Data Preparation for Big Data Analytics: Methods & Experiences

Martin Atzmueller[1], Andreas Schmidt[1], Martin Hollender[2]

[1]*University of Kassel, Research Center for Information System Design, Germany*
[2]*ABB Corporate Research Center, Germany*

ABSTRACT

This chapter provides an overview of methods for preprocessing structured and unstructured data in the scope of Big Data. Specifically, this chapter summarizes according methods in the context of a real-world dataset in a petro-chemical production setting. The chapter describes state-of-the-art methods for data preparation for Big Data Analytics. Furthermore, the chapter discusses experiences and first insights in a specific project setting with respect to a real-world case study. Furthermore, interesting directions for future research are outlined.

Keywords: Big Data Analytics, Data Mining, Data Preprocessing, Industrial Production, Industry 4.0

INTRODUCTION

In the age of the digital transformation, data has become the fuel in many areas of research and business - often it is already regarded as the fourth factor of production. Prominent application domains include, for example, industrial production, where the technical facilities have typically reached a very high level of automation. Thus, many data is typically acquired, e.g., via sensors, in alarm logs or entries into production management systems regarding currently planned and fulfilled tasks. Data in such a context is represented in many forms, e.g., as tabular metric data, also including time series. In the latter example, this data can be structured according to time and different types of measurements. With respect to textual data collected in logs or production documentation, however, we can easily see that this data does not exhibit the rich structure as in the case of the sensor data. Therefore, this unstructured data first needs to be transformed into a data representation that exhibits a higher degree of structuring, before it can be utilized in the analysis. However, this is also true for structured data, since metric data, for example, can also contain falsely recorded measurements leading to outliers and non-plausible values. Therefore, appropriate data preprocessing steps are necessary in order to provide for a consolidated data representation, as outlined in the data preparation phase of the Cross Industry Standard Process for Data Mining (CRISP-DM) process model (Shearer, 2000).
This chapter discusses state-of-the-art approaches for data preprocessing in the context of Big Data and reports experiences and first insights about the preprocessing of a real world dataset in a petro-chemical production setting. We start with an overview on the project setting, before we outline methods for processing structured and unstructured data. After that, we summarize experiences and first insights using the real-world dataset. Finally, we conclude with a discussion and present interesting directions for future research.

Preprint of Atzmueller, M., Schmidt, A., Hollender, M. (2016) Data Preparation for Big Data Analytics: Methods & Experiences. In: Enterprise Big Data Engineering, Analytics, and Management, IGI Global (In Press)

CONTEXT

Know-how about the production process is crucial, especially in case the production facility reaches an unexpected operation mode such as a critical situation. When the production facility is about to reach a critical state, the amount of information (so called shower of alarms) can be overwhelming for the facility operator, eventually leading to loss of control, production outage and defects in the production facility. This is not only expensive for the manufacturer but can also be a threat to humans and the environment. Therefore, it is important to support the facility operator in a critical situation with an assistant system using real-time analytics and ad-hoc decision support.

The objective of the BMBF-funded research project “Frühzeitige Erkennung und Entscheidungsunterstützung für kritische Situationen im Produktionsumfeld”[1] (short FEE) is to detect critical situations in production environments as early as possible and to support the facility operator with a warning or even a recommendation how to handle this particular situation. This enables the operator to act proactively, i.e., before the alarm happens, instead of just reacting to alarms.

The consortium of the FEE project consists of several partners, including application partners from the chemical industry. These partners provide use cases for the project and background knowledge about the production process, which is important for designing analytical methods. The available data was collected in a petrochemical plant over many years and includes a variety of data from different sources such as sensor data, alarm logs, engineering- and asset data, data from the process-information-management-system as well as unstructured data extracted from operation journals and operation instructions (see Figure 1). Thus, the dataset consists of various different document types. Unstructured / textual data is included as part of the operation instructions and operation journals. Knowledge about the process dependencies is provided as a part of cause-effect-tables. Information about the production facility is included in form of flow process charts. Furthermore, there is information about alarm logs and sensor values coming directly from the processing line.

METHODS

In this chapter, we share our insights with the preprocessing of a real world, industrial data set in the context of big data. Preprocessing techniques can be divided into methods for structured and unstructured data. Different types of preprocessing have been proposed in the literature and we will give an overview of the state-of-the-art methods. We first give a brief description of the most important techniques for structured data. After that, we focus on preprocessing techniques for unstructured data, and provide a comprehensive view on different methods and techniques with respect to structured and unstructured data. Specifically, we also target methods for handling time-series and textual data, which is often observed in the context of Big Data. For several of the described methods, we will briefly discuss examples for special types of problems that need to be handled in the data preparation phase for Big Data analytics, by sharing some experiences in the FEE project. In particular, this section focuses on the Variety dimension concerning Big Data - thus we do not specifically consider Volume but mainly different data representations, structure, and according preprocessing methods.

