Wildauer Konferenz für Künstliche Intelligenz WiKKI - Wildau 2025
Beiträge zur Wildauer Konferenz für Künstliche Intelligenz 2025
https://doi.org/10.52825/th-wildau-ensp.v2i.2933

Published: 12 Sep. 2025

AutoML for Industrial Process Control

Analysis of its Benefits and Impact on Real Applications

Abdelrahman Elsharkawi[1,2], Danny Krautz[2], and Erik Rodner[1]

[1]KI-Werkstatt, University of Applied Sciences, Berlin
[2]CeramTec Group
*Correspondence: Erik Rodner, Erik.Rodner@htw-berlin.de

Abstract. Due to the growing complexity of modern manufacturing, industrial process control systems generate vast amounts of data with significant potential for machine learning applications. While ML offers immense benefits, the lack of data science expertise poses challenges for adoption. AutoML frameworks tackle these barriers by automating key ML tasks, enhancing accessibility and efficiency. This study investigates their effectiveness in a ceramic industry use case, comparing preprocessing strategies and analyzing explainability with SHAP values validated by domain experts. The findings highlight AutoML's potential to streamline ML model development but also its reliance on domain expertise for effective feature selection and explainability.

Keywords: Machine Learning, AutoML, Industrial Process Control, Benchmarking

1. Introduction

1.1 Background and Motivation

In modern industrial environments, the demand for accessible machine learning (ML) tools is surging [1]. This growth is driven by the increasing availability of manufacturing data, collected through high-tech sensing devices, electronic manufacturing records, mobile sensors, and Industrial Internet of Things (IIoT) technologies [2]. The applications based on these data are vast and offer significant benefits to factories, including predictive maintenance, root cause analysis, anomaly detection, and resource management [3]. These capabilities not only improve operational efficiency, but also enable proactive decision-making, helping industries reduce downtime and optimize production processes [4].

Despite this abundance of data, many production workers and engineers lack the formal data science expertise required to design and implement ML models [5]. Building ML pipelines traditionally involves manual feature engineering, model selection, hyperparameter tuning, and data preprocessing tasks that are time-consuming, error-prone, and require advanced technical skills [6]. This gap highlights the need for democratized ML solutions that empower non-experts to leverage industrial data for actionable insights [7].

AutoML frameworks, such as AutoGluon and H2O, aim to bridge this gap by automating key aspects of the ML pipeline, including feature engineering, model optimization, and validation [8] [9]. These frameworks simplify ML workflows, improve productivity, and reduce errors, especially when used correctly. Furthermore, advances in ML cloud services, such as AWS and Google Cloud, have improved the accessibility of AutoML tools, integrating them into scalable environments that facilitate efficient experimentation [10].

This study investigates the applicability of AutoML frameworks in industrial settings using the Edge Curl dataset from high-performance ceramic production as a case study. By comparing preprocessing strategies across raw, expert-preprocessed (SILVER), and collaboratively preprocessed (GOLD) datasets by data scientists and domain experts, the research evaluates AutoML frameworks such as AutoGluon and H2O, alongside traditional methods like XGBoost. The evaluation emphasizes model performance, explainability, and the extent to which automated preprocessing aligns with expert insights. To assess trustworthiness, this study systematically inspects framework outputs, evaluating transparency and alignment with domain expertise. Explainability, particularly in industrial contexts, is addressed using SHAP (SHapley Additive exPlanations) and permutation feature importance values, validated by domain experts to ensure trust and usability.

2. Related Work

This section explores various dimensions of AutoML research, including its financial benefits, implications for human involvement, transparency challenges, automated preprocessing capabilities, and applications in industrial settings. For each topic, multiple sources were reviewed, but we selected two to three papers based on their relevance and the quality of the work. Each subsection synthesizes findings, identifies gaps, and provides context for this study’s contributions.

2.1 Literature Review

Financial Benefits of using AutoML: Ayyalasomayajula [11] analyze the cost-effectiveness of AutoML workflows in public cloud environments, comparing them with traditional machine learning methods. The study evaluates tasks like image classification and object detection, considering metrics such as model accuracy, computational cost, and resource utilization. It finds that AutoML pipelines often match or exceed manual approaches in performance, with significant efficiency gains due to advanced hyperparameter tuning and resource optimization. The authors suggest hybrid approaches combining AutoML and human expertise to address these challenges.

Similarly, Rosario [12] emphasized AutoML’s transformative business impact by reducing costs, accelerating decision-making, and improving return on investment (ROI). The study highlighted operational scalability and market adaptability as key advantages but acknowledged transparency concerns and potential job displacement as critical challenges. Together, these studies position AutoML as a driver of efficiency but underscore the need for strategic human oversight and ethical considerations.

Replacing Human Expertise: The debate around AutoML replacing human expertise has produced mixed findings. AutoML in the Wild: Obstacles, Workarounds, and Expectations [13] investigates this issue through interviews with ML practitioners to evaluate the real-world usability of AutoML tools. The study identifies significant limitations, with one of the biggest concerns among both technical and management users being the inability of AutoML to address complex domain-specific requirements. A key workaround proposed is the integration of human-in-the-loop systems, where domain experts contribute at essential stages such as feature selection and model evaluation. Another prevalent solution involves tailoring AutoML pipelines with custom configurations to meet industry-specific needs, which partially mitigates these challenges.

Crisan [14] echoed these concerns, emphasizing the balance between automation and human expertise in enterprises. Unlike AutoML in the Wild, which focused on technical issues and workarounds, this study highlighted organizational factors, showing that human involvement remains indispensable for interpreting results and ensuring ethical AI practices. Both studies agree that AutoML alone cannot fully replace human expertise but differ in their emphasis on technical versus organizational challenges.

