

Forecasting Electricity Consumption and Production in Smart Homes through Statistical Methods

Arpad Gellert^{1,*}, Ugo Fiore², Adrian Florea¹, Radu Chis¹, Francesco Palmieri³

¹ Computer Science and Electrical Engineering Department, Lucian Blaga University of Sibiu, Romania

² Department of Management Studies and Quantitative Methods, Parthenope University of Naples, Italy

³ Department of Computer Science, University of Salerno, Italy

* arpad.gellert@ulbsibiu.ro

Abstract— Over the last years, a steady increase in both domestic electricity consumption and in the adoption of personal clean energy production systems has been observed worldwide. By analyzing energy consumption and production on photovoltaic panels mounted in a house, this work focuses on finding patterns in electrical energy consumption and devising a predictive model. Our goal is to find an accurate method to predict electrical energy consumption and production. Being able to anticipate how consumers will use energy in the near future, homeowners, companies and governments may optimize their behavior and the import and export of electricity. We evaluated the ARIMA and TBATS statistical prediction methods and compared them with other models on datasets from a household equipped with photovoltaics and an energy management system. The evaluation results have shown a mean absolute error of 73.62 Watts for the TBATS model, which is far better than the one obtained with neural forecasting methods.

Keywords: ARIMA, TBATS, electricity prediction, energy management system, photovoltaic panels, smart house

I. INTRODUCTION

Environmental degradation and energy efficiency represent one of the big challenges of today's society. Taking into consideration the expansion of the IoT (Internet of Things) with more than 95 billion interconnected devices expected in 2025, some grim estimates predict that, without a reduction in the consumption for all devices, in 2040 the energy required will exceed that which will be produced [1]. Among the reasons that led to the increase in energy consumption in buildings (over 40% in the EU and US) [2] we recall: population growth, COVID-19 which limited mobility and forced people to spend more time indoors, and the global climate change.

A more efficient production and use of electricity reduces both the amount of fuel needed to generate electricity and the quantity of greenhouse gases and other air pollutants emitted as a result. Environmental specialists and the EU promote energy efficiency within the European Union by recommending alternative solutions for producing electricity from renewable resources, like solar, geothermal and wind energy that do not affect climate or air

pollution. Beyond the social character regarding the effect on the environment of the electricity production and consumption, the economic character must also be taken into account. For example, for regional electricity supply agencies, forecasting electricity consumption at agency level is very important. The average monthly deviation (AMD) represents a key performance indicator:

$$AMD = 100 \cdot (AMC - MF) / MF \quad (1)$$

where AMC means actual monthly consumption of electricity and MF stands for monthly forecast of electricity. Regional electricity distribution companies (at least in Romania) rely on the AMD indicator for an accurate estimate of the monthly consumption in order to be able to buy the exact amount of electricity needed for the region. If too much energy is bought, the distribution operator has to sell (most often in loss), and if they initially buy too little for that month, they have to buy later to cover the requests, possibly at a higher price. Under these conditions, it is desirable that the AMD is very close to 0, which means an almost perfect prediction of energy consumption for the next month.

Among the external factors that can influence the fluctuating energy consumption, we mention:

- Variations of weather conditions and implicitly of temperatures, precipitation and cloudiness. Thus, the winter that came too early or left relatively late in certain years, in comparison with the usual case (average), can cause the thermal heater to start up additionally. Analogously, a long summer can determine additional electricity consumption from heating, ventilation, air conditioning (HVAC).
- A harsher winter during legal (national) or religious winter holidays, when many people are on vacation and staying longer at home, using more electricity.
- Migration of large consumers to other distribution operators or arrival (contracting) of new consumers to the distribution operator.
- Intensification of industrial or production activities in the night shifts, especially (but not only) in the period before major events (end of year, Christmas or Easter) when firms aim to match the increased demand and generate additional revenue.
- The architecture of buildings, the character and degree of occupancy, family composition, the living behavior and the regime of use of the lighting and HVAC systems [3].
- Unexpected breakdowns in power stations or on low and medium voltage power lines.

Regional agencies aim at maintaining an average monthly deviation of the electricity consumption forecast of approx. 5% for economic reasons. For example, for a small city with a monthly consumption of approximately 120,000 MWh, an improvement in prediction accuracy by only 0.13% results in a cost saving of 1500 Euros.

Forecasting energy production and consumption can help make automated decisions in households with photovoltaic panels installed, to reduce power consumption from the grid and also ameliorate the carbon dioxide footprint. Nowadays, photovoltaics allow decentralized electricity production at a low cost. When energy storage systems are available, the consumption of self-produced electricity can be significantly increased with an intelligent energy management system able to streamline electricity production and

consumption [4]. As not all houses are equipped with photovoltaic panels and storage systems, an evaluation of the potential savings achievable could be a factor influencing the decision to invest into such systems.

