Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 180 (2021) 476–481

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

Explaining a Random Forest With the Difference of Two ARIMA Models in an Industrial Fault Detection Scenario

Anna-Christina Glock[a]

[a] *Software Competence Center Hagenberg GmbH (SCCH), Softwarepark 21, 4232 Hagenberg, Austria*

Abstract

In this paper a method is proposed to obtain explainability of the random forest model. Two Auto-Regressive Integrated Moving Average (ARIMA) models form the basis for this approach. The ARIMA models are used in a way similar to how local surrogate models are typically applied. The explanation of random forest's prediction is derived from the numerical differences of the parameters of the ARIMA models. To demonstrate the feasibility of this idea, an experiment that implements this approach is conducted. The data used for this are similar to an accumulated bathtub curve representing failure rates in a production process. The results of the experiment show that the approach is able to identify a linear trend in some parts of the data, and therefore locally provide an explanation for the functional form of the underlying failure rate.

Keywords: ARIMA; random forest; Explainable AI; surrogate model; Fault Detection

1. Introduction

In the last few years, the term Industry 4.0 gained in importance. Industry 4.0 has the goal to increase the productivity by integrating modern technologies like Machine Learning techniques into industrial processes. The vision is that those techniques do not replace humans but assist humans by taking care of low-level thinking tasks [1, 2]. Consequently, it is important that those human experts willingly work with the new techniques. One of the problems hereby is that a lot of Machine Learning solutions are black boxes and their decision-making process is incomprehensible. However, understandability is important for the acceptance of AI solutions by human experts, according to an article from the German Research Center for Artificial Intelligence (DFKI) [3]. Furthermore, there are also legal and societal implications of black box solutions that pose unsolved dilemmas in terms of accountability. Therefore, explainable AI has emerged as an important sub-subject of modern machine learning research [4, 5, 6, 7].

E-mail address: anna-christina.glock@scch.at

10.1016/j.procs.2021.01.360

In section 2, relevant literature regarding the used models and the use case of the experiment is reviewed. The subsequent section 3 presents an algorithm that enables interpretation of the random forest model. Furthermore, an experiment is conducted to show that the algorithm works as intended and allows the interpretation of the random forest model (section 4). The results of the experiments are promising since they show that there are differences in the parameters of the two ARIMA models. Moreover, the difference in one specific parameter is interesting as it helps to interpret the local random forest model.

2. Related Work

The importance of industry 4.0 can be seen by looking at the strategies different leading industrial countries proposed over the last years [1]. Machine Learning can improve different areas in the industrial manufacturing setting, like fault detection or remaining useful life estimations [8, 9]. Those information support the decision making process for problems regarding predictive maintenance or production planning [10, 11, 12]. The DFKI built a smart factory with LEGO bricks to demonstrate the potentials of AI in industrial manufacturing [13].

As mentioned before, understandability is important for the acceptance of AI solutions [3, 14]. Therefore, it is not surprising that explainable AI has grown over the years and can now be divided into different sub-fields [15]. A specific field of explainable AI are local surrogate models, such as LIME [16]. They try to approach the problem of finding a model-agnostic, (i.e., independent of the used AI-solution), post-hoc [14, Chapter 5.7] interpretation by the use of a 'surrogate' or 'proxi' [17] which due to its more conventional architecture provides interpretability of its 'proxy' solutions. The idea is: If the surrogate imitates the solutions of the AI model, then due to its interpretability the AI-solutions are inheriting these 'explanations' for all cases in which the simple model imitates the AI model well. This property is called model fidelity [16, 17].

In the present approach, a random forest model is used to identify change points in time series belonging to machine failure data [18]. The lack of interpretability due to the nature of bagging [19] is targeted by the difference of two local surrogate models based on the Auto-Regressive Integrated Moving Average (ARIMA) time series method. This is the novelty of the approach, as in contrast to the typical surrogate models no training on the data and prediction of the black box model happens. The two surrogates are trained on the data and the error between the black box model prediction and the original data (see Section 3). Changes in the surrogate model parameters are used for interpreting the random forest predictions. It is shown in an experiment (see Section 4) that locally, specifically in the part of the data with a linear regime of failure types, the fidelity is sufficient to provide interpreted change point detection solutions from the otherwise non-interpretable, but high performing machine learning algorithm represented by the random forest model.

