs, but also of warning when those outputs are no longer reliable.

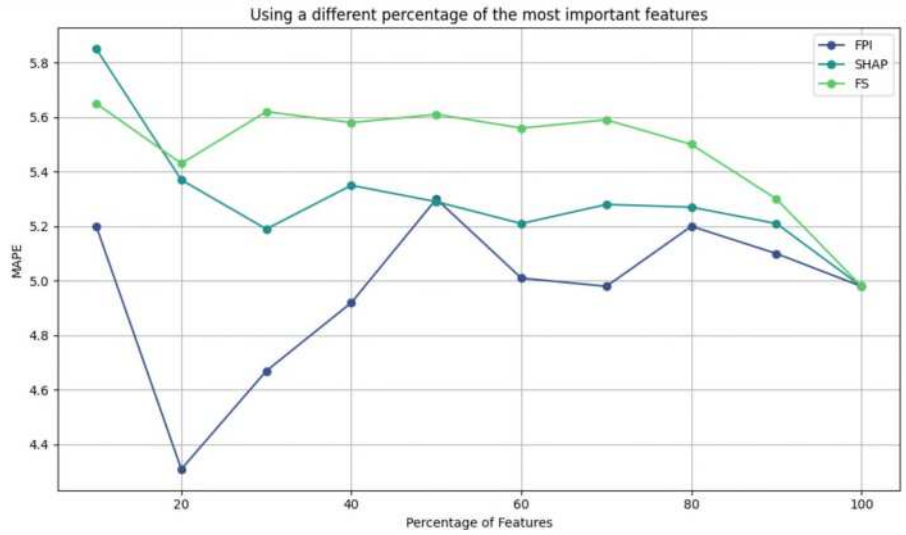

Fig. 1 Using a different percentage of the most important features. FS refers to feature selection.

5. Evaluation and Results

5.1. Metrics and Setup

To evaluate the effectiveness of the proposed explainability-driven refinement process, I follow the original BMW setup of predicting checkpoint-level hourly throughput using an 80/20 chronological split. Forecasts are generated every 5 minutes, and performance is measured using R2, the mean absolute error (MAE), the root mean squared error (RMSE), and the mean absolute percentage error (MAPE). All models underwent 5-fold cross-validation to ensure robustness across temporal regimes.

As with prior work (Jeereddy et al. 2019), exact throughput values and absolute error magnitudes cannot be disclosed due to confidentiality constraints. For this reason, all evaluation metrics are reported in relative or percentage-based terms (e.g., MAPE), which remain interpretable while preserving the data sensitivity.

First, features were ranked by three importance methods: SHAP, permutation importance, and SelectKBest (from the Scikit-learn library (Pedregosa et al. 2011)). In successive trials, models were trained using only the top $p\%$ of features. By limiting training to the most salient inputs, models became leaner and more interpretable without sacrificing predictive power. For instance, selecting only the top 20% of features (as ranked by permutation importance) improved the MAPE from 4.98% to 4.31%. This finding illustrates that explainability can be used for proactive model simplification.

As shown in Figure 1, both the SHAP and permutation importance enable effective pruning strategies, with performance plateauing at 30–40% of the total features. FS-based selection lags behind in early pruning regimes, suggesting less stable feature ranking under model-agnostic heuristics. However, the FPI achieves comparable performance with just 20% of the features, indicating a more stable and efficient attribution profile.

5.2. Performance Improvement

In assessing the quality of the predictions, the key metric employed is the mean absolute percentage error (MAPE). A prediction is considered to be of high quality if its MAPE is less than 5%.

I enhanced the ML models' performance by integrating only the most critical features into the training dataset. This approach effectively streamlined the model by removing unnecessary features, improving performance, and allowing for more transparent and explainable model operations. By choosing only the top 20% of features deemed most critical (as determined by permutation importance), I enhanced the MAPE from approximately 4.98% to 4.31%.

Even though the improvement in prediction accuracy (from 4.98% to 4.31% MAPE) is relatively modest, it comes with disproportionately larger benefits in other areas: smaller models, lower latency, and clearer logic paths. This is because the pruning process removed low-impact and unstable features identified through both SHAP and permutation analysis. These features often introduce noise or redundancy, and their removal simplifies the model without sacrificing reliability.

For reference, a standalone Decision Tree trained on the same data yielded a 5.88% MAPE, confirming the 15.3% error reduction achieved through ensemble modeling and attribution-guided pruning. A further 13.5% reduction was achieved by pruning the Extra Trees model, bringing the total improvement over the standalone Decision Tree to 26.7%.

Table 1 Prediction Error Comparison Across Models

Model Variant	MAPE
Extra Trees (Pruned)	4.31%
Extra Trees (Unpruned)	4.98%
Decision Tree	5.88%

5.3. Explainability and Predictive Mechanism Analysis

The SHAP and p ermutation methods did not always agree on feature rankings, highlighting the methodological differences between distributional versus functional attribution. However, the intersection of their stable top-ranked features showed consistent predictive value. In particular, shift production totals and specific 30- to 60-minute lag windows from upstream checkpoints were recurrent high-impact features.