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

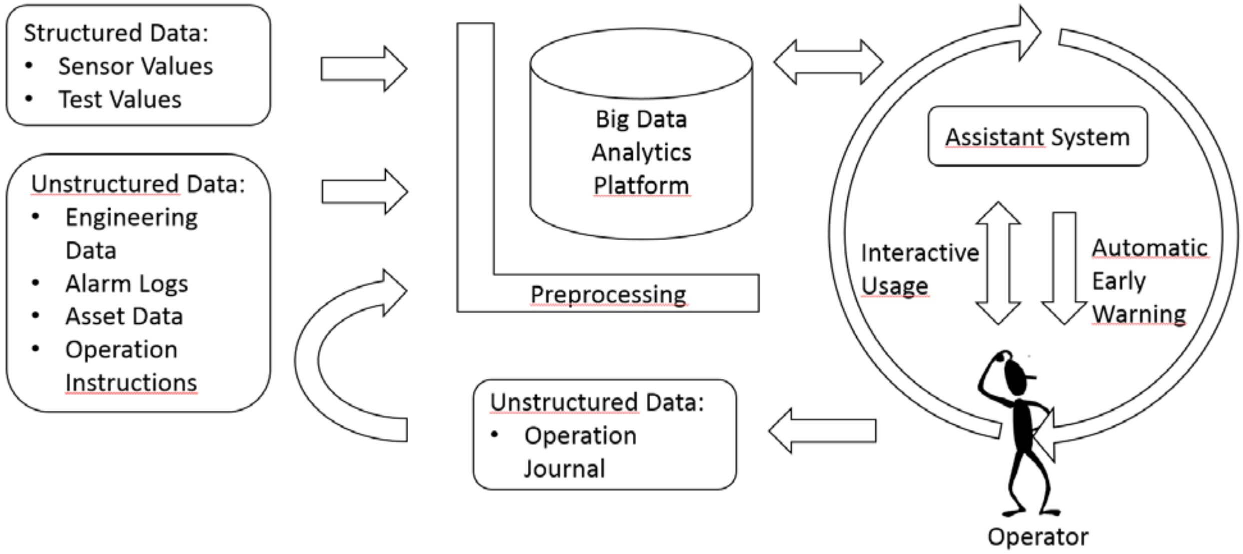

Figure 1. In the FEE project, various data sources from a petrochemical plant are preprocessed and consolidated in a big data analytics platform in order to proactively support the operator with an assistant system for an automatic early warning.

Preprocessing of Structured Data

Preprocessing techniques for structured data have been widely applied in the data mining community. Data preparation is a phase in the CRISP-DM standard data mining process model (Shearer 2000) that is regarded as one of the key factors for good model quality. In this section, we give a brief overview of the most important techniques that are widely used in the preprocessing of structured data.

When it comes to the application of a specific machine learning algorithm, one of the first steps in data preparation is to transform attributes to be suitable for the chosen algorithm. Two well-known techniques that are widely used are numerization and discretization: Numerization aims at transforming non-numerical attributes into numeric ones, e.g. for machine learning algorithms like SVM and Neural Networks. Categorical attributes can be transformed to numeric ones by introducing a set of dummy variables. Each dummy variable represents one categorical value and can be one or zero meaning the value is present or not. Discretization takes the opposite direction by transforming non-categorical attributes into categorical ones, e.g. for machine learning algorithms like Naive Bayes and Bayesian Networks. An example for discretization is binning, which is used to map continuous values to a specific number of bins. The choice of bins has a huge effect on the machine learning model and therefore manual binning can lead to a significant loss in modeling quality (Austin and Brunner 2004).

Another widely adopted method for improving the numerical stability is the centering and scaling of an attribute. By centering the attribute mean is shifted to zero while scaling is transforming the standard deviation to one. By applying this type of preprocessing, multiple attributes are transformed to a common unit. This type of transformation can lead to significant improvements in the model quality especially for outlier sensitive machine learning algorithms like k-nearest neighbors. Modeling quality can also be affected by skewness in the data. Two data transformations that reduce the skewness are Box and Cox (1964), and Yeo and Johnson (2000). While Box and Cox is only applicable for positive numeric values, the approach by Yeo and Johnson (2000) can be applied to all kinds of numerical data.

The transformations described so far are only affecting individual attributes, i.e., the transformation of one attribute does not have an effect on the value of another attribute. They can also be applied to a subset of the available attributes. In contrast to that there also exist data transformations that are affecting multiple attributes. The spatial sign (Serneels et al. 2006) transformation is well known for reducing the effect of outliers by projecting the values to a multi-dimensional sphere.

