Preprint. To appear in: Proc. IEEE International Conference on Industrial Informatics (INDIN), 2017

Explanation-Aware Feature Selection using Symbolic Time Series Abstraction: Approaches and Experiences in a Petro-Chemical Production Context

Martin Atzmueller
Tilburg University (TiCC)
& University of Kassel (ITeG)
m.atzmuller@uvt.nl

Naveed Hayat and Andreas Schmidt
University of Kassel (ITeG)
Knowledge and Data Engineering Group
{nha, schmidt}@cs.uni-kassel.de

Benjamin Kloepper
ABB Corporate Research
Center Germany
benjamin.kloepper@de.abb.com

***Abstract*—For supporting interpretation, assessment and application of data mining models, explanation-aware methods are crucial. This paper presents an approach for explanation-aware feature selection and assessment using symbolic abstractions of time series. For that, we utilize the symbolic approximate aggregation (SAX) method for data abstraction to be implemented into data mining models. We investigate several approaches and discuss experiences in the context of petro-chemical production.**

I. Introduction

A common process model applied in data analytics is the CRISP-DM model [1] for data mining. It structures the data mining process into the phases *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation* and *deployment*. Ideally, these phases are applied iteratively. A very important phase is the *evaluation* phase, where the constructed data mining models are checked before the models can be deployed. Furthermore, models for which the *deployment* phase involves interactions with the user also requires user assessment of the mined models and their output.

In addition, in data mining typically data exploration during the phases *business and data understanding*, as well as the *data preparation* phase take up 70% to 80% of the project efforts [2]. Usually, for data mining projects for process industry applications, signal data is a very important data type to be assessed. Signal data is produced by sensors (e.g. flow-meters, temperature or pressure sensors, online analyzers) and actors (e.g. valves) and can be understood as time-series data. When extracted from a process information system, the time-series are usually equidistant. Signal data presents several challenges for data mining and analytic purposes. Examples are information obscured by the sheer volume of data [3], missing values, collinearity, data outliers, measurement noise, varying sampling rates [4], high dimensionality, low accuracy, or incorrect data. All these problems are present in the FEE data sets which commonly feature around 1,000 signals.

The objective of the FEE project[1] [5] is to provide support to plant operator by means of big data analytics. Several application scenarios have been identified in workshops with application partners from the petro-chemical industry [6].

[1] http://fee-projekt.de/

Event prediction is a particular important application scenario: It is suitable for events that happen repeatedly in a similar fashion in the plant but for which the exact timing is not known beforehand. The objective of the scenario is to predict a future event with sufficient probability and prediction period (sufficient depends on the specific event) and to give operators more time to react appropriately. This class can be tackled by means of classification or regression, if combined with a suitable alarm logic.

A particular challenge in the petro-chemical application domain is the importance of domain knowledge or even plant-specific knowledge. While domain knowledge can be acquired, plant-specific knowledge is only available from engineers and operators working in the specific plant. Obviously, these persons are no data mining experts. The issues of explanation becomes thus more pressing for all steps within the CRISP-DM process. Then, data analytics needs to be able to confirm decisions during the data mining process with the plant experts and to sufficiently collect their feedback.

Objectives. The transparency of the applied models and their explanation-awareness is a major factor for supporting the user. In particular, if explanations for the complete models, or parts thereof can be provided, then the acceptance of the patterns and their evaluation can often be significantly improved, e. g., [7]. In this context, we consider both explanation-aware models and representations in order to provide for a cost effective feature selection and assessment approach.

Contribution. In this paper, we consider symbolic representations, i. e., decision tree models and sequence representations of time series given by the symbolic aggregate approximation (SAX) [8], [9] as a convenient data abstraction. In a general process model for explanation-aware data analytics, we investigate this abstraction together with a decision tree model in the context of feature selection and assessment, and present a case study in a petro-chemical production context.

The rest of the paper is structured as follows: Section II discusses related work. Next, Section III outlines the proposed approach. After that, Section IV presents results of a case study in the petro-chemical production context. Finally, Section V concludes with a summary and directions for future work.