This analysis helped isolate spurious correlations that arose from overfitting to non-causal input features in small data regimes. Pruning the low-impact, unstable, and non-causal features improved the reliability and interpretability of the model outputs.

6. Discussion

6.1. Practical Outcomes and Operational Gains

The integration of post hoc explainability methods—namely SHAP and permutation feature importance—yielded both operational and structural benefits. At a predictive level, pruning low-impact features led to modest gains in accuracy and variance reduction. But more significantly, it reduced inference cost, training time, and model complexity—making the approach viable across 360 distinct models in a production setting. In factories, smaller models yield faster inference, lower deployment costs, and clearer logic paths.

Differences between SHAP and permutation scores highlight the methodological variance inherent in explainability: SHAP assesses marginal contributions in a game- theoretic framework, while permutation reflects functional degradation upon perturbation. In practice, overlaps among the top-ranked features allow for conservative pruning while maintaining performance. Unstable or low-signal inputs—particularly stale lag windows or saturating upstream checkpoint counts—were identified as non-contributive and safely removed.

This approach also revealed spurious dependencies: overreliance on a dominant input often signaled a learned shortcut or proxy, not a causal driver. By simplifying the input space, the system achieved benefits beyond accuracy, including fewer required sensor reads, reduced preprocessing overhead, and more agile model updates.

6.2. Architectural Extensions and Model Introspection

This work directly operationalizes several unresolved gaps articulated in Jeereddy et al. (2019), who wrote, “a detailed study needs to be conducted that examines how machine learning algorithms can replace or augment current statistical process control methods. Machine learning algorithms provide more flexibility (e.g. analysis of text as a feature), can examine much longer periods of time (e.g. SPC methods typically focus on the last 30-100 units), and can detect trends with seasonality. However, evaluation of machine learning algorithms are more problematic due to their black box nature.” The current system addresses both aspects: it extends beyond SPC-style detection by embedding introspective logic into the model itself, and it mitigates black box opacity through structured attribution and pruning.

Statistical process control (SPC) assumes output stability and flags anomalies based on residual deviations over short horizons. These assumptions degrade under high-variance, weakly supervised manufacturing regimes. Rather than relying on external thresholds, this system surfaces instability from within—tracking shifts in the model’s own attribution structure as a proxy for epistemic decay. This reframes reliability as an internal property of the model’s logic, not just its outputs.

The black box limitation is addressed by applying SHAP-based attribution to tree ensemble regressors to identify which input features most consistently influence predictions

under production conditions. Attribution is not used passively; it informs architecture pruning. Features with low or unstable influence are removed, yielding smaller models with clearer decision boundaries and improved recall in ambiguous regimes. This approach tightens the alignment between model behavior and the underlying system dynamics without retraining.

The overall methodology involves training an initial model, ranking features via SHAP across time windows, and pruning those features with weak or volatile attribution signals. This produces a more compact model that is easier to audit and more stable under drift. Attribution metrics are monitored longitudinally. When variance increases or top features reorder unpredictably, the system flags this as attribution drift—an indicator of potential model logic erosion, even in the absence of output degradation. Additionally, BMW (Jeereddy et al. 2019) noted the fragility of Extra Trees models under novel disruptions: "an event that occurred during the testing showed a more difficult problem to correct. This involved an unprecedented issue with a supplier in which parts were no longer available for an extended time. As there was no historical data for such an event, the model took a significant amount of time for retraining to correct itself." Although this work does not fully resolve that issue (see future work), it introduces attribution drift as a runtime diagnostic—quantifying how and when a model's internal decision logic begins to erode under distributional shift. This allows the system to surface failure signals from within the model itself. Rather than reacting to lagging KPI deviations, early-stage instability is flagged in the causal dependencies between features and predictions.

By ranking and pruning features based on stable attribution influence—and then tracking their volatility longitudinally—this approach repositions throughput forecasting models as structural proxies for the underlying production environment. Attribution vectors function as a meta-model for system health. In this setup, explainability acts as a generative signal—a way to measure when the model's logic begins to misalign with the system it represents.

7. Future Work

In the future, I am interested in augmenting the current framework with risk-sensitive forecasting. While the existing system already raises soft flags via SHAP- based attribution drift, a dedicated risk model could formalize these alerts and escalate them through structured confidence thresholds or event-triggered protocols. This is particularly relevant for handling black swan events such as those described in the 2019 BMW study by Jeereddy et al. (2019), where an upstream supplier outage disrupted production in ways that standard forecasting models failed to anticipate. The original model lacked mechanisms to recognize when it was operating outside its training distribution. A risk model—whether through meta-prediction, distributional diagnostics, or uncertainty quantification—could provide the missing layer of awareness required to escalate anomalies before system-wide impact. This approach would enable tighter integration with operational decision systems and improve the forecasting pipeline's ability to respond to never-before-seen circumstances.