3. Algorithm

This section explains the proposed algorithm for explaining a non-explainable AI model. The general idea of the algorithm is to use the difference between two explainable models to explain a non-explainable model. As mentioned before, two ARIMA models are used as explainable models, and a random forest is used as a non-explainable model in this paper.

The execution of the following steps allows reaching the goal of explaining a non-explainable model. First, a sufficient amount of data points for training a random forest and ARIMA model are chosen. Then, an ARIMA model, A_t, is fitted to the data. The sum of the model results and the corresponding residual error, ϵ_t, are the y-values of the data, G_t. This is represented in equation (1).

$$G_t = A_t + \epsilon_t \tag{1}$$

The next step is the fitting of a random forest regression, RF_t, on the data. In equation (2), G_t again represents the y-values of the data. The error between RF_t and G_t is η_t.

$$G_t = RF_t + \eta_t \tag{2}$$

The result of the random forest regression is subtracted from the data in the next step. The result of this subtraction is the part of the data that is not explained by the random forest regression. This is the same as the error η_t.

$$G_t - RF_t = \eta_t \tag{3}$$

The last step before the interpretation of the results is the fitting of a second ARIMA model to the result of the subtraction. In equation 4, η_t is the subtraction result of 3. The fitted ARIMA model is represented by A_t'. The error between A_t' and η_t is ϵ_t'.

$$\eta_t = A_t' + \epsilon_t' \tag{4}$$

After that step, there are two ARIMA models: one fitted to the original data G_t (A_t), the other one fitted to the part of the data which is unexplained by the random forest model η_t (A_t'). A comparison of these two ARIMA models with each other is the last step. As the difference originates in the subtraction of the random forest regression from the data, being able to explain the difference between the two ARIMA models helps to understand what part of the data the random forest was able to predict.

Different parameters can be used to describe the difference between two ARIMA models. For this paper, the order has been chosen as the parameter of interest. The order of an ARIMA model consists of three parameters q, d and p [20]. The first parameter, q, originates from the AR part, d defines the number of integration steps needed to reach a stationary signal and the last parameter, p stems from the MA part. The first and last parameter define how many previous data points should be used for the AR and MA part, respectively.

4. Experiments

The idea described in the previous section originates from a project in which changes in accumulated failure counts should be detected. Therefore, similar data are used for testing the idea. Equation 5 [21] allows the generation of a curve that is similar to the curve created by accumulated failure counts. For the experiment in this section, one curve is generated using equation 5.

$$f(t) = (\alpha e^{-\theta(t-t\_0)} + c)\eta(t - t\_0) + c'\eta(t - t\_1) + b(t - t\_2)\eta(t - t\_2) \tag{5}$$

There are three parts:

- $\alpha e^{-\theta(t-t\_0)}$ this part has an influence on the curve from time t_0 until the end. α is the amplitude and θ is the change rate and allows the configuration of the running-in process.
- c and c' allow to configure two different linear failure rates. One, c, that begins at t_0, and c' beginning at t_1, represent two different linear failure rates, starting at t_0 and t_1, respectively.
- $b(t-t\_2)$ starts at t_2 and represents the superlinear failure rate that appears at the end of such process. b is there the quadratic coefficient.

With the help of a sliding window technique [22, 23], the generated curve was split into different parts: For every window, the algorithm described in section 3 was executed. The chosen window size was 200, the curve consists of 1100 data points, and the sliding window is moved one data point further per step. Therefore, a total of 900 windows have been generated and analyzed.

In 59 of the 900 windows, no change in the order of the ARIMA models occurred, in the other 841 windows at least one order parameter changed between the ARIMA models. The most changes happened in the d parameter where is changed in 625 windows, followed by the p in 616 windows and the q parameter, which changes in 564 windows. Often more than one parameter changes in the same window.

To explain the random forest one has to look at a specific window, for example the window from timestamp 431 to 631. Figure 1 shows the RF_t and the G_t for this window. The η_t is illustrated in Figure 2. Comparing those two figures, it can be noted that η_t has no trend, while G_t has one. Which can also be observed looking at the order of the

ARIMA model A_t and A'_t. The order of the ARIMA model A_t is (0,1,1), the order of the other ARIMA model A'_t is (0,0,2). Comparing those two orders, it is apparent that the order of d and p changes but the order of q stays the same. That d is 0 means that in the A'_t model the integrated part is not used. As the q is also 0, this model is reduced to a MA model. That the integrated part of the ARIMA is not needed means that η_t is stationary and has no trend any more. Which can also be observed comparing Figure 1 and 2. From these observations, it can be concluded that the random forest is able to model the trend for the data in this window. In this specific case the random forest is able to model the linear trend that exist in the data from timestamp 431 to 631.