II. Related Work

A. Explanation-Aware Computing

For software systems the ability to explain reasoning processes and their results ultimately impacts usability and acceptance. In that context, explanation-aware computing (ExaCt) is the vision of software systems being smart in interactions with their users [10]. In a general explanation scenario we can distinguish three main participants [11]: The *user* who interacts with the software system via its user interface, the *originator*, i. e., the problem solver, modeling, or prediction component (in the data mining context), which provides the functionality for the original task of the software, and the *explainer*. In a reconstructive explanation [12], for example, the explainer generates explanations by transforming a trace, e. g., based on output of the originator and/or intermediate results of the problem solving process, into a plausible explanation story. The transformation is an active, complex problem-solving process in itself using additional domain knowledge.

Similarly, for data mining and analytics explanation-aware techniques [7] can be applied for a better explanation of models and discovered patterns for the data analyst. Explanation-aware data mining is especially relevant for establishing trust in the method, e. g., [13] for explanation-aware data mining in the context of pattern mining, inspection and introspection, e. g., [14]. This paper considers explanation-aware approaches for feature selection and assessment using symbolic time series abstractions. To the best of the authors' knowledge, no approach tackling this problem has been proposed so far.

As put forward and described in the *Mining and Analysis Continuum of Explaining* [7] appropriate data representation and abstraction can facilitate explanation-awareness, also supporting and featuring different analysis and presentation levels. Then, data and models can be inspected at different levels of detail, from aggregated representations to the original ones in drill-down fashion combined with appropriate explanation capabilities. Typically, the user starts on an aggregated view that can be refined subsequently, for getting insights into the relations in the data and the constructed model.

As one prominent example of an aggregation and abstraction technique in the area of time series analysis, the symbolic aggregate approximation (SAX) technique [8], [9] is a high-level representation of time series. A key feature of the technique is its symbolic representation by discretizing time series into symbolic strings. This allows dimensionality reduction, data abstraction, effective distance computation and data summarization. These are also the key features that we build upon for enabling explanation-awareness in this paper.

B. Dimensionality Reduction and Feature Selection

High-dimensional data such as signal data (commonly more than 1,000 signals per plants) presents a challenge to existing machine learning, data mining and system identification methods also known as curse of dimensionality [15]. The issue can be addressed during the data preparation phase of CRISP-DM by dimensionality reduction methods.

For soft sensors dealing with signal data, [16] discusses multivariate statistical techniques; e. g., Principal Component Analysis (PCA) and Partial Least Squares (PLS) are used to reduce the dimensionality of process data, by projecting the large amount of original variables onto a lower number of orthogonal latent variables. Kadlec et al. [4] discuss PCA and PLS also for dealing with the collinearity present in signal data. These methods however make it very difficult to understand the process of dimensionality reduction and verify the results with domain or plant-specific knowledge.

Variable selection can be also considered as dimensionality reduction. The idea is to select variables that are most informative for a given learning or data mining problem. Applying variable selection to signal process data is not a simple task. Common methods are the ranking of variables by correlation measures, one-class classifiers, or variable subset selection by wrappers or embedded methods [17]. Correlation measures are expensive due to the required cross-correlation analysis of hundreds of signals and one-class classifiers usually do not yield results suitable for ranking. In addition, [17] give examples where variables with a partial linear correlation actually improve the performance of trained machine learning models. Thus, a filtering approach based on correlation only might result in a not optimal selection of features. Wrapper methods [18] easily become intractable considering the large number of possible process signals [17], [19]. Most suitable appear the embedded methods, where the subset selection happens within the learning algorithm. An important class of algorithms with embedded variable selection are decision trees with corresponding pruning techniques. Decision trees furthermore have the benefit of being comprehensible for humans, in contrast to other embedded machine learning algorithms like artificial neural networks.

