An Automated Machine Learning Framework for Predictive Analytics in Quality Control

Mattheos Fikardos[1](✉), Katerina Lepenioti[1], Alexandros Bousdekis[1], Enrica Bosani[2], Dimitris Apostolou[1], and Gregoris Mentzas[1]

[1] Information Management Unit (IMU), Institute of Communication and Computer Systems (ICCS), National Technical University of Athens (NTUA), Athens, Greece
{mfikardos,klepenioti,albous,dapost,gmentzas}@mail.ntua.gr

[2] Whirlpool EMEA, Benton Harbor, USA
enrica_bosani@whirlpool.com

Abstract. Developments in Machine Learning (ML) in the last years resulted in taking as granted their usage and their necessity clear in areas such as manufacturing and quality control. Such areas include case specific requirements and restrictions that require the human expert's knowledge and effort to apply the ML algorithms efficiently. This paper proposes a framework architecture that utilizes Automated Machine Learning (AutoML) to minimize human intervention while constructing and maintaining ML models for quality control. The data analyst gives the setting for multiple configurations while designing predictive quality models which are automatically optimized and maintained. Moreover, experiments are conducted to test the framework in both the performance of the prediction models and the time needed to construct the models.

Keywords: Machine learning · Automated machine learning · Quality control · Predictive quality

1 Introduction

Quality of products and processes have increasingly concerned the manufacturing firms because negative consequences do not show up until the product is actually produced or worse, until the customer returns it [1, 2]. Predictive quality moves beyond traditional quality evaluation methods towards extracting useful insights from various data sources with the use of Machine Learning (ML) in an Industry 4.0 context [1]. Even though well-known methodologies like the Cross-Industry Standard Process for Data Mining (CRISP-DM) [3] can be applied, their generic approach does not consider domain-specific requirements in manufacturing quality procedures [4]. This limitation requires the data analyst and the production expert to work alongside the AutoML pipeline.

In this paper, we propose the use of AutoML in methodologies similar to CRISP-DM that can facilitate their implementation in a predictive quality context. Despite the fact that extensive research for ML in manufacturing has already been conducted [5,

D. Y. Kim et al. (Eds.): APMS 2022, IFIP AICT 663, pp. 19–26, 2022.
https://doi.org/10.1007/978-3-031-16407-1_3

6], highlighting advantages, challenges and applications, research on AutoML in the manufacturing quality function is still in preliminary stages [4, 7–9]. However, AutoML has the potential to reduce time-consuming tasks of constructing ML models for quality procedures, allowing the data analyst to devote more time on data integration and deployment. In this way, the human intervention in ML model configuration is minimized since the algorithms are automatically updated and optimized based on new data.

The rest of the paper is organized as follows. Section 2 outlines the theoretical background on AutoML. Section 3 presents our proposed approach for AutoML in predictive quality. Section 4 describes the implementation of the proposed approach in a real-life scenario of white goods production. Section 5 concludes the paper and presents our plans for future work.

2 Theoretical Background on Automated Machine Learning

AutoML aims to simplify and automate the whole ML pipeline, giving the opportunity to domain experts to utilize ML without deep knowledge about the technologies and the need of a data analyst [10]. The most fundamental concept of AutoML is the Hyper Parameter Optimization (HPO) problem where hyperparameters are automatically tunned for ML systems to optimize their performance [10] for problems such as classification, regression and time series forecasting. As of today, further developments to the field of AutoML added additional capabilities to the AutoML pipeline: *Data Preparation, Feature Engineering, Model Generation* and *Model Evaluation* [11].

The *Data Preparation* and *Feature Engineering* steps are associated with the available data used for the ML algorithms. The former includes actions for collecting, cleaning and augmenting the data, with the latter includes actions for extracting, selecting and constructing features. In the *Model Generation* step, a search is executed with the goal of finding the best performing model for the predictions, such as k-nearest neighbors (KNN) [12], Support Vector Machines (SVM) [13], Neural Networks (NN), etc. The *Model Evaluation* step is responsible for evaluating the generated models based on predefined metrics and runs in parallel to the Model Generation step. The evaluation of the generated models is used for optimization of existing models and the construction of new models. The search procedure of AutoML terminates based on predefined restrictions, such as the performance of the models or the time passed.