The next step is to analyze the changes in all the windows to find out if this behaviour is specific for this one window or generalizable for more windows. The d for the A_t models over all windows is always 1. Figure 3 shows the d for all windows and models A'_t, there it is visible that, except for some outliers, the d is 0 in the middle and 1 in the beginning and the end. Furthermore, there are periods where the value changes very often from one window to the next. As the middle part of the curve mostly consists of a linear trend it can be deduced that the random forest models are able to model the linear trend, but have problems with modelling the nonlinear trend at the end and the beginning.

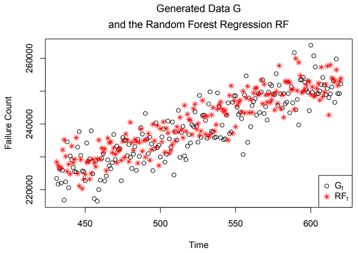

Fig. 1: The black points are the data points generated by accumulating the results of equation 5. The red stars represent the regression result of the random forest.

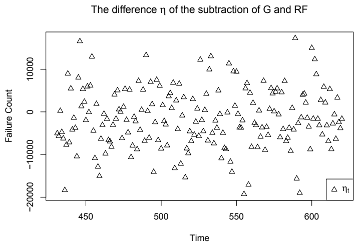

Fig. 2: The triangles are the difference between the random forest regression result and the data G.

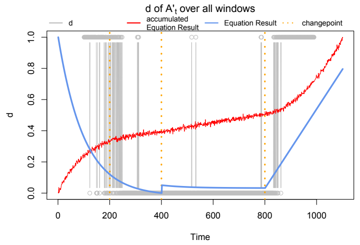

Fig. 3: The grey line and points is the order of d of the A'_t ARIMA model. The red line is the accumulated result of the equation 5 combined with a noise. The blue line represents the not accumulated result of the equation 5.

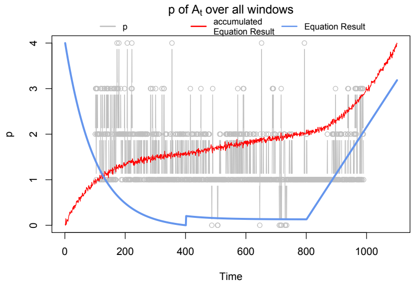

Fig. 4: The grey line and points is the order of p of the A_t ARIMA model. The red line is the accumulated result of the equation 5 combined with a noise. The blue line represents the not accumulated result of the equation 5.

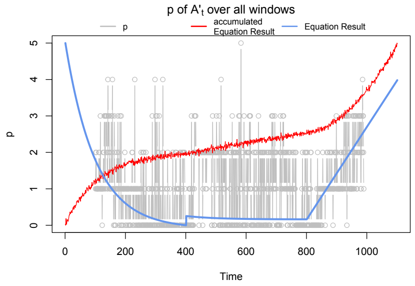

Fig. 5: The grey line and points is the order of p of the A'_t ARIMA model. The red line is the accumulated result of the equation 5 combined with a noise. The blue line represents the not accumulated result of the equation 5.

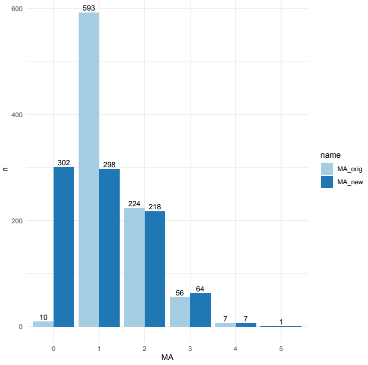

Fig. 6: The values of the left bars of every value on the x-axis stems from the p values from A_t model. The model A'_t supplies the values for the other bars.

The other parameter that changes in the window from timestamp 431 to 631 is p. The parameter increases by one, which means that not only one but two values of the past are used to calculate the MA part of the ARIMA model. It is also interesting if and how the first coefficients of the MA model changed between A_t and A'_t. How this difference can help to explain the random forest requires further research.

