Explainable AI in Manufacturing:
A Predictive Maintenance Case Study

Bahrudin Hrnjica[2][0000−0002−3142−1284] and Selver Softic[1][0000−0003−2949−8239]

[1] University of Bihac, Bosnia and Herzegovina
`bahrudin.hrnjica@unbi.ba`
[2] CAMPUS 02 University of Applied Sciences, Austria
`selver.softic@campus02.at`

Abstract. This paper describes an example of an explainable AI (Artificial Intelligence) (XAI) in a form of Predictive Maintenance (PdM) scenario for manufacturing. Predictive maintenance has the potential of saving a lot of money by reducing and predicting machine breakdown. In this case study we work with generalized data to show how this scenario could look like with real production data. For this purpose, we created and evaluated a machine learning model based on a highly efficient gradient boosting decision tree in order to predict machine errors or tool failures. Although the case study is strictly experimental, we can conclude that explainable AI in form of focused analytic and reliable prediction model can reasonably contribute to prediction of maintenance tasks.

Keywords: Explainable AI · Predictive Maintenance · Production Management.

1   Introduction

Predictive Maintenance (PdM) anticipates maintenance needs to avoid costs associated with unscheduled downtime. By connecting to devices and monitoring the data the devices produce, we can identify patterns that lead to potential problems or failures. Those insights can be used to address issues before they happen. This ability to predict when equipment or assets need maintenance allows us to optimize equipment lifetime and minimize downtime [1].

The fundamental litmus test for explainable AI (XAI) – that is, machine learning algorithms and other Artificial Intelligence systems that produce outcomes that humans can readily understand and track backwards to the origins [2].

In this case study we will consider the field of maintenance in manufacturing. More precisely we will deal with PdM by involving explainable AI outputs as base for our decisions and predictions.

2   Explainable AI - XAI

Recent success of Machine Learning (ML) led to series of application scenarios for Artificial Intelligence (AI) applications. Continued advances promise to produce autonomous systems that will perceive, learn, decide, and act on their own. However, the effectiveness of these systems is limited by the machine's current inability to explain their decisions and actions to human users.

The Explainable AI (XAI) program introduced by DARPA[3] aims to create a suite of ML techniques that:

- Produce more explainable models, while maintaining a high level of learning performance (prediction accuracy); and
- Enable human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners.

For decision makers who rely upon Data Analytics and Data Science, explainability is a real issue. If the computational system relies on a simple decision model such as logistic regression, they can understand it and convince executives who have to sign off on a system because it seems reasonable and fair. They can justify the analytical results to shareholders, regulators, and other involved stakeholders. But for "Deep Nets" and ML systems, this is no longer possible.

There is a need to find ways to explain the system to the decision maker so that they know that their decisions are going to be reasonable. The goals of explanation involves reaching a persuasion, but that comes only as a consequence of understanding the how the AI works, the mistakes the system can make, and the safety measures surrounding it.

Meanwhile, AI is increasingly allowed to make and take more autonomous decisions and actions. Justifying these decisions will only become more crucial, and there is little doubt that this field will continue to rise in prominence and produce exciting and much needed work in the future [3].

The importance of explanation, and especially explanation in AI, has been emphasized in numerous popular press outlets over the past decades, with considerable discussion of the explainability of "Deep Nets" and ML systems in both the technical literature and the recent popular press [2, 4–13].

3   Predictive Maintenance

PdM extracts insights from the data produced by the equipment on the shop floor and acts on these insights. The idea of PdM goes back to the early 1990's and augments regularly scheduled, preventive maintenance. PdM requires the equipment to provide data from sensors monitoring the equipment as well as other operational data. Humans act based on the analysis. Simply speaking, it is a technique to determine (predict) the failure of the machine component in the near future so that the component can be replaced based on the maintenance

---

[3] https://www.darpa.mil/program/explainable-artificial-intelligence

plan before it fails and stops the production process. The PdM can improve the production process and increase the productivity. By successfully handling with PdM we are able to achieve the following goals:

- Reduce the operational risk of mission-critical equipment.
- Control cost of maintenance by enabling just-in-time maintenance operations.
- Discover patterns connected to various maintenance problems.
- Provide Key Performance Indicators.

