Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen

doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P7259-cd

Failure Causality Diagnostic in Industrial Systems through Automated Machine Learning

Bahman ASKARI

Department of Electrical and Information Engineering, Polytechnic University of Bari, Italy.
E-mail: bahman.askari@poliba.it

Hasan MISAII

University of Tehran, Tehran, Iran. E-mail: hasanmisaii14@gmail.com

Amélie PONCHET DURUPT

Université de technologie de Compiègne, Roberval, Compiègne, France. E-mail: amelie.durupt@utc.fr

Mitra FOULADIRAD

Aix Marseille University, CNRS, Centrale Med, M2P2, Marseille, France
E-mail: mitra.fouladirad@centrale-marseille.fr

Antoine GRALL

Université of Technologie de Troyes, LIST3N, France. E-mail: antoine.grall@utt.fr

Negin JAFARI

University of Tehran, Tehran, Iran.. E-mail: negin.e.jafari@gmail.com

In the context of modern industrial systems, efficient failure management is crucial to maintain operational integrity, minimize downtime, and optimize maintenance. This paper explores the application of Automated Machine Learning (AutoML) to enhance both the diagnostic and the prognostic of failure causality in industrial systems. Different failure causes are detected by failure causality diagnostics, and the upcoming failure could be prevented by failure causality prognostics. In fact, future failures could be avoided by preventing their causalities. Traditional machine learning (ML) approaches require significant manual intervention for model selection, hyperparameter tuning, and feature engineering, which can be time-consuming and cost-consuming. AutoML, on the other hand, automates these processes, enabling faster and more accurate predictions while reducing the need for extensive domain expertise. AutoML could be applied for prognostics, predicting the remaining useful life (RUL) of components and foreseeing future failures. This paper integrates AutoML into real-time failure diagnostics, identifying the root causes of system malfunctions using historical and sensor data. The Steel Plates Faults industrial real-world data set is considered to be surveyed for fault detection using AutoML. The run times and accuracy acquired by AutoML are stated to clarify its superiority.

Keywords: Failure Causality, Machine Learning, Automated Machine Learning, Diagnostic, Health Monitoring, Competing Risks, Steel Plates Faults.

1. Introduction

In industrial systems, failure diagnostics and prognostics are critical to maintaining operational efficiency and minimizing downtime Yu et al. (2019). Traditional machine learning (ML) algorithms have been used to detect misbehavior, diagnose faults, and predict equipment failures by analyzing historical and real-time data Askari et al. (2023); Misaii et al. (2024); Askari et al. (2023). In recent years, ML and artificial intelligence (AI) have attracted the attention of the research community. Today ML is a necessary aspect of modern business and research for many organizations. In modern industrial systems, maintaining operational reliability and minimizing downtime are critical objectives Friederich and Lazarova-Molnar (2024). The complexity of such systems, however, poses significant challenges in identifying and addressing causes of failure. Failures can propagate across components, leading to cascading effects that compromise safety, efficiency, and productivity Smith et al. (2017); Seligmann et al. (2019). Diagnosing and predicting these failures is essential for implementing proactive maintenance strategies and enhancing system resilience. Traditional ML methods require significant human expertise for feature and model selection, and parameter tuning, which can be time-consuming and may not fully capture the complexities of industrial processes. Consequently, while traditional ML has advanced predictive maintenance and fault detection, it often requires substantial manual intervention and domain knowledge Gkioka et al. (2024).

The advent of Automated Machine Learning (AutoML) has opened new possibilities for tackling these challenges Baratchi et al. (2024); Feurer et al. (2015); Zöller and Huber (2021). By automating the process of model selection, hyperparameter tuning, and feature engineering, AutoML empowers practitioners to efficiently deploy sophisticated diagnostic and prognostic models without requiring deep expertise in data science Thornton et al. (2013); Hutter et al. (2019). This wider accessibility of ML aligns well with the needs of industrial systems, where domain experts often face constraints in time and resources to develop custom analytical solutions.