Azevedo [15], through a systematic review, found that AutoML's limitations often outweigh its advantages in replacing human expertise. Challenges include a lack of interpretability, difficulties generalizing across workflows, and reliance on extensive preprocessing. However, it highlighted AutoML's strengths in repetitive tasks and accessibility for non-experts. Collectively, these studies affirm that AutoML is best used as a complement to human expertise rather than a replacement.

Transparency in AutoML: Transparency remains a significant challenge for AutoML frameworks. Stoica [16] noted that while methods such as SHAP values improve interpretability, processes like automated feature selection and model evaluation remain opaque. Factors like ensemble techniques and insufficient documentation contribute to AutoML's black-box nature.

Drozdal [17] demonstrated that incorporating transparency features such as performance metrics and visualizations enhances trust in AutoML systems. However, they highlighted limitations, including a lack of detailed explanations for model selection and feature engineering processes. Both studies agree on the importance of transparency but emphasize different aspects: Stoica focused on technical challenges, while Drozdal addressed user trust and information needs.

Automated Data Preprocessing: Mumuni [18] surveyed automated preprocessing techniques, including data cleaning, transformation, and augmentation, emphasizing their role in improving model accuracy while reducing manual effort. However, the study noted that current systems struggle to handle context-specific requirements independently. Techniques like GAN-based synthetic data generation and meta-learning for adaptive preprocessing show promise but still require human oversight to address edge cases.

AutoML on Industrial Data: Garouani [7] introduced AMLBID, an AutoML system tailored to industrial applications. By leveraging meta-learning and interactive explainers, AMLBID simplifies pipeline selection for non-experts but relies on clean datasets due to limited preprocessing capabilities.

Chaabi [19] evaluated various AutoML frameworks, including AMLBID, AutoGluon, and H2O, on manufacturing datasets for predictive maintenance and quality management. AMLBID excelled in predictive maintenance tasks due to its meta-feature-driven model selection. However, its reliance on clean data limited broader applicability. These studies underscore AutoML's potential for industrial data but highlight preprocessing as a key limitation.

2.2 Gap Analysis

Current AutoML research reveals several limitations, particularly in industrial applications. Despite advancements, gaps persist in applying AutoML to real-world scenarios with complex, diverse datasets. This study focuses on addressing the following gaps:

1. **Lack of Real-World Industrial Data Evaluation**: Existing studies, like [17] and [18], heavily rely on curated datasets that miss industrial complexities such as noise and missing values. In particular, several works ([7], [16], [19]) highlight reliance on clean data, limiting applicability in messy, real-world settings. This research uses authentic ceramic production datasets to address these challenges.
2. **Neglect of Human Expertise in Preprocessing**: Studies such as [15] and [16] recognize the value of human input but rarely quantify it. The paper of Azevedo [15] notes AutoML struggles with domain-specific preprocessing, while [16] highlights the necessity of expert interpretation. This study evaluates the added value of expert preprocessing through RAW, SILVER, and GOLD datasets.
3. **Automated Preprocessing Limitations**: Tools often fail to address real-world issues like inconsistent placeholders or irrelevant features. For instance, Mumumi [18] explores automation but overlooks industrial-specific challenges such as managing diverse data formats and debugging preprocessing outcomes. This study analyzes these gaps, assessing their effects on performance and interpretability.

This study addresses these gaps by:

- Evaluating AutoML frameworks with real manufacturing data.
- Quantifying the impact of domain expertise on preprocessing and performance.
- Enhancing pipeline transparency and validating outputs with domain experts.
- Highlighting and addressing preprocessing challenges such as placeholder handling.

These efforts enhance the understanding of AutoML's capabilities and limitations, offering practical insights for industrial applications.

3. Method

This section details the methodology employed to evaluate the usability of AutoML frameworks in the context of industrial data, with a focus on data preprocessing, model evaluation, model performance, and explainability.

3.1 Dataset and Preprocessing

The dataset used in this study focuses on predicting the occurrence of edge curl, a production defect in high-performance ceramics, based on parameters recorded at different stages of the manufacturing process. It has been processed by domain experts and data scientists at three distinct levels, as described in Table 1. For confidentiality reasons, all feature names in this paper are anonymized.

As shown in the data flow chart in Figure 1, the datasets undergo automated preprocessing and prediction through the AutoML frameworks. XGBoost, however, could only process the GOLD dataset due to its strict requirement for clean, fully numeric or encoded data, making it suitable for structured datasets without noise or non-numeric entries. The resulting data from the automated preprocessing are then inspected to

Table 1. *Description of the datasets used in the study, including preprocessing levels and modifications.*

Dataset	Description
RAW	Unprocessed dataset. All 197 original columns are included without modifications or feature selection. The dataset consists of 5456 rows.
SILVER	Irrelevant or redundant columns are removed based on domain experts' knowledge to enhance dataset relevance.
GOLD	Fully feature-engineered dataset. Categorical variables are encoded, features are split where necessary, and dummy variables are created. Additionally, one of each pair of highly correlated columns is removed. This version is preprocessed by a data scientist to optimize model performance through advanced feature engineering.