C. Decision Tree Classifiers

Decision trees are standard methods for classification, e. g., [20], [21] that build a tree structure with decisions (selections on the attribute domain) on the inner nodes, until a leaf node is reached indicating the respective classification. Decision trees feature a fast classification performance, and can be applied both for categorical as well as numeric features. Furthermore, they provide rather interpretable models (for humans) since the tree structure (paths, nodes) can be directly inspected. Also, utilizing the structure, the importance of individual features for the classification task can be assessed.

In addition to categorical data, an important subclass of classification tasks also addresses class labels which can be (partially) ordered. For this ordinal data, there exist specific decision tree classification methods, e. g., the classification trees for ordinal responses method [22] as implemented in the *rpartScore* software package [23]. The algorithm is an adaptation of the standard CART [24] algorithm for building classification and regression trees. In this paper, we apply the rpartScore algorithm in our industrial application context for demonstrating its explanation capabilities and potential for feature selection and model understanding.

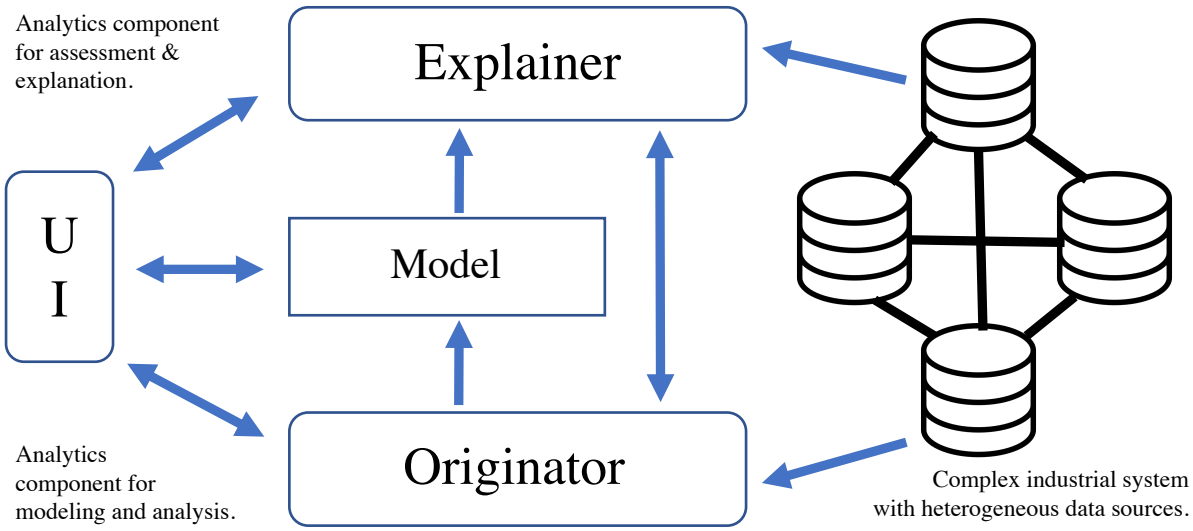

Fig. 1. **Overview – General process model for Explanation-Aware Analytics**

III. Method

Below, we describe the general process model and exemplary components for explanation-aware analytics. The process can be seen as a special instantiation of a general explanation-aware scenario in an industrial data analytics context. We first present a bird's eye view on the overall process. After that, we focus on the task of feature selection and assessment for which we briefly summarize the basics of symbolic abstraction and decision tree methods applied in our case study.

A. Process Model

In a general explanation-aware data analytics system, the user interacts via the user interface which then accordingly interfaces with the originator, the constructed model, and the explainer. For our general industrial application context, Figure 1 provides an overview on the general process. As described above, the originator targets the problem solving task, e. g., a predictive modeling or pattern detection task, while the explainer focuses on generating appropriate explanations, e. g., for making the processes of the originator more transparent, to increase trust in the decisions taken, or to support assessment and validation processes. For that, e. g., the generated artefacts such as a data mining model can be exploited.

In our application context of time series analytics and feature assessment, we primarily facilitate the inspection of the built model using the SAX and decision tree representation described below. We utilize the features of the originator and the constructed model for providing indications on the importance of features. In this way, the available feature set and the model can be inspected in detail, and important influence factors can be identified (or even uncovered). For a cost effective approach we can choose a relatively simple but interpretable model in the early phase of a CRISP-DM project for identifying important features, while the process model allows more advanced models as well which can be applied in later phases.