This paper presents an innovative framework for failure causality diagnostics in industrial systems through AutoML. Our approach leverages the capabilities of AutoML to uncover hidden patterns in operational data, identify root causes of failures, and predict potential future failures with high accuracy Salehin et al. (2024). The proposed methodology highlights correlations and uncovers causal relationships to provide actionable insights for decision-makers. The key contributions of this work are threefold:

(1) Development of a robust AutoML-based framework tailored to the complexities of industrial systems.
(2) Enhancing the interpretability and reliability of failure diagnostics Peters et al. (2017).
(3) Validation of the proposed framework through a case study in the steal plate manufacturing process to demonstrate practicality and scalability.

The rest of the paper organized as follows: Section 2 describes the proposed methodology, emphasizing the integration of AutoML and failure causality. Section 3 provides details on the case study and experimental setup. Section 4 presents the results, followed by a comprehensive analysis and discussion. Finally, Section 5 concludes the paper, highlighting key findings and proposing directions for future research.

2. Methodology

AutoML provides reasonable results in time by automating the ML workflow, including tasks such as data preprocessing, feature selection, model selection, and hyperparameter optimization. Figure 1 provides an overview of the AutoML process. Unlike traditional ML, AutoML encompasses a broader scope, integrating and automating five critical steps typically performed in traditional ML workflows.

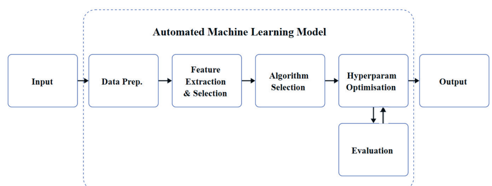


Fig. 1.: Overview of an automated ML process.

This automation significantly reduces the time and expertise required to develop effective ML models, making it accessible even to users without in-depth knowledge of ML techniques. This approach not only speeds up the model development process but also ensures that the resulting models are robust, reliable, and optimized for specific data characteristics, making it a powerful tool for industrial applications where time and accuracy are critical.

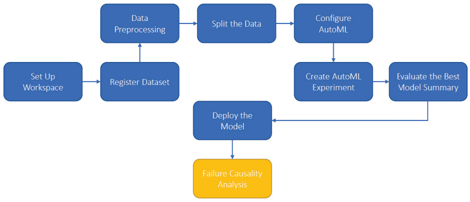


Fig. 2.: An efficient pipeline for processing data sets through AutoML.

The following steps are discussed to develop an efficient pipeline for processing data sets through an AutoML pipeline in Figure 2.

Set Up Workspace: A dedicated workspace is configured to support the development and execution of the AutoML job. This workspace is assigned to a specific resource group to ensure streamlined management and organization of associated resources.

Register Dataset: The data set is uploaded to facilitate advanced analytics and ML workflows.

Data Preprocessing: Data preprocessing involved a comprehensive selection of characteristics to ensure that the data set was optimally prepared for training.

Split the Data: Data set is divided into training and testing portions to evaluate the performance of the models.

Configure AutoML: A variety of classification methods are applied during the model selection phase to ensure robust predictive performance and adaptability to the data set.

AutoML Experiment: To run the AutoML experiments, the task settings are configured to align with the specific requirements of the data set and ML problem.

Evaluate Models: Various evaluation metrics are used to evaluate the performance of the model during experiments.

Deploy the Model: To consume the model, it must be deployed to be integrated with a service that enables applications to make real-time predictions for individual data points or small batches of data.

3. Case Study

In this paper, we use Steel Plates Faults as the case study to evaluate the performance of AutoML for fault diagnostic problems and the Microsoft Azure cloud platform is applied to streamline the process of model development, hyperparameter tuning, and evaluation.