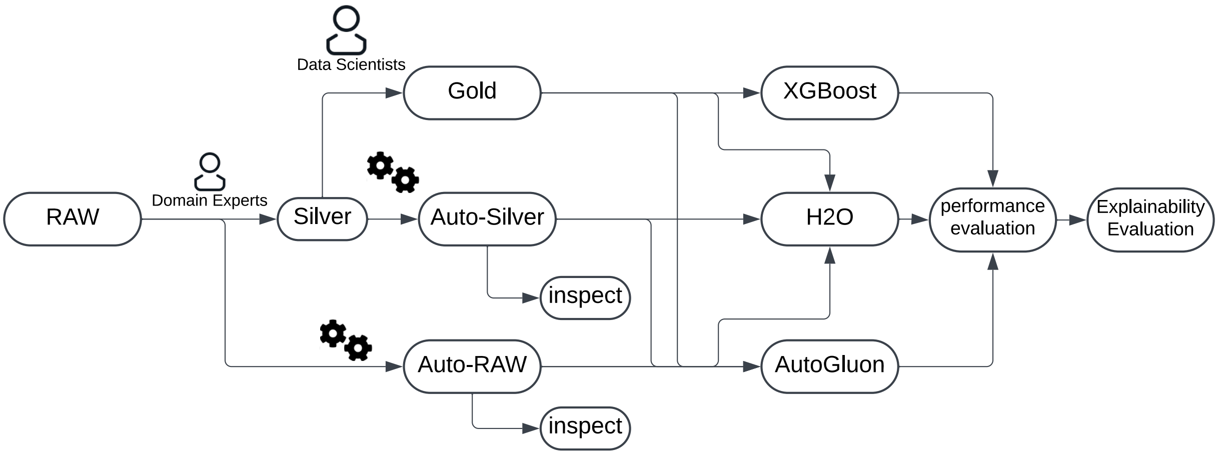


Figure 1. *The data flow chart shows the different steps the data went through during preprocessing and prediction for the* RAW*,* SILVER*, and* GOLD *datasets.*

identify whether the frameworks' automated feature engineering aligns with expert-driven preprocessing and whether it meaningfully contributes to model performance.

3.2 Evaluation Methodology of the AutoML Frameworks

Our experiments utilize two types of cross-validation: k-fold with subsequent data splits and repeated k-fold with randomized data splits. These methods are employed to test the effect of random splitting of data with temporal patterns. Mean Absolute Error (MAE) was selected as the primary evaluation metric because it offers a clear and interpretable measure of average prediction errors, expressed in the same units as the target variable. This characteristic makes MAE particularly suitable for industrial applications where transparency and interpretability are essential [20].

The default behavior of the AutoML frameworks during model evaluation is examined to determine whether any inherent issues could affect evaluation reliability or introduce biases in the results.

The study also conducts an in-depth analysis of the explainability of models generated in each experiment. Explainability is assessed through the application of SHAP values [21] and permutation feature importance. These results will be compared with SHAP explanations that are reviewed and validated by domain experts. This rigorous validation process is critical in industrial contexts, where trust and usability often depend on clear and accurate model explanations.

Through this comprehensive methodology, the study aims to reduce the black-box nature of AutoML frameworks by explaining their outputs, thoroughly inspecting and analyzing their automated data preprocessing, and evaluating their performance across datasets of varying quality.

4. Results

4.1 Automated Data-Preprocessing Inspection

This section systematically investigates how the AutoML frameworks H2O [8] and AutoGluon [9] handle data preprocessing and compares the resulting data to that processed by domain experts and data scientists. These frameworks were chosen for their widespread adoption and their ability to perform automated data preprocessing. The evaluation focuses on the transparency of their preprocessing pipelines, the appropriateness of the applied transformations in addressing the dataset's specific characteristics, and the resulting data quality.

4.1.1 H2O

The preprocessing steps in H2O AutoML encompass automatic imputation of missing values, normalization when required, and one-hot encoding tailored specifically for XGBoost models. Furthermore, H2O's tree-based models, such as Gradient Boosting Machines (GBM) and Random Forests, inherently support group-splits on categorical variables, allowing them to natively process such data without additional encoding. However, in contrast to AutoGluon, H2O's pipeline lacks an interface for interacting with individual preprocessing steps. This design choice amplifies the ”black-box” nature of the H2O framework, reducing transparency and constraining the detailed inspection of intermediate preprocessing outputs. As a result, precise examination of these internal transformations is not feasible, which can pose significant challenges in scenarios demanding fine-grained control over preprocessing workflows.

4.1.2 AutoGluon

AutoGluon's preprocessing pipeline comprises 15 specialized feature generators, each tailored to address specific preprocessing tasks. These generators can either be custom-configured to suit dataset-specific requirements or employed using default parameters provided by the framework. Leveraging metadata and dataset structure, AutoGluon intelligently selects the most suitable feature generators, enabling a dynamic and adaptive preprocessing workflow.

Table 4 provides a general overview of the key feature generators auto-selected by AutoGluon for the data under consideration, detailing their primary functionality and usage across different preprocessing stages (RAW and SILVER datasets). Detailed documentation of the generators is given in the official documentation of AutoGluon [22].

In the subsequent paragraphs, the differences between the RAW and SILVER datasets will be discussed first, followed by a detailed analysis of feature generators whose impact deviated from expected behavior based on the official documentation. Generators that performed as documented will not be examined in detail.

Differences between RAW and SILVER: The primary differences between the preprocessing generators used by AutoGluon on the RAW and SILVER datasets lie in the usage of the text and date feature generators. In the RAW dataset, AutoGluon recognized only one text feature, Position, which was absent in the SILVER dataset. Similarly, all time-related features were excluded from the SILVER dataset. These features were removed by domain experts due to their irrelevance, and the higher dimensionality of the generated features further amplified their irrelevance, adversely affecting both model performance and interpretability.

