Krome *et al. Energy Informatics* (2019), 2(Suppl 1): 17
https://doi.org/s42162-019-0076-0

Energy Informatics

RESEARCH Open Access

Clustering time series applied to energy markets

Cornelia Krome*, Jan Höft and Volker Sander

From The 8th DACH+ Conference on Energy Informatics,
Salzburg, Austria, 26-27 September, 2019

*Correspondence:
krome@fh-aachen.de
[1]FH Aachen - University of Applied Sciences, Heinrich-Mußmann-Str. 1, Jülich, Germany

Abstract

In Germany and many other countries the energy market has been subject to significant changes. Instead of only a few large-scale producers that serve aggregated consumers, a shift towards regenerative energy sources is taking place. Energy systems are increasingly being made more flexible by decentralised producers and storage facilities, i.e. many consumers are also producers. The aggregation of producers form another type of power plants: a virtual power plant.

On the basis of aggregated production and consumption, virtual power plants try to make decisions under the conditions of the electricity market or the grid condition. They are influenced by many different aspects. These include the current feed-in, weather data, or the demands of the consumers. Clearly, a virtual power plant is focusing on developing strategies to influence and optimise these factors. To accomplish this, many data sets can and should be analysed in order to interpret and create forecasts for energy systems. Time series based analytics are therefore of particular interest for virtual power plants.

Classifying the different time series according to generators, consumers or customer types simplifies processes. In this way, scalable solutions for forecasts can be found. However, one has to first find the according clusters efficiently.

This paper presents a method for determining clusters of time series. Models are adapted and model-based clustered using ARIMA parameters and an individual quality measure. In this way, the analysis of generic time series can be simplified and additional statements can be made with the help of graphical evaluations. To facilitate large scale virtual power plants, the presented clustering workflow is prepared to be applied on big data capable platforms, e.g. time series stored in Apache Cassandra, analysed through an Apache Spark execution framework.

The procedure is shown here using the example of the Day-Ahead prices of the electricity market for 2018.

Keywords: Virtual power plant, Time series, Model-based clustering, Arima, Classification

Introduction

Energy transition is a highly discussed topic. It describes the shift from fossil energy sources to renewable energy and is an important piece for a solution to the global warming problem. While carbon dioxide emissions shall be reduced, the energy supply must be sustainably secured. Formerly, electrical power was generated by a rather small amount

of power plants that adapted their production based on the aggregated demand of a large amount of consumers. Now, energy is produced by many small power plants that were geographically and, with respect to the grid, topologically dispersed. A new concept of power plants is built among these decentralised resources, called virtual power plants (VPP). To address the specific situation of small scale producers, some VPPs also integrate consumers, or so-called prosumers into their model. Their goal is to gain new flexibilities for either the business strategy on the energy stock exchange market or stable grid operation.

The German Ordinance on Electricity Network Access stipulates that energy consumption and production must be in balance at all times in an electricity grid (Stromnetzzugangsverordnung - StromNZV, section 4). Any inconsistency in the load generation balance leads to a violation of the grid parameters (Latha et al. 2011). A VPP is also subject to these requirements. To this end, the grid is continuously monitored and generation is adapted to consumption.

In order to obtain flexibility options, the adjustable control of power-consuming processes, so-called demand side management (DSM), is necessary. For this purpose, the demand for electricity is controlled by means of targeted addition and subtraction of loads on the basis of the price level. (Deutsche Energie-Agentur GmbH (dena) 2016)

Demand side management is often realised with the help of storage elements. Storage elements are a good example for the prosumer model since they can "consume" (store) or "produce" (inject) energy. Also, various consumers can shift their load. Charging an electrical car does not always have to be done as quick as possible. During their working time, employees might rather accept a deadline charging approach in which the car is charged by a specific time, i.e. 5pm.

The optimisation steps of a virtual power plant depends on many input values. The power feed-in of all producers, current consumptions, the weather conditions and energy prices are only a few examples for variables that all have to be forcasted.

Decisions on buying or selling energy can be made by knowing future prices or their behaviour. A VPP makes the energy consumption and production more flexible to reduce costs. For a profitable DSM future knowledge is indispensable.

To interpret the input parameters as time series, "a sequence of observations taken sequentially in time" (Box et al. 1994, p. 1), is much more useful. Time series models can be used for forecasting, model specification, process control and estimation.

An algorithm to analyse generic time series and cluster the results is of great interest. To process multiple time series simultaneously, this algorithm can be applied to Apache Spark as execution framework. Apache Cassandra could provide a distributed storage infrastructure for the time series, as it is done in (Krome and Sander 2018).

A VPP needs multiple forecasts for its DSM. Among others energy prices and the actual generation of different production types, i.e. offshore wind and hydro pumped storage, are used.

Based on the result of a cluster analysis potential action strategies for virtual power plants could be derived. A use case of VPPs may be clustering clients. Here, an algorithm groups a set of objects in such a way, that the objects in the same group (cluster) are more similar to each other than those in other groups (clusters). Based on historical consumptions patterns might be found, which can be used to cluster different clients. For example, some clients such as a bakery consume most energy in the morning,

while other clients such as a dining restaurant have a high consumption in the evening. Further analyses may get a better quotation for each customer, so that they have to pay less for energy and the virtual power plant can predict the consumption more precisely.

Another application is the control of loads by energy prices and the actual generation. If a VPP knows the price trend of the electricity and can react to forecasted price signals in short or medium term, the electricity costs can be reduced. Thus the value of the additional or less consumed electricity can be estimated. This enables to inform customers about potential savings. As storage option hydro pumped storages are a highly used option (Latha et al. 2011; Béguin et al. 2014). The control of the storages depends on the current state of the power grid. General statements and predictions are helpful to obtain an assessment of the storage behaviour. A VPP depends on such knowledge.

Explanatory for the described problem, the Day-Ahead prices of the electricity market for 2016 - 2018 are analysed. Every day represents one time series of 24 values. Additionally, the aggregated generation of hydro pumped storages are considered. One day is reflected in 96 quarter h of this data. Each of these can be modelled with autoregressive-integrated-moving average (ARIMA) methods. Afterwards, different quality measures are considered to cluster time series. The different years are compared with each other and searched for commonalities.

The paper is organised as follows. First, an overview to related work is presented. Ensuing, a short introduction to time series analysis is given in "Time series analysis" section. "Classification or clustering" section summarises the difference between classification and clustering and presents the used algorithm. In "Methodology" section an exemplary clustering is carried out. The results of the clustering of 2018 are given in "Empirical evaluation" section. Additionally, the Day-Ahead prices of 2016 and 2017 and the daily net generation output of hydro pumped storages are analysed. A conclusion and an outlook are provided in "Conclusion and future work" section.

Related work

Other researchers have presented techniques to cluster electricity price time series with K-means or Fuzzy C-means (Martínez-Álvarez et al. 2007) or to forecast those time series with GARCH models (Härdle and Trueck 2010), an adaptive non-parametric regression approach (Zareipour et al. 2006) or based on the Weighted Nearest Neighbours method (Lora et al. 2007). In (Zhou and Chan 2014) a model-based multivariate time series clustering algorithm is presented, where the clusters are created with K-means. However, the approaches do not use ARIMA orders to group similar models. Consequently, it is necessary to discover patterns in the electricity prices time series models to provide better forecasting capabilities.

Time series analysis

By analysing time series the focus of the analysis is the dependence between adjacent observations. A very important feature of time series is stationarity. The joint probability function of a stationary model does not change when the stochastic process is shifted in time. So, it is important to distinguish between stationary and non-stationary time series. Former are adapted with autoregressive and moving average models or a mixture of both.

