

WHODID: Web-based interface for Human-assisted factory Operations in fault Detection, Identification and Diagnosis

Pierre Blanchart and Cédric Gouy-Pailler
CEA, LIST, 91191 Gif-sur-Yvette Cedex, France

Abstract. We present WHODID: a turnkey intuitive web-based interface for fault detection, identification and diagnosis in production units. Fault detection and identification is an extremely useful feature and is becoming a necessity in modern production units. Moreover, the large deployment of sensors within the stations of a production line has enabled the close monitoring of products being manufactured. In this context, there is a high demand for computer intelligence able to detect and isolate faults inside production lines, and to additionally provide a diagnosis for maintenance on the identified faulty production device, with the purpose of preventing subsequent faults caused by the diagnosed faulty device behavior. We thus introduce a system which has fault detection, isolation, and identification features, for retrospective and on-the-fly monitoring and maintenance of complex dynamical production processes. It provides real-time answers to the questions: "is there a fault?", "where did it happen?", "for what reason?". The method is based on a posteriori analysis of decision sequences in XGBoost tree models, using recurrent neural networks sequential models of tree paths.

The particularity of the presented system is that it is robust to missing or faulty sensor measurements, it does not require any modeling of the underlying, possibly exogenous manufacturing process, and provides fault diagnosis along with confidence level in plain English formulations. The latter can be used as maintenance directions by a human operator in charge of production monitoring and control.

Keywords: production units, fault detection and identification, maintenance operator friendly, tree ensemble, gradient boosting, LSTM-RNN networks

1 Introduction

Modern factories operation and optimization rely on fine-grained monitoring of machines and products. Besides classical purposes such as energy optimization and smart production planning, there is a high demand for systems able to detect and isolate the location of faults occurring in production chains. Thus there has been a tremendous effort to design computational intelligences able to represent the underlying dynamics of such complex systems, with the goal of detecting, identifying and possibly explaining the occurrence of faults while the system is in operation. Fault detection and identification is often addressed through an explicit modeling of the system processes using supervised approaches. The first problem with this approach is that it implies learning as many models as there

are processing steps, which can be a huge number in modern factories. The second problem comes from the faulty and missing sensor measurements, which, combined with the complex and dynamical nature of some processes make such modeling highly inaccurate and unreliable for fault detection [5]. In our approach, we learn a global fault detection model (FDM) taking all sensor measurements into account for more reliable detection, and we perform a posteriori analysis of this model to perform fault identification and diagnosis. Of course, such an approach is only viable if the global model’s decisions are interpretable by any means, and those decisions can be related to the individual physical equipments, e.g. the work stations, for fault isolation/identification. We use XGBoost [1], a gradient boosting tree ensemble classification method, as a FDM since it has proved robustness and even superior performance for such unbalanced two-class classification problems as fault detection. The drawback of such a model is that it does not provide with any direct interpretability of its decision, which is a desirable feature for identification and diagnosis [2, 3]. Some approaches cope with this issue by simplifying the learned FDM to make it interpretable [4, 6], but degrading the detection performances. In a similar spirit, some models are constrained to be simple enough for interpretability, impacting the detection performance as well [7]. Unlike those, we keep the original FDM and seek interpretation from directly it using tree path analysis, thus keeping the original FDM performance.

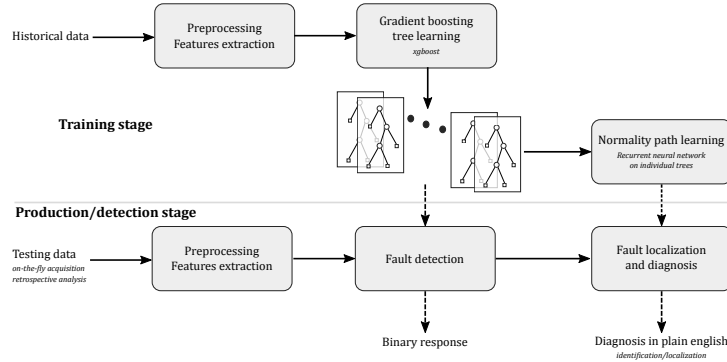
2 Fault detection, identification and diagnosis

We train the XGBoost FDM on a large set of engineered features that are related to a physical equipment or a physical entity in the factory such as a station or a production line. Features can be sensor measurements made at stations level, timestamps of products passage in a station, more evolved features such as non-linear projections of sensor measurements, features characterizing the time distribution of faults at a station. . . XGBoost is particularly suited to the scenario where we are using heterogeneous data, with various dynamics, and possibly many missing/abnormal data. Besides, it is not sensitive to redundant features, making it a very robust approach for fault detection in production industry, where we typically deal with numerical, categorical and timestamps data representing a mix of sensor measurements and feedback from human station operator, and as such very liable to be faulty/redundant or missing.