B. Feature Selection and Model Assessment

Feature selection targets the construction process of the model as well as its assessment by the domain specialist, by explicitly identifying the important features. Then, for an in-depth assessment of the model, for example, important features can be inspected on the symbolic as well as on the (original) time series level in order to validate the model before being deployed. Furthermore, the model itself can potentially be simplified using the reduced feature set, or alternative modeling approaches can be explored.

C. Symbolic Approaches

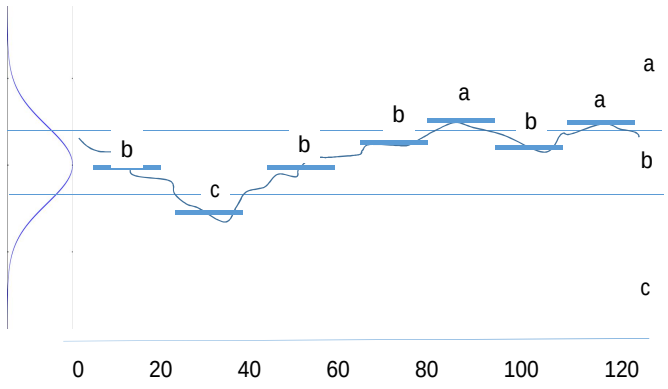

Fig. 2. **Example SAX transformation of a time-series with alphabet a-b-c (adapted from [9]). The x-axis shows the time series that is partitioned by segments of 20 time units. The segments are mapped to the alphabet on the y-axis; splits are derived by distributional analysis, cf. [9].**

The symbolic aggregate approximation (SAX) technique is a high-level representation of time series. This symbolic representation is obtained by discretizing time series into symbolic strings. That is, for a given alphabet size, the time series is split into different time windows, for example, with a length of one minute each.

To summarize, the time series is first transformed by piecewise approximate aggregation, dividing the time series into different time windows (segments). After that, each segment is replaced by the mean value of the contained values. For discretization, each such preprocessed segment is subsequently mapped to a given alphabet value. Figure 2 shows an illustrative example. Due to the limited space, we refer to Lin et al. [9] for an in-depth discussion of SAX.

The SAX representation can be simply integrated into a machine learning algorithm for categorical (symbolic) data, e. g., for a decision tree learner. Figure 3 shows an illustrating example of a decision tree: This tree has been constructed using the classification trees for ordinal responses (rpartScore) approach, and accordingly targets an ordinal class variable, utilizing SAX-based features. Using the tree, we can classify new data by traversing a path through the tree according to the features observed in our data until we reach a leaf node which provides the classification. In particular, the left branch is taken, if the condition on the inner node matches, while the right branch is chosen otherwise.

So, in the example shown in Figure 3, for the root node the right branch (with the next condition $P6315 = a$) is taken, if the observed value of the attribute *FI6316* of the new data tuple to be classified is not contained in the set $\{a, b, c, d, e, f, g, h, o, p, q, r, s, t\}$. The labels on the leaf node indicate the ordered class labels.