With the trend of embedded devices being more and more interconnected in the IoT, we can expect forecasting algorithms to be present in such devices as well, and to be able to influence and give indications on electrical energy consumption behaviors. One such application might be in photovoltaic panels. Like all "smart" devices, it may have downsides, like security vulnerabilities, but its pros outweigh the cons. These "smart" capabilities are enhanced by the increasing computational power of embedded devices, through which forecasting operations on the consumption and production of electrical energy for a house may be made.

While several methods to forecast aggregated electricity demand have been published [5], and the prediction for campuses [6], office buildings [7] and hotels [8] have also been addressed, the pattern of electricity usage in a household raises some difficulties owing to the small scale, with effects of individual behavior [9].

In this stage of our research, the main goal is to determine the best method and its optimal configuration (from analysis of different prediction methods) which can be integrated into an embedded smart energy management system. The role of such a system is to adjust and synchronize through prediction the electricity consumption and production in a smart house. Based on predictions of electricity consumption and production, the management system can make decisions in order to increase self-consumption from an energy storage system, reducing the intake from the power grid and thus decreasing the total annual operating cost, with the additional benefit of reducing losses in distribution networks. The energy management system may decide to activate some household appliances when cheap electricity is available and to delay their activation when only high-cost electricity is available.

Accurate prediction of electricity consumption and production is crucial, since it directly influences the efficiency of smart energy management systems and the amount of electricity used from self-production. Models for predicting energy demand take into account several predictor variables, such as time of day, temperature, season and other social elements. On the consumption side, the time of day reflects the diurnal pattern of human activity, while on the supply side it relates to sunlight peaks which influence the production of photovoltaic panels. The outside temperature is also important when electricity is used for heating and cooling. Seasonal cycles will influence all of the above. For the forecasting of such seasonal data the ARIMA and TBATS statistical models are good candidates. In this work we comparatively present two prediction models – ARIMA and TBATS – applied on the data recorded by the FENECON Energy Management System (FEMS) [10] to forecast electricity consumption and production in a smart home. We used the same dataset from our previous work [4] since we compare the ARIMA and the TBATS statistical prediction methods with our previous work. The AutoRegressive Integrated Moving Average (ARIMA) algorithm was selected since it is a combination of several other models and can generalize them, covering a wide range of possibilities through its parameters. We will analyze several models like the ARIMA, Seasonal ARIMA (SARIMA) and ARIMA with external regressors and possibly seasonality (SARIMAX). We predict electricity consumption and production at a

granularity of five minutes using these models. For better prediction results then we proceed by analyzing the TBATS model.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents electricity prediction using the ARIMA and TBATS models. Section IV illustrates the experimental methodology and results, studies the impacts and potential applications. Section V discusses a case study and a backcasting strategy. Finally, Section VI reports conclusions and some possible directions for further study.

II. RELATED WORK

In [11], the authors used linear regression analysis and quadratic regression analysis to predict hourly and daily energy consumption in a residential building. They observed that the longer the interval between the observations, the better the quality of the prediction model, which is explained by the fact that the differences between individual effects are averaged over the longer time periods. The use of solar radiation improves the coefficient of determination, but affects the root mean square error. The authors concluded that quadratic regression model can provide better results for shorter intervals, for example on hourly data.

In [12], the authors propose convolutional neural networks (CNN) used together with long short-term memory (LSTM) to predict residential electricity consumption. These methods combined together can extract complex electricity consumption features. The CNN can extract features of variables that affect the consumption. The outputs of the CNN layers are applied as input to the LSTM which can model the irregular trends of the electricity consumption. The output of the LSTM layer is then passed to a fully connected layer which provides the predicted electricity consumption. In [13], a similar methodology is applied but with bi-directional LSTM. In [14], LSTMs are directly used to predict electricity consumption and production in a household equipped with photovoltaic panels. In [15], CNNs and recurrent neural networks are analyzed in comparison with the SARIMAX method in terms of day-ahead building-level load forecasting. The authors concluded that the CNN proved to be competitive.

In [16], Bedi et al. evaluated an Elman recurrent neural network and an exponential model for electricity consumption forecasting in IoT-driven buildings. They exploited the correlation between the electricity consumption and the ambient temperature along with the occupancy of the building. They used for the evaluations as target building a laboratory equipped with smart monitoring and control capabilities through IoT technology.

The original ARIMA model was introduced in 1970 by Box and Jenkins [17], and it is still widely used. The forecast R package provides a convenient interface to fitting an ARIMA model to data. These fitting and forecasting steps are somewhat equivalent to the "training" and "testing" terminology from the domain of neural networks. To model complex seasonality, terms of the Fourier series can be used instead of seasonal dummies [18]. Other models for predicting the electrical energy consumption and production of a household over a period of time include Markov Models [4] and multi-layer perceptrons (MLP) [10].