3.1. *Steel Plates Faults Data Description*

The well-known Steel Plates Faults data set, proposed by Buscema and Tastle (2010), is considered. The data set, obtained from research undertaken by the Semeion Research Center of Sciences of Communication, consists of a detailed classification job focused on detecting surface imperfections in stainless steel plates. The data set comprises 1,941 instances, each characterized by 27 features that describe various aspects of steel plates, such as geometric measurements, luminosity indices, and material properties. These features are instrumental in identifying and classifying surface defects, which are categorized into six major distinct types and other faults, Figure 3:

- **Bumps:** Often appearing as little raised patches or lumps, these are imperfections on the surface of the steel plate that protrude outward. Smoothness and general quality of the plate surface might be impacted by bumps.
- **K_Scratch**: Describes scoring marks or scratches on the steel plate's surface that take the form of the letter "K." The integrity and beauty of the plate may be jeopardized by these variations in depth and severity.
- **Z_Scratch**: These are scuffs or abrasions on the surface of the steel plate that have the shape of the letter "Z." These scratches, which affect the surface polish and may cause structural flaws, can range in size and depth like K_Scratch.
- **Pastry:** Generally speaking, pastry flaws are surface defects on the steel plate that mimic patterns seen in baked products or pastries. Often the result of manufacturing oddities or flaws, these patterns may include swirls, loops, or other unique designs.
- **Stains:** Stained spots or marks on the surface

of the steel plate caused by different impurities or chemical reactions during handling or manufacturing procedures are known as stains. Small stains to more obvious imperfections can all detract from the plate's appearance and maybe its usefulness.

- **Dirtiness:** Defects in dirtiness are those in which there are foreign particles or debris on the steel plate's surface. These particles, which lessen the hygienic and high-quality surface of the plate, can be dust, oil residue or other pollutants.

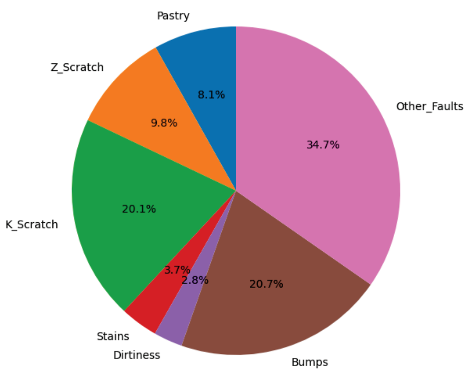


Fig. 3.: Different fault types.

The Other_Faults category encompasses defects not specified in the aforementioned types. These detailed categorizations facilitate the development and evaluation of ML models aimed at automating fault detection and classification in steel manufacturing processes.

Numerous researchers worked on this data set for fault detection and classification, evaluating ML algorithms for imbalanced data sets, and feature selection and extraction tasks, such as Lourenço et al. (1996); Buscema et al. (2010); Farahmand-Tabar and Rashid (2024); Dorbane et al. (2024). This paper attempts to predict these different causes of failure using AutoML.

3.2. *Experimental Setup*

For the AutoML pipeline, a compute cluster was provisioned with the Standard_D4s_v3 configuration, featuring 4 cores, 16 GB of RAM, 32 GB of storage, and a cost-effective pricing of \$0.24 per hour through Microsoft Azure Machine Learning (Azure ML). The Steel Plates Faults data set was uploaded as a data asset in Azure ML to facilitate advanced analytics and ML workflows. In Azure, data preprocessing includes identifying and performing key actions such as cleaning missing values, detecting and addressing data imbalances, and applying normalization to manage features with varying scales. We employed Monte Carlo Cross-Validation (MCCV) with a 5-fold approach and an 80/20 train-test split. In this approach, 20% of the data was randomly allocated to the test set, while the remaining 80% was used for training the models. The model training space includes a variety of classification methods, such as Logistic Regression, Stochastic Gradient Descent (SGD), Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Decision Tree, Random Forest, Extreme Random Forest, LightGBM, Gradient Boosting, XGBoost, and Linear SVM Classifiers. By systematically comparing these techniques, the selection process aimed to identify the most effective classifier that balances accuracy, computational efficiency, and generalization capabilities, ultimately optimizing the fault detection pipeline for steel plate manufacturing. To run the AutoML experiments, we configured the task settings to align with the specific requirements of the data set and classification problem. The task type was set to classification, focusing exclusively on a subset of the data, with the target column specified to predict the presence or absence of fault categories. These settings provided a structured and efficient framework for conducting the AutoML experiments, enabling the identification of the best-performing classification models within the specified parameters. In AutoML experiments for classification tasks, various evaluation metrics are utilized to assess model performance, including Accuracy, Precision, Recall, F1-Score, AUC-ROC, Log Loss, Matthews Correlation Coefficient (MCC), and Balanced Accuracy. Among these, one is designated as the primary metric, which is used to rank and sort models after training. The selection of the primary metric depends on the specific objectives of the task, such as F1-Score for imbalanced data sets or Accuracy for balanced ones, ensuring that the best-performing model aligns with the desired performance criteria. After identifying the best model based on the selected primary metric, the model can be deployed for practical use, such as deploying it as an endpoint to be integrated into a production environment. Furthermore, the deployed model can be updated on a predefined schedule or based on specific triggers, such as system maintenance or changes in data patterns.