A stationary behaviour of non-stationary time series can be obtained by analysing the dth difference of the process. (Box et al. 1994)

Stationary time series

Stationarity means, that the process is not evolving over time. There are two main stochastic models to analyse stationary time series: *autoregressive* (AR) and *moving average* (MA) models.

AR models represent the current value of a process as a finite, linear combination of previous values of the process and a random noise ω. The AR model of order p, $AR(p)$, is indicated as in (Box et al. 1994):

$$\tilde{x}_t = \sum_{i=1}^{p} \phi_i \tilde{x}_{t-i} + \omega_t = \phi_1 \tilde{x}_{t-1} + \phi_2 \tilde{x}_{t-2} + \ldots + \phi_p \tilde{x}_{t-p} + \omega_t \tag{1}$$

where $\tilde{x}_t$ is the time series value, ω_t is the random noise and ϕ_i are the model coefficients. A linear model relates a dependent variable or a set of independent variables and a random error term. It is referred to as a *regression model*. This model is named autoregressive because $\tilde{x}$ is regressed by previous values of itself.

While an AR model (1) relates $\tilde{x}$ to its previous values as weighted sum, MA models represent $\tilde{x}$ linearly dependent on a finite number q of previous random noise ω's. They consider the error of the process. The MA of order q, $MA(q)$ is (Box et al. 1994):

$$\tilde{x}_t = \sum_{j=0}^{q} \theta_j \omega_{t-j} = \omega_t + \theta_1 \omega_{t-1} + \ldots + \theta_q \omega_{t-q} \tag{2}$$

where θ_j are the model coefficients with $\theta_0 = 0$.

With a mix of AR (1) and MA (2) more flexibility is achieved. This leads to *autoregressive-moving average* (ARMA) models of order p and q, $ARMA(p, q)$ (Box et al. 1994):

$$\tilde{x}_t = \sum_{i=1}^{p} \phi_i \tilde{x}_{t-i} + \sum_{j=0}^{q} \theta_j \omega_{t-j}$$

where ϕ_i are the model coefficients of the AR part and θ_j are those of the MA part.

Non-stationary time series

To analyse non-stationary time series *autoregressive-integrated-moving average* (ARIMA) models are used. Non-stationary time series can be transformed so that the new time series is stationary.

The Box-Cox transformation stabilises the variance of the process, so that stationarity is obtained. It is defined as in (Box and Cox 1964):

$$x_t^{(\lambda)} = \begin{cases} \frac{(x_t + c)^{\lambda} - 1}{\lambda} & \text{for } \lambda \neq 0 \\ \ln(x_t + c) & \text{for } \lambda = 0, \end{cases} \tag{3}$$

where c is constant. The parameter λ may be estimated, i.e. with a maximum likelihood estimation. For this paper, the method of (Guerrero 1993) is used, where λ minimises the coefficient of variation for subseries.

Then, an ARIMA model can be adapted. In particular, if there are d unit roots, the ARIMA model of parameters p, d and q, $ARIMA(p, d, q)$, is indicated as in (Box et al. 1994):

$$\phi(B)(1 - B)^d x_t = \theta(B)\omega_t,$$

where B is the backward shift operator, which is defined as $Bx_t = x_{t-1}$. The functions ϕ and θ of B are defined as

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \quad \text{and} \quad \theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q.$$

$\phi(B)$ represents the AR part and $\theta(B)$ the MA part of the model.

Interpretation of model parameters

The predicted value of a time series Y is composed of a weighted sum (or constant term) of recent values of Y and a weighted sum (or constant term) of recent values of errors. The parameters of an $ARIMA(p, d, q)$ model have the following meaning:

p: number of lagged autoregressive terms,
d: number of non-seasonal differences needed for stationarity, and
q: number of lagged forecast errors in prediction equation.

The interpretation of ARIMA models depends on the concrete values of the model parameters p, d and q. Looking at the number of differences (d) subsequent remarks can be made, letting y be the dth difference of Y, $y = Y^{(d)}$:

d=0: $y_t = Y_t$,
d=1: $y_t = Y_t - Y_{t-1}$,
d=2: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$.

The value of p describes the order of the autoregressive model. With an $ARIMA(1, 0, 0)$ model Y is regressed on itself lagged by one period, e.g. the stock price of the hour before. For $p = 2$ it is regressed on itself lagged by two periods and so on. If $p = q = 0$ and $d = 1$ ($ARIMA(0, 1, 0)$) the model describes a non-stationary random walk. ARIMA and ARMA models are related. By d-fold integration of the $ARMA(p, q)$ process an $ARIMA(p, d, q)$ model is achieved.

Model selection

A fitted model is satisfactory, if the model residuals are uncorrelated and resemble white noise. Whether a model is appropriate pertaining the randomness of the time series can be tested with the Ljung-Box test (Ljung and Box 1978):

H_0: The data are uncorrelated.
H_1: The data are correlated.

The test statistic is defined as:

$$Q = n\,(n+2) \sum_{k=1}^{h} \frac{\hat{\rho}_k^2}{n-k},$$

where n is the sample size, h is the number of tested lags and $\hat{\rho}_k$ is the autocorrelation at lag k. The test is applied to the residuals of a model.

There exist multiple algorithms to fit ARIMA and ARMA models. For example, the algorithm *auto.arima* (Hyndman and Khandakar 2008) chooses the model order d by a unit root test in the beginning. Afterwards, different $ARIMA(p, d, q)$ models are fitted to the given time series.

As d represents the number of differences used to get stationarity, only p and q need to be selected for a good fit. It is suggested to use the Akaike information criterion (AIC)

(Akaike 1973) as prime criterion for model selection (Brockwell and Davis 2006). The AIC statistic is defined as:

$$AIC_{p,q} := -2\ln(\text{maximum likelihood}) + 2(p+q+1),$$

where $p+q+1$ denotes the number of parameters estimated in the model. The last part, $2(p+q+1)$, is a penalty factor for inclusion of additional parameters. (Box et al. 1994)

The algorithm *auto.arima* iterates p and q and selects the model with the lowest AIC as best fit.

Classification or clustering

The goal of classification and clustering is to organise and categorise large data sets. The big difference between both methods is their a priori knowledge. A short clarification of terminology is given based on (Ceri et al. 2013).

Classification

Classification algorithms assign an object or observation to one or more categories. Each observation is characterised by quantifiable properties or features. Based on human annotated data the supervised learner is trained initially, where it assigns a class to each data item. Some example for classification methods are

- Naive Bayes,
- Regression Classifiers,
- Decision Trees, and
- Support Vector Machines.

Clustering

A clustering algorithm is an unsupervised learner. No a priori knowledge is needed to group observations into categories. The groups are a measure of inherent similarity between the including instances. The objects within one cluster are very similar to each other (compactness property) and are very different to observations in other clusters (separateness property).

The steps of a cluster analysis are as follows:

1. Process the input data and characterise each observation with descriptors, so called features.
2. Select a similarity function and calculate the distance matrix.
3. Analyse the data set for clusters with the chosen similarity function to get clusters with best possible compactness and separateness properties.
4. Validate the clusters by so called validity indexes.
5. Label the clusters.

Description of the algorithm

The steps of the algorithm are visualised in Fig. 1. The Day-Ahead prices of the electricity market for 2018 are loaded from (ENTSO-E Transparency Platform) and each day gets transformed to a time series of 24 observations. The time series are the input data for the algorithm.