Usually PdM uses descriptive, statistical or probabilistic approach to drive analysis and prediction. There are also several approaches which used Machine Learning (ML) [1, 14]. Through the literature [15] there can be found the following types of PrM in the production: reactive, periodic, proactive and predictive (Figure 1).
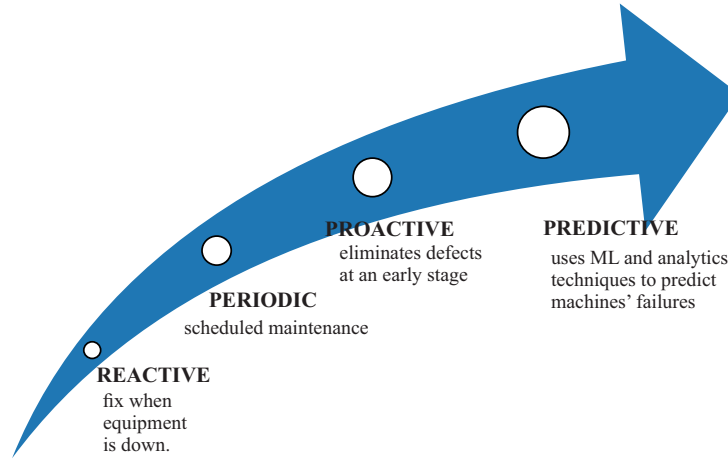

PROACTIVE
eliminates defects
at an early stage

PREDICTIVE
uses ML and analytics
techniques to predict
machines' failures

PERIODIC
scheduled maintenance

REACTIVE
fix when
equipment
is down.

Fig. 1. Different types of maintenance in the production

4   Case Study

In order to handle and use this technique we need a various data from the machines in production. In this case study we used the freely available data from a data source generated as test data set for PdM containing information about: telemetry, errors, failures and machine properties.

The data can be found at Azure blob storage. The data is maintained by Azure Gallery Article [4]. Once the data is downloaded from the blob storage, local copies will be used for further observations in this contribution.

---

[4] https://gallery.azure.ai/Experiment/Predictive-Maintenance-Implementation-Guide-Data-Sets-1

4.1 Methodology

Usually, every PdM technique should proceed by the following three main steps:

 – **Collect Data** – collect all possible descriptions, historical and real-time data, usually by using IoT (Internet of Things) devices, various loggers, technical documentation, etc.
 – **Predict Failures** – collected data can be used and transformed into ML ready data sets, and build a ML model to predict the failures of the components in the set of machines in the production.
 – **React** – by obtaining the information which components will fail in the near future, we can activate the process of replacement so the component will be replaced before it fails, and the production process will not.

4.2 Data Preparation

In order to predict failures in the production process, a set of data transformations, cleaning, feature engineering, and selection must be performed to prepare the data for building a ML model. The data preparation part plays a crucial role in the model building process because quality of the data and its preparation will directly influences the model accuracy and reliability. The data used for this PdM use case can be classified to:

 – TELEMETRY – which collects historical data about machine behavior (voltage, vibration, etc).
 – ERRORS – the data about warnings and errors in the machines.
 – MAINT – data about replacement and maintenance for the machines.
 – MACHINES – descriptive information about the machines.
 – FAILURES – data when a certain machine is stopped, due to component failure.

Errors data represents the most important information in every PdM system. The errors are non-breaking recorded events while the machine is still operational. In the experimental data set the error date and times are rounded to the closest hour since the telemetry data is collected at an hourly rate. What we get to insight is shown in the left chart of figure 2.

Failures data represents the replacements of the components due to the failure of the machines. Once the failure is happened the machine is stopped. This is a crucial difference between errors and failures. Failure distribution produced by certain component across machines is shown in the right chart of figure 2.

Maintenance data tells us about scheduled and unscheduled maintenance. The data set contains the records which correspond to both, regular inspection of components as well as failures. To add the record into the maintenance table a component must be replaced during the scheduled inspection or replaced due to a breakdown. In case the records are created due to breakdowns are called failures. Maintenance contains the data from 2014 and 2015 years.
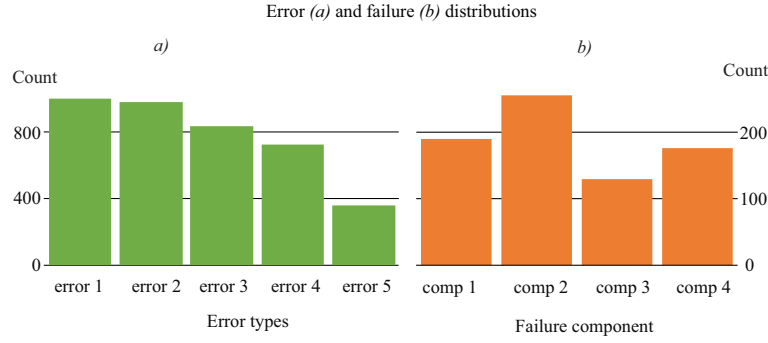
Error *(a)* and failure *(b)* distributions