4. Results and Discution

4.1. *Traditional ML Fault Detection*

Dorbane et al. (2024) used five powerful machine learning models, including Random Forest (RF), AdaBoost, Decision Tree, Support Vector Machines (SVM), and Naive Bayes to detect the failure causes. The ensemble models (i.e., RF and AdaBoost) harness the collective power of multiple weak learners to enhance discrimination capacity. Table 1 shows that Random Forest achieved the highest AUC of 0.942, with an accuracy of 0.771 and a balanced F1 score, compared to the other models. This comprehensive investigation improves fault detection efficacy and promotes informed decision-making in steel plate manufacturing processes.

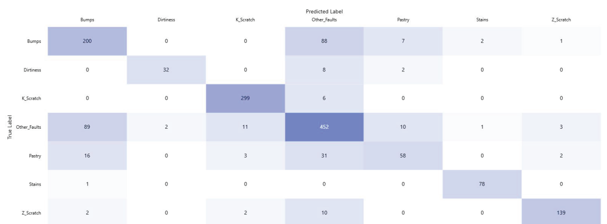
4.2. *AutoML: Multi-Class Single-Output classification*

A diverse range of algorithms was considered for model selection, each accompanied by a variety of hyperparameters to optimize. Additionally, various feature engineering techniques, such as normalization, were explored to enhance the modeling process. For example, the model selection includes a variety of classification methods, such as Logistic Regression, Stochastic Gradient Descent (SGD), Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Decision Tree, Random Forest, Extreme Random Forest, LightGBM, Gradient Boosting, XGBoost, and Linear SVM. Given the expansive search space of algorithms and hyperparameter configurations, along with feature engineering strategies, our experiments involved designing and evaluating 1003 unique combinations. Each combination was carefully tuned, and the most effective configuration was ultimately selected as the best model for the given dataset, ensuring optimal performance and generalization.

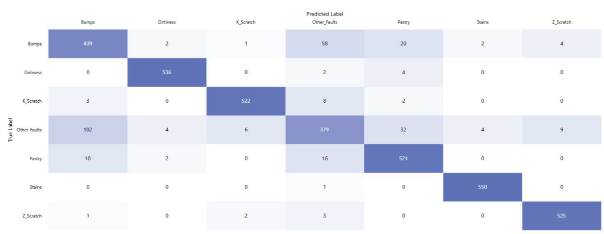
Following a comprehensive exploratory data analysis (EDA), we observed that the steel plate dataset contains seven output or class labels, each representing a binary classification. To streamline the classification task, we combined all class labels into a single column, representing the final target variable, where each sample corresponds to a specific fault type. We initially ran AutoML on the dataset, and the results for both training and testing are presented in Figures 4 and 5. The model achieved a performance of 0.80900 on the MCSO training and 0.78920 on testing data, which was lower than expected due to a significant class imbalance in the dataset. This issue is further highlighted in the confusion matrix for the testing data in Figure 4 and 5, where certain fault types were underrepresented in the predictions. To address this imbalance, we applied data augmentation techniques, including the Synthetic Minority Oversampling Technique (SMOTE) and Random Oversampling (ROS), to create a more balanced dataset. The AutoML classification pipeline was then rerun with the same configuration. As shown in Figures 4 and 5, the performance of the model improved significantly in the balanced data set, demonstrating the effectiveness of these techniques in mitigating the impact of class imbalance. For MCSO, the best model is a voting ensemble that consists of 10 XGBoost classifiers, each wrapped with a standard scaler to ensure consistent feature scaling. In the case of SMOTE, the ensemble includes 9 XGBoost classifiers, along with a LightGBM classifier that uses a MaxAbsScaler to handle data with varying magnitudes. For ROS, the ensemble incorporates 8 XGBoost classifiers, a random forest classifier, and a LightGBM classifier, with the XGBoost and random forest models using a standard scaler, while the LightGBM classifier utilizes the MaxAbsScaler.