Bouzerdoum et al. investigated a hybrid model (SARIMA-SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant [19]. Pedro and Coimbra have analyzed five different models: Persistent, ARIMA, k-Nearest Neighbors, NNs and Genetic Algorithm optimized NNs. They reported the best results with NNs and Genetic Algorithm

(or in short form: GA) optimized NNs [20]. Ding et al. have proposed in 2001 a NN system forecasting a full day of the future power output of a photovoltaic plant. They report that improvements can be made if the day of the forecast is selected to match weather data [21]. This approach of selecting the day based on weather data is similar to using dummies in ARIMAX models. We would have one set of dummies for sunny days and another one for cloudy days.

For capturing different seasonal periods of a time series, multiple runs of the ARIMA algorithm are needed, in which the appropriate seasonality is used. For capturing multiple seasonal periods at a time, Fourier terms might be used. Depending on the number of Fourier terms included in the regression, the seasonal pattern may be smoother (for fewer terms) or more precise (for many terms).

III. FORECASTING ELECTRICITY PRODUCTION AND CONSUMPTION

A. Electricity Prediction Using ARIMA

An ARIMA model is written as: $ARIMA(p,d,q)$ (see [18]), where p is the order of the autoregressive part, d is the number of non-seasonal differences applied, and q is the order of the moving average part. The model is built according to equation (2):

$$y'_t = c + \sum_{i=1}^p \phi_i y'_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t \quad (2)$$

where:

- y'_t is the differenced series, the degree of differencing being given by the d parameter of the model;
- c is a constant;
- $\phi_p y'_{t-p}$ are Auto Regressive (AR) terms, their number being defined by the p parameter of the ARIMA model and they being weighted by the ϕ_p coefficients which are computed during the fitting phase;
- $\theta_q \varepsilon_{t-q}$ are Moving Average (MA) terms, again determined in number by the q parameter of the ARIMA model, with the coefficients θ_q being computed during the fitting phase;
- ε_t are i.i.d. error terms with zero mean.

To model seasonal patterns, four other parameters should be added to (2), obtaining an $ARIMA(p,d,q)(P,D,Q)[m]$ (see [18], [22], and [23]) also known as SARIMA:

$$y'_t = c + \sum_{i=1}^p \phi_i y'_{t-i} + \sum_{h=1}^P \Phi_h y'_{t-m-h} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t + \sum_{k=1}^Q \Theta_k \varepsilon_{t-m-k} + \varepsilon_{T-m} \quad (3)$$

where:

- P, D, Q are the orders of the autoregressive, differencing and moving average for the seasonal part of the data;
- $\Phi_p y'_{t-m-p}$ represents one or more Seasonal Auto Regressive terms (SAR), their number is defined by the Φ_p parameter of the SARIMA model and are weighted by the coefficients which are computed during the fitting phase of the SARIMA model;
- $\Theta_Q \varepsilon_{t-m-Q}$ represents one or more Seasonal Moving Average terms (SMA), again, their number is defined by the Q parameter of the ARIMA model and are weighted by the Θ_Q coefficients which are computed during the fitting phase of the SARIMA model,

similarly to the SAR coefficients Φ_p ;

- m is fixed to the seasonal period of the data.

ARIMA and SARIMA models only include information about the time series itself in the model. The (S)ARIMAX algorithm incorporates external regressors, writing $y'_t = \beta x_t + n'_t$, where x_t is the external regressors time series and β is the coefficient, and reinterpreting Eqs. (2) and (3) as written for the residuals n'_t (see [24], [25] and [18]).

B. Electricity Prediction using TBATS

The next analyzed model is TBATS (Trigonometric Seasonal, Box-Cox Transformation, ARMA residuals, Trend and Seasonality), introduced in [26]. The TBATS($\omega, \varphi, p, q, \{m_1, k_1\}, \dots, \{m_T, k_T\}$) model has the following parameters: ω is the Box-Cox parameter, φ is the dumping parameter, p and q are the ARMA parameters, m_1, \dots, m_T are the seasonal periods, and k_i is the number of harmonics necessary for the i^{th} seasonal component.