Categorical features: The `'CategoryFeatureGenerator'` is tasked with label encoding for categorical features, handling data types such as `'object'`, `'category'`, and `'bool'`. This process utilizes Pandas functions like `'cat.categories'` and `'astype(CategoricalDtype())'` to achieve the desired encoding. In the analyzed dataset, certain features were misclassified as `'object'` due to inconsistent representations of missing values, such as the placeholder `'TBD'`. AutoGluon failed to identify `'TBD'` as a representation for missing values in a predominantly numeric column, treating the feature as an object type and label-encoding the `'TBD'` entries.

For instance, the feature `'F92'` contained seven unique floating-point temperature values along with `'TBD'` entries. This led to the automatic conversion of the column to the `'object'` data type, followed by label encoding. Using cat.categories ensures that lower temperature values receive lower index labels if categories are explicitly defined as ordered (ordered=True) and sorted accordingly. However, inconsistencies arise when temperature features with placeholders like `'TBD'` are inferred as `'object'` in some instances but as 'float' in others. Such discrepancies render the encoding schema invalid.

Moreover, the validity of this approach depends on all temperature-related features receiving consistent labels for identical temperature values representation throughout the dataset. Any misalignment in label assignments across columns representing the same temperature scale can lead to encoding inconsistencies, adversely affecting downstream machine learning tasks reliant on uniform numerical representations.

To address these challenges, columns containing placeholder strings like `'TBD'` must be explicitly handled before encoding to ensure uniform data types across all features with the same units or representing the same concept. This involves converting placeholder values into a standardized representation, such as NaN, aligning with the dataset's semantic expectations and facilitating accurate label encoding.

Text features: Text columns are identified by evaluating the uniqueness of their rows. Specifically, a column is classified as text if the ratio of unique entries to total rows exceeds one percent and most rows contain multiple distinct words [9]. In the dataset used for this study, only one feature, `'F184'`, met these criteria and was encoded by the `'TextSpecialFeatureGenerator'` into the derived features: `['F184.char_count', 'F184.word_count', 'F184.capital_ratio', 'F184.lower_ratio', 'F184.digit_ratio', 'F184.special_ratio', 'F184.symbol_count', 'F184.symbol_ratio']`.

As previously mentioned, domain experts identified the `Position` feature as irrelevant in the cleaned dataset versions. As a result, all derived features also inherited this irrelevance. Retaining this feature in the automatically preprocessed raw data and amplifying its dimensions introduces unnecessary noise, inflates computational costs, and reduces overall interpretability. Furthermore, its presence may cause the model to identify spurious correlations, thereby compromising both predictive performance and generalization on unseen data.

Unique features: At the initial stage of automated preprocessing, the `'AsTypeFeatureGenerator'` removed the unique features originally present in the RAW dataset (those not generated by any feature generators). Subsequently, additional features were generated by the `'DatetimeFeatureGenerator'`, `'TextSpecialFeatureGenerator'`, and `'TextNgramFeatureGenerator'`. Among these newly created features, some were identified as unique. These features were removed by the `'DropUniqueFeatureGenerator'`, effectively eliminating 31 unique features introduced by these generators.

Dropping duplicates: The `'DropDuplicatesFeatureGenerator'` is designed to identify and remove duplicated features, retaining only the first feature in each group of duplicates. While this approach ensures the elimination of redundant information, it also introduces a potential error: the automated removal of an original feature in favor of a generated feature. Although this issue was not observed in our specific case, an inspection of the framework revealed no safeguards to prevent such occurrences.

If an original column is replaced by a generated column during the automated deduplication process, the predictive accuracy of the model remains unaffected. However, this replacement could significantly reduce the interpretability of the model. Original feature names carry semantic meaning that aids in understanding model decisions, whereas generated feature names can obscure this clarity. For instance, if a duplication occurs between an original temperature feature and a generated date feature, the expert interpreting the model may struggle to identify that a generated date feature appears in the explanation plot because the original temperature feature was removed.

Ensuring that original features are prioritized during the deduplication process is crucial for maintaining both the interpretability and reliability of machine learning models.

4.1.3 Conclusion

AutoML frameworks like AutoGluon and H2O streamline data preprocessing but require domain expertise for optimal outcomes. H2O's preprocessing pipeline offers limited transparency due to its black-box nature, whereas AutoGluon provides a structured and traceable approach through its feature generators. However, AutoGluon struggles with domain-specific nuances, such as handling inconsistent missing values (e.g., "TBD") or filtering irrelevant features like Position and time-related columns, which were retained in the RAW data.

While these frameworks significantly reduce manual effort, they lack the capability to autonomously address domain-specific preprocessing challenges. This underscores the importance of integrating domain knowledge with automated preprocessing to ensure robust and interpretable models.

4.2 Performance Comparison of AutoML and Traditional ML

This section presents the Mean Absolute Error (MAE) of model predictions across different datasets using the cross-validation methods described in the methodology section.

Table 2 demonstrates that repeated k-fold cross-validation generally yields better scores compared to standard k-fold cross-validation. However, whether this improvement indicates genuinely enhanced performance will be analyzed in subsequent sections. XGBoost was unable to process the RAW or SILVER datasets due to its requirement for clean, fully numeric or encoded data. In contrast, both AutoML frameworks successfully handled these datasets after automatically preprocessing them.

Among the results, AutoGluon outperformed H2O on the silver dataset, while H2O achieved the best results for the RAW dataset using standard 3-fold cross-validation. AutoGluon, however, excelled in repeated 3-fold cross-validation across all datasets. For the GOLD dataset, AutoGluon achieved superior results with repeated k-fold cross-validation, while XGBoost performed best using standard 3-fold cross-validation.