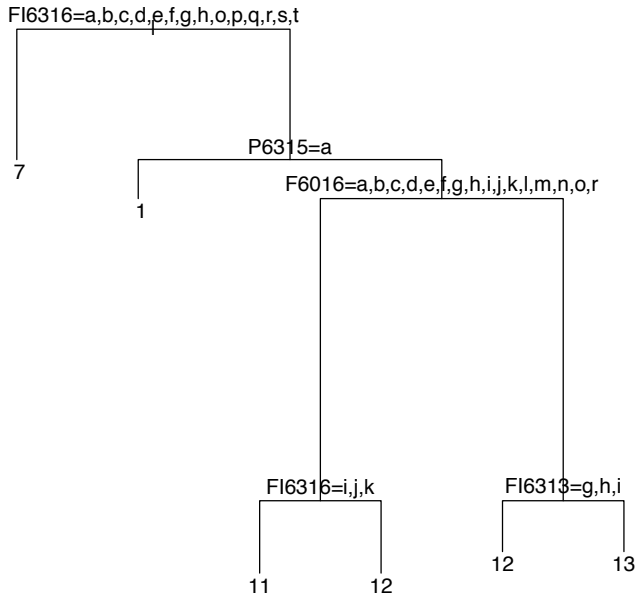

Fig. 3. **Illustrative example of a decision tree classifier built using the rpartScore method using symbolic features and an ordinal class (response) variable. For classification of new data, the tree is traversed starting from the root node; the respective left branch is taken, if the condition on the inner node matches, while the right branch is chosen otherwise, until a leaf node is reached. The classification is then indicated on the respective leaf node.**

IV. Case Study

Below, we provide a case study using a real-world dataset from a petro-chemical production context. We first describe the applied dataset before we present our experiments and discuss our results in detail.

A. Dataset

In the context of the FEE project, we obtained a dataset from one of the application partners, i. e., data from a petro-chemical plant. The dataset contained measurements of different sensors from April to June 2014, and January to March 2016. In particular, the dataset contains so-called foaming events, that occurred in 2014 and 2016, which we target for classification.

For our experiments, we split the dataset by year and obtained two partitions of equal size for 2014 with 130024 instances and 2016 with 129876 instances, respectively. As features, we utilized 195 time series of sensor measurements (signals) that were present in both datasets (2014 and 2016). The data was preprocessed by computing the SAX representation for each signal, yielding 20 classes (symbols: 'a' - 't').

B. Results and Discussion

In our industrial application context, we instantiated the process model with a decision tree classifier for obtaining the model since the embedded approach seems most suited for the process industries: First, all signals can be utilized, and information gain is obtained also for signals with collinearities, since collinearity is considered implicitly. Second, a model (result) can always be obtained, which features introspection and results in explanation-aware fashion. Third, the approach is modular and scalable using available Big data implementations, e. g., [25].

For decision tree construction, we applied the rpartScore algorithm [23] using squared differences in scores for estimating the misclassification costs in splitting, and the total misclassification rate in pruning. For the complexity parameter it of the algorithm, we empirically determined the value $it = 0.38$ using the automatic estimation method of rpartScore during training.

The process engineer provided us with background knowledge about a specific signal, namely *P6315A*, which is correlated with our targeted class, i. e., the foaming event, but so far the technical reasons for the correlation are unknown. Therefore, we used our method in order to identify other relevant signals in the dataset. As an indication of a foaming event, we the used *P6315A* signal together with a threshold as a class label and the other 195 signals as features for the classification model. As a metric for the model performance, we used the Kendall-tau-b metric to measure the correlation on the held out data in terms of changed labels.

Therefore, for all pairs test instances, we counted the concordant pairs in terms of a higher label, e. g. $a \rightarrow c$ in the model prediction and $a \rightarrow e$ in the held out data. Discordant pairs were counted in terms of opposite label directions, e. g. $c \rightarrow e$ in the model prediction and $c \rightarrow a$ in the held out data.

TABLE I
RESULTS OF THE DECISION TREE CLASSIFIER USING DIFFERENT COMBINATIONS OF THE 2014 DATA (D2014) AND THE 2016 DATA (D2016) FOR TRAINING AND TEST, RESPECTIVELY. THE KENDALL-TAU-B SCORE (*Kendall*) IS USED AS A PERFORMANCE MEASURE.

Model Train / Test (CV)	Kendall
D2014 / D2014	0.83
D2016 / D2016	0.80
D2014 / D2016	0.20
D2016 / D2014	0.28

First, we performed 10-fold cross validation on each partition individually, to measure the model quality by year. Then, for each year, the best model of the cross validation was applied to the respective other year, to measure the generalization of the model. The results are shown in Table I. We observe, that although the correlation within the same year is relatively high ($\geq 80\%$), switching the models, i. e., applying the 2014 model on the 2016 data and vice versa does not perform as well. This indicates, that the data for 2014 and 2016 have different characteristics, that were captured by the model, i. e., that there is some sort of "concept drift" which can be caused, for example, by different load profiles of the plant or other modifications in the production process.