Another area of future work is defect detection, which is a strong candidate for explainability methods. Prior work (Brusa et al. 2023) has suggested that XAI can help isolate root causes. Attribution methods could highlight which inputs triggered a warning, giving operators something to verify rather than just a red light. Defect-per-unit forecasting feeds directly into the rolled throughput yield (RTY)—a cumulative measure of how defects

compound across process steps. It is one of the more actionable quality metrics, and is central to identifying where inefficiencies stack up in production. Previously reported work by BMW used a standard Kalman filter for defect trend estimation, motivated by concerns about noise in the data and false alarms in predictions—common issues when applying machine learning to manufacturing and IoT systems. The Kalman filter performs denoising and smoothing before predictions, which helps reduce false alarms. The method is simple and computationally efficient, but it is constrained by linear dynamics. This means that it can oversmooth signals, potentially masking real defects. This tradeoff was intentional: the goal was to avoid false alarms, which would waste operator and maintenance time chasing phantom issues. However, these constraints also limit responsiveness and resolution—especially for detecting short-lived or nonlinear defect patterns.

8. Conclusion

This work builds on the throughput prediction architecture of BMW (Jeereddy et al. 2019) by layering explainability-driven refinement on top of an existing Extra Trees ensemble framework. Shapley values and permutation-based feature attribution are used to eliminate redundant, unstable, or low-signal inputs. This pruning process improves MAPE from 4.98 to 4.31, reduces the model size, reduces the inference time, and strengthens interpretability. Compared to a standalone Decision Tree baseline, the final pruned Extra Trees model achieves a 26.7% reduction in error.

These improvements enable smaller, faster, and more transparent models suitable for real-time deployment across manufacturing checkpoints. Attribution tracking also acts as an early warning system—surfacing internal logic degradation before severe misalignment occurs between predictions and reality, without needing labeled anomalies. This supports that the just-in-time model overrides or retrains when the system detects that its own reasoning is drifting.

Together, these enhancements move forecasting models from opaque, high- performing black boxes toward self-aware infrastructure that can audit its own logic, reduce false confidence, and adapt to drift. In volatile production environments, this kind of structural introspection supports more robust, explainable, and operator-trustworthy automation at scale.

References

Syafrudin, M., Alfian, G., Fitriyani, N. L., & Rhee, J. (2018). Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. *Sensors*, 18(9), 2946.

Jeereddy, S., Kennedy, K., Duffy, E., Walker, A., & Vorster, B. (2019). Machine learn- ing use cases for smart manufacturing KPIs. In *Proc. IEEE Int. Conf. Big Data* (pp. 1651–1658).

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42.

Sundararajan, M., & Najmi, A. (2020). The many Shapley values for model explanation. In *Proc. Int. Conf. Mach. Learn. (ICML)* (pp. 9269–9278).

Gross, D., Spieker, H., Gotlieb, A., & Knoblauch, R. (2024). Enhancing manufacturing quality prediction models through the integration of explainability methods. *arXiv preprint* arXiv:2403.18731.

Sofianidis, G., Roˇzanec, J. M., Mladenic, D., & Kyriazis, D. (2021). A review of explainable artificial intelligence in manufacturing.

Yoo, S., & Kang, N. (2021). Explainable artificial intelligence for manufacturing cost estimation and machining feature visualization. *Expert Systems with Applications*, 183, 115360.

Senoner, J., Netland, T. H., & Feuerriegel, S. (2022). Using explainable artificial intel- ligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*, 68(8), 5704–5723.

Lampathaki, F., Agostinho, C., Glikman, Y., & Sesana, M. (2021). Moving from 'black box ' to 'glass box 'artificial intelligence in manufacturing with XMANAI. In *Proc. IEEE Int. Conf. Eng. Technol. Innov. (ICE/ITMC)*.

Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19(1), 3–26.

Bins, J., & Draper, B. A. (2001). Feature selection from huge feature sets. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)* (Vol. 2, pp. 159–165).

Bento, V., Kohler, M., Diaz, P., Mendoza, L., & Pacheco, M. A. (2021). Improving deep learning performance by using explainable artificial intelligence (XAI) approaches. *Discover Artificial Intelligence*, 1(1), 1–11.

Sun, H., et al. (2022). Utilizing explainable AI for improving the performance of neural networks. In *Proc. IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)* (pp. 1775–1782).

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Montgomery, D. C. (2007). *Introduction to Statistical Quality Control* (6th ed.). Hoboken, NJ: Wiley.

Brusa, E., Cibrario, L., Delprete, C., & Di Maggio, L. (2023). Explainable AI for machine fault diagnosis: Understanding features' contribution in machine learning models for industrial condition monitoring. *Applied Sciences*, 13(4), 20–38.