

# Data Analytics for Advanced Energy Management in Real-Time based on Open Source Technologies

Marcel Graus, Philipp Niemietz, Michaela Hiller, Mohammad Touhidur Rahman  
*Information Management*  
*FIR at the RWTH Aachen University*  
*Aachen, Germany*

## Abstract

*Due to the climate change, the need of sustainable technological progress is growing. Renewable energies undergo a significant upturn, but they come with the price of nearly uncontrollable and hard-to-predict fluctuating energy generation. This forces a paradigm change in handling the energy resources in the society. Especially in the industry, there will appear a need to manage energy in the sense of planning and controlling power consumption. If a company wants to make use of its flexibility in energy consumption - for example by managing a local virtual power plant - there is a need to make decisions in the required pace. Thus, a real-time decision-making process needs an adequate IT-infrastructure and algorithms tailored to the specific use case. This paper gives an overview of the necessary components to realize real-time analytics for industrial energy management. The concepts are expatiated on a scalable real-time computing architecture and a mathematical approach for short term prediction of energy consumption and pattern recognition that are included in a Business Intelligence Tool for decision support using only open source components.*

## 1. Introduction

The worldwide attention to the UN climate summit in Paris 2015 showed the increasing importance of sustainability in the global development of the society. The irreversibility of these trend-setting decisions has been proven in the recently shown massive resistance against any try of the American President to reverse the movement towards a more sustainable economy. Another powerful development that has an impact on nearly every part of the economy is the digitalization: the use of new digital technological possibilities induces a large chance for the sustainability movement. This applies in particular in the manufacturing sector with many interesting use cases for both - the digitalization as well as the sustainability process. In Germany for instance, there are two initiatives, Industry 4.0 and the energy transition, to support the innovation process in this area. Whereas the first one promotes the fourth industrial revolution by an integration of the internet

of things into the industrial value chain [1], the latter one tries to speed up the transition to a low carbon, environmentally sound, reliable and affordable energy supply. [2]

For industrial energy management, the potential synergies of these two developments are close to obvious. Here, innovations are increasingly based on a continuous data acquisition as promoted in ISO 50001. This, along with the expansion of sensors and data, provides the basis of nearly every industry 4.0 scenario. [3] As a consequence, a huge amount of gathered data has to be transformed into information on how to lower energy consumption or greenhouse-gas emission. Hence, data processing becomes the digital shadow for many recent innovations. Decisions reached on the basis of data [4] can enable a holistic and thus complex optimization of the energy use within a company. Therefore, the usage of data plays a key role in industrial energy management that corresponds to the characteristic feature of Industry 4.0, which is the creation of value from data. [2]

In brief summary this means that the industrial management needs to develop tools to handle the increasing complexity based on data. Additionally, corresponding information systems are required to overcome the stage of a pure monitoring device, but need to provide an active system that integrates IoT concepts and can handle new challenges that arise with the sustainability movement. Consequently, this paper follows the vision of an energy management system that appears as a digital expert supporting the energy manager in a manufacturing company. This expert has not only to process data at a large scale but also to break down its complexity to achieve comprehensibility and graspability for the user. The described tool is technically a decision support system that bases its output on an optimization over all possible scenarios. That requires the ability to predict the behavior of the system given certain parameters from the user. The whole approach is introduced in [5] with the predictive model as central component.

In order to provide those functionalities, there is a need of powerful user interfaces with a visualization supporting the user permanently and guaranteeing the possibility to react to recent developments in the

system. This requires real-time reaction, as decision-makers are used to immediate feedback from human expert collaboration. The intelligence of the system depends mainly on fast data processing and the adaptability to previous experiences in the predictive model. Thereby induced general requirements on the architecture of the system can be summarized as the ability to analyze large sets of historical data and rapidly process streams of data from manufacturing and energy sensors. [6]

## 2. Related Work

A lot of work about has already been done about IT infrastructures that can handle real-time data processing. For example, the usage of Apache Spark and Apache Storm for real-time applications are well understood. In [7] Agneeswaran shows the advantages of using these tools in context of machine learning algorithms. Furthermore, a big data architecture with an in-memory data base was presented in [8]. There are many publications available not only about adequate IT infrastructures but also about mathematical models and algorithms that can be used for real-time systems. The application of machine learning algorithms in the domain of energy management is already focused in the special case of buildings in [9]. In addition, predictions of energy consumption for large-scale systems are analyzed in [10].