From a technical perspective, AutoML attracted a lot of research interest resulting in several AutoML frameworks, such as: Autokeras [14], FEDOT [15] and TPOT [16]. Additionally, research focusing on benchmarking several AutoML frameworks [17, 18] concludes that they do not outperform humans yet but give promising results.

3 The Proposed Approach for Automated Machine Learning in Predictive Quality

The proposed approach focuses on the development of dynamic ML algorithms using AutoML to minimize human intervention in model configuration. The proposed approach is divided into two phases: the ***Design phase*** and the ***Runtime phase***, as depicted in Fig. 1. In a nutshell the *Data analyst* based on technical and case specific knowledge designs the ML models used for the predictions which are then used by the *Quality Expert* for predictions and are automatically updated when new data are available for training. Compared with the traditional process of creating and maintaining ML models for quality control, the *Data Analyst* would spend valuable time constructing models. Even though the data preprocessing algorithms may exist, the HPO and fine tuning of the models would have been performed via trial and error from the *Data Analyst*.

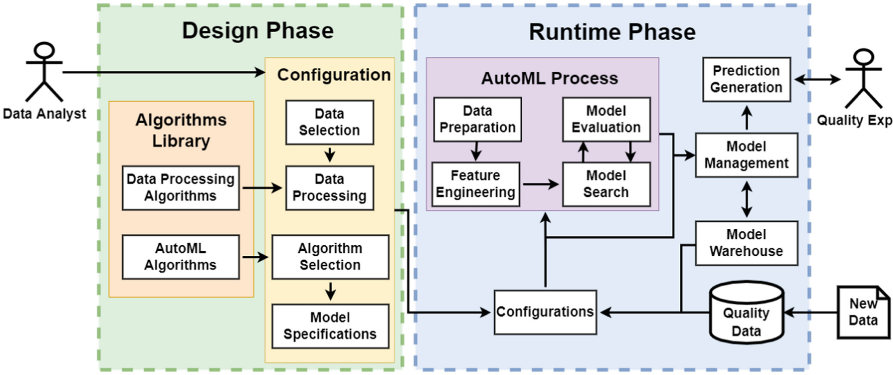

Fig. 1. The architecture of the proposed approach.

3.1 Design Phase

The Design phase is executed by the data analyst, who is responsible for defining the necessary configurations that bootstrap the analysis solving the predictive quality problem under examination by employing the available *AutoML algorithms*. The Design phase consists of two components, *Configuration* and *Algorithms Library*. During the *Configuration*, the data analyst must first select which of the available quality data are required for the predictions to be carried out. After the *Data Selection*, if necessary, the data analyst can apply *Data Processing Algorithms* from the *Algorithms Library*, which may include data cleaning, data augmentation, feature extraction and feature selection.

Regarding the ML algorithms, the data analyst specifies the AutoML algorithm that will search for the best predictive model, also found in the *Algorithms Library*. For the *Algorithm Selection*, the data analyst can define the configuration of the selected AutoML algorithm, e.g., construction parameters for the model, metrics for evaluation, and termination conditions. With the *Model Specifications*, additional case specific configurations can be made, such as model acceptance conditions and output formats, that will be used by the *Model Management* process during the Runtime phase.

3.2 Runtime Phase

The Runtime phase is responsible for executing the *AutoML process* and the *Model Management* of the constructed models. It can start either when the data analyst creates new configurations or when new data become available for existing models.

In the first case, after the data analyst completes the configuration for the predictive quality problem, the *Configurations* are stored for later use and the *AutoML process starts* searching for a model. During that process, additional data processing actions may be executed from the AutoML algorithm at the Data Preparation and Feature Engineering steps. After the input data transformations have been completed, the algorithm starts the search by constructing several models followed by the evaluation and optimization of the candidate ones. When the search step finishes, a single model is selected as the model with the best performance. The selected model is passed to the *Model Management* process, where it will be stored in the *Model Warehouse* or discarded based on the acceptance conditions configured in the Design phase.

