Information modeling and knowledge extraction for machine learning applications in industrial production systems

Stefan Windmann[1] and Christian Kühnert[2]

[1] Fraunhofer IOSB-INA, Fraunhofer Center for Machine Learning
Campusallee 6, 32657 Lemgo, Germany
stefan.windmann@iosb-ina.fraunhofer.de
[2] Fraunhofer IOSB, Fraunhofer Center for Machine Learning
Fraunhoferstraße 1, 76131 Karlsruhe, Germany
christian.kuehnert@iosb-ina.fraunhofer.de

Abstract. In this paper, a new information model for machine learning applications is introduced, which allows for a consistent acquisition and semantic annotation of process data, structural information and domain knowledge from industrial productions systems. The proposed information model is based on Industry 4.0 components and IEC 61360 component descriptions. To model sensor data, components of the OGC SensorThings model such as data streams and observations have been incorporated in this approach. Machine learning models can be integrated into the information model in terms of existing model serving frameworks like PMML or Tensorflowgraph. Based on the proposed information model, a tool chain for automatic knowledge extraction is introduced and the automatic classification of unstructured text is investigated as a particular application case for the proposed tool chain.

Keywords: machine learning, information modeling, model serving, knowledge extraction

1 Introduction

Data in industrial production systems is usually stored in a heterogeneous way, using a large variety of data formats and semantics. The integration of these data sources, which cover besides process data also structural information, domain knowledge and process documents, is an essential prerequisite for the successful application of machine learning and optimization methods in the context of industrial production. Therefore, different data sources for machine learning applications (see Fig. 1) need to be fusioned and semantically annotated with structural information and domain knowledge.

J. Beyerer et al. (Hrsg.), *Machine Learning for Cyber Physical Systems*, Technologien für die intelligente Automation 13,
https://doi.org/10.1007/978-3-662-62746-4_8

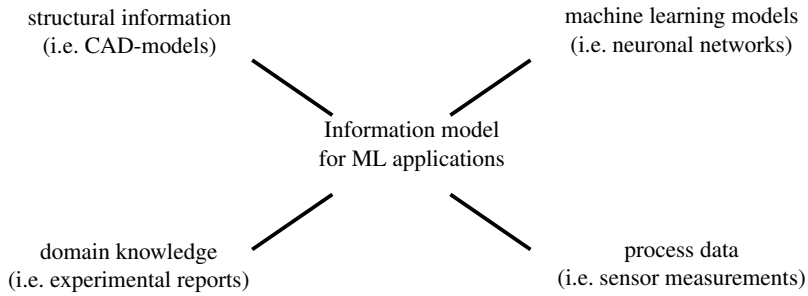

Fig. 1. Knowledge needed about a production process when using machine-learning algorithms. Fusioning all information leads to the proposed information model for ML applications.

This paper makes the following contributions to this objective:

- A unified information model for machine learning applications in production systems is proposed.
- Methods for the transformation of plant knowledge into the unified information model are introduced.

The development of appropriate information models for industrial production systems is an essential subject of the current research in the context of Industry 4.0 [4]. Examples are the reference architecture Industry 4.0 (RAMI4.0) [3], modeling languages, which allow for a structured and component-based of production plants, such as AutomationML [9], and industrial communications standards with information modeling capabilities such as OPC-UA [6]. However, such general approaches are usually not tailored to the specific requirements of machine learning approaches. In addition to these approaches, specialized information models exist, e.g. models for the sensor data acquisition like the OGC SensorThings model [1] or standardized XML-descriptions of machine learning models such as PMML [10].

In this paper, an information model for machine learning applications in production environments is proposed, which is build upon general I4.0 components and specialized information models for process data and machine learning models. In addition, a tool chain is introduced, which enables information to be automatically extracted from sensor data and other information sources and to be stored in the form of a corresponding information model. In particular, the extraction of features such as parallel automata and the automatic classification of documents from production environments are considered.

The remaining part of this paper is structured as follows: The proposed information model for machine learning applications is introduced in section 2. The tool chain for knowledge extraction is described in detail in section 3. Finally, a conclusion is given in section 4.

