Making Industrial Analytics work for Factory Automation Applications

Markus Koester

Weidmüller Group, Klingenbergstr. 16, 32758 Detmold, Germany
markus.koester@weidmueller.com

Abstract. In this contribution, we give an insight in our experiences in the technical and organizational realization of industrial analytics. We address challenges in implementing industrial analytics in real-world applications and discuss aspects to consider when designing a machine learning solution for production. We focus on technical and organizational aspects to make industrial analytics work for real-world applications in factory automation. As an example, we consider a machine learning use case in the area of industry compressors. We discuss the importance of scalability and reusability of data analytics pipelines and present a container-based system architecture.

Keywords: Industrial Analytics, Anomaly Detection, Development Process.

1 Introduction

In factory automation maintainers and operators constantly ask themselves if their assets are operating well or what measures they should take to keep up a good operation and to avoid unforeseen downtimes. Classical condition monitoring approaches, such as signal tracing and threshold mechanisms, only apply for a reactive maintenance scenario, where machine operators usually get informed, when it is already too late to avoid a machine failure. Inspired from recent advance in other areas such as e-commerce and finance, industrial analytics based on machine learning algorithms is gaining attention as a mean to get a deeper insight into the current state of machines or plants. Machine learning is promised to be the key technology to deliver a glimpse into the future of the machine behavior, predicting if and when components are supposed to fail from a statistical point of view under the current operational conditions. In the context of factory automation, machine learning is a relatively new topic, such that the know-how and experience of machinery experts in implementing data analytics pipelines is still limited. Adapting machine learning in the field of factory automation requires not only a sound understanding of the underlying mechanisms of the various algorithms, but also software engineering skills to implement suitable data analytics pipelines for the target machine. Working examples of machine learning implementations at a production level are still rare [1].

This paper gives an overview of the experiences we have gained in creating industrial analytics solutions in the area of machinery and factory automation. The focus of

J. Beyerer et al. (Eds.), *Machine Learning for Cyber Physical Systems*, Technologien
für die intelligente Automation 9, https://doi.org/10.1007/978-3-662-58485-9_13

this paper is more on the challenges in implementing these solutions. Section 2 gives a high-level overview over the functionality of the industrial analytics pipeline, which was considered in the implementations. This pipeline describes the data flow starting from the raw data created by the target machine to the visualization of the analytics results. Section 3 covers the main scope of this contribution by highlighting the challenges, which are a) design considerations to allow for scalability of the solution, b) our underlying process from the first idea of the solution to the final production-ready software, and c) a continuous integration (CI) and continuous delivery (CD) pipeline for automatically building the software solution. In Section 4 we give an overview over an example application, which we have implemented the industrial analytics solution for.

2 Overview of the Industrial Analytics Pipeline

The core concept of the analytics pipeline is present in Fig. 1. Collecting machine data is highly use case dependent and requires to be tailored according to the given data sources and accessibilities of the target machine. To simplify the data processing of the following analytics steps the raw data requires being collected and stored centralized if the target architecture allows. Having a single data source for further data operations of the pipeline, such as a centralized data base, greatly simplifies the data handling.

Preprocessing of the data is a key step to filter out data that has little or even no impact on the modeling success and to create relevant features that represent the actual state of the target machine. As described in the context of data dependencies in [2] the quality of the result of an analytics model greatly depends on the given input features. Besides statistical and data centric approaches, we consider domain knowledge provided by the machine user in the creation of features. Thus we combine expert know-how from the industry application domain and from the data science domain.