As seen above, there is already much research done about all subcomponents requisite for real-time analytics. However, to the knowledge of the authors the application domain of industrial energy management integrated in a holistic approach from data basis to the decision support has not been examined nor published yet.

### 2.1. Approach

The goal of this paper is the presentation of a concept that closes the gap identified in the related work. An approach for a holistic procedure model was already presented in [5, 6], whereby data analytics is introduced as a process of transformation from data to information usable for decision-making. Additionally, a procedure consisting of four steps (data modelling, analysis, simulation and optimization) is presented *ibid*. In this paper, the analysis and simulation step as well as their mutual influence are focused, whilst the decision support is realized through visualization of the prediction. This includes the future effects of an intervention. Thus, it behoves the user to perform the optimization step manually.

As motivated above, the digital expert needs to be fast and adaptive. The implied advantage of the decision support system as a digital expert is its general capability to handle any desired amount of data sources on an arbitrary fine time scale. This

entails a huge scalability of the system and determines an absence of scalability as central limitation of data-driven energy management.

The prerequisite for this capability is a predictive model by which data processing at high speed and integration of all available data is possible. This imposes high demands to the infrastructure and motivates the introduction of a corresponding big data architecture. According to the presented approach, the next two chapters concern the big data architecture and the predictive model, before a suitable visualization environment is discussed. The paper concludes with an outlook on the future research work. The aim of this paper is the introduction of a holistic approach, wherefore the complexity of the presented components is consciously restricted and details about every submodel need to be found in the referenced papers.

## 3. Scalable Real-Time Computing Architecture

To realize the above-mentioned concepts there is a need to process sensor data at a massive scale while providing real-time insight into the current state of the system, as well as predicting trends in near future. Thus, an important step is the construction of an adequate infrastructure. By keeping the modularization of the architecture as high as possible, the system is able to adjust to new challenges. It is possible to separate such an architecture into five distinct parts: Data Integration, Storage, Access, Analytic Processing and Visualization. [11] A sixth aspect is Security which is not focused in this paper. Figure 1 shows a graphical view of the now presented architecture.

### 3.1. Data Integration

The first stage of building a data processing architecture is to identify components needed to acquire the desired data. In order to generate data, energy sensors are connected to all machines of interest. Moreover, energy data is complemented by information from machine data acquisition that is either accessible via a Representational State Transfer (REST)-compliant Web services or is periodically sent by a bus system to the cluster. If the data provided by the sensors and machines is homogenous, it is possible to directly hand it over to a general-purpose message broker with complex functionalities like RabbitMQ [12] or a lightweight one which comes with very high throughput like Apache Kafka. [13] If the data dealt with is highly heterogeneous or the system needs to be able to adjust to format changes quickly, Apache NiFi [14] is an adequate tool for constantly evolving devices, metadata and data formats and can handle any file types that may occur.

It uses flow-based programming and provides a web-based user interface to observe and manage data flows in real-time. Moreover, Kafka offers build-in functionalities for an Extract-Transform-Load (ETL) approach, meaning that the initial data cleaning logic is applied to the data before it enters the system architecture. [15] For the system introduced in this paper, an Extract-Load-Transform (ELT) approach is used, since the data cleaning logic is applied after the data have been loaded into the cluster (see Figure 1).