Fig. 2. Error and failure distributions across machines

Machine data includes information about 100 machines which are subject of the PdM analysis. The information includes: model type, and machine age. Distribution of the machine age categorized by the models across production process is shown in the following figure 3.
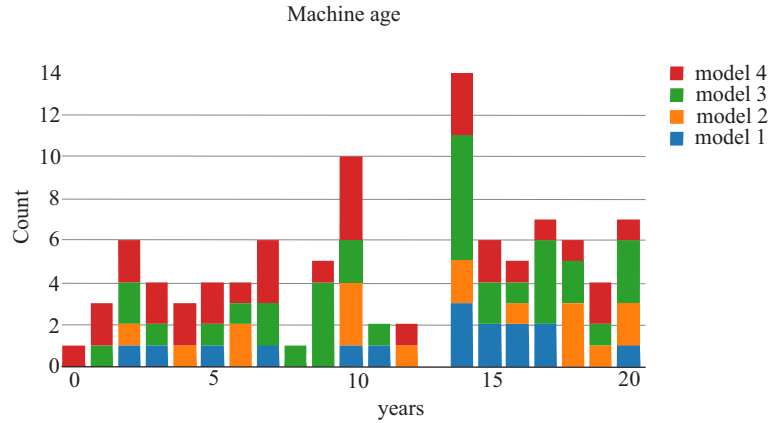
Machine age

Fig. 3. Machines overview grouped by different types of models and age.

4.3   Feature Engineering

First, several lagged telemetry data was created, since telemetry data are classic time series data. In the following, the rolling mean and standard deviation of the telemetry data over the last 3-hours lag window is calculated for every 3 hours. For capturing a longer term effect 24 hours lag features rolling average and standard deviation were calculated. Once we have rolling lag features calculated, we can merge them into one data frame. Now that we have basic data frame,

we merged previously calculated data frames with this one. At the end of the merging process, the relevant columns are selected.

Unlike telemetry that had numerical values, errors have categorical values denoting the type of error that occurred at a time-stamp. This was used to aggregate categories of the error with different types of errors that occurred in the lag window. The main task here was to create a relevant feature in order to create a quality data set for the machine learning part.

One of the good features that has been chosen was the number of replacements of each component in the last 3 months to incorporate the frequency of replacements. Furthermore, we calculated how long it has been since a component is last replaced as that would be expected to correlate better with component failures since the longer a component is used, the more degradation should be expected. The machine data set contains descriptive information about machines like the type of machines and their ages which is the years in service.

As the last step in feature engineering, we are performing merging all features into one data set. The label in PdM should be the probability that a machine will fail in the near future due to a failure certain component. If we take 24 hours to be the period (task) for this problem, the label construction consists of a new column in the feature data set which indicate if certain machine will fail or not in the next 24 hours due to failure one of several components.

In this way, we are defining the label as a categorical variable containing: – none – if the machine will not fail in the next 24 hours, – comp1 to comp4 if the machine will fail in the next 24 hours due to the failure of certain components. Since we can experiment with the label construction by applying different conditions, we can implement methods that take several arguments in order to define the general problem.

4.4   Preliminary Results

We analyzed 5 data sets with information about telemetry, data, errors and maintenance as well as failure for 100 machines. The data were transformed and analyzed in order to create the final data set for building a machine learning model for PdM.

Once we created all features from the data sets, as a final step is to create the label column so that it describes if a certain machine will fail in the next 24 hours due to failure a comp1, comp2, comp3, comp4 or it will continue to work. In this part, we performed a part of the ML task and start training a ML model for predicting if a certain machine will fail in the next 24 hours due to failure, or it will be in functioning normal in that time period.