In the other case, models already used for predictions are automatically retrained or optimized and changed based on new data that are available without any human intervention. As soon as the new data become available, the related models are retrieved from the *Model Warehouse* and are automatically forwarded to repeat the aforementioned *AutoML Process*. Before the *AutoML process* starts using the stored Configurations, the data selection and data processing actions are executed, feeding the *AutoML process* with all the available data in the correct format. As with the previous case, after the AutoML process is finished, the new model is passed to the *Model Management* process where it will be compared with the existing model. If the new model performs better and fulfills the acceptance conditions, it replaces the existing model, otherwise it gets discarded.

In both cases the *Model Management* process can retrieve the corresponding model for a prediction and pass the model to the *Prediction Generation* process to execute predictions. The generated predictions are then communicated to the quality expert, in order to support the predictive quality-related decisions.

4 Application to a White Goods Production Use Case

4.1 Use Case Description

In the Whirlpool production model, the whole white goods production is tested from quality and safety point of view in order to ensure a high standard level of product quality to final customers. The use case under examination deals with the microwave production line. At the end of the production line, random inspections are made from employees to detect defective products, which are subsequently repaired or replaced. During the quality control, several features of the products and the tests are recorded, including their Defect Groups which are used as categories for similar Defects. In this scenario, we opted to predict the Defect Group of the defective products and the number of orders found with defects for the following days. The former is a classification problem while the latter is a Regression/Timeseries forecasting problem.

4.2 Dataset

Based on the available data sources, we constructed a Quality Control Data Model as shown in Fig. 2. The main entity in the Quality Control Data Model is the Defect Instance which maps all the entries from the data. Common attributes are used as reference fields for other entities such as the Product, the Part and the Defect Type, which provide further information about the Defect Instance. This Data Model gave us the ability to better manage the available data and retrieve additional information if needed. The experiments were performed based on a limited amount of data. The dataset consisted of 25655 entries during a span of 270 Days and included a total of 38 features from which we extracted each entity as a Defect Instance.