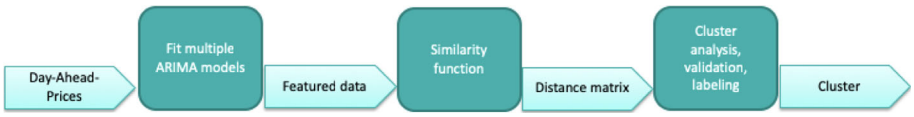

Fig. 1 Workflow of the algorithm

First, all time series are tested for stationarity. If the augmented Dickey-Fuller test (Said and Dickey 1984) declines stationarity, a Box-Cox transformation (Box and Cox 1964), using (3), is performed. Afterwards, multiple ARIMA models are fitted with *auto.arima* (Hyndman and Khandakar 2008). The parameters of each model serve as features of the time series. The values are saved as featured data.

The similarity function for the ARIMA models is the corresponding AIC. A small modification can be used to put more weight on the complexity of the model, represented by p and q (see (4)).

Using AIC as similarity function and the ARIMA parameters as features of the time series, the main algorithm detects clusters within the vector of assigned models. A cluster is valid if there is at least one time series, which is properly described by the model corresponding to the cluster. This is tested with a Ljung-Box test (Ljung and Box 1978), which analyses the autocorrelation of the model residuals.

The valid clusters get labeled by their group name in form of triplets of (p, d, q), the ARIMA orders of the associated models.

Categorisation of the algorithm

The developed algorithm groups time series based on features without a priori knowledge. A training phase with human annotated observations is not necessary. Instead of statistical classification methods the AIC serves as similarity function. The detected groups get validated and labeled. Overall, the algorithm performs a model-based clustering approach as described in (Warren Liao 2005), where the time series are modelled and coefficients or residuals get clustered (Warren Liao 2005, Fig. 1). While a priori knowledge is not necessary, the interpretation of the clusters should rely on domain specific features.

Methodology

This section explores the data and summarises the methods used for analysis. The algorithm described here has been implemented in *R* and evaluated for the Day-Ahead prices. The aim is to analyse clusters identified by the algorithm.

Data

For analysis, three sets of Day-Ahead prices have been used, collected by ENTSO-E Transparency Platform (ENTSO-E Transparency Platform). These are the electricity prices for an hour per day for the years 2016, 2017 and 2018. In the following sections a detailed analysis of the data of 2018 are given. The data for 2016 and 2017 will be evaluated later. The results will be compared with those of 2018. Furthermore, the aggregated generation of hydro pumped storages for 2018 are clustered.

The energy price relies on many aspects and has a very special behaviour (Härdle and Trueck 2010). Scarcity and abundance have high impacts. Additionally, the price is related to performance. Supply and demand have to coincide at every point in time at every place in Germany (Stromnetzzugangsverordnung - StromNZV, "Classification or clustering" section). So, shortages can not be equalised by profusion at another place or time. (Borchert et al. 2006) As displayed in Fig. 2a the Day-Ahead price is as volatile as stocks.

Having a look at the prices throughout a year, some conspicuous features arise. Every day has to be analysed separately and holidays vary greatly from normal weekdays. The mean price within a week is mostly constant. A systematic decrease at weekends can be observed. (Borchert et al. 2006) In Fig. 2b one exemplary week (calender week 2) is displayed. The mentioned effects can be recognised.

There is a high demand fluctuation that is originated by business hours at a daily level as seen in Fig. 2b or changing climate conditions. Consequently, energy prices have a strong seasonal component (Härdle and Trueck 2010). There are big differences between winter and summer. In summer the energy around noon is more expensive. In winter there is a peak in the evening. (Borchert et al. 2006) This phenomenon can be seen in Fig. 3.

The Day-Ahead price on January 30, 2018 increases into the evening hours, where a peak is reached around 7pm. The plot of the Day-Ahead prices of July 21, 2018 peaks around noon. April 1, 2018 was Easter Sunday. The Day-Ahead prices differ greatly from those of normal weekdays. So, holidays need to be analysed separately. While clustering can be handled without a priori knowledge, domain specific knowledge is either used to filter the data before clustering or during the interpretation process.

Fitting models

The loaded Day-Ahead prices from (ENTSO-E Transparency Platform) are converted into $n = 365$ time series with 24 knots each. In effect, the vector

$$\mathbf{ts} := (ts_1, \ldots, ts_n)$$

is considered. The bold font indicates vectors. Before models can be adapted stationarity must be checked and if necessary be transformed with the Box-Cox transformation (Box and Cox 1964), (3). One example for a non-stationary time series are the values for December 23. The variance differs throughout the day. The variance-stabilisation transformation is applied to the time series to bring the non-normal dependent variables into

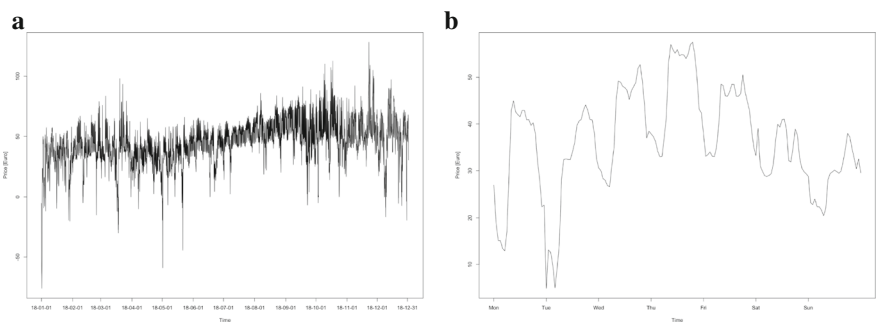

Fig. 2 Volatility of energy prices. **a** Day-Ahead prices in 2018. **b** Day-Ahead prices of one week

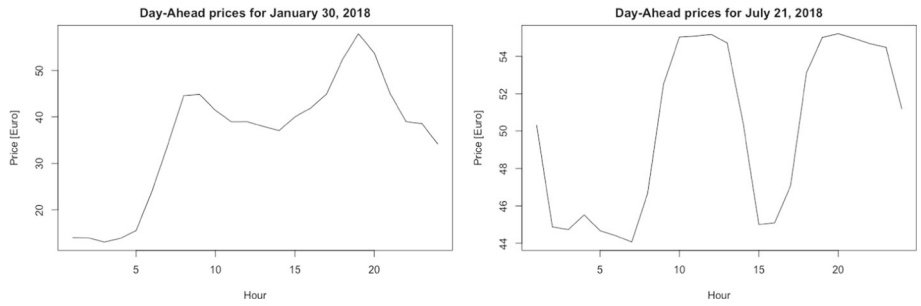

Fig. 3 Two examples for Day-Ahead prices 2018

normal shape. Since the logarithm is applied to the data, the new values differ greatly from the original ones in magnitude.

In Fig. 4 the time series before and after transformation is displayed. While the original prices wobble between 10*e* and 70*e*, the transformed prices wobble between 30*e* and 270*e*. The transformation is necessary for a model adaption, but the values are not relevant for the ARIMA parameters. Those values would be important for prediction, in which case the inverse Box-Cox transformation should be applied.

After transformation, multiple ARIMA models are fitted with *auto.arima* (Hyndman and Khandakar 2008). To get all tested models, the option *trace* is set to *TRUE* and *stepwise* to *FALSE*. A drift is not allowed. The *trace* of *auto.arima* yields all models. Those with an infinite AIC are eliminated and a validation with the Ljung-Box test (Ljung and Box 1978) is performed. Models, whose AIC is greater than the median of all AICs, are discarded.