### 3.2. Data Storage

The raw data stream produced by energy sensors and machine data acquisition is stored in the column-based database Apache Cassandra® [16], which allows fast reads, writes and good scalability for time-series data sets. [17] Both the cleaned data set and the analytics results will be stored in the in-memory computing and storage platform Apache Ignite. [18] Ignite's in-memory storage paradigm and built-in support for Apache Spark [19] as well as in-memory MapReduce computation can significantly boost performance of complex algorithms run on historical data sets. Since in-memory storage is not persistent, Cassandra can be used as the persistent storage for the Ignite cache.

### 3.3. Data Access

In the architecture there is a need to perform stream processing for real-time data analytics as well as batch processing to analyze historical data at a large scale. For this sake the open source cluster-computing framework Apache Spark can be used either on top of Ignite's in-memory computing platform or just with Cassandra to perform the analysis of historical data sets. Spark offers implicit parallelism as well as fault-tolerance computing and comes with various programming interfaces (for Scala, Java, Python and R) that ease the use of the framework.

For stream analytics, the stream processing framework Apache Storm [20] provides distributed and fault-tolerant real-time computation. A Storm topology consumes streams of data and processes those streams in arbitrarily complex ways.

### 3.4. Analytic Processing

The Analytic Processing builds the core component of the here presented digital expert for industrial energy management. In a first step, historical data have to be analyzed to extract useful information that will be used to perform real-time classification and prediction. Thus, Spark is needed to do the initial data mining and machine learning on large historical data sets. Next, each tuple in the data stream is classified using the data mining information and the next tuples are predicted based on the model

derived from the machine learning process. It is important to keep in mind that not all data mining and machine learning methods can be parallelized such that they can be implemented with Spark efficiently. This also holds for the classification and prediction part which needs to fit into the processing model of Storm. Therefore, while choosing the methods applied to different steps in the analytic processing, the restrictions given by the selected technology must be taken into account. The procedure in the analytic processing and possible used methods are discussed in detail in the Section III.

### 3.5. Visualization

The information resulting from the real-time analytic processing needs to be transformed into a view that provides all important information for the specific use case in real-time. The real-time view has to picture the short term prediction of every time unit, such as seconds, meaning that for each time unit the next short term prediction of that point has to be loaded and displayed by the visualization tool within strict time requirements. In Section IV a real-time intelligence tool is introduced that provides this feature together with some illustrations. Moreover, R is used as a general tool to visualize and analyze historical data sets [21].

An additional component, which is not listed explicitly in the paragraphs above, is Apache Zookeeper. [22] It is implicitly needed to take care of the coordination and synchronization between the components of Storm and Kafka.

All components of the technology stack in the presented architecture are selected to potentially support data analytics and real-time processing at a massive scale. Moreover, all of them run distributed on a cluster of flexible size and therefore scale horizontally to the amount of data and resources. As mentioned above, most components of the technology stack used in the architecture can be replaced by components fulfilling similar roles as long as the interfaces between the components are provided. As an example, instead of Storm, Apache Flink® [23] can simply be used to do the real-time analytics if it yields a better performance for the specific use case. Since Kafka offers an API for Flink, the data can be pushed into the Ignite cache with a Flink streamer. Similar, Spark can be replaced by MapReduce computation schema on top of Ignite if it fits the developer experience better. This flexibility enables the developer team to adjust each part of the architecture in a way that fits best the team skills and analytic algorithms in a specific use case.





Figure 1. An hierarchical architecture of a real-time data analytics architecture, covering the main functions of big data systems, including data integration, storage, access, process analytics and visualization

### 3.6. Open Source Paradigm

This subchapter reflects the experience of the authors during their data analytics projects and explains the reasons why the presented architecture was built completely with open source technologies. First, open source technologies ensure maximal flexibility in the independent specific design of a software solution due to its provision of the source code. In the development of new concepts this can predominate the potentially higher implementation effort. Second, the main development environments for data analytics like R and Python are based on open source principles and provide domain specific algorithms for a variety of analytical tasks. Due to the large online community and many contributors, these tools offer good support and are continuously improved.

A central aspect for the decision for the open source approach is the velocity of the development of data analytics and corresponding technologies today.