The model which we built was multi-class classification model since it has 5 values to predict: comp1, comp2, comp3, comp4 or none – means it will continue to work. We used the DART Booster hyper-parameter tuning along with Light-GBM[5] [16] which is a gradient boosting framework that uses tree based learning

---

[5] https://lightgbm.readthedocs.io/en/latest/

algorithm. It is especially efficient on small data sets. We evaluated the trained model first with training data set (see Table 1).

PREDICTED TRUTH	none	comp1	comp2	comp3	comp4	Recall
none	165371	0	0	0	0	1,0000
comp1	0	844	26	4	8	0,9588
comp2	0	22	1097	8	31	0,9473
comp3	0	4	8	576	13	0,9548
comp4	0	16	25	11	772	0,9369
Precision	1,0000	0,9546	0,949	0,9616	0,9369	

Table 1. Results on training data set.

As can be seen the model predicts the values from the training data set correctly in most cases. In order to see how the model predicts unknown data entries we used the test data. The result is shown in Table 2. We can see, that

PREDICTED TRUTH	none	comp1	comp2	comp3	comp4	Recall
none	120313	15	0	0	6	0,9998
comp1	2	464	24	24	14	0,8788
comp2	0	22	1097	8	39	0,9382
comp3	0	0	0	412	4	0,9904
comp4	1	10	17	4	552	0,0,9452
Precision	1,0000	0.9489	0.9532	0.9035	0.8976	

Table 2. Results on test data set.

the model has overall accuracy 99%, and 95% average per class accuracy which is very promising for experimental case.

5   Conclusions, Limitations and Outlook

In this paper we conducted a case study in the field of Predictive Maintenance (PdM) with sample machine data to demonstrate how explainable AI can be reached in the field of manufacturing. Although the study is strictly experimental, we can conclude that explainable AI in form of reliable prediction model and visualizations can reasonably contribute to avoiding unnecessary costs associated with unscheduled downtime caused through machine errors or tool failures.

The basic limitation of this contribution is that this experiment was conducted with generic data set, however the presented concept shows high maturity with promising results.

The next step in the future would be to engage the trained model with some data collected directly in real world manufacturing settings and involving data

from different manufacturer. In this way the reliability of presented results could be approved and through the comparison of results from different data sources and adjustment of the prediction model.

References

1. Lee, W.J., Wu, H., Yun, H., Kim, H., Jun, M.B., Sutherland, J.W.: Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data. Procedia CIRP **80** (2019) 506 – 511 26th CIRP Conference on Life Cycle Engineering (LCE) Purdue University, West Lafayette, IN, USA May 7-9, 2019.
2. Kim, T.W.: Explainable artificial intelligence (xai), the goodness criteria and the grasp-ability test. arXiv preprint arXiv:1810.09598 (2018)
3. Biran, O., Cotton, C.V.: Explanation and justification in machine learning : A survey or. (2017)
4. Bornstein, A.M.: Is artificial intelligence permanently inscrutable?
5. Clancey, W.J.: Intelligent tutoring systems: A tutorial survey. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE (1986)
6. Core, M.G., Lane, H.C., Van Lent, M., Gomboc, D., Solomon, S., Rosenberg, M.: Building explainable artificial intelligence systems. In: AAAI. (2006) 1766–1773
7. Hawkins, J.: Special report: Can we copy the brain?-what intelligent machines need to learn from the neocortex. IEEE Spectrum **54**(6) (2017) 34–71
8. Marcus, G.: Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631 (2018)
9. Monroe, D.: Ai, explain yourself. Commun. ACM **61**(11) (October 2018) 11–13
10. Sheh, R.K.M.: " why did you do that?" explainable intelligent robots. In: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence. (2017)
11. Voosen, P.: How ai detectives are craking open the black box of deep learning. Science (2017)
12. Wang, D., Yang, Q., Abdul, A., Lim, B.Y.: Designing theory-driven user-centric explainable ai. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. (2019) 1–15
13. Weinberger, D.: Our machines now have knowledge we'll never understand. Backchannel https://www. wired. com/story/our-machines-now-have-knowledge-well-never-understand (2017)
14. Susto, G.A., Schirru, A., Pampuri, S., Mcloone, S., Beghi, A.: Machine learning for predictive maintenance: A multiple classifier approach. Industrial Informatics, IEEE Transactions on **11** (06 2015) 812–820
15. Selcuk, S.: Predictive maintenance, its implementation and latest trends. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **231**(9) (2017) 1670–1679
16. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems 30. Curran Associates, Inc. (2017) 3146–3154