The Box-Cox transformation $y_t^{(\omega)}$ can be computed using the following formula, with the parameter ω :

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^{(\omega)} - 1}{\omega} & , \omega \neq 0 \\ \log(y_t) & , \omega = 0 \end{cases} \quad (4)$$

$$y_t^{(\omega)} = l_{t-1} + \varphi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t \quad (5)$$

$$l_t = l_{t-1} + \varphi b_{t-1} + \alpha d_t \quad (6)$$

$$b_t = (1 - \varphi)b + \varphi b_{t-1} + \beta d_t \quad (7)$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t \quad (8)$$

$$d_t = \sum_{i=1}^p \gamma_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (9)$$

where b is the long-run trend, b_t is the short-run trend in period t , l_t is the level component at time t , $s_t^{(i)}$ is the i^{th} seasonal component at time t , d_t is prediction error, ε_t is a Gaussian white noise process with zero mean and constant variance σ^2 , and α, β, s and γ_i are the smoothing parameters. For a higher flexibility, the trigonometric representation of the seasonal components based on Fourier series have been introduced:

$$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)} \quad (10)$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \quad (11)$$

$$s_{j,t}^{*(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \quad (12)$$

$$\lambda_j^{(i)} = \frac{2\pi j}{m_i} \quad (13)$$

where $\gamma_1^{(i)}$ and $\gamma_2^{(i)}$ are the smoothing parameters, $s_{j,t}^{(i)}$ is the stochastic level of the i^{th} seasonal component, and $s_{j,t}^{*(i)}$ is the stochastic growth of the i^{th} seasonal component.

The TBATS models can decompose seasonal time series into trend, seasonal and irregular components. The trigonometric terms from TBATS do not need normalization and the overall seasonal component can be decomposed to multiple seasonal components having

different frequencies. As it is shown in [26], TBATS allows automated model selection, which makes it easier to apply in comparison with ARIMA.

IV. EXPERIMENTAL RESULTS

The experimental installation consists of two 12.24 kWp photovoltaics and an 8.5 kWh “FENECON by BYD PRO Hybrid” energy storage system, supplying the electricity for a household. The data has been collected with a previously developed energy monitoring system, called FEMS, and stored in a computer through FENECON Online-Monitoring, as it is described in [10]. We used the dataset presented in [10], which consists of five time series, called Ph1, Ph2, Ph3, PV1 and PV2. The first three correspond to each phase for three-phase current while the last two represent the electrical energy of two photovoltaic panels installed in the same household. The sampling frequency is five minutes. Data were captured over five months, from January 2015 to May 2015. This means that for each time series we have 150 days’ worth of data, with 288 samples per day or about 43200 data points in a single time series. For the considerations about granularity explained previously in Section III and because the sheer amount of data points makes impractical to apply ARIMA to the original time series, we have further transformed the initial data by downsampling it. For each of the five time series we created other two, one where we keep a data point every hour and one where we keep a data point every two hours from the original time series. As we will see, downsampling did not lose too much information, since the results of the ARIMA models applied to them will be very similar with those obtained on the initial data.

Fig. 1 illustrates how the statistical prediction methods are integrated into the energy management system. In the first stage, historical data about the produced and consumed electricity is collected. In the next stage, the collected data is checked for erroneous values and corrected. The data is also encoded in this preprocessing stage. In the forecasting stage, the predictor computes the electric power for the next period, based on the available historical data. The predicted electricity production and consumption are used to make the appropriate decisions in the view of efficient energy management, with positive environmental, social and economic impact.

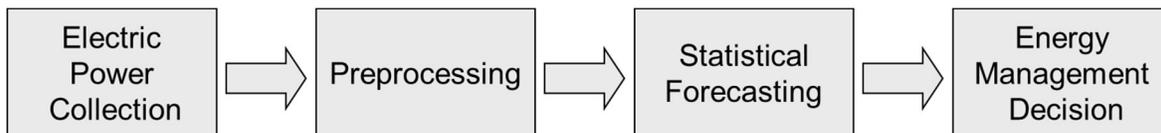


Fig. 1. Energy management system

As experimental methodology, we analyzed the data by looking at pieces of it, each piece being approximately the size of a week, and for avoiding assumptions about the representativeness of that week, we always model with ARIMA a single week and produce forecasts for several days. This approach also falls in line with the Pareto principle, having approximately a week for training and several days for testing. The exact size for the training days and test days is determined based on the so-called 80/20 rule [27]. This "week by week" forecasting approach is also useful in parallelizing the code. The code is embarrassingly parallel since every data chunk is independent of the previous or the next chunk. We can

create an ARIMA model for a week and produce forecasts from it without influencing in any way the model and predictions for the previous or next week, for which all the data is independent. Of course, the parallelization was done using the *foreach* and *doParallel* R packages.

Forecasting errors were used for comparing models. When a model is tested on N data samples, several measures can be used to evaluate its performance [28], including the Mean Absolute Error:

$$MAE = \frac{1}{N} \sum_{t=1}^N |y_t - \hat{y}_t| \quad (14)$$

Further, this section presents the main results and provides some insight into the causes that led to these results. All the experiments were run on a quad-core Intel Core i5 processor with 8GB of RAM.

