Claudio Hartmann, Martin Hahmann, Dirk Habich, Wolfgang Lehner

CSAR: The Cross-Sectional Autoregression Model

Erstveröffentlichung in / First published in:

2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA). Tokyo, 19.-21.10.2017. IEEE, S. 233-241. ISBN 978-1-5090-5004-8.

DOI: http://dx.doi.org/10.1109/DSAA.2017.27

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

CSAR: The Cross-sectional Autoregression Model

Claudio Hartmann, Martin Hahmann, Dirk Habich, and Wolfgang Lehner
Technische Universität Dresden
Database Systems Group
01062 Dresden, Germany
e-mail: <firstname.lastname>@tu-dresden.de

***Abstract*—The forecasting of time series data is an integral component for management, planning, and decision making. Following the Big Data trend, large amounts of time series data are available in many application domains. The highly dynamic and often noisy character of these domains in combination with the logistic problems of collecting data from a large number of data sources, imposes new requirements on the forecasting process. A constantly increasing number of time series has to be forecasted, preferably with low latency AND high accuracy. This is almost impossible, when keeping the traditional focus on creating one forecast model for each individual time series. In addition, often used forecasting approaches like ARIMA need complete historical data to train forecast models and fail if time series are intermittent. A method that addresses all these new requirements is the cross-sectional forecasting approach. It utilizes available data from many time series of the same domain in one single model, thus, missing values can be compensated and accurate forecast results can be calculated quickly. However, this approach is limited by a rigid training data selection and existing forecasting methods show that adaptability of the model to the data increases the forecast accuracy. Therefore, in this paper we present CSAR a model that extends the cross-sectional paradigm by adding more flexibility and allowing fine grained adaptations to the analyzed data. In this way, we achieve an increased forecast accuracy and thus a wider applicability.**

I. Introduction

Nowadays, forecasting of time series data has become an irreplaceable part of management, planning and decision making. Especially, current developments in Smart Grid technologies, where thousands of Smart Meters monitor the energy consumption of individual households, energy providers require reliable forecasts to balance the transmission grid [1], [2]. Another current development is the Internet of Things where thousands of objects are monitored with a multitude of sensors. Here, thorough predictions are necessary to increase the efficiency of production processes and manage stock keeping, all with reduced human intervention [3].

The large number of monitoring devices combined with an intensive data gathering leads to new requirements towards the task of forecasting. As the monitoring granularity becomes finer in structure and time, data sets now consist of thousands of very long time series. This makes the timely provision of forecast results a growing problem. Moreover, the increasingly fine time granularity leads to much noisier and less regular time series, which are smoothed by aggregation effects on coarse granularities. The ability to work with such noisy data is another emerging requirement for forecasting techniques. Furthermore, the growing number of monitoring devices increases the risk for sensor drop outs and transmission errors. This causes missing values and incomplete or intermittent time series which also is an arising requirement.

These new requirements hamper the application of traditional forecasting techniques like ARIMA or Exponential Smoothing, although, they are successfully applied in a wide variety of application scenarios [4]. These techniques only focus on the prediction of one individual time series. This makes the handling of large data sets very time consuming since a large number of models has to be optimized to properly represent each and every time series. Furthermore, it is not possible to compensate for noise or missing values of individual time series since these methods do not involve other data sources during the model creation. Therefore, traditional forecasting techniques are not suited to meet the requirements that we are confronted with by the prediction of large scale time series data sets.

An approach that already addresses the aforementioned requirements is cross-sectional forecasting (CS) [5]. It creates only one single model for all time series of an entire data set assuming that time series from the same domain share a common behavior. But, this approach lacks modeling versatility since it is limited by a very rigid data selection. CS does not represent the actual behavior of every individual time series, but the relative change from one point in time to the next one (like an AR(1) model) for the whole data set and predicts all time series with the same model. Therefore, this technique can compensate noise and missing values in individual time series, since other series from the same data set contribute to the model creation. Furthermore, forecasting a large number of time series in reasonable time is possible as well since only one model has to be optimized. We want to keep these qualities and make the model more adaptable. Because, as the research on other prediction methods shows, adaptability of the model towards the analyzed data leads to a higher forecast accuracy [6], [7].

In this paper we present CSAR the **C**ross-**S**ectional **A**uto**R**egression model. It combines the qualities of cross-sectional forecasting with the adaptability of ARIMA. Therefore, we create a model which meets all the aforementioned requirements and is adaptable to wide range application scenarios. In detail, our contributions are:

- We give an overview on the forecasting process and illustrate the new requirements for the forecasting of large scale time series data sets in more detail in Section II.

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

- Subsequently, we review existing forecast techniques and discuss their abilities and shortcomings regarding the new requirements in Section III. Thereby, we show that none of the existing techniques entirely satisfies all requirements.
- We describe our new CSAR model and how it combines properties of cross-sectional forecasting and ARIMA in Section IV. Furthermore, we introduce the parameters which can be used to adapt CSAR to a specific data set.
- We extensively evaluate our approach on three different real world data sets to show its high forecast accuracy and short execution time in Section V.

Finally, we conclude the paper in Section VI with a short summary and a brief overview on future research directions for the prediction of large scale time series data.

II. Time Series Forecasting

In many application scenarios a lot of data is collected from multiple sources, such that there is a large number of time series originating from the same domain which have to be analyzed, modeled, and forecasted. We refer to such a collection of time series as a data set $\mathcal{Y}$ where all time series Y^n – n is a unique series identifier – are recorded at the same points in time $1, ..., t$.

Usually, the task of forecasting focuses on only one time series Y. A time series is a sequence of values $Y_1, \ldots, Y_t$ where the subscript marks the time at which a value was recorded. Additionally, a time series is assumed to be sorted in ascending order by time, complete, i.e., has no missing values, and equidistant, i.e., all time series values are recorded at regular time intervals. For the forecasting, a model is optimized on the values of the time series to represent them as good as possible and then this model is used to calculate the requested forecast values [8]. Fig. 1 shows an example of a time series represented by the connected black crosses $\times$. The x-axis of the diagram denotes the time and the y-axis denotes the corresponding measure values. The red crosses $\times$ mark the forecast values $\hat{Y}_{t+1}, \ldots, \hat{Y}_{t+h}$. The exact number of requested forecast values is called forecast horizon h. In the example three values are predicted $h = 3$. The prediction of long forecast horizons entails additional problems beyond the focus of this work. Therefore, we limit ourselves to one-step ahead forecasts with $h = 1$ and address the topic of long range forecasting in a later work.