Table 2. *Comparison of 3x Repeated 2-Fold Cross-Validation and 3-Fold CV. In 3x repeated 2-fold cross-validation, the dataset is randomly split into two equal parts for each fold, with 50% of the data used for training and 50% for testing. The sampling is random and non-sequential (the samples in the figure are hypothetical). In contrast, 3-fold cross-validation divides the data into three consecutive and fixed folds, with each fold serving as the test set once while the remaining two are used for training.*

Cross Validation	Dataset	AutoGluon	H2O	XGBoost
3 folds CV	RAW	0.022208	0.018066	-
	SILVER	0.022716	0.022862	-
	GOLD	0.022449	0.025377	0.020470
Repeated 3 folds CV with 2 splits	RAW	0.006049	0.006088	-
	SILVER	0.009871	0.009902	-
	GOLD	0.009903	0.009938	0.009908

4.3 Evaluating Model Validation Strategies in AutoML Frameworks

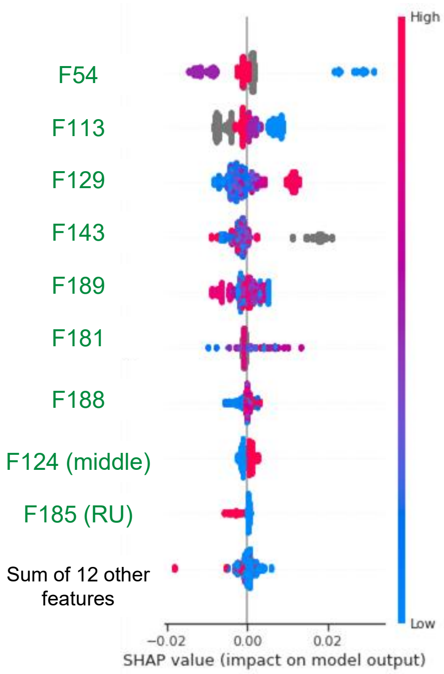
This section evaluates how the frameworks handle model evaluation and examines any potential negative effects on the resulting models.

H2O's AutoML framework employs k-fold cross-validation as its default strategy for evaluating model performance, utilizing 6 folds by default. A notable feature is its `'split_column'` parameter, which allows users to specify a column, such as a time-related column, for splitting. This ensures that data from the same time frame is not shared between training and validation sets.

AutoGluon's cross-validation strategy employs 8 folds by default, using random splitting of the dataset into training and validation sets to assess model performance.

Random splitting of data in temporally ordered contexts introduces bias in model evaluation by disregarding the temporal dependencies inherent in the data. This can lead to data leakage, where information from the same time context influences the training process, compromising the integrity of model evaluation, and resulting in overly optimistic performance estimates [23]. While the use of a separate test set that does not overlap temporally with training or validation data can mitigate data leakage during testing, it does not address overfitting to temporal patterns while training or the mismatch between training conditions and real-world scenarios where data from different time frames are used [24].

The `'split_column'` option provided by H2O is a useful feature, but its effectiveness depends on the user's awareness of the need for structured splitting. However, the default settings of both frameworks, which rely on random splitting, can have a negative effect on model evaluation when data contains temporal or contextual dependencies. This default behavior disregards the structured nature of the data and risks introducing



	Feature being analyzed	Known impact by Domain Experts	Interpretation
	F54	TBD	Low F45 pushes edge curl value up
★	F113	new-meaningful	Low F113 pushes edge curl value up
★	F129	new-meaningful	High F129 pushes edge curl value up
	F143	Hypothesized	Unclear impact
☑	F189	Proven	F189 has both positive and negative influence on edge curl value across samples
☑	F181	Proven	Same interpretation as F189
☑	F188	Proven	Low F188 values push the edge curl value down
☑	F124 (middle)	Hypothesized	Middle F124 push the edge curl values up
☑	F185 (RU)	Proven	Substrates in F185 (RU) push the edge curl values down

★ New findings ☑ Confirmatory

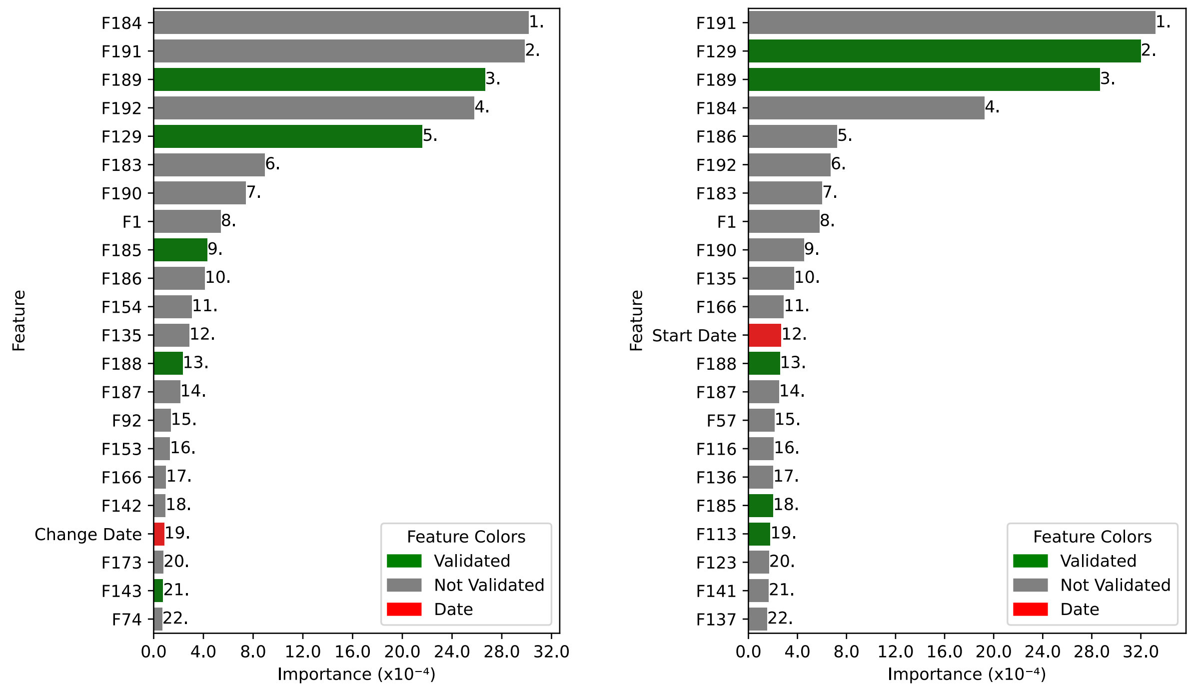
Figure 2. *SHAP values for the XGBoost model trained on the* GOLD *dataset. And the validation of the outcomes by domain experts*