On average, the median AIC of the different models per time series is greater than 52% of all AICs. It is also about 7% larger than the minimum AIC per time series. The restriction of the models to all those whose AIC is smaller than the median corresponds to a confidence interval with a flexible confidence level for each time series (day). On average, the level is beneath 10%.

The resulting models represent a set of m "proper models". Here m can differ for each day, because of the elimination process described above. The model parameters, as well as the AIC for each model, are saved in vectors **p**, **q** and **AIC**, where

$$\mathbf{p} := [p_1, \ldots, p_m]^T, \quad \mathbf{q} := [q_1, \ldots, q_m]^T, \text{ and } \quad \mathbf{AIC} := [AIC_1, \ldots, AIC_m]^T.$$

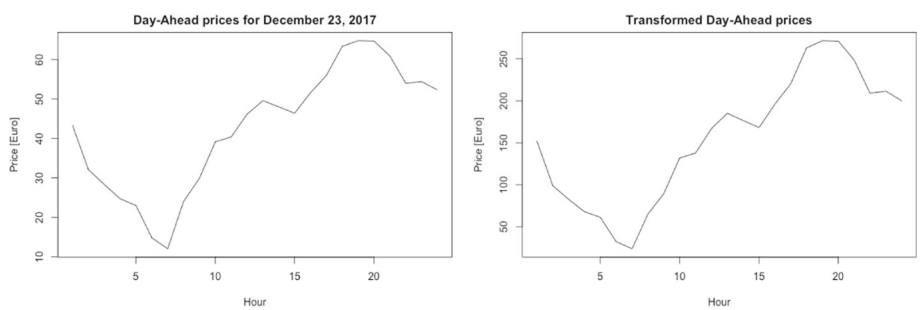

Fig. 4 Day-Ahead prices of December 23, 2018 and transformed time series

Cluster analysis

After all time series have been analysed, the data sets are scrutinised for clusters. The different time series are categorised by the ARIMA parameter d. An algorithm is performed which selects the amount of clusters so that every time series can be adapted to one model. The resulting clusters for 2018 are listed in Table 1, grouped by the value of d.

While there are five or six clusters for $d = 0$ and $d = 1$ there exists only one cluster for $d = 2$. Only five time series were adapted by an $ARIMA(p, 2, q)$ model. After the above clusters are determined, every time series is mapped to a cluster by a similarity function. The first approach is, to use the AIC as similarity function to select the best model. So, any fixed day k is classified as $ARIMA(p_l, d_k, q_l)$, where

$$l := \text{argmin}(\text{AIC}),$$

where $\text{argmin}(\mathbf{x})$ for any vector $\mathbf{x} \in \mathbb{R}^m$ is defined as

$$\{l : x_l = \min(\mathbf{x})\}.$$

The above index is assumed to be unique.

As stated by (Ruppert et al. 2003), "all models with reasonably small [...] AIC values should be considered as potentially appropriate and evaluated according to their simplicity[...]". Hence, one might rather want select a less complex model over the one with the lowest AIC to gain more stability.

The ARIMA orders p and q indicate the complexity of a model. In order to reduce the model complexity, another similarity function is contemplated, namely the mAIC. First, define for any vector $\mathbf{x} = [x_1, \ldots, x_m]^T$ the map

$$\bar{\mathbf{x}} : \mathbb{R}^m \ni \mathbf{x} \longrightarrow \frac{\mathbf{x}_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \in \{\mathbb{R} \cap [0, 1]\}^m, \forall i \in \{1, \ldots, m\}.$$

Now, the mAIC is defined by a linear combination of the normalised versions of AIC and the model complexity as

$$\mathbf{mAIC} = (1 - \alpha) \cdot \overline{\mathbf{AIC}} + \alpha \cdot \overline{\mathbf{p} + \mathbf{q}}. \tag{4}$$

As a result of the elimination process the AIC of all models tend to be very similar. The average standard deviation is approximately two. Additionally, the AIC involves the logarithm of the likelihood function. Thus, the parameters p and q might be undervalued, because of their low valence. Only models with low AIC (smaller than the median) are included in the analysis. The attention is already focused on good approximations. The results are stable for $\alpha \geq 0.75$. For this consideration $\alpha = 0.9$ was chosen. It puts the complexity of the models in foreground, but the measure still includes the AIC in the rating.

Here, $ARIMA\ (p_{l^*}, d_k, q_{l^*})$ is assigned to time series k, where

$$l^* := \arg\min(\mathbf{mAIC}).$$

Table 1 Clusters grouped by d

d	amount of cluster	associated models as (p,d,q)
0	5	(2,0,0), (3,0,0), (4,0,0), (1,0,1), (2,0,1)
1	6	(1,1,0), (2,1,0), (3,1,0), (0,1,1), (1,1,1), (0,1,2)
2	1	(1,2,0)

Next, the resulting models are saved as triplets

$$\mathbf{model}_k := (p_l, d_k, q_l) \quad \text{and} \quad \mathbf{model}_k^* := (p_{l^*}, d_k, q_{l^*}).$$

This way, two vectors for the assigned models are obtained

$$\mathbf{model} := [\mathbf{model}_1, \ldots, \mathbf{model}_n]^T \quad \text{and} \quad \mathbf{model}^* := [\mathbf{model}_1^*, \ldots, \mathbf{model}_n^*]^T.$$

The modification in 4 allows to trade small increases in AIC for the selection of less complex models.

Empirical evaluation

The labeled clusters are returned and are displayed for further analyses. This section elucidates these plots and analyses and compares the different clusterings.

Visual analysis

For visualisation purposes, the time series are displayed according to their weekday. Each graph shows one specific weekday. An extra plot shows the adapted models for German public holidays:

- 01/01/18 - New Year's Day
- 03/30/18 - Good Friday
- 04/01-02/18 - Easter Sunday and Monday
- 05/01/18 - Labor Day
- 05/10/18 - Ascension
- 05/20-21/18 - Whit Sunday and Monday
- 10/03/18 - German Unification Day
- 10/31/18 - Reformation Day
- 11/01/18 - All Saints' Day
- 12/24/18 - Christmas Eve
- 12/25-26/18 - Christmas Days
- 12/21/18 - New Year's Eve

Those groupings are made for presentation and optical analysis. This a priori knowledge is not required for the algorithm.

Each time series is represented by an item in the plot. Their coordinate is a triplet of their cluster, labeled by (p, d, q), and the date. In addition, there is a binary property of the models: zero mean or non-zero mean. Accordingly, the time series is represented as a point or a triangle.

A tick on the main y-axis (left) marks the first day of each month. The German school holidays (see Table 2) are marked by horizontal lines and are named on the right y-axis:

Table 2 Earliest start and latest end of school holidays in Germany 2018

Start	End	Holidays
	01/07/18	Christmas holidays
03/24/18	04/07/18	Easter holidays
06/30/18	09/17/18	Summer holidays
09/29/18	10/28/18	Autumn holidays
12/20/18		Christmas holidays

As only five days are adapted with $d = 2$ the cluster $(1, 2, 0)$ is not shown for clarity reasons. It is interesting that those are winter days. The vertical line between clusters $(2, 0, 1)$ and $(1, 1, 0)$ splits the plot into models with $d = 0$ on the left and $d = 1$ on the right.