There are two very important characteristics of time series that a model should address in order to produce accurate forecasts. The first one is the trend characteristic which sums up all long term changes without reoccurring patterns. The second is the seasonality which describes regularly reoccurring patterns within fixed intervals. The example time series in Fig. 1 has a seasonality with a season length of $s = 12$, which is recognizable at the reoccurring peaks, but lacks a clearly visible trend, e.g., a continuous rise or decline in the measured values. Between the time series and the x-axis of Fig. 1 there is a secondary representation of the same time series. Each square represents one time series value. Historical values are

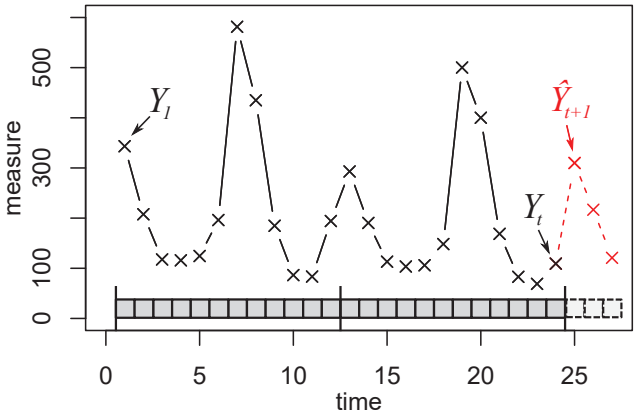


Fig. 1. Example time series and forecast with a forecast horizon of $h = 3$.

marked by gray squares with a solid contour, forecast values are marked by light gray squares with a dashed contour. In the remainder of this paper, we will use this representation to visualize how time series values are used to calculate forecasts.

Changes in the way of data collection, towards more and more time series that are recorded on increasingly fine structural and temporal granularities, lead to new requirements for the forecasting process [5]. We would like to emphasize these new requirements and illustrate them with an example data set from the energy domain which has to be predicted with high accuracy despite its properties. It consists of Smart Meter data monitored in Ireland over one and a half years in 30min time granularity. The data was recorded by the Commission for Energy Regulation (CER) and made publicly available by the Irish Social Science Data Archive (ISSDA) [9].

R1 – Numerous Series The high number of time series that has to be modeled, makes the application of models that only predict one series at a time very difficult, since there is a large number of models which has to be created. The example data set consists of 6433 individual time series which request an equally high number of forecast models. This is a very time consuming task as we will show later on in our evaluation. In order to suit the prediction of large scale time series data sets, a modeling technique has to provide forecast values for a large number of time series in reasonable time.

R2 – Incomplete Data The individual time series originate from a multitude of different data sources, e.g., Smart Meters of individual households and enterprises. Technical malfunction of the monitoring devices can lead to missing values and, thus, to incomplete time series which many forecasting techniques are not able to work with. In the example data set only 71% of all time series have a complete history and compared to a complete data set where all time series were recorded over the full monitoring time, 5% of the overall data is missing. When the focus lies on only a few time series it often is possible to ensure complete data, e.g., by the application of imputation methods or the search of compensation values from similar time series [10]. However, regarding data sets with thousands of time series this is associated with very high cost if it is feasible at all. Hence, a forecasting technique has to

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

be able to provide accurate predictions regardless whether the data set is complete or not.

R3 – Increasingly Fine Granularity The increasingly fine structural and temporal granularity at which time series are monitored leads to very long historical data. Most forecasting techniques use all the available data for the model optimization which makes the forecasting of large scale data sets even more time consuming. On top of that, such fine grained time series are very prone to noise originating from external influences which can hardly be monitored and can only be seen at a very fine granularity, e.g., the operation times of domestic appliances which are unique per household and will vary on a daily basis [11]. This makes time series especially hard to model and forecast since their behavior does not seem to be deterministic and describable. Therefore, a modeling technique has to be able to compensate noisy behavior of time series and provide accurate forecast values.

In the next section we review existing forecasting techniques and discuss their ability to address these requirements.

III. Related Work

The topic of time series forecasting is an old field with a lot of published research available. In this section we give an overview on the most commonly used methods and analyze their advantages and shortcomings related to the just depicted requirements.

A. Univariate Forecasting Methods

Univariate forecasting techniques focus on the prediction of one single time series. The ARIMA model is one of the most commonly used techniques from this class [4], [6]. We describe this model slightly more detailed than other approaches since we will pick up on its properties later on.

ARIMA models a time series using three basic concepts: autoregression **AR**, integration **I**, and moving average **MA**. These three different concepts make the ARIMA model adaptable to different time series characteristics and applicable in a wide range of use cases. The modeling process begins with the integration step. The time series is differentiated to eliminate trend and seasonal characteristics and make it stationary. A time series is called stationary if it has a constant expectation value and a constant variance over time. Afterwards, both, one, or none of the two predictive model parts AR and MA are applied. The autoregressive part models future time series values based on the most recent historical time series values. The moving average part models future values based on error terms which are the result of the simulated prediction of the available time series history. This type of error correction uses historical errors to create better predictions. For every model part there is a distinction between a non-seasonal and a seasonal case. Whether a seasonal model has to be applied or not depends on the presence of a seasonal pattern in the time series. For every analyzed time series the optimal ARIMA model has to be configured, i.e., the degree of differentiation has to be determined and the right number of AR and MA components has to found. There are some very helpful guides available [6], [12], which show how to find the right ARIMA model for a specific time series but the search for the optimal model configuration is not part of this paper and therefore not detailed any further.

Another often applied univariate technique is Exponential Smoothing, i.e., the Holt-Winters model [4], [13]. This model applies a smoothing mechanism on the entire time series and continues the process in order to generate forecast values. This model is also capable of handling time series with seasonal and trend characteristics. Both techniques, ARIMA and Holt-Winters, apply smoothing mechanisms in parts of their modeling and, therefore, rely on complete time series. Missing values clearly limit their applicability whereby they fail to meet the requirement R2.

Two more recent developments towards this topic are Multivariate Adaptive Regression Splines (MARS) [14] and Gradient Boosting Machines (GBM) [15]. MARS uses a set of external influences of which it autonomously extracts the most relevant ones to forecast the currently analyzed time series. Furthermore, it is capable of modeling non-linear relationships between the external influences and the target series. The GBM model uses an ensemble of weak predictors, typically regression trees, to combine their predictions into the final forecast for the analyzed time series. The input for the decision trees may be historical data of the time series itself or external influences.