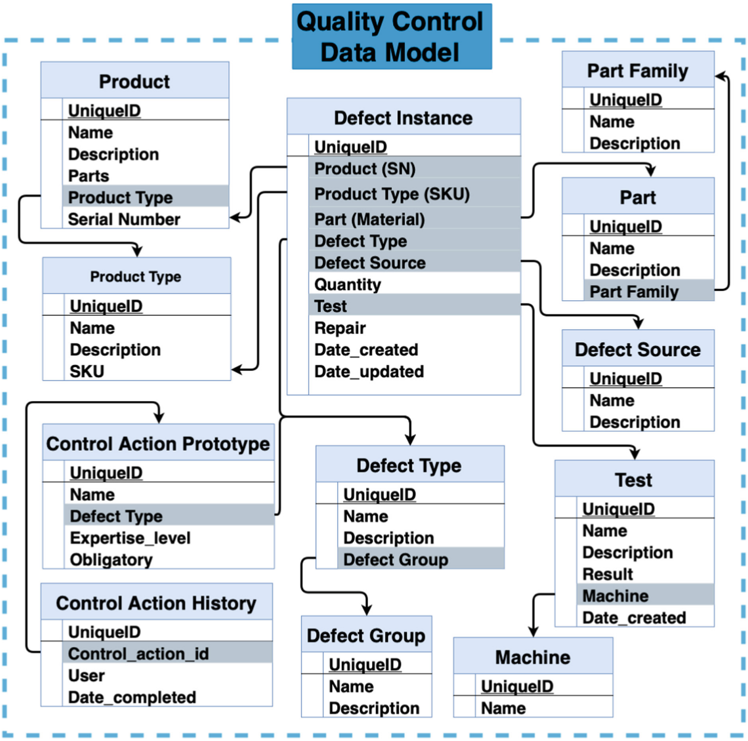

Fig. 2. Quality control data model

4.3 Results

The proposed approach was implemented with the Python libraries AutoKeras and Fedot. Specifically, the Structured Data Regressor and the Structured Data Classifier are used from the AutoKeras library, in order to find the best performing Neural Network (NN) for the predictions, and the Fedot library in order to compose a chain of data-preprocessing and ML models. In our experiments, 3 models were implemented for each algorithm as follows: 1) an *initial model* trained only with the 80% of the available data, 2) a *retrained model* which was the initial model retrained with all the available data and 3) a *new model* that started the AutoML process from scratch with all the available data. The *retrained* and the *new* model were automatically trained following the proposed approach with the configurations made by the *initial model*. Additionally, regarding the execution time of the experiments that follow the values are based on configured stopping conditions, which can be changed resulting in different values.

Predict Defect Group. Starting with the configuration, data processing algorithms were used to select 6 features of the Defect Instances: The Date Created, the Product Type (SKU), the Defect Source, the Station ID and the Part Family. From the first one, the Date Created, additional features were extracted by splitting up the timestamp into the Year, Month, Hour and Minute of the recorded defect. Two models were constructed for this classification problem, which use Fedot and AutoKeras respectively. The models are evaluated with 4 metrics: F1-macro, F1-micro, Receiver Operating Characteristic Area Under Curve (ROC-AUC) and the execution time of the AutoML algorithms. We also compared them with a manually constructed Decision Tree (DT) classifier, which had performed significantly better than other classifiers tested. These results are presented in Table 1.

Table 1. Evaluation metrics for the defect group prediction

	Models	Metrics			
		F1-macro	F1-micro	ROC-AUC	Execution time (s)
Initial models	Decision tree	**0.5347**	0.8331	0.8466	–
	FEDOT	0.5055	**0.8363**	**0.9023**	1212.50
	AutoKeras	0.4509	0.7813	0.7141	1019.81
Retrained models	Decision tree	**0.6052**	**0.8525**	**0.9041**	–
	FEDOT	0.4969	0.8368	0.9019	9.35
	AutoKeras	0.4510	0.7813	0.7141	0.58
New models	Decision tree	**0.5170**	**0.8403**	0.8514	–
	FEDOT	0.4909	0.8277	**0.8722**	91.96
	AutoKeras	0.4631	0.7681	0.7177	953.36

Even though the DT classifier outperformed the other models in almost all cases, the AutoML proposed models with an acceptable performance, while the FEDOT model has a similar performance with the DT. The models trained with all the available data performed slightly better than the initial ones and the execution time for AutoML algorithms is also acceptable, especially in the case of the *retrained models*. Finally, it is important to note the significance of choosing the evaluation metric for the model acceptance conditions, since this may affect the selection of the model.

Predict Defective orders. With data preprocessing, the Defect Instances were summed based on the attribute Date Created to produce the needed timeseries. Two models were constructed by selecting two AutoML algorithms: the Fedot by configuring the problem as timeseries, and the TimeSeriesForecaster of the AutoKeras. The performance of these models is evaluated with the Mean Square Error (MSE), the Mean Absolute Error (MAE) and the execution time of the AutoML algorithms, as shown in Table 2.

From the evaluation metrics we observe that both AutoML algorithms have performed well. In the case of the *initial models*, the metrics values are worse due to the inadequate data for training. As in the previous results the execution time follows the same pattern and all the models were proposed in a reasonable amount of time.

Table 2. Table captions should be placed above the tables.

	Models	Metrics		
		MSE	MAE	Execution time (s)
Initial models	FEDOT	0.2338	0.2017	101.88
	AutoKeras	**0.0402**	**0.1624**	187.39
Retrained models	FEDOT	0.1361	**0.1002**	0.32
	AutoKeras	**0.0201**	0.1082	2.57
New models	FEDOT	0.1391	**0.0988**	127.48
	AutoKeras	**0.0191**	0.1036	11.17

5 Conclusions and Future Work

In this paper we proposed a framework for predictive quality using AutoML algorithms, where the human supervision is decreased as existing prediction models are automatically optimized based on new data. By reducing the effort needed to construct and maintain prediction models, the data analyst can devote more time to inspect and understand case-specific requirements. From the experimental results we concluded that by leveraging AutoML algorithms, good performing models can be acquired and automatically optimized in a reasonable amount of time.