bias and overfitting, making it essential for users to intervene with appropriate expertise to mitigate these issues and ensure proper handling of data.

4.4 Model Explainability

Figure 2 displays a SHAP values plot generated by explaining an XGBoost model trained on the GOLD dataset without cross-validation. This plot illustrates the influence of individual features on the model's predictions, where features with higher SHAP values have a greater impact on the output. The validation by domain experts, as shown in Figure 2, revealed a strong alignment between the model's results and the input provided by the experts.

Figures 3a, 3b, and 4 present the permutation feature importance [9] derived from the predictions of different AutoGluon models. Figure 3a, for the model trained on the RAW dataset using 3-fold cross-validation, shows 5 matches with the validated features, with the first date feature appearing at position 19. The role of date variables will be discussed later. Figure 3b also displays 5 matches, but the date feature appears earlier, at position 12. Figure 4 represents the feature importance by the predictions of the model trained on the SILVER dataset using 3-fold cross-validation, showing a closer alignment with the validated features, with 8 out of 9 validated features appearing in higher positions. No date variables are present here, as they were removed from the SILVER dataset based on expert recommendations. Lastly, Figure 5 illustrates the SHAP values for the H2O model trained on the SILVER dataset, showing 8 matches and similar directional trends to the validated SHAP plot.



***(a)** Feature Importance for AG on the* RAW *dataset with 3 folds cv*

***(b)** Feature Importance for AG on the* RAW *dataset with repeated 2 splits 3 fold cv*

***Figure 3.** Comparison of permutation feature importance derived from AutoGluon models with different Cross validation methods.*

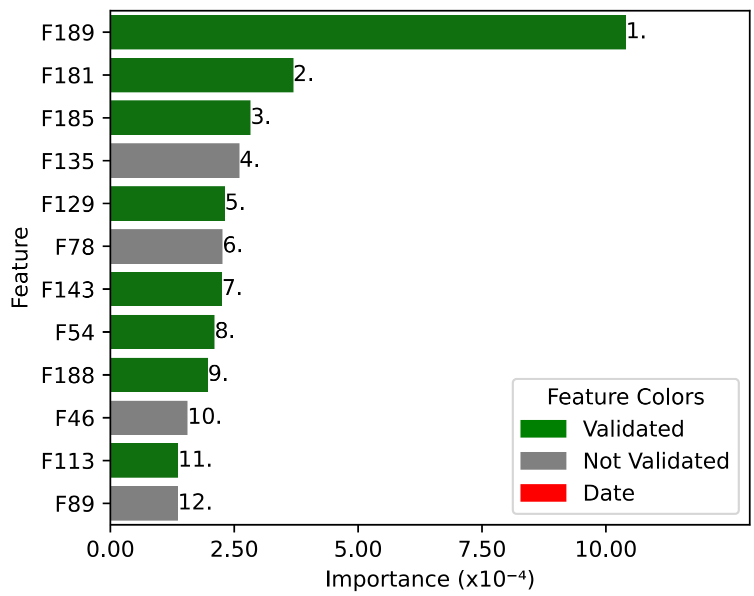


***Figure 4.** Permutation feature importance derived from the AutoGluon model trained on the* SILVER *dataset using 3-fold cross-validation.*

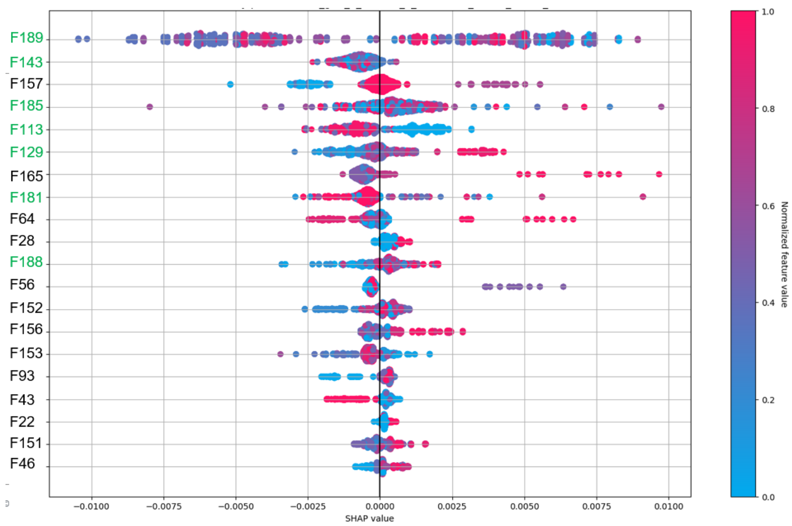


Figure 5. *SHAP values for the H2O model trained on the* SILVER *dataset. Green features indicate matches with the validated features from the XGBoost validated SHAP Plot.*