All of these techniques have in common that they only focus on one time series. This makes their application very time consuming when it comes to the prediction of thousands of individual time series. Therefore, they do not meet the requirement R1. Additionally, they are not capable of compensating noise in the modeled time series, when its behavior can neither be explained properly by its own history nor by external influences. This often leads to an insufficient model fit and subsequently high forecast errors. Therefore, these techniques also fail to the meet requirements R3.

B. Techniques for Incomplete Time Series Data

The forecasting of incomplete or intermittent time series has already been discussed in the forecasting literature. Croston's method is the most widely used approach for this kind of data and especially designed for time series of intermittent occurrence [16], [17]. The model involves two simple exponential smoothing processes. The first one models the interval between the non-zero observations of the time series. The second one models the actual time series values when they occur.

Actually, since it only models one time series, Croston's method is a univariate technique with all the associated problems. However, we mention it in a separate category since it is mostly referenced in the context of intermittent series. Although, Croston's method might be able to calculate forecasts for incomplete time series, and therefore meets the requirement R2, it requires missing or zero values to occur regularly to properly model them. A regular pattern of the missing observations is usually not given in the aforementioned scenarios. On top of that, for complete time series

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

Crostons's method becomes simple exponential smoothing which is hardly ever an accurate method since it misses the ability to work with trends and seasonalities.

C. Hierarchical Forecasting

Another possibility to overcome the issues of univariate modelling is to exploit the often hierarchical structure of large data sets [18]. Time series are transferred to a more coarse grained aggregation level by the application of an aggregation function, e.g., sum. For example, time series representing the energy consumption of many individual consumers are aggregated to a higher aggregation level representing a whole city or a certain district. This can help to compensate the effects of incomplete and noisy data (requirements R2 and R3). Values of other time series cover the points where values are missing in one time series and fluctuations of several time series can equalize each other. Therefore, hierarchical forecasting addresses all the requirements R1 to R3 since it reduces the number time series that has to be forecasted by aggregating them and compensates for missing values and noisy behavior.

As convenient as this approach might seem, it does not solve the problems entirely. Even at higher aggregation levels it is possible to experience missing values for groups that consist of only a few time series. On top of that, by the utilization of aggregation we loose the opportunity to conduct analyses on the base level of the data set. This is not acceptable since we assume that in many domains data is collected and stored in a granularity that is relevant for the analysis. In fact, it is possible to overcome this issue by the use of disaggregation methods where the forecasted aggregates are split and distributed to lower aggregation levels. However, this involves a second modeling step to generate the disaggregation keys and will lead to another source of forecast errors, especially keeping noisy and incomplete time series in mind.

D. Multivariate Forecasting Methods

Multivariate forecasting techniques focus on the analysis and prediction of multiple time series within one model. The most commonly referenced approaches are VARMA models [19], [20]. They use the concept of autoregression and moving averages of ARIMA and apply them to a set of time series. Thereby, they do not model each time series individually but explicitly express how the time series in a data set interfere with each other. Econometric models build up on this idea and add external influences which affect the time series that should be predicted but are not affected themselves by other time series [8].

Multivariate models address requirement R3 by using the information of many time series. They can compensate noise of individual time series much better, because some effects which may not be entirely explained by a time series itself may be explained from others. However, in order to apply this kind of model succesfully the time series have to directly influence each other. This is not given in the described scenarios, i.e., the energy consumption in one household is not directly affected by the energy consumption of others. Rather, they are influenced by the day-night rhythm, which is a trait that is hard to measure and to model. In those cases multivariate modeling techniques will not work since there is no inter time series relationship worth modeling. Furthermore, these models are not capable of handling thousands of time series. The optimization process of the quadratic number of parameters to model the influences of all time series on one another becomes too time consuming and requires a very long history of training data [5], therefore, they miss the requirement R1. Ultimately, these models also fail to meet requirement R2 since the time series have to be complete in order to properly model the interference of all analyzed time series.

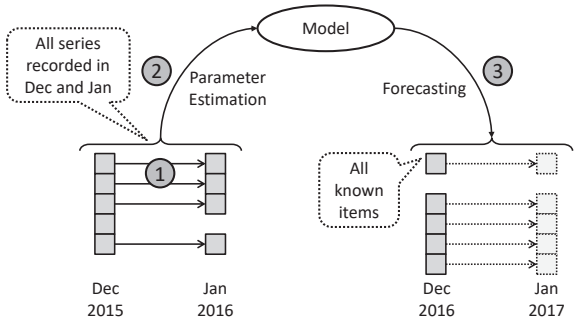


Fig. 2. Cross-sectional forecasting

E. Cross-sectional Forecasting

Cross-sectional forecasting is a modeling technique which is designed to address the requirements R1 to R3 of forecasting large scale time series data sets [5]. CS follows three core steps which are shown in Figure 2. First of all, the model assumes that the time series from the same data set follow a common behavior and that this behavior – the changes from one period t to the next period $t+1$ – is stable over several seasons. In Fig. 2 this is the change from December to the following January. The data for the model creation is extracted from the historical data of all time series of the data set in the form of so called cross-sections. A cross-section is a time slice which contains the values of all time series at a certain period t.

In the second step the model parameters are optimized. Equation 1 shows the cross-sectional forecast function where $\vec{y}_t$ and $\vec{y}_{t+1}$ are the cross-sections over all time series at time t and $t+1$. ϕ_1 and c are the parameters which are optimized.

$$\hat{\vec{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t \tag{1}$$

The model now represents the average transition from one period to the next one for all time series, e.g., as shown in Fig. 2 the transition from Dec 2015 to Jan 2016.

Finally in step three, the optimized model is applied to the last monitored cross-section of the data set to obtain a forecast value for every single time series in one step. The cross-sections for model training and forecast calculation are situated in the distance of exactly one season to properly represent a reoccurring seasonal behavior. This approach is always applied on the most fine grained aggregation level (base level) of the

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

data set. This ensures on the one hand a sufficiently broad training set with many time series contributing to the model optimization. On the other hand, all possible aggregation levels which might be of interest can be predicted using the same model by aggregating the forecasts after the model application.