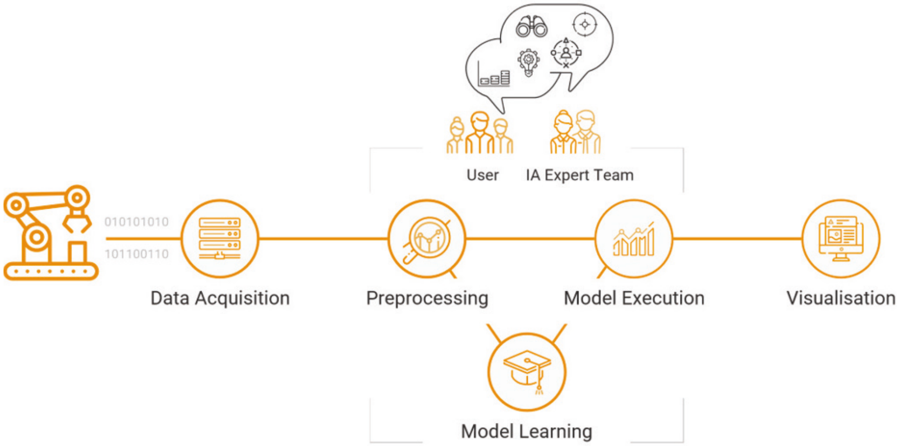

Fig. 1 Industrial Analytics Pipeline

The selected features are used in the two branches model learning and model execution. The selection of the underlying machine learning algorithm highly depends on the target application. Once a model is created it can be used in the model execution branch to compute analytics results. These can be numerical indicators for anomaly detection or contextual information reflecting the current state of the machine. For the scenario of predictive maintenance the output of the model can be e.g. the likelihood of a failure in a given future time interval. This information is finally visualized to support the user in taking decisions for optimizing the efficiency of the machine and for avoiding unplanned down-times.

3 Challenges in implementing Industrial Analytics

In the context of machinery and factory automation industrial analytics is a relatively new topic, where experiences in the technical and organization realization are still rare. In this section we provide an insight into our experiences in implementing industrial analytics in real world applications and discuss aspects to consider when designing a machine learning solution for production.

3.1 Scalability and reusability of data analytics pipelines

In contrast to classical big data application such as natural language processing or image classification, machinery applications typically suffer from little amount of historic data. On the one hand automation technology for collecting machine data at high sampling rates needed for machine learning applications was hardly available. On the other hand there was simply no need to store large amounts of machine data for the given automation application. With the growing awareness of the value of historic data, machine builders and operators start to implement more and more sensor technology to improve the data quality. Thus the amount of data generated by machines will increase in the future as the cost for implementing sensor and storage technology decrease. However, in the current machinery applications, data sets tend to be in the Mega-Byte to Giga-Byte range, allowing for applying small data processing architectures, which should be prepared for scalability to allow for processing larger data sets in the future. To achieve that, we designed a container-based architecture, where the key functions such as the analytics pipeline, frontend user interface, etc. are implemented in separate software containers as shown in Fig. 2.

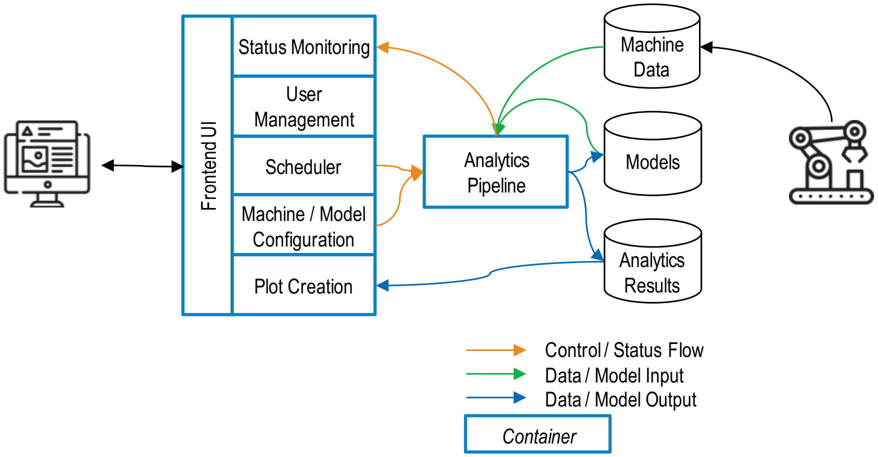

Fig. 2 Container-based Industrial Analytics Architecture

The fronted user interface holds different functionality, which is described in the following: Status monitoring is used to track the state of the analytics pipeline and to inform the user about abnormal behavior. The analytics architecture is designed to handle different users and to provide authentication and user grouping functionalities. Analytics functions, such as model scoring or model learning can be executed in different time intervals, which can be configured in a scheduler. The user can select different models out of the ones given in the model data base and configure and tune the models according to the target machine. The plot creation container is used to generate user-defined plots based on the resulting analytics data.

The machine data is collected and stored in a corresponding data base, which is used as source for the data analytics pipeline container. Besides the machine data the architecture additionally comprises a model data base, where different machine learning models with its pre-processing pipelines are stored.