This leads to a continuous change of choice of different software tools that all provide specific advantages. The idea of many commercial platforms is to propose all kind of technologies, so that the user is invited to build a full technology on a single platform. This can potentially lead to a so called vendor-lock-in, where the user can only change general components of the architecture with a large effort because of missing interfaces to other technologies. For a single use case, there will not appear any problem, since one can choose the commercial software that fits the problem best. However, for this work the simple addition of other little related use cases that can be realized on the same data basis takes in a mayor role. The example in this work is the realization of a short-term prediction for industrial energy management and a pattern recognition for condition monitoring on the basis of the same big data architecture. In general, the rise of the digitalization and internet of things (IoT) just started and on the long run every company will have

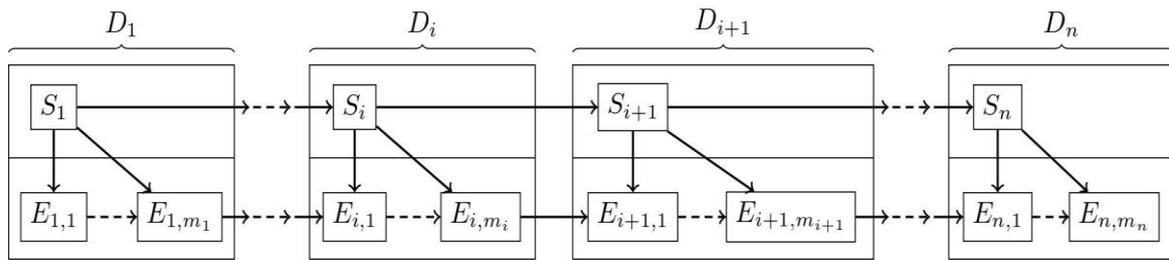


Figure 2. Directed acyclic graph consisting of machine states  $S_1, S_n$  with durations  $D_1, \dots, D_n$  and energy clusters  $E_{(i,1)}, \dots, E_{(i,m_i)}$  during state  $S_i$  represented as nodes

many completely different use cases. The current rise of the internet of things (IoT) and digitalization just started and comes with a variety of new use cases. Thus, no single software company provides the best solutions for every specific task. Here, an open source paradigm has the advantage that in most of the cases interfaces between the most common tools are given, so that one can put together the class-best-components.

#### 4. Predictive Analytics Model

Pursuing a data-driven approach the objective is to learn a model which has to comply with real-time and parallelization capability only using the machine's output data. The supposed approach for unsupervised learning consists of two steps which include a number of single modules performing algorithms for specific analytical subtasks. First, the historical state and energy data are examined to extrapolate a clustering and the transition probabilities. Second, forward propagation and a short-term forecast is calculated based on the real-time data.

##### 4.1. Historical Data Analytics - Model Definition

As the complex coherences cannot be described by some simple physical model, a probability model including a directed acyclic graph (DAG) is set up. The procedure is illustrated on the example of an industrial machine which provides its state and power consumption. In order to keep the model as simple as possible, a dimension reduction or feature extraction method can be applied at the beginning.

Retrieved historical energy data are pre-processed in a data mining module that can perform for instance scaling, a segmentation and clustering algorithms. For this sake, an approach such as a segmental k-means algorithm can be chosen, depending on power value and further by feature extraction acquired data, e.g. gradient and arithmetic average power value over the latest  $r$  timestamps. The result is a sequence of energy clusters. Thus, through data mining the complexity of the input data from energy sensors is reduced from the

over-countable state space of the power consumption to the discrete state space of energy clusters.

In the machine learning module the correlations between different data streams need to be mapped. In the exemplary stochastic approach mentioned above, each state of the machine data and each energy cluster from the data mining module is considered as a node of a two-layer probabilistic graphical model (PGM), as shown in Figure 2. Furthermore, a state is assumed to affect its duration along with the energy clusters during its term and the successive state, whereas an energy cluster has an impact merely on the following energy cluster. The resulting transition probabilities between these random variables yield a prediction of the future data based on the current values. Our example model possesses the  $r$ -th order Markov property, as the mean power value for each time stamp purely depends on the  $r$  previous values. Furthermore, state changes do not occur in equal periods of time and hence this certain model defines a semi-Markov process.