By optimizing only one model for an entire data set CS meets requirement R1 and by incorporating data from many time series into the model creation it can compensate incomplete data and noise of individual series and also meets R2 and R3. However, compared to other modeling techniques CS is limited in its modeling versatility by a very rigid data selection. This also limits the accuracy the model can achieve on data sets with different characteristics, because, as we have learned in the related literature, adaptability of the model towards the data can significantly improve its accuracy [6], [7].

IV. CSAR-Modelling

In this section, we describe in detail how our new CSAR model is working. It combines the qualities of cross-sectional forecasting and ARIMA, thus CSAR meets all the requirements R1 to R3 and is adaptable to different data sets with their unique characteristics. We follow the structure of the ARIMA model and highlight the adaptations we introduced to combine every individual part with the paradigm of cross-sectional forecasting. For us, the ARIMA model is the logical choice for this combination since it is a mature forecasting technique and suits the paradigm of CS.

A. Integration

The integration part serves, as in the ARIMA model, as a preparation step to remove trend and seasonal characteristics from the time series and make them stationary. In our approach every time series is differentiated individually, such that the properties of each time series are preserved and not changed by the influence of other series. When the integration is used, every time series is differentiated first, then the data set is forecasted, and finally the series are integrated to obtain the forecast values for the original time series. As in the ARIMA model there is a distinction between a non-seasonal and a seasonal case.

The non-seasonal differentiation is used to eliminate trend characteristics. The first degree of numeric differentiation is shown in Equation 2. The value y_t of the differentiated time series is calculated as the difference of the original time series value Y_t and an earlier value Y_{t-d} divided by their distance d. In the usual case when there are no missing values we set $d = 1$ and y_t is calculated directly from the corresponding value Y_t and its predecessor Y_{t-1}. If there are one or more missing values directly before Y_t then d is increased, such that the next available value Y_{t-d} is used for the differentiation. The division by their distance is necessary to represent the trend change from one period to the next one.

$$y_t = \frac{Y_t - Y_{t-d}}{d} \tag{2}$$

This kind of numeric differentiation is called backward differentiation since we are looking backwards from the current point in time t. Alternatives like forward differentiation $y_t = (Y_{t+d} - Y_t)/d$ or a symmetrical approach $y_t = (Y_{t+d} - Y_{t-d})/2d$ are not suited for the task of forecasting since the differentiation of the last value of the time series is not possible; and this value is crucial for the forecast calculation in most if not all forecast methods. The absence of the first value as it is the case for the backward differentiation is not a problem at all if the time series is long enough to train a model without depending on this first value, which is usually the case.

The seasonal differentiation is used to eliminate reoccurring seasonal patterns of the time series Y. Equation 3 shows how y_t is calculated by the difference of Y_t and its corresponding value in a previous season $Y_{t-D \cdot s}$. s is the seasonality of the data set which is either known from the contextual information about the data set or can be determined, e.g., using the autocorrelation function. When the value in the direct preseason is available we set $D = 1$, otherwise D is increased such that the next available corresponding seasonal value of Y_t is used to calculate the differentiated value.

$$y_t = Y_t - Y_{t-D \cdot s} \tag{3}$$

The seasonal differentiation does not need a division by the size of the gap that is bridged, since it is assumed that the seasonal behavior is stable over time and should be removed entirely.

B. Autoregression

The autoregressive part of the CSAR model is a combination of the cross-sectional forecasting approach and the autoregressive part of the ARIMA model. It predicts all time series of a data set based on their most recent historical observations. As in the ARIMA model the autoregressive part of CSAR consists of non-seasonal and seasonal components. Though, the optimized weights are not applied to only one time series but to cross-sections which span over all time series in the data set. Hence, every time series is forecasted based on a weighted sum of its own historical values while the model parameters (the weights) are optimized on all time series of the data set which have the necessary historical data.

The Equations 4 and 5 show how the predictions are calculated in the non-seasonal and seasonal case. In the non-seasonal case (Equation 4), the forecast values $\hat{\vec{y}}_{t+1}$ are calculated as the weighted sum of their direct predecessor values $\vec{y}_t$ to $\vec{y}_{t-(p-1)}$ weighted with the model parameters ϕ_1 to ϕ_p. p denotes the number of non-seasonal autoregressive model components. $\vec{y}_t$ refers to the cross-section at time t which contains the historical values y_t^n of every individual time series y^n. Additionally, there is a constant part c which is also optimized during the model training and used for every time series. c can be excluded in order to fit the optimal model to a data set.

In the seasonal case (Equation 5) $\hat{\vec{y}}_{t+1}$ is calculated by the corresponding seasonal historical values $\vec{y}_{t-s+1}$ to $\vec{y}_{t-P \cdot s+1}$

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

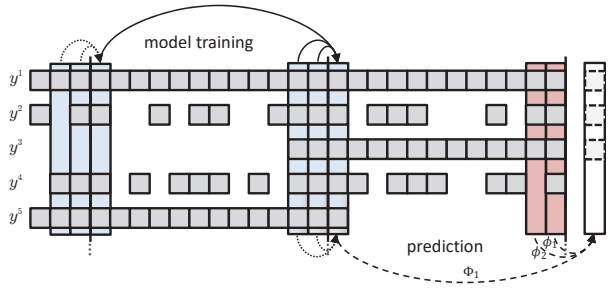


Fig. 3. CSAR model with one seasonal and two non-seasonal AR components.

with a time distance of s periods. P denotes the number of seasonal autoregressive model components. The seasonal weights are represented by Φ_1 to Φ_P.

$$\hat{\vec{y}}_{t+1} = c + \phi_1 \cdot \vec{y}_t + \cdots + \phi_p \cdot \vec{y}_{t-(p-1)} \quad (4)$$

$$\hat{\vec{y}}_{t+1} = c + \Phi_1 \cdot \vec{y}_{t-s+1} + \cdots + \Phi_P \cdot \vec{y}_{t-P \cdot s+1} \quad (5)$$