2 Information modeling

The essential purpose of the proposed information model is to describe production plants in such a way that machine learning applications can be straightforwardly applied to them. Figure 2 shows the complete entity relationship diagram of the proposed information model. The information model is build upon Industry 4.0 components [5]. Following a hierarchical order, components can be arranged in a tree form using subcomponents, which are specified as attributes of the entity *Component*. The information required to describe a production plant

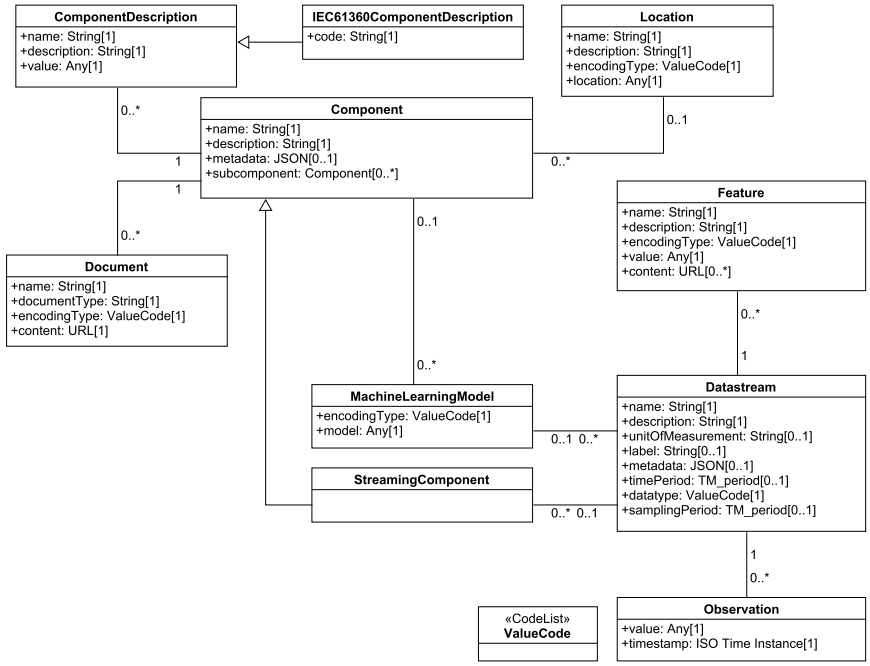

Fig. 2. Proposed information model

for machine learning applications can be roughly divided into four classes, which determine the structure of the proposed information model:

– **Structural information:** Machine learning approaches require information about the construction of the production plant, e.g. about the positions of sensors and actuators. Structural information of the production plant, which comes e.g. from CAD models, is mainly stored in the component tree using the entity *Component*. To store geospatial data for the individual components, the entity *Location* has been adopted from the OGC SensorThings model [1].

- **Domain knowledge:** In a production process, lots of domain knowledge is is available, which can be exploited for machine learning. This covers e.g. component descriptions, experimental reports, log-files, thresholds for sensor signals or information about maintenance cycles. The incorporation of such domain knowledge is mainly implemented by means of the entities *ComponentDescription*, *Feature* and *Document*. Information about the individual components (i.e. sensor replaced, sensor cleaned, etc.) is stored in the entity *ComponentDescription*, where it is possible to add e.g. the Industry 4.0 admin shell [5] and the IEC 61360 component descriptions [2] but also domain specific component descriptions. The entity *Feature* is used to add information to a *Datastream* (i.e. by adding labels, standard deviations, mean values etc.). Text documents are integrated into the proposed information model using the entity *Document*. In doing so, either links to the particular documents are used or the documents are made machine-readable by following the approach described in section 3.
- **Process data:** The process data, which is acquired and fusioned from different data sources, forms the basis for most of the existing machine learning approaches for production plants. The information model has to provide detailed information on the type and the usage of the measurements. For this purpose, the concept of data streams has been taken from the OGC SensorThing model [1]. The entity *StreamingComponent* is used to model components containing a *Datastream*. A *Datastream* describes in detail what and how the component is measuring (i.e. the unit, feature of interest, metadata). Hereby, it is worth noting that a streaming component can also be an experimental protocol, a log-file or even the operator controlling the process. To store the individual measurements, the entity *Observation* has been incorporated into the information model, which is connected using a one to many relationship to *Datastream*.
- **Machine learning models:** When using machine learning algorithms in a production environment those algorithms need to be served correctly. This serving can be in its simplest form the description of a numerical processing pipeline or in a more complex way a graph, which is stored in an XML file. Machine learning models are incorporated into the information model by using the entity *MachineLearningModel*. The proposed information model for ML applications has the possibility to store those models in a standardized way using XML descriptions such as the predictive markup model language (PMML) [10], Tensorflow or Pytorch, but it is also possible to store the served model in terms in a tailor-made, non-standardized format.

The information model covers in summary 10 entities. Details on the particular entities are described in the appendix.