An intervention in the state process corresponds to the adaption of the machine program, which enables simulations of improvements in machine control. Regardless of the specific model used for the data mining and machine learning module the result of the historical data analytics steps should be a predictive model that allows simulations of interventions in established processes.

##### 4.2. Properties and Limitation of the Machine Learning Model

After defining the model an important step is the comparison of its properties to the specifics of the empirical data. Adapting the model to the properties of our application certain courses of energy data can be described, such as alternating high and low power consumption, local peaks or a plateau phase with virtually constant values, followed by an increase/decrease in use of energy, or vice versa. More complex correlations like a particular curve progression, however, need to be captured with a more complex model.

A characteristic property of industrial machines are long stand-by times followed by sudden rises in energy consumption with sharply fluctuating energy data. Thus, a semi-Markov assumption with a separate stochastic process for dwell time in one state delivers an appropriate model for machine states as latent cause. For further pattern recognition, observing the cluster sequence might be an appropriate approach that could yield an indication of machine states.

Although computing-intensive medium-term forecasts based on historical data can be performed with this model using Spark, this paper focuses on real-time prediction which is discussed in the following subsection.

### 4.3. Real-Time Data Analytics – Predictions

As shown in Figure 1 the first step is performed by a data cleaning module that ensures an adequate data quality. The correlations are represented in the machine learning model premised on historical data analytics, while the complexity of real-time processing can be reduced by replacing the elaborate clustering procedure by a simple classification in the data stream. Thus, using the output data of the historical data analytics yields real-time capability in spite of the significant higher complexity of our model compared to the mostly used regression approach.

Incoming energy data are processed first in the classification module which is connected with the data mining module for the historical data analytics. In the presented example, energy data are classified and assigned to the  $k$  partition sets by applying the minimal distance to cluster centroids regarding the same metric as used for the  $k$ -means clustering. Next, a prediction module based on the machine learning model derived from historical data analytics is used. Here, the following state is predicted using the current one and its transition probability. Thereafter, the expected next power value  $\hat{p}_{t+1}$  is calculated with the expected value depending on the current state and previous energy cluster. If  $P[l, i, j]$  defines the probability that with a present machine state  $l$  and previous energy cluster  $i$  the current energy cluster will be  $j$  and  $\mathcal{N}_{i,j}(p)$  indicates the normal distribution within power values of state  $l$  and energy cluster  $j$ , it holds

$$\hat{p}_{t+1} = \sum_{j=1}^k P[l, i, j] \cdot \int_p p \mathcal{N}_{i,j}(p) dp.$$

The resulting power value is classified to its corresponding cluster and thereby a forecast of the state and energy cluster for the succeeding timestamp is obtained. The complexity of the prediction step is determined with the machine learning model. In this case, the Markov property of the stochastic model ensures the iterative nature of the prediction model.

In addition, the short term prognosis of the power value  $p_{t+1}$  is refined by computing its  $(1 - \alpha)$ -confidence interval  $[p_{\min}, p_{\max}]$  which fulfils

$$\begin{aligned} &P[p_{\min} \leq p_{t+1} \leq p_{\max}] \\ &= \sum_{j=1}^k P[p_{\min} \leq p(j) \leq p_{\max}] \cdot P[l, i, j] \\ &\geq 1 - \alpha, \end{aligned}$$

where  $p(j)$  denotes the power values within cluster  $j$ .

Thus, the future energy consumption of a machine for the following few time units can be forecasted with a defined certainty, and as a long term goal adjust its schedule to the expected available amount of energy.