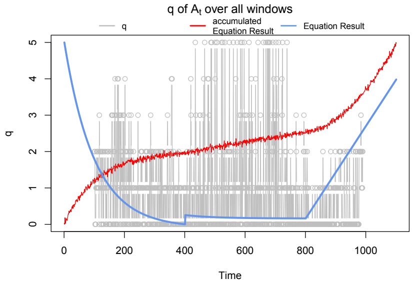

Fig. 7: The grey line and points is the order of q of the A_t ARIMA model. The red line is the accumulated result of the equation 5 combined with a noise. The blue line represents the not accumulated result of the equation 5.

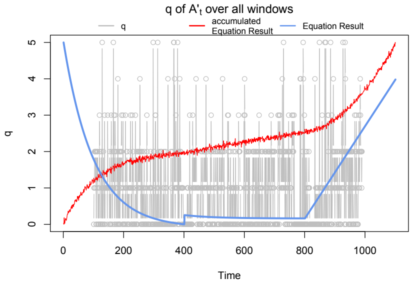

Fig. 8: The grey line and points is the order of q of the A'_t ARIMA model. The red line is the accumulated result of the equation 5 combined with a noise. The blue line represents the not accumulated result of the equation 5.

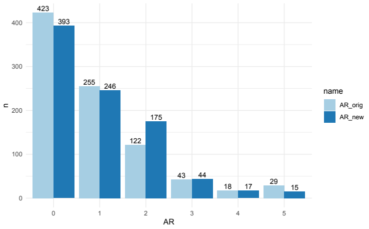

Fig. 9: The values of the left bars of every value on the x-axis stems from the q values from A_t model. The model A'_t supplies the values for the other bars.

Similar to the analysis of the parameter d it was looked into the change over all the windows. Figure 4 visualizes the values of the order p of the A_t models for all the windows. In Figure 5 the same is shown for the A'_t models. The total amount of different values for parameter p for A_t and A'_t respectively can be seen in figure 6. From those figures, it can be deduced that for the A_t models the value of p was 1 significantly more often than the other values. The values of p of the A'_t models are distributed evenly between the values 0, 1 and 2.

The last parameter of ARIMA that is analyzed is q. This parameter defines how many past values are used for the AR part of an ARIMA model. For the window from timestamp 431 to 631 q is 0 for the A_t model and the A'_t model. There is no change that can be used to explain the random forest.

In the next step, it was evaluated if there are windows where the q changes or if the q is always unaffected. Viewing the other windows showed that there are windows where q changes (Figure 7 and 8). Figure 9 illustrates that distribution of the values of q is very similar between the A_t and the A'_t models. As with the parameter q, there is still more research to do, for example, how the coefficients of q change between model A_t and model A'_t.

As mentioned at the beginning of this section, the idea for this algorithm stems from a project to detect points in time where the behavior of given data changes. The dashed lines in figure 3 represents the point in time where a change happens in the data represented by the red curve. In this figure, an interesting observation can be made: a correlation between two of the three change points and changes in the parameter d can be seen. Future research is necessary to determine if this correlation between change points and the value of parameter d is generalizable for similar curves or only occurs for this specific curve. In case it is generalizable, the question, if it is better than other change point detection methods, needs to be addressed.

5. Conclusion

The main question of this paper revolves around explainable AI. An algorithm is proposed that uses the difference of two local surrogate models to explain a non-explainable model locally. An experiment was conducted, with a random forest model as the non-explainable model and ARIMA models as the explainable models. During the experiment the data was split into parts with the help of the sliding window technique. The results of this experiment have shown that a random forest models the linear trend in the data, as the ARIMA parameter d changed from 1 to 0 in those parts of the data where a linear trend exists. Furthermore, changes in q and p have been registered. There future research is necessary to use these changes to improve the interpretation of the random forest model. Overall it can be said that the proposed algorithm can explain a non-explainable model by utilizing the difference of two explainable models.

Acknowledgements

”The research reported in this paper has been funded by the COMET InTribology, FFG.-No 87176 (project coordinator AC2T research GmbH, Austria) and by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Federal Ministry for Digital and Economic Affairs (BMDW), and the Province of Upper Austria in the frame of the COMET - Competence Centers for Excellent Technologies Programme managed by Austrian Research Promotion Agency FFG.”

References

[1] J. Zhou, P. Li, Y. Zhou, B. Wang, J. Zang, L. Meng, Toward new-generation intelligent manufacturing, Engineering 4 (1) (2018) 11–20. doi:10.1016/j.eng.2018.01.002.