Another data preprocessing technique that is having an effect on multiple attributes is feature extraction. A variety of methods have been proposed in literature and we will only name Principle Component Analysis (Hotelling 1933), short PCA, as the most popular one. PCA is a deterministic algorithm that transforms the data into a space where each dimension (Principle Component) is orthogonal, i.e., not correlated, but still captures most of the variance of the original data. Typically, PCA is applied to reduce the number of dimensions by using a cutoff for the number of Principle Components. PCA can only be applied to numerical data, which is typically centered and scaled beforehand.

Another popular preprocessing method for reducing the number of attributes is feature reduction. It is apparent that attributes with variance close to zero are not helping to separate the data in the machine learning model. Therefore, attributes with variance near zero are often removed from the dataset. Highly correlated attributes capture the same underlying information and can therefore be removed without compromising the model quality. Feature reduction is typically used to decrease computational costs and support the interpretability of the machine learning model. A special case of feature reduction is feature selection where a subset of attributes is selected by a search algorithm. All kinds of search and optimization algorithms can be applied and we will only name Forward Selection and Backward Elimination. In Forward Selection, search starts with one attribute adding one attribute at a time as long as model quality improves with respect to an optimization criterion. Backward Elimination has the same greedy approach starting with all attributes removing one attribute at a time. In addition to feature reduction, the feature selection method has also the motivation of preventing overfitting by disregarding a certain amount of information.

Finally yet importantly, feature generation is a preprocessing technique for augmenting the data with additional information derived from existing attributes or external data sources. Of all the presented methods feature generation is the most advanced one, because it enables the induction of background knowledge into the model. Complex combination of the data has been considered in Forina et al. (2009).

So far, only the preprocessing of attributes has been covered. When it comes to the attribute values, there is a lot of effort in order to eliminate missing values. The most obvious approach is to simply remove the respective attribute, especially when the fraction of missing values is high. In the case of numeric data, another approach is to "fill in" missing values utilizing the attribute mean, which is not changing the centrality of the attribute. Approaches that are more sophisticated use a machine learning model to impute the missing values, e.g., by using a k-nearest neighbors model (Troyanskaya et al. 2001). Alternatively, one can also not address the missing value problem and simply select a machine learning model that can deal with missing values, e.g., Naïve Bayes and Bayesian Networks.

In the case of supervised learning, one can also face the problem of unevenly distributed classes leading to an overfitting of the model to the most frequent classes. Popular methods for balancing the class distribution are under- and over-sampling. When performing under-sampling the number of the frequent classes is decreased. The dataset gets smaller and the distribution of classes becomes more similar. In contrast to that over-sampling is increasing the number of infrequent classes by replication. The dataset gets bigger and again the distribution of classes becomes more similar. This problem can also be addressed in the training phase of the machine learning model by using instance-weighting. Instance weighting is a technique for dealing with unevenly distributed classes by introducing a penalty for misclassification giving the infrequent classes more weight.

Preprocessing of Time Series Data

Numerical process values like flows, pressures and temperatures are typically digitized with a limited resolution like, for example, 16 bit. Sampling rates of one value per second are well mastered by today's Distributed Control Systems (DCS). Some processes like metal rolling mills require faster sampling rates whereas in other areas one value in ten minutes might be sufficient. Modern smart instruments implement sophisticated self-diagnosis mechanisms telling about the quality and reliability of the measured signal. It is almost certain to assume that in a larger plant with many thousands of sensors, some of the sensors will deliver wrong signals. It is the task of the maintenance department to make sure that all sensors are well calibrated and properly functioning, but some of the sensors might not have highest priority or the next scheduled maintenance might be relatively far away.

Usually the operators and the plant engineers will make sure that the key signals they monitor in their trend displays are available in high quality. However, data analysts are often also interested in other signals that were not in the focus so far. Quite often the deadbands for exception-based storage are chosen much too wide for such signals (the next value is only stored once the deadband is left) (Hollender, 2010). Consequently, some parts of the signals have been filtered out and the signal might not be suitable for the intended analysis (Thornhill 2004). If such problems are discovered, the configured deadbands need to be optimized and new data needs to be collected with the new settings. If large amounts of data are required, this might mean to wait several months until the new data is available.