Table 1.: Traditional ML results.

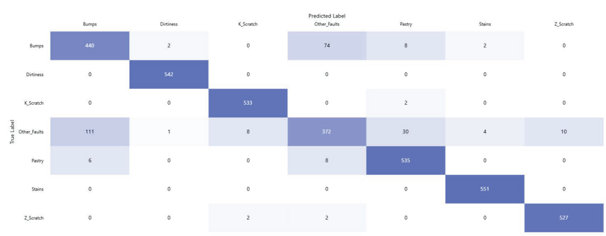
Model	AUC	Accuracy	F1	Precision	Recall	MCC
RF	0.942	0.771	0.771	0.776	0.771	0.705
AdaBoost	0.829	0.735	0.734	0.737	0.735	0.661
Decision Tree	0.823	0.706	0.708	0.713	0.706	0.630
SVM	0.908	0.673	0.670	0.671	0.673	0.580
Naive Bayes	0.890	0.626	0.625	0.672	0.626	0.556



(a) Confusion Matrix - Training-MCSO



(b) Confusion Matrix - Training-SMOTE

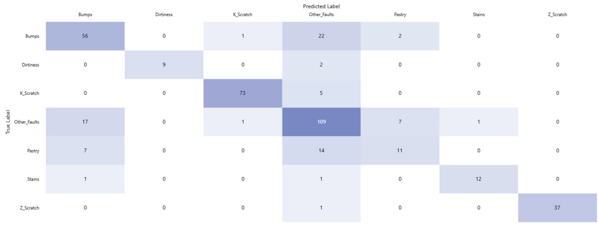


(c) Confusion Matrix - Training-ROS

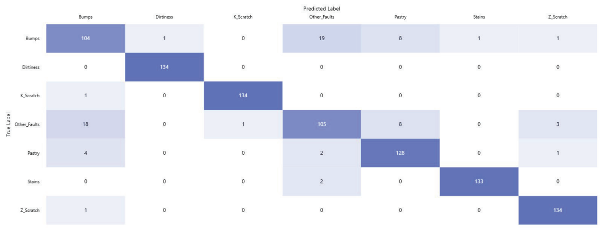
Fig. 4.: Confusion Matrices for Training MCSO and Different Sampling Methods: SMOTE and ROS

5. Conclusion

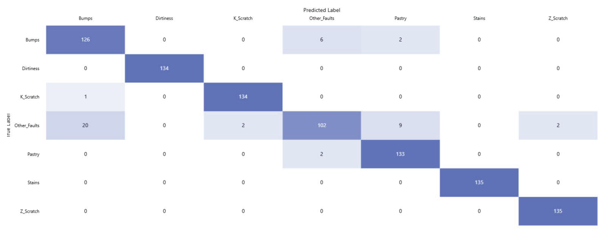
In conclusion, an AutoML approach was successfully applied for fault detection and failure diagnosis in the steel plate manufacturing process. The results from various experiments demonstrate that AutoML outperforms traditional machine learning (ML) techniques, primarily due to its capabilities in automated feature engineering, hyperparameter tuning, and model selection while the running time for AutoML is longer compared to conventional methods. Future work could explore the integration of deep learning techniques, building upon the promising results achieved thus far. Additionally, incorporating maintenance strategies within the AutoML workflow for predictive maintenance could help automate the entire pro-



(a) Confusion Matrix - Testing-MCSO



(b) Confusion Matrix - Testing-SMOTE



(c) Confusion Matrix - Testing-ROS

Fig. 5.: Confusion Matrices for Testing MCSO and Different Sampling Methods: SMOTE and ROS

cess, offering a more comprehensive and efficient solution for fault detection and maintenance planning in the steel plate industry.