5. Conclusions

This study examined AutoML frameworks in industrial process control using a real-world ceramic production dataset. While AutoML accelerates machine learning model development, it heavily relies on the data quality and pre-processing applied beforehand. In our case, this has been investigated with three levels of involved manual preprocessing, which we term RAW, SILVER, and GOLD. With unprocessed RAW data, AutoML failed to remove irrelevant features that domain experts had identified for exclusion leading to a reduced model performance. Using the SILVER dataset, it made errors in feature engineering, mishandled NaN values, and incorrectly encoded data, demonstrating its limitations in automatically processing data at different refinement stages.

Nonetheless, AutoML excels in model selection, ensembling, and hyperparameter optimization (HPO), consistently achieving competitive Mean Absolute Error (MAE) across all dataset versions. It effectively automates the identification of high-performing models and fine-tunes them for optimal accuracy. These results show that despite preprocessing challenges, AutoML optimizes architectures and parameters, often generating strong predictive models even with corrupted or invalid data. This resilience makes it a valuable tool for rapid model development in industrial applications, particularly for model selection and tuning, though it still requires high-quality, representative data for reliable and interpretable results.

While AutoML lowers the technical barrier to machine learning in manufacturing, it is not a substitute for expert-driven preprocessing. Future research should focus on enhancing AutoML's preprocessing capabilities, better integrating domain knowledge, and refining validation techniques to improve alignment with real-world industrial constraints.

6. Appendix

Data availability statement

The data used in this study is not publicly available. The dataset belongs to CeramTEC GmbH and cannot be shared due to confidentiality restrictions. Access to the data is strictly limited to authorized personnel.

Author contributions

Table 3. *Author contributions following the CRediT taxonomy, detailing individual roles in the research process.*

Role	Contributors
Conceptualization	Abdelrahman Elsharkawi, Erik Rodner
Methodology	Abdelrahman Elsharkawi
Software	Abdelrahman Elsharkawi
Validation	Abdelrahman Elsharkawi, Erik Rodner
Formal Analysis	Abdelrahman Elsharkawi
Investigation	Abdelrahman Elsharkawi
Data Curation	Abdelrahman Elsharkawi
Resources	CeramTec GmbH
Writing – Original Draft	Abdelrahman Elsharkawi, Erik Rodner
Writing – Review and Editing	Abdelrahman Elsharkawi, Erik Rodner
Visualization	Abdelrahman Elsharkawi
Supervision	Danny Krauz, Erik Rodner
Project Administration	Abdelrahman Elsharkawi, Erik Rodner, Danny Kautz

Competing interests

The authors declare that they have no competing interests.

Funding

This research was supported by CeramTec GmbH (Danny Kautz, Abdelrahman Elsharkawi; data, compute, and personnel costs) as well as the University of Applied Sciences Berlin (Erik Rodner; personnel costs).

Acknowledgments

We acknowledge the data science team at CPP Investments for creating the dataset used in this study and the domain experts at CeramTec GmbH for validating the feature importance. This study was also supported by KI-Werkstatt (University of Applied Sciences, Berlin).

Underlying and related material

Table 4. *Overview of AutoGluon's feature generators [9], their primary functionality, and their usage across different stages of the Edge Curl dataset (RAW and SILVER).*

Generator	Description	RAW	SILVER
AsTypeFeatureGenerator	Enforces type conversion to match types seen during fitting. - Encodes features with only two unique values (including nan) to binary features (0,1). - Drops unique Features	Used	Used
FillNaFeatureGenerator	By default takes only features with the data type object and replaces the nan with an empty string. The final data frame still contained Na	Used	Used
IdentityFeatureGenerator	Identifies the features, that will not be processed in the subsequent generators and passes them without change.	Used	Used
CategoryFeatureGenerator	Converts object types to categories and removes rare ones. -	Used	Used
DatetimeFeatureGenerator	Transforms datetime features into numeric representations. (Unix time stamp in nanoseconds, year,month,day,day of week)	Used	Used
TextSpecialFeatureGenerator	Extracts specific attributes from raw text features.	Used	Not Used
TextNgramFeatureGenerator	Creates n-gram features from text data.	Used	Not Used
DropUniqueFeatureGenerator	Drops features with only one or mostly unique values.	Used	Used
DropDuplicatesFeatureGenerator	Removes duplicate features, retaining one instance.	Used	Used

References

[1] G. Martínez-Arellano and S. Ratchev, "Towards frugal industrial ai: A framework for the development of scalable and robust machine learning models in the shop floor," *The International Journal of Advanced Manufacturing Technology*, pp. 1–23, 2024.

[2] C. Bulla and M. N. Birje, "Anomaly detection in industrial iot applications using deep learning approach," *Artificial Intelligence in Industrial Applications: Approaches to Solve the Intrinsic Industrial Optimization Problems*, pp. 127–147, 2022.

[3] A. Tsanousa et al., "A review of multisensor data fusion solutions in smart manufacturing: Systems and trends," *Sensors*, vol. 22, no. 5, 2022, ISSN: 1424-8220. DOI: 10.3390/s22051734. [Online]. Available: https://www.mdpi.com/1424-8220/22/5/1734.