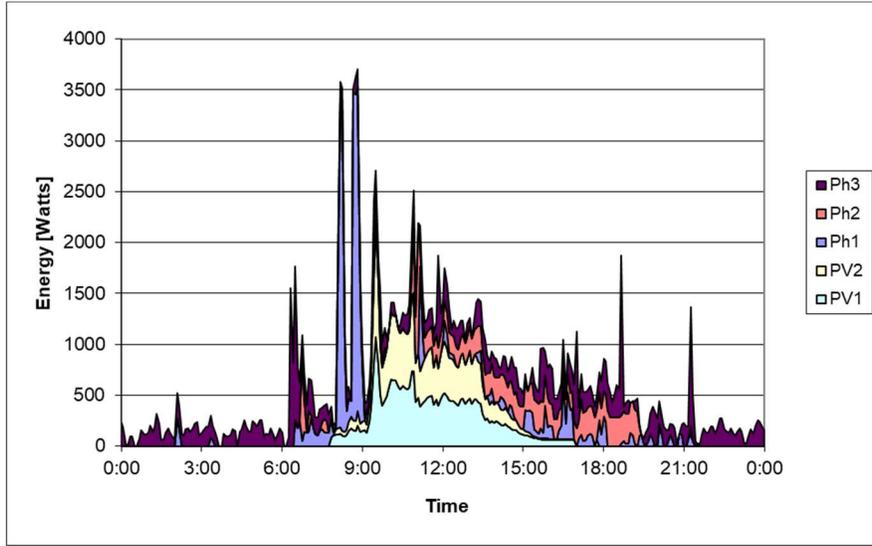


Fig. 2. The electrical energy consumption and production for one day (7th January 2015)

First of all, we note (Fig. 2) that the energy consumption (Ph1, Ph2 and Ph3) and photovoltaic production (PV1 and PV2) curves have different patterns. The photovoltaic production curves are typically smooth, showing peaks around noon depending on weather conditions, whereas electricity consumption often shows peaks in the morning and evening. This confirms the potential for an energy management system that matches production and consumption through smart automated decisions. What follows next are tables that provide the main results for each of the time series, together with their downsampled versions, and descriptions for each set of results. All the tables from this section will have the same format and meanings for the columns, namely:

- Time series: the series for which we applied the model;
- Model: the forecasting model applied to the series (this will respect the $ARIMA(p,d,q)$ or $ARIMA(p,d,q)(P,D,Q)$ notations shown in the rest of the paper, where p is the order of the autoregressive part, d is the number of non-seasonal differences applied, and q

is the order of the moving average part, whereas P, D, Q are the orders of the autoregressive, differencing and moving average for the seasonal part of the data);

- K : the number of Fourier terms, a hyphen being shown instead if none were used;
- Dummies: the dummies configuration in the form of “*start:length*”, where *start* represents the hour of the start dummy and *length* shows how many dummies were used (in hours), a hyphen being shown instead if none were used;
- Train and test days: the number of training days for the fitting phase of the algorithm and the number of the test days on which the accuracy was tested;
- MAE.

Table 1 shows the models chosen for the Ph1 data series, first the series sampled every two hours is shown, then the series sampled every one hour, and finally the original series sampled every 5 minutes.

Table 1. Models for the Ph1 data series

Time series	Model	K	Dummies (start:length)	Train days	Test days	MAE
2hrs Ph1	ARIMA(1, 0, 0)	2	-	7	2	220.87
2hrs Ph1	ARIMA(1, 0, 0)	2	12:6	7	2	218.81
2hrs Ph1	ARIMA(1, 0, 0)(1, 0, 0)	2	-	7	2	223.40
2hrs Ph1	ARIMA(1, 0, 0)(1, 0, 0)	2	12:6	7	2	222.06
1hrs Ph1	ARIMA(1, 0, 0)	2	-	7	2	217.44
1hrs Ph1	ARIMA(1, 0, 0)	2	11:1	7	2	215.51
1hrs Ph1	ARIMA(1, 0, 0)(1, 0, 0)	2	-	7	2	218.36
1hrs Ph1	ARIMA(1, 0, 0)(1, 0, 0)	2	11:1	7	2	217.18
5min Ph1	ARIMA(1, 0, 0)	2	-	7	2	223.15
5min Ph1	ARIMA(1, 0, 0)	2	11:1	7	2	218.04

For each series, several models have been fit and used for predictions. The results for the Ph1 series show that the ARIMA(1,0,0) with $K=2$ (two Fourier terms) and dummies starting at 11:00 and ending at 12:00 is best suited for forecasting the Ph1 data series with the lowest errors. The reason for choosing to use the same dummy values as for the hourly sampled data set, is simple: the more frequently sampled data set (i.e., the hourly Ph1 series) is a closer approximation of the real data than are the series sampled every two hours.