TABLE II
RESULTS: TOP 20 SIGNALS OF THE BEST PERFORMING MODEL IN CROSS-VALIDATION FOR THE 2014 AND THE 2016 MODEL, RESPECTIVELY. THE SIGNALS MARKED WITH (*) ARE PART OF THE SUBPLANT AFFECTED BY FOAMING, WHILE THE SIGNALS IN BOLD ARE CONTAINED IN THIS SELECTION BOTH FOR 2014 AND 2016.

2014		2016	
Signal	Score	Signal	Score
FI6316*	100.0	**FI6313**	100.0
FV6316	93.0	**FI6316***	77.91
LV6308	91.78	T6021	68.36
T6314	91.71	**FV6316**	64.19
T6315*	79.66	T6335	61.80
TV6317	69.85	PC6601B	52.68
P6315*	25.56	T6321	39.77
U6212	25.51	L6304	26.58
TI6410	24.46	FI6311	26.09
TC6402	16.48	T6266	25.96
TI6402	16.38	T6260	25.89
P6220	8.48	L6309A	22.68
F6016	7.59	F6310	13.20
PC6001	5.05	FI6303	11.33
P6005	4.56	L6017	10.88
F6403B	4.49	F6235	10.54
F6403A	4.43	FC6235	10.50
FI6313	3.27	FV6224	6.24
FP6630A	2.31	F6319	5.06
FC6235	1.84	PV6316A	5.04

As indicated in the table, we see rather different profiles of the features. The abstraction capability of this representation can be observed in Figure 3 indicating a strongly pruned decision tree for the 2014 model during cross validation. Here, we observe the most important feature (*FI6316*) at the root node, while the features up to the contained feature *P6315* are slightly less important alternatives for the first condition that were considered in the construction of the tree.

TABLE III
RESULTS OF THE ROBUSTNESS EXPERIMENT: RESULTS OF DECISION TREE CLASSIFIERS LEARNED FROM DIFFERENT SAMPLE SHARES FROM THE RESPECTIVE 2014 AND 2016 DATASETS, AND THE RESPECTIVE KENDALL-TAU-B (*Kendall*) PERFORMANCE SCORES.

Model: Train / Test (CV)	Kendall
10% D2014, 90% D2016 / 90% D2014, 10% D2016	0.79
20% D2014, 80% D2016 / 80% D2014, 20% D2016	0.79
30% D2014, 70% D2016 / 70% D2014, 30% D2016	0.80
40% D2014, 60% D2016 / 60% D2014, 40% D2016	0.85
50% D2014, 50% D2016 / 50% D2014, 50% D2016	0.85
60% D2014, 40% D2016 / 40% D2014, 60% D2016	0.82
70% D2014, 30% D2016 / 30% D2014, 70% D2016	0.81
80% D2014, 20% D2016 / 20% D2014, 80% D2016	0.85
90% D2014, 10% D2016 / 10% D2014, 90% D2016	0.77

As outlined above, the individual models for 2014 and 2016 did not score well on the respective complementary dataset, taking the full dataset into account. Therefore, we performed another experiment in order to assess the robustness of the proposed symbolic approach using SAX and the rpartScore ordinal decision tree classifier for trend detection. For that, we constructed mixed (train and test) datasets by sampling the 2014 and 2016 datasets with varying proportions from 10% to 90% and 90% to 10%, respectively. Thus, we trained our model on this mixed data containing portions of both the 2014 and 2016 data characteristics, and applied it on the respective (mixed) hold-out sets that would have also been encountered in a typical cross-validation setting. Table III shows the partitioning of the data and the respective results of the individual runs.