Typical pre-processing problems for the data analyst include:

1. Outliers (e.g. the signal jumps to maximum value for several samples and then returns to the previous range). Many algorithms for outlier detection and removal exist (Liu 2004).
2. Frozen signals (signal stops moving, typically the current value will stay at the last known good value in case of an error). After maintenance has fixed the problem, the signal starts moving again. The intervals where a signal is frozen need to be identified and excluded from the analysis.
3. Noise like electromagnetical interference needs to be filtered out. Low pass and Median filters can be used for this.
4. Sampling rate is either too high (unnecessary calculation load) or too low (not enough information contained in the signals). The Nyquist-Shannon sampling theorem says that the sampling frequency should be twice the highest frequency of interest. This also means that for slow phenomena in the area of days or weeks it does not make sense to work with one second samples. A typical down sampling algorithm is to take the average of the values inside an interval.
5. Exception-based sampling means that the distance between samples varies. Most analysis algorithms require equidistant sampling rates. An interpolation is required to get to an equidistant sampling rate

Many big data algorithms assume 100% clean and valid data. One single outlier can completely destroy complex calculations. It is therefore very important to have an adequate preprocessing in place to either remove or at least identify intervals with measurement problems.
It is very important to carefully take the quality attributes of a signal into account. For example these attributes can contain the results of a self diagnosis in an intelligent field device or if a cable is broken. However, a less than perfect quality attribute does not always mean that a signal cannot be used because it might only be indication that a maintenance should be performed.
Data reconciliation (Crowe 1996) uses mathematical models of the process to discover and remove errors in the measured signals.

Preprocessing of Unstructured Data

Structured Data deals with data in a pre-defined fixed format. In contrast to that, unstructured data involves free formats, e.g., the text that an operator has written into a digital operating journal. Techniques for the preprocessing of unstructured data have been proposed in the research areas of Information Retrieval and Natural Language Processing and have been widely adopted in the Text Mining community.

Typically, unstructured data is organized into multiple documents containing free text. The preprocessing of unstructured data starts with a tokenization step. For each document, the text is cleaned by removing non-word characters, e.g., punctuation and special characters, and then split at each whitespace to create a set of words for the document. The union of all words in the document collection yields the dictionary.

One of the most commonly used representations for a document is the bag-of-words model. In this model, a document is represented by a multiset of words, i.e., a set of words with corresponding frequencies. The order of the words is not taken into account, which means that this type of model is not able to capture the relation between words, e.g. the co-occurrence of words in a sentence.

A popular implementation of the bag-of-words model is the vector space model (Salton 1975) where each document is represented by vector containing weighted frequencies of the words. Each dimension of the vector corresponds to a term and the value of the vector component is given by a weighting scheme. Multiple types of weighting schemes have been proposed in literature and two of the most popular ones are TF (Luhn 1957), short for term frequency, i.e., giving a high weight to frequent terms, and TF-IDF (Sparck 1972), short for term frequency combined with inverted document frequency, i.e., giving a high weight to frequent terms that only occur in few documents. Finally, a document-term-matrix is obtained by using the document vectors as rows in the matrix. The TF can be computed in a single run having linear time complexity. For IDF a second run over all terms is required. With D documents, T terms and the use of appropriate data structures TF-IDF can be computed in O (D+T), making it applicable for big data applications.

When working with unstructured data, especially with text data, one has to deal with two types of problems:

1. High dimensionality of the data: With a vector representation, each term in the dictionary becomes a dimension in a vector space. Typically, the dictionary size is large and thus the vector space has a high dimensionality.

2. Sparsity of the data: The distribution of terms corresponds to a power law distribution. There are a few, very frequent terms while most of the terms occur only a single time. This so-called “long tailed” distribution results in a high sparsity, because most of the vector components will be zero when words are absent.

In order to deal with these two types of problems techniques have been proposed to reduce the size of the dictionary, either by removing unimportant terms, e. g. words that do not hold any information in the domain context, or by mapping semantically related words to a common notation, i.e., equivalence classes of terms.

When it comes to the application of machine learning algorithms to unstructured data, one has to deal with upper and lower case terms. One way to model case-insensitivity is to simply substitute all upper case letters according to their lower case counterpart. This operation is appropriate for big data applications, because it can be performed in linear time.

Another popular preprocessing technique coming from the information retrieval community is the filtering of stop words (Rijsbergen 1979). Hans Peter Luhn introduced the term "stop word" in 1958 for non-keywords, i.e., words with a lack of content that do not help to distinguish information in the documents. This involves articles such as “a” and “the”, pronouns such as “that” and “my”, as well as language-dependent frequent words. Typically, stop words are filtered by using a standard stop word list for the corresponding language. In the FEE project, a German stop word list is being used. Filtering of stop words is also appropriate for big data because of linear time complexity.

One of the challenges in the preprocessing of unstructured data is to make use out punctuations and numbers. This is a problem related to the natural language processing community and involves advanced analysis of sentences, terms and term order, e.g. segmentation methods, POS tagging and named entity recognition. As a simplification, one can just ignore punctuations and numbers by removing the corresponding characters. This operation can also be performed in linear time and is appropriate for big data applications.