Identification and diagnosis are then performed in a joint manner by analyzing the trees in the XGBoost model. The idea is to learn sequential models of paths followed by non-faulty data inside the trees. Thus for each node of a tree, we want to have a model able to say what is the most likely path to be followed subsequently by a non-faulty data, i.e. we want to model what is the probability to go to the left branch, to the right branch or to end in a leaf. Those models have a sequential nature since, in a given node, they are conditioned by the path followed from the root to this node. And there is a combinatorial aspect induced by all the possible paths in the tree. We address this aspect by learning recurrent models of tree paths, using long-short term memory recurrent neural

networks [8]. Numerical data is used along with the node index, to make the learning problem easier and break the combinatorial aspect, since, numerically speaking, not all tree paths figure in the data: only tree paths potentially existing are learned. We train as many tree path models as there are trees in the XG-Boost model, and for each faulty data, we look inside each tree in which node(s) its tree path **diverges** from the "normal" tree path learned from non-faulty data. KL-divergence is used as a measure of divergence in a node between the predicted distribution by our normal path model (probability of "left", "right", "leaf"), and the observed distribution, *i.e.* in which branch the fault data goes. This gives us an indication as to where and why a fault happened, since the faulty data obviously follow paths in the decision trees which at some point diverge from normality. Identification and diagnosis are straightforward to obtain since each node of a decision tree makes direct reference to a feature, and, defines a "normality regime" on this feature thanks to the split value associated to the node. The feature being related to a precise physical equipment, we can easily output as a potential fault identification the concerned equipment, and, as a diagnosis the interval of normality defined by the node split along with the abnormal measure. Such an identification/diagnosis pair can be formulated in plain English and enriched with informations on the sensor(s) measures associated with the node where the divergence was observed. This last part is mostly the responsibility of the industrial actor and has no genericity.

Fig. 1. Processing workflow of the fault detection, identification and diagnosis system.



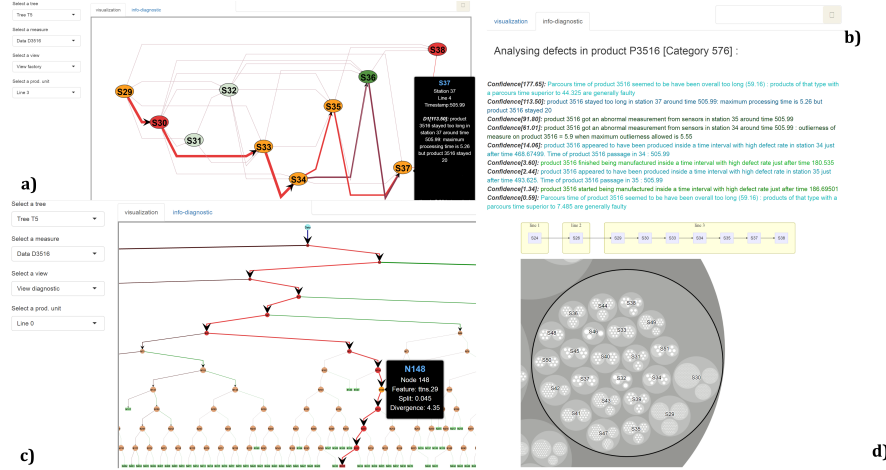
To rank identification diagnosis pairs according to relevance, observed node divergences are aggregated across all the trees of the global defect model by computing in which proportion an individual tree score contributes to the global defect score and reweighing accordingly. It enables a ranking of potential fault diagnosis by decreasing order of relevance. This human readable output then allows an operator in charge of production chain maintenance and control to address the problem in the right place.

3 Interface operation

The interface operation is demonstrated in fig. 2: the operator selects a production line in the hierarchical view in 2.d and a faulty product in the side menu

in 2.a, and obtains a view of the selected line which shows the product parcoures through stations along with fault diagnosis shown as tooltips in the stations where a problem was identified (2.a). A full fault report in plain English is displayed in panel 2.b. The view in 2.c shows algorithmic insights about the model and would not be visible to a production monitoring operator.

Fig. 2. User-Interface overview. a) Faults (orange stations) are reported on the path of product P3516 through stations in line 3 (red), and detailed in the tip. b) Full fault diagnosis of P3516 with their respective confidence levels in brackets. c) Decision path of P3516 (in red) in tree T5, with one node divergence (in orange) referring to a fault in station S29. d) Hierarchical view of the factory (lines – stations).



References

- Chen, T. and Guestrin, C.: XGBoost: A Scalable Tree Boosting System. ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining (2016)
- Túlio Ribeiro, M. and Singh, S. and Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining (2016)
- Lipton, Z. C.: The Mythos of Model Interpretability. ICML Workshop on Human Interpretability in Machine Learning (WHI 2016) (2016)
- Hara, S. and Hayashi, K.: Making Tree Ensembles Interpretable. ICML Workshop on Human Interpretability in Machine Learning (WHI 2016) (2016)
- Sobhani-Tehrani, E. and Khorasani, K.: Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach. Springer (2009)
- Gallego-Ortiz, C. and Martel, A. L.: Interpreting extracted rules from ensemble of trees: Application to computer-aided diagnosis of breast MRI. ICML Workshop on Human Interpretability in Machine Learning (WHI 2016) (2016)
- Letham, B., Rudin, C., McCormick, T. H. and Madigan, D.: Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model. The Annals of Applied Statistics, Vol. 9(3), 1350-1371 (2015)
- Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory. Neural Comput., Vol. 9(8), 1735-1780 (1997)