As we can observe in the table, the proposed method is relatively robust, since the obtained accuracies are in line, or even better than the performance on the respective original datasets. While these increases in predictive performance were not statistically significant, nevertheless we get some confirmation on the appropriate selection of our classification method in our process model since the instantiation with the decision tree approach proved rather adequate in all demonstrated settings with robust results. Thus, our results support the applicability of our proposed method, since it could construct robust models even in the (large) presence of noise.

Overall, our results indicate, that the proposed approach using SAX and ordinal decision trees is able to capture the dependencies between the signal features and the target (prediction) variable indicating the foaming event. We were able to capture the trends in our model, and experiments with noisy data (by mixing different partitions of the data) showed the robustness of the approach. Furthermore, the resulting decision trees were typically rather small (see Figure 3 for an example), and thus easy to interpret. Thus, the applied models are well suited for inspection and feature selection, since the decision tree provides a clear view on the important factors, as a cost effective method for feature selection. The factors can then provide information or justification to the process engineer or support the interpretability of the applied method. Furthermore, they can also be used in further engineering of other predictive approaches.

V. Conclusions

In this paper, we considered both explanation-aware models and representations in order to provide for a cost effective feature selection and assessment approach. We proposed an explanation-aware process model for data analytics, and showed its instantiation for feature selection and assessment in the industrial domain. Specifically, we considered symbolic representations, i. e., decision tree models and sequence representations of time series given by the symbolic aggregate approximation (SAX) as a convenient data abstraction. In a case study in a petro-chemical production context, we investigated the proposed process model and its instantiation in the context of feature selection and assessment.

Our results indicate the applicability of the proposed approach in the industrial context. The method is able to capture the important dependencies as indicated by the applied trend detection technique regarding the anomalous event. Furthermore, the applied method proved rather robust when using different samples of different situations as could be observed for our applied datasets. In addition, since collinearity is considered implicitly in the selected decision tree classifier we were able to utilize all signals for the feature selection which was also reflected in the results regarding the feature importance. Altogether, the method provides a cost effective option for feature selection and assessment in process industries. Due to its modularity, simple models can be utilized in an early stage of the data mining process, while they can be refined subsequently also including (intermediate) results of the process later, as needed.

For future work, we aim to extend the approach towards a more comprehensive coverage of the CRISP-DM process, such that, e. g., also the modeling phase can be supported by explanation-aware techniques. Here, both appropriate modeling methods as well as data representations and introspection methods [7], [26] are relevant starting points. Here, especially the refinement of Big Data into Smart Data, and its utilization in process industries [27] requires adequate modeling and explanation capabilities. Also, we aim to extend and evaluate the approach towards Big data implementations, e. g., [25] for providing an effective and scalable approach for large-scale analytics. The connection with graph-based (semantic) data representations [5], [28] and according data mining methods [29] is another interesting option for future research.

Acknowledgment

This work was funded by the BMBF project FEE under grant number 01IS14006.

References

[1] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, *CRISP-DM 1.0: Step-by-Step Data Mining Guide*, CRISP-DM consortium: NCR Systems Engineering, DaimlerChrysler AG, SPSS Inc. and OHRA Verzekeringen en Bank Groep B.V, 2000.

[2] T. Dasu and T. Johnson, *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, 2003, vol. 479.

[3] D. Flynn, J. Ritchie, and M. Cregan, "Data Mining Techniques Applied to Power Plant Performance Monitoring," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 369–374, 2005.

[4] P. Kadlec, B. Gabrys, and S. Strandt, "Data-Driven Soft Sensors in the Process Industry," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.

[5] M. Atzmueller, B. Kloepper, H. A. Mawla, B. Jäschke, M. Hollender, M. Graube, D. Arnu, A. Schmidt, S. Heinze, L. Schorer, A. Kroll, G. Stumme, and L. Urbas, "Big Data Analytics for Proactive Industrial Decision Support: Approaches & First Experiences in the Context of the FEE Project," *atp edition*, vol. 58, no. 9, 2016, (In Press).