### 4.4. Real-Time Data Analytics – Pattern Recognition

Besides the usual energy management tasks, there are other valuable applications of the real-time data analytics framework. Instead of short-time predictions, the induced classification module offers possibilities for different approaches. Assume that we have a pattern that is associated to a critical state of the regarded machine. This association could be learned by correlating patterns in energy data with for example critical machine states. Based on this knowledge, it would be advantageous to detect occurrences of this critical pattern. Since the presented machine learning approach is based on stochastics, it is possible to compute probabilities of seeing a critical pattern.

Independent from the clustering module and thus, independent from the process to generate stochastic descriptions of the energy consumption time series, the result can be summarized as the following. If  $(p_{t-d+1}, \dots, p_t)$  are the last  $d$  power values, then the stochastic description corresponds to the translation into energy clusters  $(c_{t-d+1}, \dots, c_t)$ . Depending on the stochastic model and the underlying assumption like the markov or semi-markov-property the method to detect patterns is different.

Here, we explain the idea in the case of uncorrelated observations, i.e. each power value belongs to a certain cluster, which can be described via Gaussian mixture model or in the easiest case as Gaussian distributed. Using a sliding window approach with window size  $d$  at every time step  $t$  a partial time series  $(p_{t-d+1}, \dots, p_t)$  is regarded. Because of the assumptions, it holds

$$P(p_{t-d+1}, \dots, p_t) = \prod_{s=1}^d \left( \sum_{m=1}^M c_m N_{\mu_m \sigma_m}(p_{t-d+s}) \right).$$

If this value is below a certain threshold, an alarm can be triggered. Thus, partial time series or windows can be classified into critical and non-critical patterns.

Before applying this method, the time series consisting of power values could be normalized in several ways to avoid offset translation and other principal problems in time series. Alternative approaches to detect patterns can be derived with the translation of the energy clusters into symbols. For this case, a symbolic aggregate approximation (SAX) could be applied. Based on this representation of each window every text mining algorithm could be applied in general. The task is to search correlations of words and critical machine states.

The real-time pattern recognition can be used to detect critical machine states, which lead to a waste of energy. Together with the short-time prediction, this aspect of real-time data analytics insures higher energy efficiency in industrial companies.

#### 4.5. Architecture – Model

As mentioned in Section II, there is a need to choose a model and algorithms that fit the selected processing paradigm. Computing the transition probabilities for the model described in the previous paragraphs and performing a  $k$ -means clustering can be efficiently implemented in Spark. Moreover, the real-time classification and forecast depend on the last  $r$  values in a time series as well as some precomputed parameters and transition probabilities. This information can be loaded initially with the startup of the Storm topology. Thus, both classification and forecast can be implemented within the Storm framework using a sliding window approach.

### 5. Real-Time Intelligence

For this project an internal visualization website showing the real-time scenario of various attributes has been developed.

#### 5.1. Tools and Technologies

At first, the requirements for the tools used to build and design each part of the website have to be defined. Clearly, there is a need for a database-backend tool that should be able to manipulate the data according to our requirements. To present the data in a visually attractive manner on the website, a good frontend tool is demanded. Finally, the selected visualization tool needs to provide many different visualization possibilities. For example, there should be support for line graphs or timeline graphs and it should provide the feature to customize the visualization according to the specific use case. For all of those components it is important to consider real-time ability of the website in order to provide the functionalities described in Section I. In the following paragraph tools are discussed which can be used to achieve the requirements stated above.

In general, there are many tools available to build a website such as PHP, Python, Java and Ruby on Rails, to name but a few. For the specific use case described in this paper, Ruby on Rails has been chosen, since it is suitable for rapid prototyping with its provided program skeletons. [24] This framework is already configured and ready for manipulation to fit the use case without any initial configuration hassles. Further, common features like a good documentation or an open source license are provided in addition to a healthy community.