Next, we will present the results for the Ph2 data series. Table 2 presents the models and their errors on the original and down-sampled versions of the Ph2 time series. As shown in Table 2, the best results for the original data series are provided by a model that consists of a combination of dynamic harmonic regression (in this case with $K = 3$), non-seasonal ARIMA errors with one autoregressive term and dummies starting in the morning hours, lasting until noon (8:00 to 11:00). For this series the best results are achieved if we fit the model on four days’ worth of data and predict three, thus going a bit against the Pareto principle, which stated that we should use 70% or 80% of the data for learning and the rest for testing, here the proportions are closer to 60% for the training data and 40% for the test data.

Table 2. Models for the Ph2 data series

Time series	Model	K	Dummies (start:length)	Train days	Test days	MAE
2hrs Ph2	ARIMA(1, 0, 0)	1	-	7	3	185.72
2hrs Ph2	ARIMA(1, 0, 0)	1	8:8	7	3	178.53
2hrs Ph2	ARIMA(1, 0, 0)(1, 0, 0)	1	-	7	3	184.51
2hrs Ph2	ARIMA(1, 0, 0)(1, 0, 0)	1	8:8	7	3	178.84
1hrs Ph2	ARIMA(1, 0, 0)	3	-	4	3	159.89
1hrs Ph2	ARIMA(1, 0, 0)	3	8:7	4	3	157.91
1hrs Ph2	ARIMA(1, 0, 0)(1, 0, 0)	3	-	4	3	165.76
1hrs Ph2	ARIMA(1, 0, 0)(1, 0, 0)	3	8:7	4	3	164.03
5min Ph2	ARIMA(1, 0, 0)	3	-	4	3	163.15
5min Ph2	ARIMA(1, 0, 0)	3	8:3	4	3	161.40

The results for the Ph3 data series are not presenting any surprise, the best model for this series is shown in Table 3, it consists of dynamic harmonic regression with two Fourier terms ($K = 2$) and dummies between 9:00 and 11:00 with a first-order autoregressive ARIMA error.

Table 3. Models for the Ph3 data series

Time series	Model	K	Dummies (start:length)	Train days	Test days	MAE
2hrs Ph3	ARIMA(1, 0, 0)	2	-	7	3	183.13
2hrs Ph3	ARIMA(1, 0, 0)	2	10:2	7	3	179.03
2hrs Ph3	ARIMA(1, 0, 0)(1, 0, 0)	2	-	7	3	182.83
2hrs Ph3	ARIMA(1, 0, 0)(1, 0, 0)	2	10:2	7	3	179.90
1hrs Ph3	ARIMA(1, 0, 0)	2	-	7	2	179.01
1hrs Ph3	ARIMA(1, 0, 0)	2	9:2	7	2	172.99
1hrs Ph3	ARIMA(1, 0, 0)(1, 0, 0)	2	-	7	2	179.53
1hrs Ph3	ARIMA(1, 0, 0)(1, 0, 0)	2	9:2	7	2	174.30
5min Ph3	ARIMA(1, 0, 0)	2	-	7	1	187.96
5min Ph3	ARIMA(1, 0, 0)	2	9:2	7	2	183.96

For this data series, it appears best to fit the model on seven days and predict maximum two days or even a single one. For this model, as for the others, the hourly sampled data dummies apply best to the original time series. An intermediary conclusion that we can draw is the fact that all the phases have similar behaviors, all three of them having similar models that fit best. All observations can be deduced from the previous one (first-order autoregressive ARIMA model), all are seasonal and the observations that influence the results are situated in the first part of the day, morning to noon. This is unsurprising, since at that time people prepare for work or have lunch.

The next two sets of results deal with the PV1 and PV2 data series. These represent the electrical energy production of a household that has two photovoltaic panels installed. First,

Table 4 shows the results for the PV1 time series and the down-sampled series. What is fundamentally different is the time covered by the dummies. Here the interval is directly proportional with the interval the sun shines most, which, as shown here by the best model, is between 11:00 in the morning and 17:00 in the afternoon (start dummy is 11, and length is 6). In this case, we consider the best model to be the one with the lowest overall errors, hence: ARIMA(1,0,0) with three Fourier terms and dummies starting at the 11th hour and lasting 6 hours. These dummies are necessary to capture the peaks created by the high production of electrical energy during the afternoon hours when the sun shines brightest.