Fig. 3 shows an example of five time series y^1 to y^5 with a season length of $s = 12$ which are predicted using the same CSAR model. The model shown in this example is comparable to an autoregressive (AR) model with two non-seasonal and one seasonal component. The difference is, that this model is not applied to only one individual time series but, following the idea of the cross-sectional forecasting approach, to cross-sections which are highlighted by vertical boxes that stretch over all time series. The solid arrows show how the involved cross-sections (highlighted in blue) contribute to the model training. The two short arrows represent the non-seasonal components of the model and the long arrow reaching back exactly one season represents the seasonal component. Every time series which has values in all involved cross-sections contributes to the model creation. Thus, the model represents how in average the target cross-section of the training data can be composed from the historical values for *all involved time series*. In the example, these are the time series y^1, y^2 and y^5. Series y^3 for example does not contribute to the model creation since it has no value for the seasonal component. The dotted arrows represent correction terms which are necessary due to the combination of seasonal and non-seasonal components. They subtract the direct predecessor values from $\vec{y}_{t-s+1}$ in the same way as the direct predecessors of $\hat{\vec{y}}_{t+1}$ would add up in the forecast value. In doing so, the seasonal component only represents the actual seasonal change. There are always as many correction terms to every seasonal component as the model contains non-seasonal components and they are also mandatory for a time series in order to contribute to the model creation. The data for the model creation is still, as in the cross-sectional forecasting model, situated exactly one season before the model application which is represented by the dashed arrows in Fig. 3. The optimized model is now used to calculate the forecast values for the target period $t + 1$. A forecast value can be calculated for every time series which has historical values in all involved cross-sections (highlighted in red). In the example, these are the time series y^1 to y^3. Series y^4 for example can not be forecasted because it has no values for the second non-seasonal component.

Equation 6 shows the corresponding formula to the model of the example. Next to the constant part c there are two non-seasonal components with the respective weights ϕ_1 and ϕ_2 and one seasonal component with its weight Φ_1 followed by the two corresponding correction terms.

$$\begin{aligned}\hat{\vec{y}}_{t+1} = c &+ \phi_1 \cdot \vec{y}_t + \phi_2 \cdot \vec{y}_{t-1} + \Phi_1 \cdot \vec{y}_{t-s+1} \\ &+ (-\Phi_1\phi_1) \cdot \vec{y}_{t-s} + (-\Phi_1\phi_2) \cdot \vec{y}_{t-s-1} \end{aligned} \quad (6)$$

Considering this example, it becomes clear how a CSAR model is created on a multitude of time series like a cross-sectional forecast model but offers higher flexibility in the selection of the underlying data. The model keeps the positive properties of the cross-sectional forecasting model. This means, it is still possible to compensate for high levels of noise and to handle time series with missing values since the creation of a model does not solely depend on the historical data of only one time series. Please note, albeit CSAR can compensate for missing values a higher model complexity (more non-seasonal and/or seasonal model components) increases the risk of cases where missing values forbid the forecast calculation of individual time series when they occur in the base for the forecast calculation. Hence, for very sparse data sets a less complex model may lead to better forecasting results. Although it might not represent the data set as good as possible, it can predict more of the incomplete series.

C. Error Terms

Moving average components as in the ARIMA model are not applicable in combination with a cross-sectional model where the core idea is that the same model parameters are used for all time series of a data set. In contrast to the autoregressive components, the moving average does not only rely on the most recent historical values but applies a smoothing process to the full history of a time series y. This is done by using error terms as shown in Equation 7 where the error e at time t is the difference of the original time series value y and the corresponding forecast $\hat{y}$. Equation 8 shows the calculation of a forecast value using moving average components. The forecast $\hat{y}_{t+1}$ results from subtracting the error terms e_t to e_{t-q+1} from the constant part c, respectively from a forecast calculated by autoregressive components. Every error component is weighted with a corresponding parameter θ, q denotes the number of error terms which influence the current forecast.

$$e_t = y_t - \hat{y}_t \quad (7)$$

$$\hat{y}_{t+1} = c - \theta_1 \cdot e_t - \cdots - \theta_q \cdot e_{t-q} \quad (8)$$

Considering this calculation, it becomes clear that missing values make this approach impossible since the calculation of a forecast value depends on the *error terms of all historical values* of the time series.

There are already concepts available to apply smoothing techniques such as exponential smoothing to incomplete time series [21]. In the proposed solution the next available historical predecessor value is used instead of the direct predecessor

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

and its weight is lowered depending on how many missing values are bridged. Although, this solution could be transferred to the moving average part of the ARIMA model, the missing values of different time series are not evenly distributed in the data set (see the example in Fig. 3) and, thus, an individual adaptation of the model parameters for every time series with missing values is required. This contrasts the core idea of the cross-sectional forecasting approach to train a single model with one single set of parameters for a multitude of time series.

For the CSAR model we introduce an alternative way to incorporate the error terms into the forecast calculation. Instead of applying the moving average function of Equation 8 we use the average of the error terms of each individual time series of the data set and include these into the forecast calculation. Equation 9 shows how the average error $\overline{e^n_{t+1}}$ for time series n at time t is calculated. The first sum collects the non-seasonal forecast errors for the periods directly prior to period t. The second sum collects all seasonal forecast errors which are situated exactly one or more full seasons prior to t. The error terms of the individual time series are summed up and divided by the number of non-seasonal f and seasonal error terms F. If a time series misses values to calculate either forecast or error these specific values are neglected during the error calculation and f or F are lowered accordingly for this time series. Finally, Equation 10 shows how the error is incorporated into the forecast calculation by subtracting it from the constant part c or the corresponding forecast calculated by a CSAR model without error terms.

$$\overline{e^n_{t+1}} = \frac{1}{f+F}\left(\sum_{i=1}^{f} e^n_{t-i} + \sum_{j=1}^{F} e^n_{t-j\cdot s}\right) \tag{9}$$

$$\hat{y}_{n,t+1} = c - \overline{e^n_{t+1}} \tag{10}$$

In this way, it is possible to compensate the forecast errors for time series which are systematically mispredicted. Actually, we have assumed, that a high number of error terms would be necessary in order to obtain a reliable error component. As the evaluation in the next section will show, this is not the case and small numbers of error terms already lead to improvements of the forecast accuracy.

V. Evaluation

We conduct an experimental study to evaluate the accuracy and execution time of our CSAR model. We start by giving an overview of the experimental setting, including the data sets we used for the evaluation. This is followed by a detailed description of the experiments and the discussion of the results.