In a typical flow of the analytics functionality the scheduler triggers the execution of the analytics pipeline, which loads the selected model from the database and applies the model to the specified input machine data. For model scoring the resulting data are written to the analytics result data base, which holds the data for result visualization. For a model learning scenario, the result of the analytics pipeline is a new or updated machine model, which is stored in the model data base, and which can be used for scoring in the future.

The architecture is designed for horizontal scalability and platform independence. Instead of using a single analytics pipeline, the architecture allows for running several analytics pipelines at the same time, which can be used to speed up the execution, or to run different models concurrently. Its container-based implementation allows the architecture to be deployed locally on a single PC (with reasonable amount of available resources) as well as on virtual environments in the cloud.

3.2 Process from idea to production

The industrial analytics solution touches various fields, such as data engineering, machine learning, UI design and systems engineering. Covering the variety of these topics requires an interdisciplinary development team. Typical roles are:

- Application Engineer: They cover the domain knowledge required to get a deep understanding of the target machine application.
- Data Scientist: The data scientist is developing the analytics pipeline from an algorithmic point of view. This involves the feature preparation and selection, as well as the algorithm selection and tuning.
- System Architect: System architects are required to help in defining a layout of the system architecture best suited for the target analytics application.
- Full-Stack SW-Developer: The SW developer implements the selected analytics pipeline to the target system architecture, which can e.g. be an IPC on machine level up to a cloud or hybrid solution.
- UX/UI-Designer: The user interface requires being designed for ease of use providing analytics result information at the right level of detail prepared for the target user group. The UX/UI designer creates the expected UI features and designers the

All project management related tasks are realized by a corresponding project manager. As shown in Fig. 3, we follow a development process, which is inspired by the CRISP-DM process [3]. We tailored the process to meet the special requirements of industrial analytics. Starting from the target definition of the machine learning application, we investigate the quality and quantity of data of the given application and prepare suitable analytics models in the proof of concept phase. In the pilot phase, the model is implemented on the target platform and in the final development phase the missing software features, such as UI functionality and interoperability features are finalized.

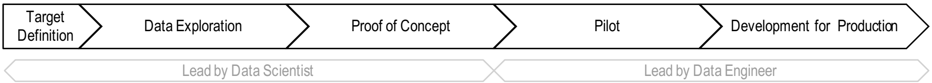

Fig. 3 Development process for data analytics solutions

3.3 CI/CD Pipeline

The development of the proposed container-based analytics solution is realized by a development team consisting of data scientist, data engineers and application engineers. One of the key challenges is to maintain team efficiency in such an interdisciplinary team. A challenge in the development process of an industrial analytics application is the implementation effort for migrating the selected machine learning model from the proof of concept phase to the final software solution in a production environment. A means to reduce this effort is to automate the software build process by continuous integration and continuous delivery (CI/CD). We have implemented a

CI/CD pipeline for the industrial analytics solution and were able to significantly reduce the development effort.

4 Practical Example

To discuss these aspects on a practical example, we consider a real-world machine learning use case in the area of industry compressors. There, we used machine learning algorithms to automatically learn the sensor data distributions of a normal behaving compressor. Consecutively, our models detect deviations from these data distributions and label them as specific anomalies. These anomalies are then predicted by an additional machine learning model to forecast component failures and to prevent unforeseen downtimes.

5 Summary

In this contribution, we focus on technical and organizational aspects to make industrial analytics work for real-world applications in factory automation. As an example, we consider a machine learning use case in the area of industry compressors. We discuss the importance of scalability and reusability of data analytics pipelines and present a container-based system architecture. Furthermore, we share the experience of our development process to bring industrial analytics solutions from idea to production. Based on that process, we present a suitable CI/CD pipeline, which supports our development team to easily bring a machine learning model from the proof of concept phase to production.

References

1. Gatica, C. P.; Koester, M.; Gaukstern, T.; Berlin, E.; Meyer, M.:"An industrial analytics approach to predictive maintenance for machinery applications," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, (2016).
2. Sculley, D.; Holt, G.; Golovin, D. et al.: Hidden technical debt in Machine learning systems. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15), C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 2. pp. 2503-2511. MIT Press, Cambridge, MA, USA (2015).
3. Wirth, Rüdiger: CRISP-DM: Towards a Standard Process Model for Data Mining. In: Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining, pp. 29-39 (2000).