References

Askari, B., G. Cavone, R. Carli, A. Grall, and M. Dotoli (2023). A semi-supervised learning approach for fault detection and diagnosis in complex mechanical systems. *IEEE Transactions on Automation Science and Engineering*, 1–6.

Askari, B., Y. Langeron, G. Cavone, R. Carli, A. Grall, and M. Dotoli (2023). An integrated approach for failure diagnosis and analysis of industrial systems based on multi-class multi-output classification: A complex hydraulic application. In M. P. Brito, T. Aven, P. Baraldi, M. Čepin, and E. Zio (Eds.), *Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)*, Singapore. Research Publishing.

Baratchi, M., C. Wang, S. Limmer, J. N. van Rijn, H. Hoos, T. Bäck, and M. Olhofer (2024). Automated machine learning: past, present and future. *Artificial Intelligence Review 57*(5), 1–88.

Buscema, M., S. Terzi, and W. Tastle (2010). A new meta-classifier. In *2010 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), Toronto, ON, Canada*, pp. 1–7.

Buscema, M, T. S. and W. Tastle (2010). Steel Plates Faults. DOI: https://doi.org/10.24432/C5J88N.

Dorbane, A., F. Harrou, and Y. Sun (2024). Detecting faulty steel plates using machine learning. In *International Conference on Advances in Computing and Data Sciences*, pp. 321–333. Springer.

Farahmand-Tabar, S. and T. A. Rashid (2024). Steel plate fault detection using the fitness-dependent optimizer and neural networks. In *Handbook of Formal Optimization*, pp. 1–19. Springer.

Feurer, M., A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter (2015). Efficient and robust automated machine learning. *Advances in neural information processing systems 28*.

Friederich, J. and S. Lazarova-Molnar (2024). Reliability assessment of manufacturing systems: A comprehensive overview, challenges and opportunities. *Journal of Manufacturing Systems 72*, 38–58.

Gkioka, G., M. Dominguez, and G. Mentzas (2024). An automl-based approach for automatic traffic incident detection in smart cities. *Intelligent Decision Technologies 18*(2), 1101–1122.

Hutter, F., L. Kotthoff, and J. Vanschoren (2019). *Automated machine learning: methods, systems, challenges*. Springer Nature.

Lourenço, P. M., M. J. Abreu, and J. M. Ribeiro (1996). Fault diagnosis in steel plates using ultrasonic measurements. In *Proceedings of the European Congress on Ultrasonics, Lisbon, Portugal.*

Misaii, H., M. Fouladirad, B. Askari, and A. Durupt (2024). Predictive degradation modelling using artificial intelligence: Milling machine case study. In *Advances in Reliability, Safety and Security*, Gdynia, Poland. European Safety and Reliability (ESREL).

Peters, J., D. Janzing, and B. Schölkopf (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Salehin, I., M. S. Islam, P. Saha, S. Noman, A. Tuni, M. M. Hasan, and M. A. Baten (2024). Automl: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence 2*(1), 52–81.

Seligmann, B. J., J. Zhao, S. G. Marmara, T. C. Corbett, M. Small, M. Hassall, and J. T. Boadle (2019). Comparing capability of scenario hazard identification methods by the pic (plant-people-procedure interaction contribution) network metric. *Safety science 112*, 116–129.

Smith, D., B. Veitch, F. Khan, and R. Taylor (2017). Understanding industrial safety: Comparing fault tree, bayesian network, and fram approaches. *Journal of Loss Prevention in the Process Industries 45*, 88–101.

Thornton, C., F. Hutter, H. H. Hoos, and K. Leyton-Brown (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 847–855.

Yu, W., T. Dillon, F. Mostafa, W. Rahayu, and Y. Liu (2019). A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics 16*(1), 183–192.

Zöller, M.-A. and M. F. Huber (2021). Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research 70*, 409–472.