A. Experimental Set-Up

We implement our CSAR model in the statistical computing environment R [22], which provides us with efficient built-in functions for model parameter estimation and commonly used forecasting techniques. The experiments are executed on a server machine with a Six-Core AMD Opteron(tm) Processor 2435@2.6GHz processor and 32GB of RAM. For the evaluation we use three real world data sets:

Energy The first one is the example data set from Section II. The data represents the energy consumption of 6433 individual households and small and medium enterprises. The time series are monitored in 30 min granularity over one and a half years. We use the last complete week of data as evaluation part. For this data set we executed our experiments on different time granularities from 30 min to daily energy consumption. Due to space limitations, we only present the 6 hour granularity in this paper since this shows the effects we want to emphasize during the evaluation best. Time series on this time granularity have a history length of 2144 values. However, the results can readily be transferred to other time granularities.

Payment The second data set is taken from the IJCAI-2017 Data Mining Contest [23]. It consists of payment transactions of 2000 distinct shops in daily granularity monitored over 494 days. We use the number of payments per shop and day of the last 14 days as evaluation period. None of the time series has a complete history which means that every time series is either not monitored right from the beginning or has missing values in its history. Compared to a complete data set, 40% of data is missing.

Sales The third data set is taken from the sales domain and is provided to us as a private data set by a market research company. It contains 6266 time series of items from the field of home appliances sold in Germany recorded in a monthly granularity over 3 years. We use the last six month for our evaluation. Only 6% of the time series have a complete history and overall there are 59% of data missing. Since the time series are very short, this data set does not allow the creation of very complex models because more complex models require a longer history of training data.

B. Forecast Accuracy

In the first experiment, we evaluate the accuracy of our forecasting approach on two different aggregation levels per data set. We begin with the top aggregation level where all base time series are aggregated only grouped by the time. This represents for example the overall energy consumption of all households and enterprises in the Energy data set. Afterwards, we analyze the base aggregation level where every base time series is evaluated individually. As comparison methods we use the ARIMA model as implemented in the auto.arima-function of the forecast package of R [7] and the cross-sectional forecasting model (CS) as presented in [5]. For ARIMA we filled missing values of all data sets with zero values to enable its application. CS is represented by a CSAR model with only one non-seasonal autoregressive component and the constant part. Additionally, we include the naïve forecast where every predicted period shows the same value as its predecessor $\hat{y}^n_{t+1} = y^n_t$. This is the baseline, if a forecasting technique performs worse than the naïve forecast, it is not suited for the specific data set, as it does not properly represent its characteristics. For a comparison with other forecasting techniques, i.e., Triple Exponential Smoothing, Vector Autoregression, Croston's Method, and Hierarchical Forecasting, please refer to the evaluation of [5] where the

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

authors show that CS already achieves a higher accuracy than these methods.

The ARIMA model is always applied at the same aggregation level the data is evaluated on, which means the data is aggregated first and then forecasted. CS and CSAR are applied on the base level and the forecasts are aggregated afterwards to obtain the top level. The optimal metaparameters for CSAR, i.e., number of seasonal and non-seasonal autoregressive components and error terms as well as the degree of integration, were optimized manually. The model parameters, i.e., the weights of the autoregressive model components and the constant c, were optimized using the optim-function of R.

All data sets are divided into a training and an evaluation part as mentioned in the data set description. All data preceding the evaluation part may be used for the model training. We apply a rolling forecast, where we create a new model for every period t in the evaluation part of every data set to calculate the forecast values. Then we compare the forecasts to the corresponding time series values and calculate the forecast error with the SAPE measure (Symmetric Absolute Percentage Error):

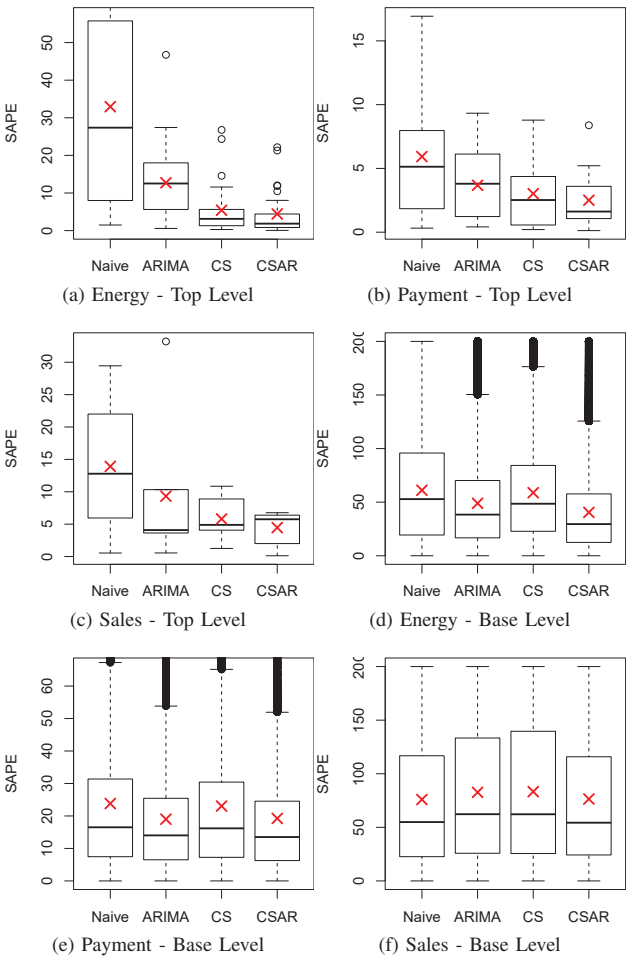
$$SAPE = \frac{|y - \hat{y}|}{(|y| + |\hat{y}|)/2} \cdot 100, \tag{11}$$

y denotes the real time series value and $\hat{y}$ is the corresponding forecast of one of the evaluated techniques. We use SAPE because as a relative measure it is easier to interpret and compare than absolute error measures. Furthermore, it can be applied even when the real time series value equals zero, where other relative error measures are not defined. If the time series value and the corresponding forecast both equal zero we assume a forecast error of zero. Values in the evaluation part that a method was not able to predict are filled with a zero forecast and, therefore, punished with a maximum error.

The results of this experiment are shown in Fig. 4. Each diagram presents the forecast errors for one data set and aggregation level as a Box-Whisker-Plot. The y-axis denotes the SAPE forecast error. Each box represents the forecast errors of one forecasting technique. The red cross in each box $\times$ denotes the corresponding average error.