The growing interest in the AutoML field in the last few years provides a promising future for its development and applications. As the automated steps of AutoML are improved, their ability to adapt or incorporate case specific requirements or restrictions paves the way for its extensive application to predictive quality. In our future work, we plan to examine in depth various configurations in the proposed framework, and test more AutoML algorithms in the predictive quality context.

Acknowledgements. This work is partly funded by the European Union's Horizon 2020 project COALA "COgnitive Assisted agile manufacturing for a LAbor force supported by trustworthy Artificial Intelligence" (Grant agreement No 957296). The work presented here reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

References

1. Zonnenshain, A., Kenett, R.S.: Quality 4.0—the challenging future of quality engineering. Qual. Eng. **32**(4), 614–626 (2020)
2. Bousdekis, A., Wellsandt, S., Bosani, E., Lepenioti, K., Apostolou, D., Hribernik, K., Mentzas, G.: Human-AI collaboration in quality control with augmented manufacturing analytics. In: Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., Romero, D. (eds.) Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part IV, pp. 303–310. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-85910-7_32
3. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T.P., Shearer, C., Wirth, R.: CRISP-DM 1.0: Step-by-step data mining guide (2000)
4. Krauß, J., Pacheco, B.M., Zang, H.M., Schmitt, R.H.: Automated machine learning for predictive quality in production. Procedia CIRP **93**, 443–448 (2020)
5. Wuest, T., Weimer, D., Irgens, C., Thoben, K.D.: Machine learning in manufacturing: advantages, challenges, and applications. Prod. Manuf. Res. **4**(1), 23–45 (2016)
6. Dogan, A., Birant, D.: Machine learning and data mining in manufacturing. Expert Syst. Appl. **166**, 114060 (2021)
7. Ferreira, L., Pilastri, A., Sousa, Vítor., Romano, F., Cortez, P.: Prediction of maintenance equipment failures using automated machine learning. In: Yin, H., et al. (eds.) Intelligent Data Engineering and Automated Learning – IDEAL 2021: 22nd International Conference, IDEAL 2021, Manchester, UK, November 25–27, 2021, Proceedings, pp. 259–267. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-91608-4_26
8. Gerling, A., Ziekow, H., Hess, A., Schreier, U., Seiffer, C., Abdeslam, D.O.: Comparison of algorithms for error prediction in manufacturing with automl and a cost-based metric. J. Intell. Manuf. **33**(2), 555–573 (2022). https://doi.org/10.1007/s10845-021-01890-0
9. Ribeiro, R., Pilastri, A., Moura, C., Rodrigues, F., Rocha, R., Cortez, P.: Predicting the tear strength of woven fabrics via automated machine learning: an application of the CRISP-DM methodology (2020)
10. Hutter, F., Kotthoff, L., Vanschoren, J. (eds.): Automated Machine Learning: Methods, Systems, Challenges. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5
11. He, X., Zhao, K., Chu, X.: AutoML: a survey of the state-of-the-art. Knowl.-Based Syst. **212**, 106622 (2021)
12. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
13. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
14. Jin, H., Song, Q., Hu, X.: Auto-keras: an efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1946–1956 (2019)
15. Nikitin, N.O., et al.: Automated evolutionary approach for the design of composite machine learning pipelines. Future Gener. Comput. Syst. **127**, 109–125 (2022)
16. Olson, R.S., Moore, J.H.: TPOT: a tree-based pipeline optimization tool for automating machine learning. In: Workshop on Automatic Machine Learning, pp. 66–74. PMLR (2016)
17. Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., Vanschoren, J.: An open source AutoML benchmark. arXiv preprint arXiv:1907.00909 (2019)
18. Zöller, M.A., Huber, M.F.: Benchmark and survey of automated machine learning frameworks. J. Artif. Intell. Res. **70**, 409–472 (2021)