Generally spoken, time series that are adapted with a model with $d = 0$ have constant mean over time. Models with $d = 1$ include a linear increase (decrease) as trend in the process. That could be an indicator for low (high) prices in the morning and rising (plummeting) levels throughout the day.

In a model the mean is a constant term. When differentiating a function, a constant term is omitted and the mean is zero. This is how it behaves with ARIMA models with $d > 0$. As a result, time series adapted to models with $d > 0$ are plotted as dot (zero mean). Only time series with an $ARMA(p, q)$ or $ARIMA(p, 0, q)$ can have a non-zero mean. Those are displayed as triangle. The color of each item represents the corresponding cluster.

Clustering with AIC

In Fig. 5 the Day-Ahead prices clustered with AIC as similarity function are displayed. Overall, twelve clusters were discovered.

The amount of corresponding time series per cluster and day is presented in Fig. 6. Four main clusters can be found: $ARIMA(2, 0, 0)$, $ARIMA(1, 1, 0)$, $ARIMA(0, 1, 1)$ and $ARIMA(0, 1, 2)$.

For each day the distribution of the clusters differ. But some similarities can be found.

For Mondays and Sundays many time series are adapted by models with $d = 1$. If $d = 0$ then the time series are mostly modelled with $ARIMA(2, 0, 0)$. This is more likely in summer months.

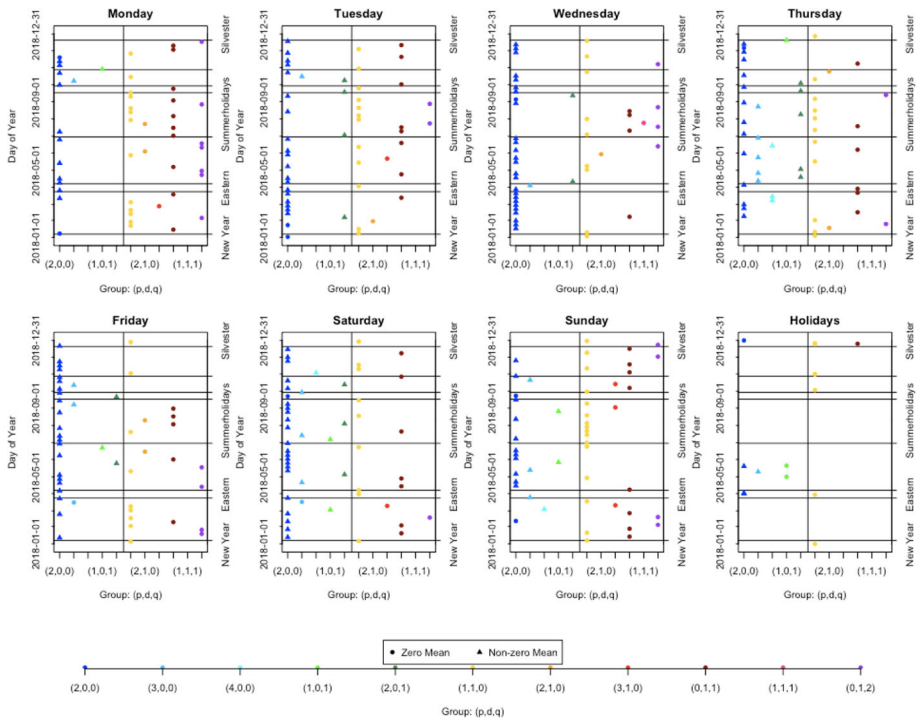

Fig. 5 Cluster of the Day-Ahead prices with AIC as similarity function

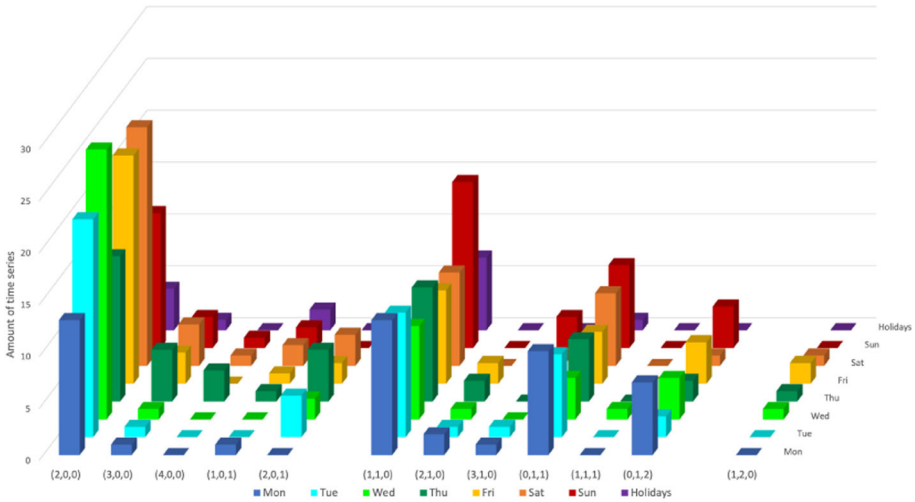

Fig. 6 Amount of time series per cluster (AIC) per weekday

For Tuesdays through Saturdays most time series are adapted with an $ARIMA\ (2,0,0)$ model. Tuesdays are evenly divided between $ARIMA(2,0,0)$ and models with $d = 1$. On Wednesdays $ARIMA(2,0,0)$ is very dominant as well, but during summer some days are adapted by $d = 1$-models. For Thursdays $ARIMA(1,1,0)$ models are very likely, too. Fridays appear to be very similar to Tuesdays. So, in summer months many time series are adapted with $d = 1$ models. This does not apply to Saturdays. There are nearly no observations with $d = 1$.

For public holidays models with $d = 1$ are dominant in winter. During summer all time series are adapted by models with $d = 0$.

Clustering with mAIC

The second clustering uses the mAIC (4) as similarity function, where the model complexity receives greater attention, than in AIC itself. The new clusters are presented in Fig. 7.

Three groups attract attention: $ARIMA(2,0,0)$, $ARIMA(1,1,0)$ and $ARIMA\ (0,1,1)$ (see Fig. 8). They are simpler than those of AIC-clustering. Overall, 337 days are modelled with one of these. Half of the year (185 days) are modelled with an $ARIMA(p,0,q)$.

As seen in the previous part, the distributions of clusters differ for each day (see Fig. 7). There are some main differences, compared to Section 1. For all days simpler models are chosen more often. In this clustering Fridays attract attention. A shift from models with $d = 1$ at the beginning of the year to models with $d = 0$ in the end can be observed.

Comparison of similarity measures

An overall comparison between both clusterings by the amount of time series per cluster is given in Table 3. A distinction is made between the two similarity functions (AIC and mAIC).