The first three diagrams (Fig. 4a - 4c) show the results of the top aggregation level. Our new CSAR model (rightmost box) performs best on all three data sets. For the payment data set all approaches perform well since the number of overall payments does not fluctuate very strong on a daily basis. For the energy and sales data sets the naïve forecast performs significantly worse since there is a strong seasonality which this approach can not model. The cross-sectional forecasting model performs better than the ARIMA model because it incorporates the knowledge of all base time series which leads to a better representation of the overall data sets. CSAR performs even better and profits from the higher adaptability.

The results for the base aggregation level are shown in the Fig. 4d - 4f. Every time series is evaluated individually and, therefore, it is much harder to achieve a high accuracy which causes the generally higher forecast errors. The diagrams show



(a) Energy - Top Level (b) Payment - Top Level (c) Sales - Top Level (d) Energy - Base Level (e) Payment - Base Level (f) Sales - Base Level

Fig. 4. Forecast error on top and base aggregation level.

again that the adaptability of the CSAR model leads to the most accurate forecasting results of all tested methods. The forecast errors of CSAR are the overall best for the energy data and on par with the best comparison method for the Payment and the Sales data. For Energy and Payment the time series on the base level still have some predictable properties. This leads to also acceptable results for ARIMA which is only outperformed by CSAR. For the Sales data set the individual time series are extremely noisy. Therefore, ARIMA and CS perform even worse than naïve forecast. The optimal CSAR model for this data set resembles the naïve forecast as it only uses one non-seasonal autoregressive component and, therefore, is the only technique that can keep up with the naïve forecast.

In summary our CSAR model performs best for all data sets without requiring a manual preparation of the data or any missing value treatment, although, all data sets have different characteristics, especially their levels of sparsity. Thus, of all compared techniques CSAR is suited best to derive accurate forecasts for noisy and incomplete data sets and satisfies the requirements R2 and R3.

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

TABLE I
OPTIMAL CSAR MODELS

	AR	SAR	ER	SER	const c
Energy Top	1	1	2	10	TRUE
Payment Top	3	0	0	1	TRUE
Sales Top	2	0	0	1	TRUE
Energy Base	4	0	0	0	FALSE
Payment Base	0	1	1	10	TRUE
Sales Base	1	0	0	0	FALSE

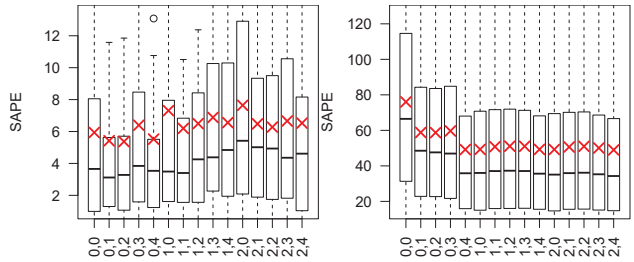
Finally, Table I shows the metaparameters of the best CSAR model for every data set and aggregation level. The columns AR and SAR show the numbers of non-seasonal and seasonal autoregressive model components, ER and SER show the number of non-seasonal and seasonal error terms. For different data sets with their unique properties different model components have to be used in order to achieve the optimal forecasting result. Furthermore, even for the same data set on top and base aggregation level the optimal model components can differ significantly and there is no general pattern which model components lead to an accurate forecast.

Please note, for none of the data sets in our evaluation CSAR could profit from a preceding differentiation. This happens since trend and seasonality characteristics are not equally strong represented in all time series of the data sets. Thus, some time series profit from differentiation and others do not, but there is no overall improvement. Apart from this, all other model components are used in any of the optimal models of our experiment and, therefore, are definitively meaningful parts of our CSAR model.

C. Model Complexity

In the second experiment we evaluate the influence of the model complexity, i.e., the number of non-seasonal and seasonal autoregressive components of the model, to the accuracy of the forecast results. Using the set-up from the previous experiment, we calculate forecasts for the Energy data set at top and base level and increased the number of model components. The results of the experiment are shown in Fig. 5. The x-axis shows the model complexity by the number of *seasonal,non-seasonal* parameters. We start with a model without autoregressive components followed by an increasing number of non-seasonal (0,1 ...0,4) and finally seasonal components. The y-axis, again, denotes the SAPE forecast error.

The model complexity clearly has a significant influence on the forecast accuracy but a higher complexity does not necessarily lead to better forecasts. For the top aggregation level the forecast error even increases with higher model complexity. For the base level more autoregressive model components lead to an increased accuracy. For both diagrams there is no systematic pattern in the fluctuations of the forecast accuracy, when the model complexity is increased. Thus, finding the optimal CSAR model for a data set is no trivial task. More research into this direction is necessary, but as the next experiment will show, the search for the optimal CSAR model can easily afford the creation of several models to test their accuracy.



(a) Energy - Top Level (b) Energy - Base Level

Fig. 5. Forecast error for increasing model complexities.

D. Execution Time

In the last experiment we evaluate the execution time for different complexities of our CSAR model and compare them with those of ARIMA. Using the set-up of the previous experiments, we calculate forecasts for the base level of the Energy data set and monitor the execution time for the prediction of one one-step ahead forecast for all time series of the data set. We execute the experiment ten times and use the average time of all ten passes for the comparison.

The results of the experiment are presented in Table II. The first five columns show different complexities of our CSAR model named with the notation of the previous experiment. First of all, there is no difference between the execution times for seasonal and non-seasonal models. They access the same amount of data and have to optimize the same number of model parameters. Hence, I/O cost and computation times are very close to each other. The addition of more model parameters increases the execution time in two ways. More data for model training and forecast calculation has to be accessed and the optimization process of the model parameters takes longer when more parameters are optimized. The combination of seasonal and non-seasonal components further increases the execution time since the correction terms have to be taken into account (ref. Section IV-B). A further increase of the number of model components leads to a higher execution time with super linear growth. Again, this is caused by the number of correction terms which grows in a multiplicative manner since there are as many correction terms for every seasonal component as there are non-seasonal components.

The sixth column shows the execution time of ARIMA. Note, the execution time of the ARIMA model was measured on ten individual time series and the overall execution time was extrapolated by the number of time series in the data set. Moreover, we did not use the auto.arima function which includes the search for the optimal metaparameters and would have lead to a much higher execution time. We evaluated the pure execution of R's arima function to create a model and calculate one forecast value per time series. Compared to

TABLE II
COMPARISON OF EXECUTION TIMES

CSAR					ARIMA
0,1	1,0	0,2	1,1	2,2	
0.4s	0.4s	0.8s	0.9s	5.9s	42min38s

Final edited form was published in "IEEE International Conference on Data Science and Advanced Analytics. Tokyo 2017", S. 233-241, ISBN 978-1-5090-5004-8
http://dx.doi.org/10.1109/DSAA.2017.27

the ARIMA model CSAR has a significantly lower execution time which is the result of the creation of only one model for an entire data set instead of modeling each time series individually. For the Energy data set on its original 30min granularity ARIMA is even not able to provide forecasts in time since the execution time exceeds the monitoring granularity by more than 40%.