Semantic relations of words can be identified by stemming and lemmatization. Stemming refers to the process of removing pre- and suffixes by applying a set of rules. A popular algorithm for stemming is the porter stemmer algorithm (Porter 1980). With stemming, the result is not guaranteed to have a valid form. Typical problems of stemming are over-stemming, i.e., too many characters are removed leading to an overlap in semantically non-related terms, and under-stemming, i.e., not enough characters are removed so that semantically related terms are not overlapping. A more complex approach is lemmatization, which only tries to map inflectional forms of terms to their base form.

Another technique for the reduction of words is the index term selection (Witten 1999). The idea behind this technique is to use only the most representative words of each document in order to reduce the dimensionality of the vector space model even further. The most representative words can be determined by an information measure on the word frequencies relative to the frequencies in the document collection, i.e., how well the documents can be separated by the words. Only the most separating words are then used to create the vector space.

Finally, one can also use matrix factorization techniques for the reduction of dimensions in the vector space. One of the most popular techniques is Latent Semantic Analysis (Deerwester 1990), which projects the original vectors to a vector space with “latent” semantic dimensions. Dimensions in the latent vector space correspond to concepts, which are shared by co-occurring terms. The dimensionality reduction is performed with respect to keeping the greatest variation in the dimensions and is closely related to PCA (see section Preprocessing of Structured Data).

EXPERIENCES AND LESSONS LEARNED

One key aspect prior to an analysis is the anonymization of the data. In order to protect personal data of individuals from further analytics, person names should be made unrecognizable. One way to achieve this is to simply remove those names from the document. This follows the principle "privacy by design" which means that anonymization should be performed as early as possible, so that person names cannot be the subject of further analytics any more. When it comes to automated anonymization, it is also a requirement to convert the files from binary to text formats. One should take into consideration that such a conversion could cause a loss of information. For example, in the case of graphical documents, it is obvious that the visual information cannot be captured by a text format. Such documents have to be manually anonymized by hand (e.g., by blackening person names). For text processing documents (e.g., word format) most of the information can be preserved by choosing an html format over a plain text format. This way, the document structure (e.g., headlines, bold words) is still available for analytical processing such as generating warnings for abnormal situations.

An assistant system that generates early warnings for abnormal situations actually has to solve two types of tasks. At first, the system has to identify events based on the long history of data. A burst in the frequency of the alarm logs, for example, could be an indicator for an unexpected situation, which corresponds to an event. Secondly, after identifying events, the system needs to extract features that help to predict this type of event as early as possible. Coming back to the frequency of alarm logs example, small fluctuations in the frequency distribution could be an indicator for a specific event characterized by a burst in the alarm log frequency.

In our analysis, we focused on unstructured data, i.e., free text entered by operators into the system, but we found that text data could also be part of the sensor data as well. For example, a sensor value is only recorded when it is within a specific range of electricity current, e.g. 24mA. An electricity current above the threshold cannot be recorded and will result in a specific error code translated to a label, e.g. the character string "bad value". There exists a variety of error codes for sensor data and one has to consider how to deal with these error values.

When analyzing free text entered into the system by an operator one has also to consider two types of situations. Text messages that are really typed by the operator, e.g. because the text is about irregular situation in the production facility, and text messages that are just copied from a template, e.g. the text is about a standard procedure like weekly maintenance work. Both situations show different characteristics and should be considered separately. In the former case, the free text is prone to typos and different spellings of the words, e.g. abbreviations and acronyms. In an industrial environment, there also exist some special wordings and a domain-specific vocabulary that has to be taken into account. One of the key challenges when working with industrial text data is to define a semantical relatedness between the words due to the lack of domain-specific concept hierarchies.

In the FEE project, we tried to overcome this problem by using two approaches. First, we used stemming (see section "Preprocessing of Unstructured Data") in order to find semantically related words by removing of prefixes and suffixes. Furthermore, we used a micro-worker approach, where ordinary persons should mark similar words in order to improve the set of related words. We found that the micro-worker approach was difficult due to the lack of domain-specific knowledge by the micro-workers. With stemming, there is also the problem of over- and under-stemming making it necessary to manually inspect stemming results.

Concerning numerical data, methods like outlier detection can help to check the data quality and to ensure meaningful episodes, e.g., in the case of time series. In addition, value imputation methods can be helpful, e.g., in the case of missing values. Advanced methods for data preprocessing, like filter approaches of course can result in a loss of information and should therefore always be targeted with respect to the analytical goals.