When using the AIC as similarity function there is a lot of variation between clusters and complex models are chosen more often. The modified version uses simple models

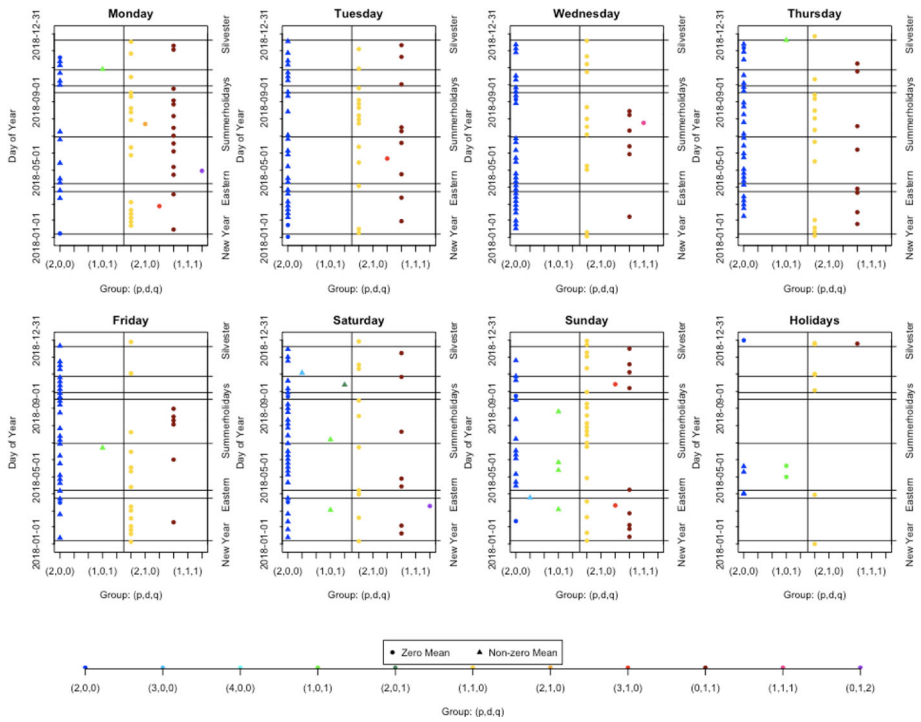

Fig. 7 Cluster of the Day-Ahead prices with mAIC as similarity function

instead. This shift can be seen in Table 3 and Fig. 9. An arrow symbolises the change from one model to another. A dot represents no change.

For $d = 0$ clustering with the AIC reveals 136 days in cluster $(2, 0, 0)$, 19 in $(3, 0, 0)$, 16 in $(2, 0, 1)$ and 9 in $(1, 0, 1)$. With the mAIC as similarity function two days are displaced into cluster $(1, 0, 1)$, one from $(3, 0, 0)$ and one from $(4, 0, 0)$. These are strong simplifications. Additionally, 35 days are shifted into group $(2, 0, 0)$, which is much easier than the others.

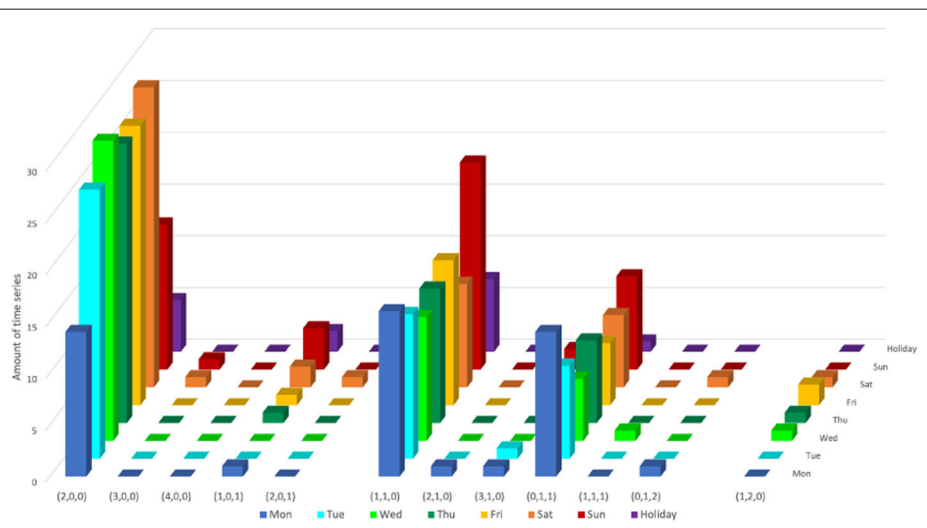

Fig. 8 Amount of time series per cluster (mAIC) per weekday

Table 3 Amount of time series per cluster

Model	AIC	mAIC
(2,0,0)	136	171
(3,0,0)	19	2
(4,0,0)	5	0
(1,0,1)	9	11
(2,0,1)	16	1
(1,1,0)	86	106
(2,1,0)	8	1
(3,1,0)	6	4
(0,1,1)	49	60
(1,1,1)	1	1
(0,1,2)	24	2
(1,2,0)	5	5

Now, no day is clustered into $(4, 0, 0)$. Only two days are adapted by an $ARIMA(3, 0, 0)$ and one with $(2, 0, 1)$.

An analogous behaviour can be seen for $d = 1$. Originally, three clusters were chosen. With mAIC only two groups are favoured: $(1, 1, 0)$ and $(0, 1, 1)$. Eleven days are shifted into group $(0, 1, 1)$ and 20 days into $(1, 1, 0)$.

In clusters with $d = 2$ no change has happened, because there is only one cluster. In general, the higher d must be chosen, the lower p and q are. Accordingly, the models for higher ds become simpler.

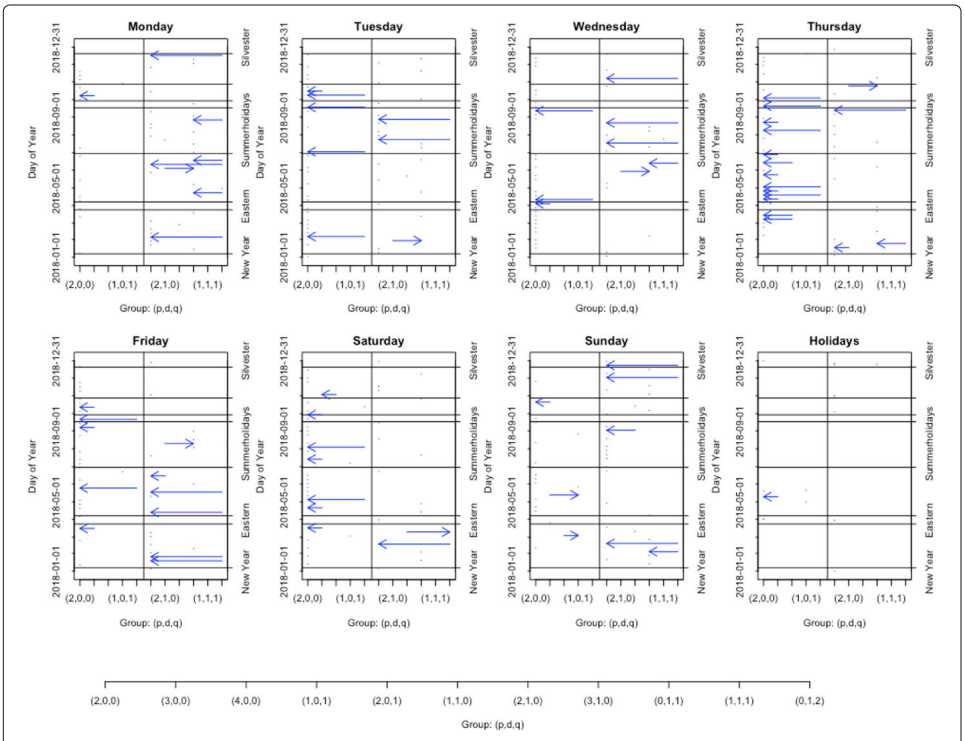

Fig. 9 Cluster difference between AIC and mAIC

So, there is a higher concentration on clusters $(2, 0, 0)$, $(1, 1, 0)$ and $(0, 1, 1)$. 337 days are grouped into those clusters. This amounts to 92% of all days in a year. The order of those models is lower and an analysis easier.