Therefore, we can show that CASR meets the requirement R1 and in combination with the results from the first experiment satisfies all the requirements (R1 to R3) on the prediction of large scale time series data sets.

VI. Conclusion

In this paper, we presented CSAR, a new forecasting technique designed to meet the requirements of forecasting large scale time series data. We have framed these new requirements originating from data sets which consist of thousands of time series monitored on fine grained structural and temporal granularity. In the discussion of related work on the topic of forecasting, we showed that none of the existing techniques adequately addresses all of the requirements. Our new CSAR model combines the qualities of the cross-sectional forecasting approach and the ARIMA model. Therefore, it is able to handle large data sets in reasonable time while offering adaptability to the analyzed data. Additionally, CSAR is robust against noise and missing values of individual time series. Our experimental evaluation showed, that CSAR achieves a higher accuracy than other forecasting techniques and greatly benefits from the increased adaptability. Furthermore, the creation of one model for many time series ensures the timely creation of forecasts, even for very large data sets, and makes our model several orders of magnitude faster than for example ARIMA. This work is part of an ongoing research process and offers many interesting directions for future research. The most relevant topics we will address next are the following:

Instantiation As our experimental study has shown, the choice of the right metaparameters of the CSAR model has a significant impact on the forecast accuracy. Right now, the optimal model has to be found manually by trying different sets of metaparameters which is very time consuming. Therefore, a set of guidelines that helps to find the optimal model, e.g., as it exists for ARIMA, is an important way to continue our research in this direction.

Long Range Forecasting In this work we focused on the calculation and evaluation of one-step ahead forecasts. However, there are many domains, where forecasts for more than one period are necessary to properly plan for future developments. This is why the extension of CSAR to long range forecasting will be a major goal of our future research.

Partitioning Currently, CSAR creates one model which represents all time series of a whole data set. This is the opposite extreme to univariate models like ARIMA where one model only represents one single time series. For the future, we plan to apply partitioning techniques before the actual modeling takes place to divide the data set into partitions of time series with similar characteristics and create one CSAR model for every partition. In doing so, we expect a further increase in the forecast accuracy by a model that better represents all its underlying time series.

Acknowledgment

This work is partly funded by the German Research Foundation (DFG) in the context of the project “Flash-Forward Query Framework” (LE-1416/17-2).

References

[1] “GOFLEX Project,” http://goflex-project.eu/, 11.02.2017.
[2] “Universal Smart Energy Framework (USEF),” www.usef.energy, 11.02.2017.
[3] O. Vermesan and P. Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Aalborg: River Publishers, 2013.
[4] T. M. McCarthy, D. F. Davis, S. L. Golicic, and J. T. Mentzer, “The Evolution of Sales Forecasting Management: A 20-Year Longitudinal Study of Forecasting Practices,” *Journal of Forecasting*, vol. 25, no. 5, pp. 303–324, aug 2006.
[5] C. Hartmann, M. Hahmann, F. Rosenthal, and W. Lehner, “Exploiting Big Data in Time Series Forecasting : A Cross-Sectional Approach,” in *IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2015.*, Paris, 2015, pp. 1–10.
[6] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control (Wiley Series in Probability and Statistics)*. John Wiley & Sons Inc, 2008.
[7] R. J. Hyndman and Y. Khandakar, “Automatic Time Series for Forecasting: The Forecast Package for R,” *Journal of Statistical Software*, vol. 27, no. 3, 2008.
[8] C. Chatfield, *Time-series forecasting*. Chapman & Hall/CRC Press, 2000.
[9] Irish Social Science Data Archive (ISSDA), *CER Smart Metering Project*, The Commission for Energy Regulation (CER), 28.04.2015, www.ucd.ie/issda.
[10] VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V., “Messwesen Strom (Metering Code); VDE-AR-N 4400,” 2011.
[11] B. Neupane, T. B. Pedersen, and B. Thiesson, “Towards Flexibility Detection in Device-Level Energy Consumption,” in *Data Analytics for Renewable Energy Integration : Proceedings of the Second ECML PKDD Workshop, DARE 2014*, Nancy, 2014, pp. 1–16.
[12] Robert Nau, “Statistical forecasting: notes on regression and time series analysis,” http://people.duke.edu/~rnau/411home.htm, accessed 09.11.2016.
[13] C. C. Holt, “Forecasting seasonals and trends by exponentially weighted moving averages,” *International Journal of Forecasting*, vol. 20, no. 1, pp. 5–10, 2004.
[14] J. H. Friedman, “Multivariate Adaptive Regression Splines,” *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.
[15] Friedman, Jerome H, “Greedy function approximation: A gradient boosting machine.” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
[16] J. D. Croston, “Forecasting and Stock Control for Intermittent Demands,” pp. 289–303, 1972.
[17] L. Shenstone and R. J. Hyndman, “Stochastic models underlying Croston’s method for intermittent demand forecasting,” *Journal of Forecasting*, vol. 24, no. 6, pp. 389–402, 2005.
[18] G. Fliedner, “Hierarchical forecasting: issues and use guidelines,” *Industrial Management & Data Systems*, vol. 101, no. 1, pp. 5–12, 2001.
[19] T. Riise and D. Tjostheim, “Theory and practice of multivariate arma forecasting,” *Journal of Forecasting*, vol. 3, no. 3, pp. 309–317, jul 1984.
[20] G. C. Tiao and G. E. P. Box, “Modeling Multiple Time Series with Applications,” *Journal of the American Statistical Association*, vol. 76, no. 376, pp. 802–816, 1981.
[21] M. Aldrin and E. Damsleth, “Forecasting Non-seasonal Time Series with Missing Observations,” *Journal of Forecasting*, vol. 8, no. 2, pp. 97–116, 1989.
[22] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014, http://www.R-project.org/.
[23] IJCAI, *IJCAI 2017 - Data Mining Contest*, 08.02.2017, http://tb.am/s0a3o.