[2] A. Angelopoulos, E. T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, T. Zahariadis, Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects, Sensors 20 (1) (2019) 109. doi:10.3390/s20010109.

[3] F. Peter, M. Nijat, Xai 4.0 – explainable artificial intelligence für industrie 4.0, DFKI - Deutsches Forschungszentrum für Künstliche Intelligenz. URL https://www.dfki.de/web/news-media/news-events/events/hannover-messe-2019/xai-40/

[4] D. Gunning, A. D. W., Darpa's explainableartificial intelligence program, AI Magazine 40 (2019) 44 – 58. doi:10.1609/aimag.v40i2.2887.

[5] R. S, Explainable machine learning — 5 must read papers, Medium. URL https://medium.com/@rs134/explainable-machine-learning-5-must-read-papers-95660d9f0c72

[6] B. Wilson, J. Hoffman, J. Morgenstern, Predictive inequity in object detection (2019). arXiv:1902.11097.

[7] A. Holzinger, Explainable AI (ex-AI), Informatik-Spektrum 41 (2) (2018) 138–143. doi:10.1007/s00287-018-1102-5. URL https://doi.org/10.1007/s00287-018-1102-5

[8] N. Amruthnath, T. Gupta, A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance, in: 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), IEEE, 2018. doi:10.1109/iea.2018.8387124.

[9] B. Freudenthaler, Predictive maintenance: Projektbeispiele aus der anwendungsorientierten forschung, in: ÖTG-Symposium 2019 Tribologie in Industrie und Forschung, 2019, pp. 5–10.

[10] J.-R. Rehse, N. Mehdiyev, P. Fettke, Towards explainable process predictions for industry 4.0 in the dfki-smart-lego-factory, KI - Künstliche Intelligenz, German Journal on Artificial Intelligence - Organ des Fachbereiches ”Künstliche Intelligenz” der Gesellschaft für Informatik e.V. (KI) 33 (1) (2019) 181–187.

[11] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, Deep learning for smart manufacturing: Methods and applications, Journal of Manufacturing Systems 48 (2018) 144–156. doi:10.1016/j.jmsy.2018.01.003.

[12] I. Nunes, D. Jannach, A systematic review and taxonomy of explanations in decision support and recommender systems, User Modeling and User-Adapted Interaction 27 (3-5) (2017) 393–444. doi:10.1007/s11257-017-9195-0.

[13] J.-R. Rehse, S. Dadashnia, P. Fettke, Business process management for industry 4.0 – three application cases in the dfki-smart-lego-factory, IT - information technology (IT) 60 (3) (2018) 133–141.

[14] C. Molnar, Interpretable Machine Learning, 2019, https://christophm.github.io/interpretable-ml-book/.

[15] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai (2019). arXiv:1910.10045.

[16] M. T. Ribeiro, S. Singh, C. Guestrin, ”why should i trust you?”: Explaining the predictions of any classifier (2016). arXiv:1602.04938.

[17] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning (2018). arXiv:1806.00069.

[18] A.-C. Glock, Detection of changes in wear-behaviour in data from continuous wear analysis, Master thesis, FH Oberösterreich, Hagenberg (2020).

[19] U. Johansson, C. Sönströd, U. Norinder, H. Boström, Trade-off between accuracy and interpretability for predictive in silico modeling, Future medicinal chemistry 3 (6) (2011) 647–663. doi:10.4155/fmc.11.23.

[20] R. Adhikari, R. K. Agrawal, An introductory study on time series modeling and forecasting, LAP Lambert Academic Publishing. URL http://arxiv.org/pdf/1302.6613v1

[21] A.-C. Glock, F. Sobieczky, M. Jech, Detection of anomalous events in the wear-behaviour of continuously recorded sliding friction pairs, in: ÖTG-Symposium 2019 Tribologie in Industrie und Forschung, 2019, pp. 30–40.

[22] H. Tawfeig, V. S. Asirvadam, N. Saad, Sliding-window learning using MLP networks with data store management, in: 2011 National Postgraduate Conference, IEEE, 2011. doi:10.1109/natpc.2011.6136391.

[23] A. Helwan, D. U. Ozsahin, Sliding window based machine learning system for the left ventricle localization in MR cardiac images, Applied Computational Intelligence and Soft Computing 2017 (2017) 1–9. doi:10.1155/2017/3048181.