Table 4. Models for the PV1 data series

Time series	Model	K	Dummies (start:length)	Train days	Test days	MAE
2hrs PV1	ARIMA(1, 0, 0)	2	-	5	2	241.34
2hrs PV1	ARIMA(1, 0, 0)	2	16:4	5	2	238.33
2hrs PV1	ARIMA(1, 0, 0)(1, 0, 0)	2	-	5	2	244.64
2hrs PV1	ARIMA(1, 0, 0)(1, 0, 0)	2	16:4	5	2	242.84
2hrs PV1	ARIMA(1, 0, 1)	2	-	5	2	242.13
2hrs PV1	ARIMA(2, 0, 0)	2	-	5	2	241.65
2hrs PV1	ARIMA(2, 0, 0)	2	16:4	5	2	239.15
1hrs PV1	ARIMA(1, 0, 0)	2	-	5	2	237.59
1hrs PV1	ARIMA(1, 0, 0)	2	11:6	5	2	234.52
1hrs PV1	ARIMA(1, 0, 0)(1, 0, 0)	2	-	5	2	232.71
1hrs PV1	ARIMA(1, 0, 0)(1, 0, 0)	2	11:6	5	2	230.87
1hrs PV1	ARIMA(2, 0, 0)	2	-	5	2	237.27
1hrs PV1	ARIMA(2, 0, 0)	2	11:6	5	2	234.58
1hrs PV1	ARIMA(3, 0, 0)	2	-	5	2	237.09
1hrs PV1	ARIMA(3, 0, 0)	2	11:6	5	2	234.80
5min PV1	ARIMA(1, 0, 0)	2	-	5	2	238.26
5min PV1	ARIMA(1, 0, 0)	2	16:4	5	2	237.10
5min PV1	ARIMA(1, 0, 0)	3	11:6	5	2	225.91

The results produced by the models on the PV2 data can be seen in Table 5. The best model in this case is ARIMA(1,0,0) with three Fourier terms for modeling the seasonality in the data. What is different in this case is the fact that we have chosen as the "best" model the one without dummies. Taking into account the similar errors, the reason behind this decision was the complexity of the model, hence choosing the simplest one. Apparently, this case is modeled sufficiently well by capturing the seasonality with the three Fourier terms and one autoregressive term for the errors. Both models for the PV1 and PV2 data work best when they are fit on five training days and when predicting two days.

Table 5. Models for the PV2 data series

Time series	Model	K	Dummies (start:length)	Train days	Test days	MAE
2hrs PV2	ARIMA(1, 0, 0)	2	-	5	2	234.55
2hrs PV2	ARIMA(1, 0, 0)	2	8:8	5	2	235.44
2hrs PV2	ARIMA(1, 0, 0)(1, 0, 0)	2	-	5	2	238.04
2hrs PV2	ARIMA(1, 0, 0)(1, 0, 0)	2	8:8	5	2	239.05
2hrs PV2	ARIMA(2, 0, 0)	2	-	5	2	234.83
2hrs PV2	ARIMA(2, 0, 0)	2	8:8	5	2	235.60
1hrs PV2	ARIMA(1, 0, 0)	3	-	5	2	220.40
1hrs PV2	ARIMA(1, 0, 0)	3	13:2	5	2	219.32
1hrs PV2	ARIMA(1, 0, 0)(1, 0, 0)	3	-	5	2	219.20
1hrs PV2	ARIMA(1, 0, 0)(1, 0, 0)	3	13:2	5	2	219.01
1hrs PV2	ARIMA(2, 0, 0)	3	-	5	2	220.30
1hrs PV2	ARIMA(2, 0, 0)	3	13:2	5	2	219.54
5min PV2	ARIMA(1, 0, 0)	3	-	5	2	220.00
5min PV2	ARIMA(1, 0, 0)	3	13:2	5	2	220.02
5min PV2	ARIMA(1, 0, 0)	2	8:8	5	2	232.03

Since the residuals from the regression with the ARIMA models are still presenting some higher values, which can lead to lower prediction accuracy, we will further apply the TBATS forecasting method (presented in Section III) for an automated selection of the best suited model (including the ARMA parameters).

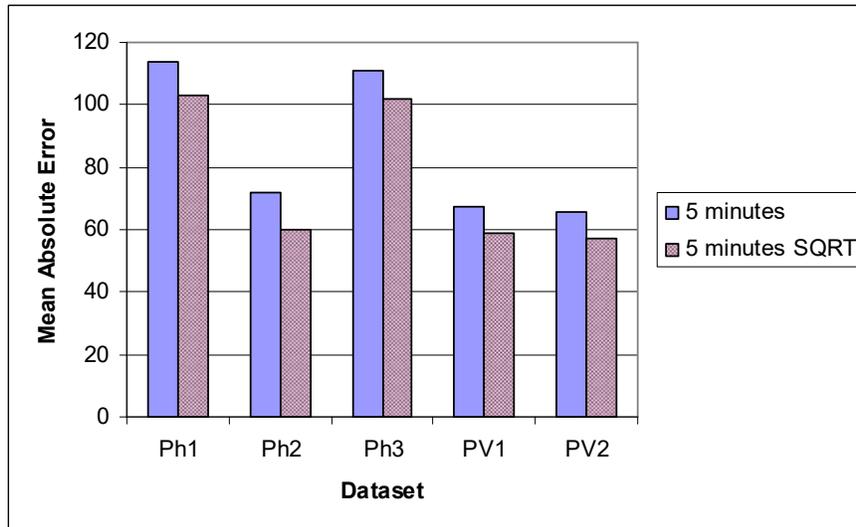


Fig. 3. MAE measured with TBATS using a history length of 500 data points on the original vs. SQRT data (5 minutes)