While the clustering results achieved by mAIC lead to models with a significant reduction in complexity, their deviation to the originally used similarity function (AIC) is less than 0.2% increase on average. If the models are assessed to be similar, mAIC helps on finding simpler models.

Using the domain specific knowledge, the presented results can be used to formulate statements that can be of significant use for specifying a business strategy: many Saturdays are adjusted with an $ARIMA(2, 0, 0)$ model. Thus, the next hour can be predicted from the previous two hours. The error terms are not considered in the model. On Sundays it is noticeable that also the $ARIMA(2, 0, 0)$ model is frequently used. In winter months, however, a trend is found in the data. The use of an $ARIMA(p, 1, q)$ model is suggested. Many Mondays are adapted to models with a trend ($d = 1$). No general statements can be made for Thursdays and public holidays. The Day-Ahead prices of these days are approximated quite differently.

Analysis of Day-Ahead prices for 2016 and 2017

To review the algorithm, the Day-Ahead prices for 2016 and 2017 are analysed. Of course, the holidays and vacation periods were adjusted for both years.

For both years similar observations can be made. As in 2018 most days in 2016 (82%) can be modelled by $ARIMA(2, 0, 0)$, $ARIMA(1, 1, 0)$ and $ARIMA(0, 1, 1)$. All days are grouped into ten clusters. As seen be before, clustering with AIC results in more groups with complex models than with mAIC, where simpler models are preferred (see Table 4). The distribution of clusters per weekday is similar to 2018, too (compare Fig. 10). Most Saturdays are adjusted with $ARIMA(2, 0, 0)$. On Sundays models with parameters $(2, 0, 0)$

Table 4 Amount of time series per cluster 2016 and 2017

Model	AIC	mAIC
(2,0,0)	159	172
(3,0,0)	17	12
(4,0,0)	5	2
(1,0,1)	9	9
(2,0,1)	9	4
(1,1,0)	82	89
(2,1,0)	18	15
(0,1,1)	40	42
(0,1,2)	23	17
(1,2,0)	4	4
(2,0,0)	148	171
(3,0,0)	20	5
(4,0,0)	8	0
(1,0,1)	11	11
(1,1,0)	153	167
(2,1,0)	21	7
(2,2,0)	4	4

are used, but a trend is often found in the data. The use of a model with $d = 1$ is suggested. As in 2018, the days mapped with $ARIMA(1, 2, 0)$ are only during winter.

Looking at the clusters for 2017 only seven of them can be found. This also results in less favoured models: only two models are preferred: $ARIMA(2, 0, 0)$ and $ARIMA(1, 1, 0)$ (see Table 4 and Fig. 11). However, similar statements as for 2018 and 2016 can be made regarding parameter d. Saturdays are often modelled without a trend and an $ARIMA(2, 0, 0)$ should be chosen. As in the other years, the values on Sundays include a trend. An $ARIMA(1, 1, 0)$ is suggested. Approximately 93% days of 2017 can be expressed with one of the most favoured models. As in 2016 and 2018, the days corresponding to a model with $d = 2$ are only during winter.

If a VPP knows the price trend of the electricity and can react to forecasted price signals in the short or medium term, the electricity costs can be reduced. Thus the value of the additional or less consumed electricity can be estimated. This enables to inform customers about potential savings.

Generation of hydro Pumped Storage

A VPP does not only use price data for DSM. Generation data is also included in the analysis. One example is the daily net generation output (MW) per market time unit and per generation unit of 100 MW or more installed generation capacity of hydro pumped storages (aggregated by (ENTSO-E Transparency Platform)). The daily generation can be analysed with the introduced algorithm. Since this behaviour differs from that of Day-Ahead prices, other models are chosen for approximation and other statements are made. As outlined in Fig. 12 there are three main models: with $ARIMA(3, 0, 0)$, $ARIMA(1, 1, 0)$ and $ARIMA(1, 1, 1)$ 264 days (more than 70% of a year) can be described.

In contrast to Day-Ahead prices, models with $d = 0$ are much less frequently selected than models with $d = 1$. Almost no Saturday or Sunday is modelled with an $ARIMA(p, 0, q)$. Almost all days are adjusted with an $ARIMA(p, 1, q)$. This can also be observed for Mondays and Thursdays. Only on a few days in summer a model with $d = 0$ is recommended.

Conclusion and future work

This paper presents a clustering workflow of time series that is targeting the operation of virtual power plants in which optimisation strategies for multivariate problems have to be developed. The paper provides a model-based clustering approach based on ARIMA.

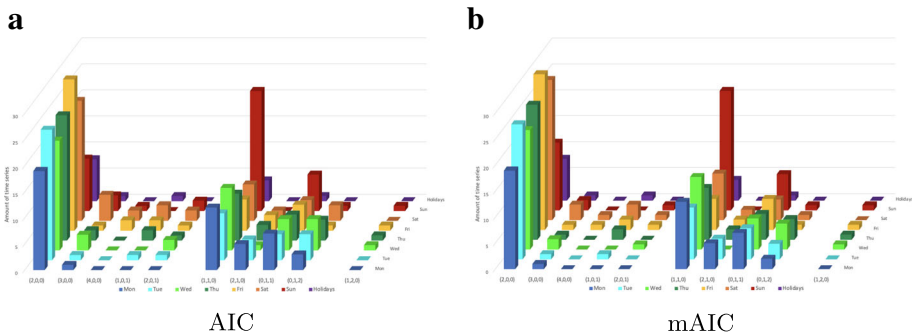

Fig. 10 Amount of time series per cluster per weekday 2016. **a** AIC. **b** mAIC

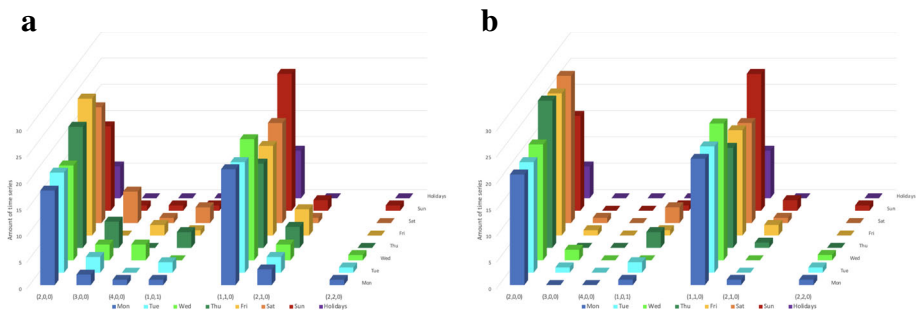

Fig. 11 Amount of time series per cluster per weekday 2017. **a** AIC. **b** mAIC

The first step computes ARIMA models for each time series. The second step computes the clusters based on two different similarity measures, AIC and mAIC. Each cluster is represented by global parameters (p, d, q) for fitted ARIMA models.

However, instead of purely following an *auto.arima*-approach based on AIC, the model selection is optionally done based on an adapted AIC measure: mAIC. The proposal is, however, to not fully replace AIC as indicator, instead, the mAIC approach is introduced, based on AIC, for selecting between models that are assessed to be nearly equivalent in terms of AIC, i.e. to models with a reasonably small AIC. mAIC basically prioritises simpler models for the clustering.

A first analysis of the Day-Ahead prices of the energy market in Germany 2018 resulted in three main clusters. The paper presented a full walk through of the clustering workflow, including transformation of time series to achieve stationarity and the consideration of specific constraints, i.e. assumptions about Saturdays, Sundays and holidays. With those, over 90% of a year can be modelled. Following the proposed mAIC-approach, significantly simpler models were selected during the clustering, with a mean deviation of 0.2% with respect to the original similarity function (AIC).