Having different data sources in FEE project, we also had to deal with different file formats. All the binary formats were converted to text formats in order to be able to further process the files with standard command line tools. For Excel documents, we choose the CSV format and Word / PDF documents were converted to HTML format in order to keep as much information about the formatting as possible. With the PDF documents, we found that text spanning multiple lines is divided by line when converted to a text format. This could lead to artefacts that have to be further processed, e.g. by dense-based clustering, to connect associated character strings again.

We also like point out, that when dealing with data from different data sources one has to consider the time dependency of the data. In a production environment, there are multiple production applications that have been introduced and expanded over multiple years. These systems are typically not synchronized by a local time server making it necessary to inspect time offsets between different data sources. We found that log entries in the operation journals had an offset of approximately one hour when compared to events in the sensor data (see Figure 2). Reasons for time offsets can be found in differing time zones as well as data types not capable of dealing with daylight saving time. Furthermore operators do not have time for documentation when a critical situation is about to happen, because they have to react to bring the system to a stable state again. Therefore, most of the documentation for critical situations is done after the event has happened.

CONCLUSION

In this chapter, we have discussed several methods and techniques for data preprocessing in the context of Big Data. We have reported experiences and first insights about the preprocessing of a real world dataset in a petro-chemical production setting.

Overall, the principle “privacy by design” is extremely important in a big data environment in order to protect personal data of individuals from further analytics. We also identified two types of tasks that have to be addressed separately in order to create an assistant system for early warnings.

Our experiences show that unstructured data can be found in various places in a production environment containing shift reports, alarm data and even some error codes in the sensor data. For structured data, always the relation between filtering and information loss needs to be balanced. Furthermore, one simple, but important preprocessing techniques for the analysis of natural language text is the mapping of different notations of the same word to a common form, i.e., to a common terminology so that the different entities can be correctly resolved.

Furthermore, we have found that the conversion of file formats can lead to further preprocessing steps, especially when the data is fragmented. Finally, it is important to consider that data is not always coming from one system, making it necessary to check for time offsets and the reasons behind it, connecting that to the business and data understanding phases of CRISP-DM.