3 Tool chain for knowledge extraction

Machine learning methods used to optimize industrial production systems usually require the evaluation of huge and diverse data sets for successful operation. In such application cases, the manual extraction of information is time consuming and error-

prone. Hence, a tool chain has been developed in the present work, which allows for automatic knowledge extraction using the proposed information model (see Fig. 2).

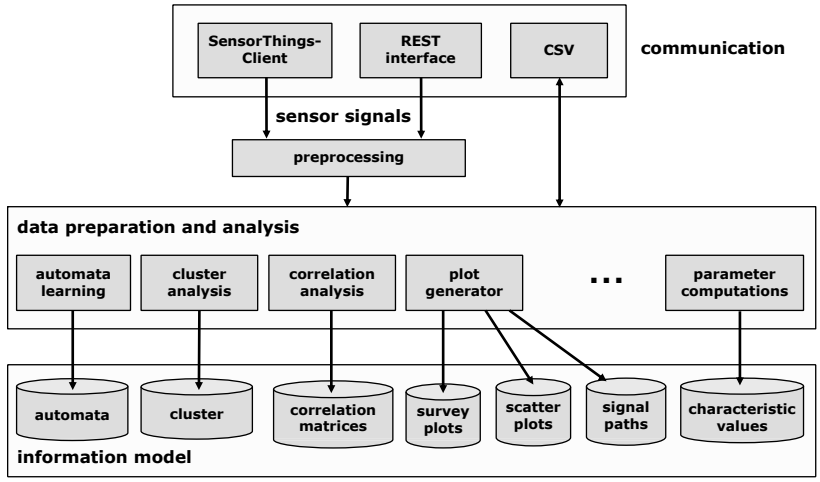

Fig. 3. Tool chain for knowledge extraction

The core of the tool chain consists of distinct ML and statistic methods, which accomplish automatic data extraction (e.g. the extraction of machine learning models such as automata [11], or features such as correlation matrices, data clusters, characteristic values, surveys, or scatter plots). Besides, different interfaces for data acquisition are provided, particularly a client for the OGC SensorThingsAPI, a REST interface and an interface for the offline import of CSV files. The extracted knowledge is integrated in the proposed information model, particularly in the entities *MachineLearningModel* and *Feature*, and can be used for plant optimization.

The proposed tool chain is also used for the automatic classification of unstructured text. In this use case, a data pool with several types of text documents is available, which contains e.g. operation manuals, shift books and repair instructions. In particular, documents from nine classes are available: operation manuals (MNL), glossaries (GLS), tables of contents (CNT), labels (LBL), security instructions (SEC), indexes (IDX), parts lists (PRT), technical data (DAT) and service notes (SRV). A document classification is used to assign an appropriate document type to each document to allow for structured access to the particular document. Based on the document type, the documents are automatically inserted into the *Document* entity of the proposed information model.

Initial evaluations of the document classification have been conducted in [12]. In doing so, K-Nearest-Neighbor classification [7] and Bayesian classification [8] have been investigated for documents of the SmartFactoryOWL, an evaluation platform for cyber-physical production systems (see Fig. 4). Altogether, each classifier has been learned

Fig. 4. SmartFactoryOWL

from 194 documents. Evaluations have been conducted on 66 documents. In this process, 65 of the 66 documents have been correctly classified for the k-NN classifier. For Naive Bayes, only 38 documents have been correctly classified. Furthermore, the k-NN classifier has been observed to significantly outperform the Naive Bayes classifier with respect to computational complexity. The runtime of the k-NN classifier on 66 documents amounts to 11s, while the Naive Bayes classifier requires 70s for the same task. Altogether, the k-NN classifier has been shown to be more suitable for the investigated application case.

4 Conclusion

In this article, a new information model was introduced, which makes it possible to record and present in a structured way the information relevant for the use of machine learning methods in production environments. Furthermore, it was shown that parts of the modeled information such as process models, features or document types can be automatically extracted from the available data. In a next step it is planned to instantiate the information model as well as the tool chain at a glass bending plant and at the SmartFactoryOWL, a demonstration and evaluation platform for cyber-physical production systems.

5 Acknowledgement

This work was partly developed in the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies".