Here, Ruby on Rails is mainly used for the backend purpose, i.e. to connect with the database and collect all the required data for the visualization. For the frontend, HTML, CSS and JavaScript have been selected. The website and all its views should be accessible for differently sized devices. Therefore, Bootstrap is a good choice as it provides the functionality to develop a responsive website. [25] Vis.js, a dynamic, browser-based visualization library written in JavaScript, comes with several components for visualization. [26] As mentioned in the sections above the current energy consumption, classification and forecast need to be clearly visualized in a graph. The functionality for line graphs is provided by the component Graph2D of Vis.js and allows different customization options like color manipulation. By utilizing Server-Side Events together with the Ruby on Rails module, real-time updates of the server can be implemented efficiently. From the perspective of the implementation achieving the real-time scenario was relatively simple due to the preconfigured skeletons of Ruby on Rails, as configuration is commonly the most time-consuming task.

#### 5.2. Visualization

In order to give an impression of the developed real-time intelligence tool, two images of different views are presented.

The first picture in Figure 3 shows the pattern of energy consumption of the considered machine. After a period of idle time a common job starts with a steep uprise, followed by three plateaus with different energy consumption rates. About 35 seconds later, the machine changes back into the idle state. This illustrates that the energy consumption for a typical job on this machine shows a distinctive pattern. The goal for the analytic model is to capture such specific pattern and to predict the next few seconds after a job has been started. The second image illustrates the visualization of a ten second prediction for a point in time, after a job has just started on the machine. The blue line is the actual energy consumption, where the red one is the predicted power value. Moreover, the grey lines limit the white space which demonstrates the confidence interval for the actual power value, which gives us a 95% chance that the actual value is



Figure 3. Common pattern of energy consumption divided into six clusters calculated by the k -means clustering and a ten second prediction of the power value (red) and the printer state (B = ready, W = warming up, D = printing, E = energy-saving mode), its 95%-confidence interval along with the actual course (blue)

within that range based on the historical analytic processing.

## 6. Conclusion

For the testing of the concepts several environments and data sources are used. The central test environment for the prototyping was built around a printer as can be found in every office. First, it was necessary to install some sensors and scrap information from the machine data acquisition. Since

a general data model is used the concepts can be transferred to other industrial machines. The results can be seen as proof-of-concept, since all components played well together.

The next steps of the research is the further development of the mathematical models, inducing more complexity in the algorithms to ensure adaptivity and their integration in the architecture. Adaptivity in this context means a continuous improvement of the prediction to establish trust in the system. The inclusion of feedback through the

deviation of real power consumption from the forecasted one into the predictive model yields an extension of the presented unsupervised learning approach in the analysis step to a reinforcement learning technique. Another aspect that is not regarded in this work so far is the real-time decision-making process. Based on the prediction and pattern recognition the user needs to make decisions. In the field of industrial energy management, a short-term prediction can help to avoid load peaks that lead to costs for the company. The user needs to decide which energy consumer can be reduced or shifted or which energy generator or storage can be adapted. In the field of condition monitoring, pattern recognition can be used to predict critical machine states. With this information, the user can try to avoid the critical states or take measures to handle the situation in an optimal way. Further, the concepts will be tested in several industrial environments in context of current research projects.

## 7. Acknowledgment

This article arose during the work of authors, within the framework of the research project EWIMA (project number: EFRE-0800681) funded by the European Regional Development Fund and Ministry of Economic Affairs and Energy of the State of North Rhine-Westphalia in Germany. The authors want to thank all donors, supporter and critics.