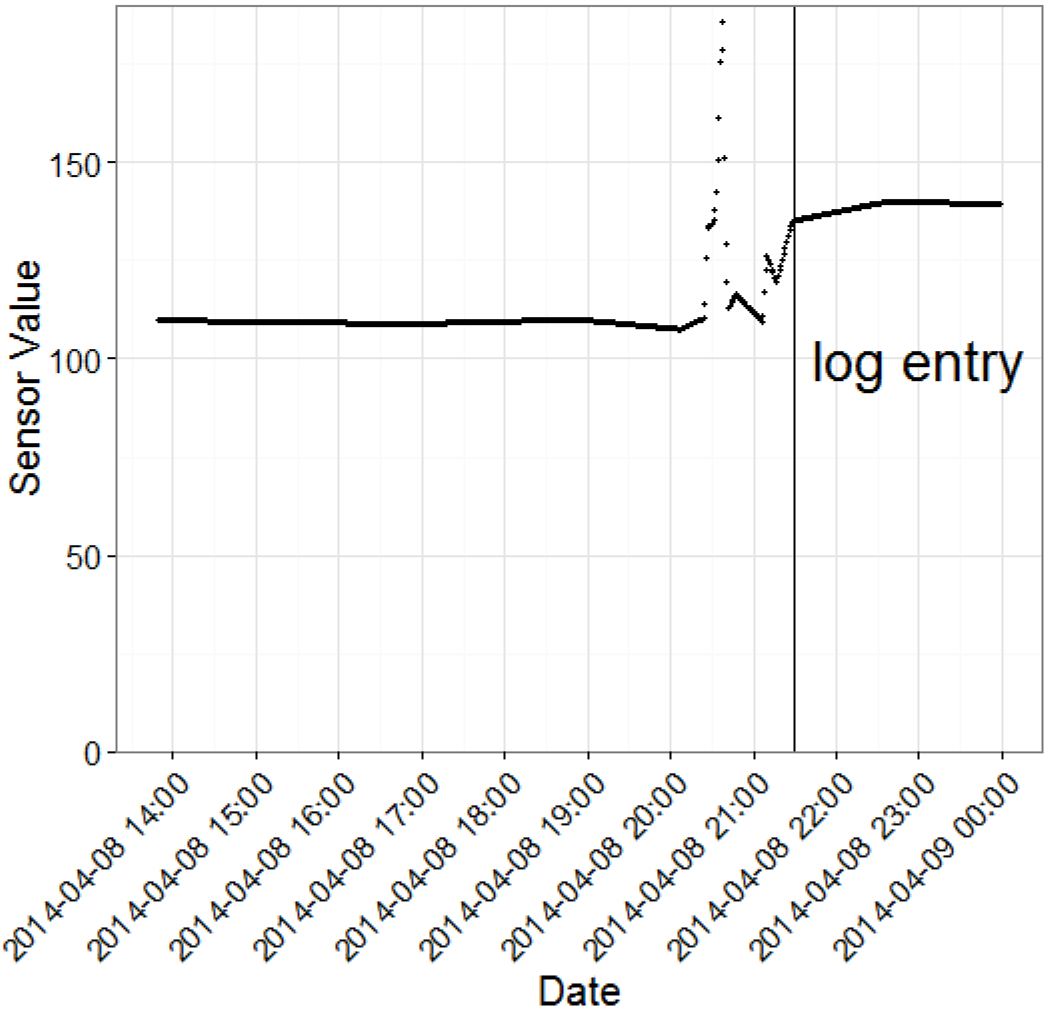

Figure 2. Log entries in the operation journals had an offset of approximately one hour when compared to events in the sensor data. Reasons for time offsets can be found in differing time zones as well as data types not capable of dealing with daylight saving time. Furthermore operators do not have time for documentation when a critical situation is about to happen, because they have to react to bring the system to a stable state again. Therefore, most of the documentation for critical situations is done after the event has happened.

FUTURE RESEARCH DIRECTIONS

Future work will include the analysis of extraordinary characteristics in the industrial real world dataset. One algorithm that can be applied for exploratory analysis is Exceptional Model Mining (Atzmueller 2015, Leman & Feelders & Knobbe 2008), as a variant of subgroup discovery (Kloesgen 1996, Wrobel 1997, Atzmueller 2015) focusing on complex target properties. For that, there are fast implementations available, e.g., (Atzmueller & Lemmerich 2012, Atzmueller & Puppe 2006). This technique can both be applied in the modeling phase as well as in the preprocessing phase, e.g., for attribute construction and data aggregation. Here, also knowledge-based approaches (e.g., Atzmueller 2007) can be utilized.

Furthermore, we plan to apply techniques from information extraction (Grishman 1997) to the unstructured / textual information for event detection (Melton & Hripcsak 2005), e.g., using rule-based techniques (Atzmueller & Kluegl & Puppe 2008, Kluegl et al. 2009). By applying NLP techniques, for example, we will analyze the potential of extracted information for indicating upcoming events. Then, by using advanced methods for information extraction, we can closely link unstructured as well as structured data, also with respect to data quality. The latter is a major issue that can also targeted using the techniques described above, including appropriate measures (cf. Atzmueller et al. 2005). Furthermore, Exceptional Model Mining can be applied on the given multi-dimensional dataset in order to detect implausible correlations, which can then indicate problematic episodes. Here, techniques for profiling expected relations, e.g., (Atzmueller et al. 2005) can be a good starting point for future research, which can also be supported by appropriate visualization, inspection and explanation methods, see e.g., (Atzmueller & Puppe 2008, Atzmueller & Roth-Berghofer 2010).

Acknowledgements: The underlying project of this contribution was sponsored by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF), under the sponsorship reference number 01IS14006. The authors are responsible for the contents of this contribution.

REFERENCES

Atzmueller, M. (2015). *Subgroup Discovery – Advanced Review.* WIREs: Data Mining and Knowledge Discovery, 5(1):35–49.

Atzmueller, M. (2007) *Knowledge-Intensive Subgroup Mining -- Techniques for Automatic and Interactive Discovery*. Dissertations in Artificial Intelligence-Infix (Diski), (307) IOS Press

Atzmueller, M., Baumeister, J. & Puppe, F. (2005) *Quality Measures and Semi-Automatic Mining of Diagnostic Rule Bases*. Proc. 15th International Conference on Applications of Declarative Programming and Knowledge Management, Springer, Berlin/Heidelberg

Atzmueller, M. & Kluegl, P. & Puppe, F. (2008). *Rule-Based Information Extraction for Structured Data Acquisition using TextMarker.* In Proc. LWA 2008. University of Wuerzburg, Germany.

Atzmueller, M. & Lemmerich, F. (2012). *VIKAMINE - Open-Source Subgroup Discovery, Pattern Mining, and Analytics.* In Proc. ECML/PKDD 2012: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer, Berlin / Heidelberg.

Atzmueller, M. & Puppe, F. (2008). *A Case-Based Approach for Characterization and Analysis of Subgroup Patterns*. Journal of Applied Intelligence, (28)3:210-221, 2008.

Atzmueller, M. & Puppe, F. (2006). *SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery.*In Proc. PKDD 2006, number 4213 in LNAI, pages 6–17. Springer, Berlin / Heidelberg.