Comparing the results from 2018 with 2016 and 2017 data, similar observations can be made. The same models were favoured: $ARIMA(2,0,0)$, $ARIMA(1,1,0)$ and $ARIMA(0,1,1)$. In all years it is evident that Day-Ahead prices on Saturdays do not include a trend. That's why Saturdays are usually approximated with an $ARIMA(2,0,0)$. On Sundays, in contrast to Saturdays, a trend is noticeable. Therefore, these days are

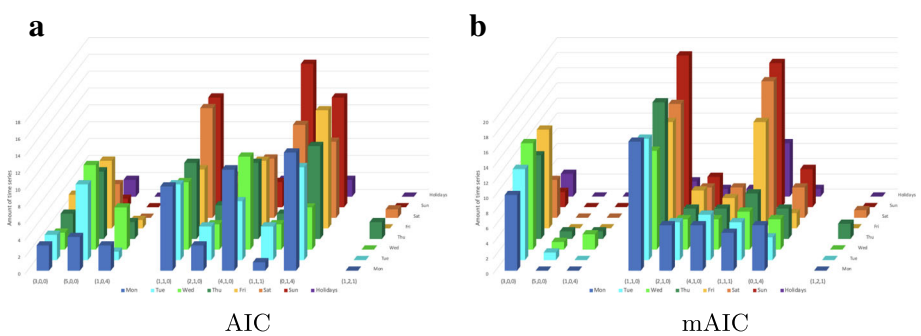

Fig. 12 Amount of time series per cluster for hydro pumped storage **a** AIC. **b** mAIC

mostly modelled with $ARIMA(1, 1, 0)$. With regard to the seasons, similarities can also be identified. Winter days are more often adjusted with models with $d = 0$. On summer days, the time series are often based on a trend, which is why models with $d = 1$ are more likely to be used.

With the introduced algorithm simpler clustering models can now be transformed to business strategies for operating virtual power plants. In order to obtain more meaningful results, the composition of the mAIC will be researched in the future. The current heuristic value for α will be checked for mathematical correlations. Especially for generation data a consideration of other time intervals, e.g. day/night, makes sense. Similarly, the price analysis will be performed on different time scales, i.e. weekly intervals.

The analysis of aggregated generation of hydro pumped storages demonstrated that the algorithm is not only applicable for Day-Ahead prices. In addition to the applications presented it is intended to apply the described workflow to the energy management data of clients of a virtual power plant, to consumption, renewable generation and other market data. Here, the related work (Krome and Sander 2018) on a scalable analytics platform based on Apache Spark and Cassandra will come into place. Apache Cassandra is a database designed to store and handle large amounts of time series data, while the *lapply*-function of Spark will allow an efficient and scalable execution of the presented analysis workflow.

Acknowledgements
We thank our colleague Gerhard Dikta from FH Aachen who provided insight and expertise that greatly assisted the research.

About this supplement
This article has been published as part of Energy Informatics Volume 2 Supplement 1, 2019: Proceedings of the 8th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-2-supplement-1

Authors' contributions
All authors developed the basic idea. CK and JH created the basic concept, carried out the implementation and prepared the first draft of the paper. VS participated in the refinement of the concept and the paper revision. All authors read and approved the final manuscript.

Funding
Publication of this article was partly funded by EFRE.NRW grant. Publication of this supplement was funded by Austrian Federal Ministry for Transport, Innovation and Technology.

Availability of data and materials
Datasets related to this article can be found at https://transparency.entsoe.eu/, under *Transmission* and *Day-Ahead Prices* with area Germany, BZN|DE-AT-LU and years 2016, 2017 and 2018 for the Day-Ahead prices. For the aggregated generation of hydro pumped storages the points *Generation* and *Actual Generation per Production Type* with country *Germany* and year 2018 are to be chosen. (ENTSO-E Transparency Platform)

Competing interests
The authors declare no competing interests.

Published: 23 September 2019

References

Akaike H (1973) Information Theory and an Extension of the Maximum Likelihood Principle. Springer, New York
Béguin A, Nicolet C, Kawkabani B, Avellan F (2014) Virtual power plant with pumped storage power plant for renewable energy integration. IEEE. pp 1736–1742. https://ieeexplore.ieee.org/document/6960417
Borchert J, Schemm R, Korth S (2006) Stromhandel: Institutionen, Marktmodelle, Pricing und Risikomanagement. Schäffer-Poeschel, Stuttgart
Box GEP, Cox DR (1964) An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological) 26(2):211–252. https://www.jstor.org/stable/2984418
Box GEP, Jenkins GM, Reinsel GC (1994) Time Series Analysis: Forecasting and Control. 3rd ed.. Prentice Hall, Englewood Cliffs, NJ
Brockwell PJ, Davis RA (2006) Time Series: Theory and Methods, 2. ed., nachdr. edn. Springer series in statistics. Springer, New York, NY

Ceri S, Bozzon A, Brambilla M, Della Valle E, Fraternali P, Quarteroni S (2013) Classification and Clustering. Springer, Berlin, Heidelberg
Deutsche Energie-Agentur GmbH (dena) (2016) Demand Side Management - Unternehmen als Anbieter für Flexibilität im Energiesystem
ENTSO-E Transparency Platform ENTSO-E Transparency Platform. https://transparency.entsoe.eu/. Accessed 23 Apr 2019
Guerrero VM (1993) Time-series analysis supported by power transformations. J Forecast 12(1):37–48
Härdle W. K., Trueck S. (2010) The dynamics of hourly electricity prices. http://sfb649.wiwi.huberlin.de/papers/pdf/SFB649DP2010-013.pdf
Hyndman R, Khandakar Y (2008) Automatic time series forecasting: The forecast package for r. J Stat Softw Artic 27(3):1–22
Krome C, Sander V (2018) Time series analysis with apache spark and its applications to energy informatics. Energy Inform 1(1):40
Latha PG, Anand SR, Ahamed TPI (2011) Improvement of demand response using mixed pumped storage hydro plant. In: ISGT2011-India. IEEE. pp 183–186. https://ieeexplore.ieee.org/document/6145380
Ljung GM, Box GEP (1978) On a measure of lack of fit in time series models. Biometrika 65(2):297–303
Lora AT, Santos JMR, Exposito AG, Ramos JLM, Santos JCR (2007) Electricity market price forecasting based on weighted nearest neighbors techniques. IEEE Trans Power Syst 22(3):1294–1301
Martínez-Álvarez F, Troncoso A, Riquelme J, Santos J (2007) Discovering patterns in electricity price using clustering techniques. Renew Energy and Power Quality 1
Ruppert D, Wand MP, Carroll RJ (2003) Semiparametric Regression. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, New York
Said SE, Dickey D (1984) Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika 71:599–607
Warren Liao T (2005) Clustering of time series data-a survey. Pattern Recogn. 38(11):1857–1874
Zareipour H., Bhattacharya K., Canizares C. A. (2006) Forecasting the hourly ontario energy price by multivariate adaptive regression splines. In: 2006 IEEE Power Engineering Society General Meeting. IEEE. p 7. https://ieeexplore.ieee.org/document/1709474
Zhou P-Y, Chan K (2014) A model-based multivariate time series clustering algorithm. Springer. https://link.springer.com/chapter/10.1007/978-3-319-13186-3_72

Publisher's Note