[4] A. Ahuja and M. Gupta, "Optimizing predictive maintenance with machine learning and iot: A business strategy for reducing downtime and operational costs," 2024.

[5] A. Scriven, D. J. Kedziora, K. Musial, and B. Gabrys, *The technological emergence of automl: A survey of performant software and applications in the context of industry*, 2022. arXiv: 2211.04148 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2211.04148.

[6] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," *KI-Künstliche Intelligenz*, vol. 29, pp. 329–337, 2015.

[7] M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, and A. Lewandowski, "Towards big industrial data mining through explainable automated machine learning," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 1, pp. 1169–1188, May 1, 2022, ISSN: 1433-3015. DOI: 10.1007/s00170-022-08761-9. Accessed: Nov. 29, 2024. [Online]. Available: https://doi.org/10.1007/s00170-022-08761-9.

[8] E. LeDell and S. Poirier, "H2o automl: Scalable automatic machine learning," in *Proceedings of the AutoML Workshop at ICML*, ICML San Diego, CA, USA, vol. 2020, 2020.

[9] N. Erickson et al., *Autogluon-tabular: Robust and accurate automl for structured data*, 2020. arXiv: 2003.06505 [stat.ML]. [Online]. Available: https://arxiv.org/abs/2003.06505.

[10] M. A. Al Alamin and G. Uddin, "How far are we with automated machine learning? characterization and challenges of automl toolkits," *Empirical Software Engineering*, vol. 29, no. 4, p. 91, 2024.

[11] M. M. T. Ayyalasomayajula, S. K. Chintala, and S. Ayyalasomayajula, "A cost-effective analysis of machine learning workloads in public clouds: Is automl always worth using," *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 7, no. 5, pp. 107–115, 2019.

[12] A. T. Rosário and A. C. Boechat, "How automated machine learning can improve business," *Applied Sciences*, vol. 14, no. 19, p. 8749, Jan. 2024, Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app14198749. Accessed: Nov. 12, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/19/8749.

[13] Y. Sun, Q. Song, X. Gui, F. Ma, and T. Wang, "AutoML in the wild: Obstacles, workarounds, and expectations," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23, New York, NY, USA: Association for Computing Machinery, Apr. 19, 2023, pp. 1–15, ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3581082. Accessed: Oct. 22, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3544548.3581082.

[14] A. Crisan and B. Fiore-Gartland, "Fits and starts: Enterprise use of AutoML and the role of humans in the loop," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21, New York, NY, USA: Association for Computing Machinery, May 7, 2021, pp. 1–15, ISBN: 978-1-4503-8096-6. DOI: 10.1145/3411764.3445775. Accessed: Nov. 27, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3411764.3445775.

[15] K. Azevedo, L. Quaranta, F. Calefato, and M. Kalinowski, *A multivocal literature review on the benefits and limitations of automated machine learning tools*, Jan. 21, 2024. DOI: 10.48550/arXiv.2401.11366. arXiv: 2401.11366. Accessed: Nov. 27, 2024. [Online]. Available: http://arxiv.org/abs/2401.11366.

[16] F. Stoica, L. F. Stoica, F. Stoica, and L. F. Stoica, *AutoML Insights: Gaining Confidence to Operationalize Predictive Models*. IntechOpen, Jun. 5, 2024, ISBN: 978-0-85466-835-9. DOI: 10.5772/intechopen.1004861. Accessed: Nov. 27, 2024. [Online]. Available: https://www.intechopen.com/online-first/1179126.

[17] J. Drozdal et al., "Trust in automl: Exploring information needs for establishing trust in automated machine learning systems," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, ser. IUI '20, ACM, Mar. 2020, pp. 297–307. DOI: 10.1145/3377325.3377501. [Online]. Available: http://dx.doi.org/10.1145/3377325.3377501.

[18] A. Mumuni and F. Mumuni, "Automated data processing and feature engineering for deep learning and big data applications: A survey," *Journal of Information and Intelligence*, Jan. 2024, ISSN: 2949-7159. DOI: 10.1016/j.jiixd.2024.01.002. [Online]. Available: http://dx.doi.org/10.1016/j.jiixd.2024.01.002.

[19] M. Chaabi, M. Hamlich, and M. Garouani, "Evaluation of AutoML tools for manufacturing applications," in *Advances in Integrated Design and Production II*, L. Azrar et al., Eds., Cham: Springer International Publishing, 2023, pp. 323–330, ISBN: 978-3-031-23615-0. DOI: 10.1007/978-3-031-23615-0_33.

[20] M. Naser and A. Alavi, "Insights into performance fitness and error metrics for machine learning," *arXiv preprint arXiv:2006.00887*, 2020.

[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon et al., Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[22] A. G. Team, *Autogluon feature engineering documentation*, 2025. [Online]. Available: https://auto.gluon.ai/stable/api/autogluon.features.html.

[23] F. Liu, L. Chen, Y. Zheng, and Y. Feng, "A prediction method with data leakage suppression for time series," *Electronics*, vol. 11, no. 22, 2022, ISSN: 2079-9292. DOI: 10.3390/electronics11223701. [Online]. Available: https://www.mdpi.com/2079-9292/11/22/3701.

[24] A. Prakash, R. Tuo, and Y. Ding, *The temporal overfitting problem with applications in wind power curve modeling*, 2022. arXiv: 2012.01349 [stat.AP]. [Online]. Available: https://arxiv.org/abs/2012.01349.