Atzmueller M., Puppe, F. & Buscher, H.-P. (2005). *Profiling Examiners using Intelligent Subgroup Mining*. Proc. 10th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2005), 46--51, Aberdeen, Scotland.

Atzmueller, M. & Roth-Berghofer, T. (2010). *The Mining and Analysis Continuum of Explaining Uncovered*. Proc. 30th SGAI International Conference on Artificial Intelligence (AI-2010).

Austin, P, & Brunner, L. (2004). *Inflation of the Type I Error Rate When a Continuous Confounding Variable Is Categorized in Logistic Regression Analyses.* Statistics in Medicine, 23(7), 1159–1178.

Box, G. & Cox, D. (1964). *An Analysis of Transformations*. Journal of the Royal Statistical Society. Series B (Methodological), pp. 211–252.

Crowe, C. M. (1996). *Data reconciliation—progress and challenges*. Journal of Process Control, 6(2), 89-98.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). *Indexing by latent semantic analysis*. JAsIs, 41(6), 391-407.

Grishman, R. (1997). *Information extraction: Techniques and challenges.* In Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. Springer.

Hollender, M. (2010). *Collaborative Process Automation Systems*. ISA, p. 300

Hotelling, H. (1933). *Analysis of a complex of statistical variables into principal components*. Journal of educational psychology, 24(6), 417.

Klösgen, W. (1996). *Explora: A multipattern and multistrategy discovery assistant.* In Advances in Knowledge Discovery and Data Mining, pages 249–271. AAAI.

Kluegl, P., Atzmueller, M. & Puppe, F. (2009) *Meta-Level Information Extraction*. The 32nd Annual Conference on Artificial Intelligence, Springer, Berlin/Heidelberg.

Leman, D. & Feelders, A. & Knobbe, A. J. (2008). *Exceptional model mining.* In W. Daelemans, B. Goethals, and K. Morik, editors, ECML/PKDD (2), volume 5212 of Lecture Notes in Computer Science, pages 1–16. Springer, Berlin / Heidelberg.

Liu, H., Shah, S., & Jiang, W. (2004). *On-line outlier detection and data cleaning*. Computers & chemical engineering, 28(9), 1635-1647.

Luhn, H. P. (1957). *A statistical approach to mechanized encoding and searching of literary information*. IBM Journal of research and development, 1(4), 309-317.

Melton, G. & Hripcsak, G. (2005). *Automated detection of adverse events using natural language processing of discharge summaries.* Journal of American Medical Informatics Association, 12(4):448–57.

Porter, M. (1980). An Algorithm for Suffix Stripping. *Program*, 14, 130-137.

Rijsbergen, C. V. (1979). *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA.

Salton, G., Wong, A. & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18, 613-620.

Serneels, S. & Nolf, E. & Espen, P. (2006). *Spatial Sign Pre-processing: A Simple Way to Impart Moderate Robustness to Multivariate Estimators.* Journal of Chemical Information and Modeling, 46(3), 1402–1409.

Shearer, C. (2000). *The CRISP-DM Model: The New Blueprint for Data Mining*. Journal of Data Warehousing, 5(4), 13-22.

Sparck Jones, K. (1972). *A statistical interpretation of term specificity and its application in retrieval*. Journal of documentation, 28(1), 11-21.

Thornhill, N. F., Choudhury, M. S., & Shah, S. L. (2004). *The impact of compression on data-driven process analyses*. Journal of Process Control, 14(4), 389-398.

Witten, I. H., Moffat, A., & Bell, T. C. (1999). *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann.

Wrobel, S. (1997). *An algorithm for multi-relational discovery of subgroups.* In J. Komorowski and J. Zytkow, editors, Principles of Data Mining and Knowledge Discovery, volume 1263 of Lecture Notes in Computer Science, pages 78–87. Springer, Berlin / Heidelberg.

Yeo, I. & Johnson, R. (2000). *A new family of power transformations to improve normality or symmetry.* Biometrika, 87, 954-959.

KEY TERMS AND DEFINITIONS

CRISP-DM: The Cross Industry Standard Process for Data Mining describes the common phases of a data mining workflow.

Dummy Variable: A variable that holds the information about one categorical value which either can be present (1) or absent (0).

Unstructured Data: Data that is not organized in a pre-defined format. Typically, this involves natural language text.

Vector Space Model: A model that represents text documents in a vector space where each dimension corresponds to a term.

TF-IDF: Weighting of terms according to the term frequency (TF) and inverted document frequency (IDF). Frequent terms that only occur in a few documents get the highest weight.

Stop words: Frequent words with a lack of content, like pronouns and prepositions, that do not help to distinguish documents from each other. Typically, stop words are filtered during the processing of natural language text.

Anonymization: Process of removing personal information from a document to protect the identity of a person.