Appendix: Entities of the proposed information model

In this section, the entities of the proposed information model (Figure 2) are described in detail:

- **Component:** In the information model, the entity *Component* represents an object that can either be a single part or a composition of several other components (i.e. tubes, motors, valves). A component can therefore be seen as a part of the overall system. The attributes are (1) name: meaningful name and (2) description: detailed description of the component; (3) metadata: additional information about the component.
- **Location:** The entity *Location* describes the position of the component. This can be a geo position, a postal address or a position of a component inside the production system or somewhere else. The attributes are (1) name and (2) description: meaningful name and description of the location; (3) encodingType: value code of the location, i.e. GeoJSON (application/vnd.geo.json) oder plain text (text/plain) (4) location: value of the location depending on value code.
- **Document:** This entity contains information about documents for a component. The attributes are (1) name: meaningful name of the document; (2) documentType: type of the document, e.g. hand book, service entry or part list (3) encodingType: type of the encoding, e.g. PDF, and (4) content: storage path of the document.
- **ComponentDescription:** The component description contains information about the component, e.g. units, permitted values or schemes for the data transfer in streaming components. The attributes are (1) name, (2) description and (3) value, giving a meaningful name and description to the corresponding value.
- **IEC61360ComponentDescription:** In this entity, component descriptions according to the IEC 61360 norm [2] can be stored. For this, the entity inherits from *ComponentDescription* and contains an additional code, which is used to identify the corresponding IEC 61360 class. Further attributes can be added to the entity according to the IEC 61360 norm.
- **StreamingComponent:** The entity *StreamingComponent* inherits from *Component* and additionally contains the entity *Datastream*. The entity is in particular used to model streaming components like sensors and actors, but it is also suitable for the modeling of users or operators of the production processes.
- **Feature:** This entity is used for a data stream or a streaming component to add further information like a mean value of a data stream, quartile distances or information about the underlying probability distributions. The attributes are (1) name and (2) description: meaningful and description of the feature; (3) encodingType: value code of the feature like double for thresholds, or matrix of double values for histograms; (4) value: value of the feature depending of the defined encodingType and (5) content: path for a more detailed description of the feature.
- **Datastream:** This entity contains data streams, which are captured by a streaming component. The attributes are (1) name and (2) description: meaningful name and description of the data stream; (3) unitOfMeasurement: physical unit of the values being in the *Observation* entity; (4) label: type of the data stream, e.g. faulty or error-free data (5) metadata: additional metadata containing further information about the data stream (6) timeperiod: Time interval in which the observations take place; (7) datatype: datatype of the observations depending on the value code (i.e. double, boolean); (8) samplingPeriod: Used sampling period for the Observations.

- **Observation:** The entity *Observation* contains the process data with the two attributes (1) value: the current value and (2) timestamp: the timestamp of the measurement.
- **MachineLearningModel:** This entity integrates machine learning models. The attributes are (1) encodingType: the encoding determines how the model is stored according to a defined value code (i.e. Predictive Model Markup Language "PMML") and (2) model: the stored model.

References

1. OGC SensorThings API Part 1: Sensing; http://docs.opengeospatial.org/is/15-078r6/15-078r6.html (2016)
2. IEC International Standard 61360: Standard data element types with associated classification scheme (2017)
3. Adolphs, Bedenbender, et al., D.: Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0). Tech. rep., VDI, ZVEI (2015)
4. Bangemann, Bauer, et al., B.: Industrie 4.0 - Technical Assets Grundlegende Begriffe, Konzepte, Lebenszyklen und Verwaltung. Tech. rep., VDI (2015)
5. Bundeswirtschaftsministerium für Wirtschaft und Energie (BMWi): Struktur der Verwaltungsschale - Fortentwicklung des Referenzmodells fuer die Industrie 4.0-Komponente, vol. 1. Spreedruck (2016)
6. Mahnke, W., Leitner, S., Damm, M.: OPC Unified Architecture. Springer Publishing Company, Incorporated, 1 edn. (2009)
7. Runkler, T.A.: Data Analytics Models and Algorithms for Intelligent Data Analysis. Springer Vieweg (2012)
8. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (2003)
9. Schleipen, M., Schick, K., Hövelmeyer, T., Okon, M., Wei, J.: Leitfaden "Interoperable semantische Datenfusion zur automatisierten Bereitstellung von sichtenbasierten Prozessführungsbildern (IDA)". Fraunhofer Verlag (2011)
10. The Data Mining Group: PMML: http://dmg.org/pmml/v4-3 (2019), PMML: http://dmg.org/pmml/v4-3
11. Windmann, S., Lang, D., Niggemann, O.: Learning parallel automata of PLCs. In: International Conference on Emerging Technologies and Factory Automation (ETFA) (2017)
12. Windmann, S., Niggemann, O.: Information Retrieval in Industrial Production Environments. In: International Conference on Emerging Technologies and Factory Automation (ETFA) (2018)