## 8. References

- [1] M. Hermann, T. Pentek, B. Otto, "Design Principles for Industrie 4.0 Scenarios", in 2016 49<sup>th</sup> Hawaii International Conference on System Science (HICSS), vol. 00, no., 2016, pp. 3928-3937.
- [2] Federal Ministry of Economics and Technology (BMWi); Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (BMU), "Energy concept for an environmentally sound, reliable and affordable energy supply", Berlin, Germany: Federal Ministry of Economics and Technology (BMWi). Retrieved 2016-05-01.
- [3] G. Förtsch, H. Meinholz, Energiemanagementsystem nach DIN EN ISO 50001. Springer Fachmedien: Wiesbaden, 2014 .pp. 376-417.
- [4] H. Wortmann, W. Mulder, "Reflections on Identity Management in Smart Industry: The Paradox of Theseus' Ship and Beyond" in Advances in Production Management Systems. Initiatives for a Sustainable World, I. Nääs et al, IFIP Advances in Information and Communication Technology, vol 488, Springer, Cham, 2016, pp. 299-306.
- [5] M. Graus, K. Westhoff, X. Xu, „Procedure Model for Data-Driven Decision Support Regarding the Integration of Renewable Energies into Industrial Energy Management“, World Academy of Science, Engineering and Technology, International Science Index 121, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 11(1), 2017, pp. 25-30.
- [6] M. Graus, "Integration einer Datenanalytik in Energieinformationssysteme produzierender Unternehmen" in Smart Energy 2016, U. Großmann and I. Kunold and C. Engels, Eds. Glücksstadt: Verlag Werner Hülsbusch, 2016, pp. 44–55.
- [7] V. S. Agneeswaran, Big Data Analytics Beyond Hadoop: Real-Time Applications with Storm, Spark, and More Hadoop Alternatives. Upper Saddle River New Jersey, Pearson Education, 2014.
- [8] J. Krumeich, D. Werth, P. Loos, J. Schimmelpfennig, S. Jacobi, "Advanced planning and control of manufacturing processes in steel industry through big data analytics: Case study and architecture proposal" in IEEE International Conference on Big Data, J. Lin, Washington. 2014, pp. 16-24.
- [9] E. Mocanu, P.H. Nguyen, M. Gibescu, W.L. Kling, "Comparison of Machine Learning Methods for Estimating Energy Consumption in Buildings" in 2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), Durham. 2014, pp. 1-6.
- [10] T.Samak, C. Morin, D. Bailey, "Energy Consumption Models and Predictions for Large-scale Systems" in 2013 IEEE 27<sup>th</sup> International Symposium on Parallel & Distributed Processing Workshops and PhD Forum, Boston. 2013, pp. 899-906.
- [11] Data, BITKOM-Arbeitskreis Big Data. "Big-Data-Technologien–Wissen für Entscheider." *Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., Berlin* (2014).
- [12] RabbitMQ, URL: [rabbitmq.com](http://rabbitmq.com) as of 20.05.2017.
- [13] Apache Kafka, URL: [kafka.apache.org](http://kafka.apache.org) as of 20.05.2017.
- [14] Apache NiFi, URL: [nifi.apache.org](http://nifi.apache.org) as of 20.05.2017.
- [15] Ganchev, Z. Ji and M. O'Droma, "The creation of a data management platform for use in the UCWW," *2016 SAI Computing Conference (SAI)*, London, 2016, pp. 585-588. doi: 10.1109/SAI.2016.7556040
- [16] Apache Cassandra®, URL: [cassandra.apache.org](http://cassandra.apache.org) as of 20.05.2017.
- [17] Dunning, Ted, and Ellen Friedman. "Time Series Databases." 2015.
- [18] Apache Ignite, URL: [ignite.apache.org](http://ignite.apache.org) as of 20.05.2017.
- [19] Apache Spark, URL: [spark.apache.org](http://spark.apache.org) as of 20.05.2017.
- [20] Apache Storm, URL: [storm.apache.org](http://storm.apache.org) as of 20.05.2017.

[21] R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL [R-project.org](http://R-project.org).

[22] Apache Zookeeper, URL: [zookeeper.apache.org](http://zookeeper.apache.org) as of 20.05.2017.

[23] Apache Flink®, URL: [flink.apache.org](http://flink.apache.org) as of 20.05.2017.

[24] Ruby on Rails, URL: [rubyonrails.org/doctrine](http://rubyonrails.org/doctrine) as of 20.05.2017.

[25] Bootstrap, URL: [getbootstrap.com](http://getbootstrap.com) as of 20.05.2017.

[26] Vis.js, URL: [visjs.org](http://visjs.org) as of 20.05.2017.