[6] B. Klöpper, M. Dix, L. Schorer, A. Ampofo, M. Atzmueller, D. Arnu, and R. Klinkenberg, "Defining Software Architectures for Big Data Enabled Operator Support Systems," in *Proc. IEEE International Conference on Industrial Informatics. IEEE, Boston, MA, USA*, 2016.

[7] M. Atzmueller and T. Roth-Berghofer, "The Mining and Analysis Continuum of Explaining Uncovered," in *Proc. SGAI International Conference on Artificial Intelligence (AI-2010)*, London, UK, 2010.

[8] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," in *Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. New York, NY, USA: ACM, 2003, pp. 2–11.

[9] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: A Novel Symbolic Representation of Time Series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[10] T. Roth-Berghofer, S. Schulz, D. Leake, and D. Bahls, "Explanation-Aware Computing," *AI Magazine*, vol. 28, no. 4, 2007.

[11] T. R. Roth-Berghofer and M. M. Richter, "On Explanation," *Künstl. Intelligenz*, vol. 22, no. 2, pp. 5–7, May 2008.

[12] M. R. Wick and W. B. Thompson, "Reconstructive Expert System Explanation," *Artif. Intell.*, vol. 54, no. 1-2, pp. 33–70, 1992.

[13] K. Behrenbruch, M. Atzmueller, C. Evers, L. Schmidt, G. Stumme, and K. Geihs, "A Personality Based Design Approach Using Subgroup Discovery," in *Human-Centred Software Engineering*, ser. LNCS. Heidelberg, Germany: Springer Verlag, 2012, vol. 7623, pp. 259–266.

[14] M. Atzmueller, J. Baumeister, A. Hemsing, E.-J. Richter, and F. Puppe, "Subgroup Mining for Interactive Knowledge Refinement," in *Proc. AIME*, ser. LNAI 3581. Heidelberg: Springer, 2005, pp. 453–462.

[15] S. Alelyani, J. Tang, and H. Liu, "Feature Selection for Clustering: A Review." *Data Clustering: Algorithms and Applications*, vol. 29, pp. 110–121, 2013.

[16] B. Lin, B. Recke, J. K. Knudsen, and S. B. Jørgensen, "A Systematic Approach for Soft Sensor Development," *Computers & Chemical Engineering*, vol. 31, no. 5, pp. 419–425, 2007.

[17] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *JMLR*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[18] R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[19] G. Doquire, G. De Lannoy, D. François, and M. Verleysen, "Feature Selection For Interpatient Supervised Heart Beat Classification," *Computational Intelligence and Neuroscience*, vol. 2011, p. 1, 2011.

[20] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[21] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[22] P. R, "Classication Trees for Ordinal Variables," *Computational Statistics*, vol. 23, pp. 407–427, 2008.

[23] G. Galimberti, G. Soffritti, M. D. Maso *et al.*, "Classification trees for ordinal responses in R: the rpartScore package," *Journal of Statistical Software*, vol. 47, no. i10, 2012.

[24] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[25] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, "MLLib: Machine Learning in Apache Spark," *JMLR*, vol. 17, no. 34, pp. 1–7, 2016.

[26] M. Atzmueller, J. Baumeister, and F. Puppe, "Introspective Subgroup Analysis for Interactive Knowledge Refinement," in *Proc. FLAIRS*. Palo Alto, CA, USA: AAAI Press, 2006, pp. 402–407.

[27] J. Folmer, I. Kirchen, E. Trunzer, B. Vogel-Heuser, T. Pötter, M. Graube, S. Heinze, L. Urbas, M. Atzmueller, and D. Arnu, "Challenges for Big and Smart Data in Process Industries," *atp edition*, vol. 01-02, 2017.

[28] P. Ristoski and H. Paulheim, "Semantic Web in Data Mining and Knowledge Discovery: A Comprehensive Survey," *Web semantics: Science, Services and Agents on the World Wide Web*, vol. 36, pp. 1–22, 2016.

[29] M. Atzmueller, S. Doerfel, and F. Mitzlaff, "Description-Oriented Community Detection using Exhaustive Subgroup Discovery," *Information Sciences*, vol. 329, pp. 965–984, 2016.