We have used the original 5 minutes datasets. We have also tried to provide the square root (SQRT) of the data to the TBATS model which provides the squared value of its prediction, with the goal of reducing the spikes' magnitude. Thus, we hoped to obtain a lower prediction error. Fig. 3 presents the obtained results. As Fig. 3 shows, the root-squared data has been better modeled by TBATS, the prediction errors being significantly lower. Further, we have analyzed the differences between the 5 minutes, 1 hour and 2 hours SQRT data.

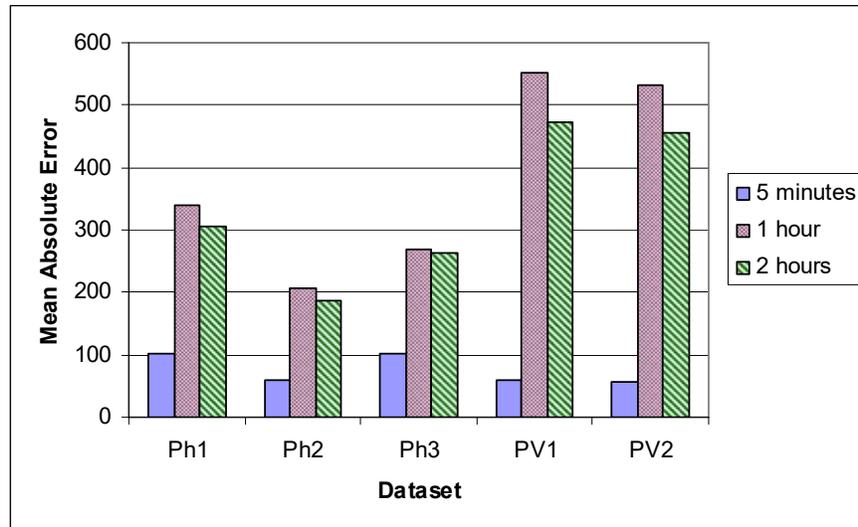


Fig. 4. MAE measured with TBATS using a history length of 500 data points on the SQRT data and considering different sampling frequencies

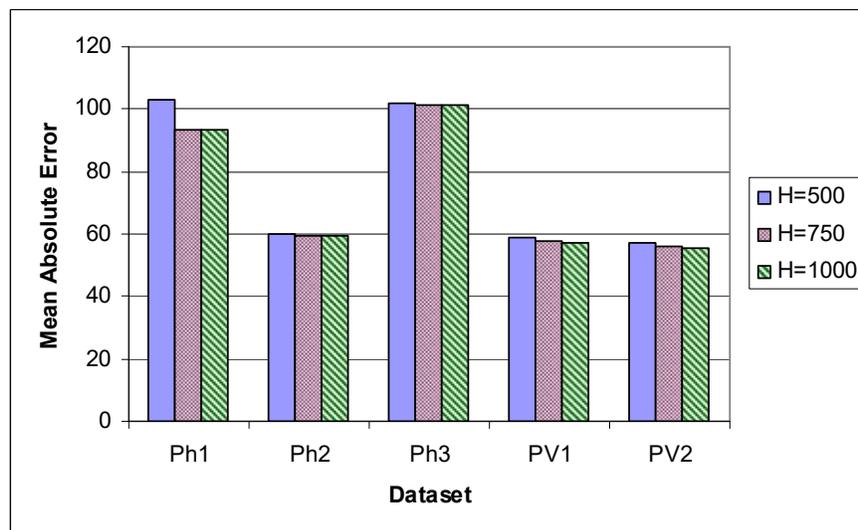


Fig. 5. MAE measured with TBATS using different history length on the 5 minutes SQRT data

As we can see in Fig. 4, the down-sampled data is much worse than the original 5 minutes data, on the TBATS model. Consequently, by using down-sampled data the model is losing essential details, which makes it less efficient in predicting electricity production and consumption. On the 5 minutes SQR data we have varied the history length. Fig. 5 presents the results. As Fig. 5 shows, a data history length of 750 is better than 500 in terms of MAE. If we further increase the data history length to 1000, on Ph2, PV1 and PV2 the results are just slightly better, whereas on Ph1 and Ph3 the results are worse. Therefore, we consider the optimal data history length as being 750.

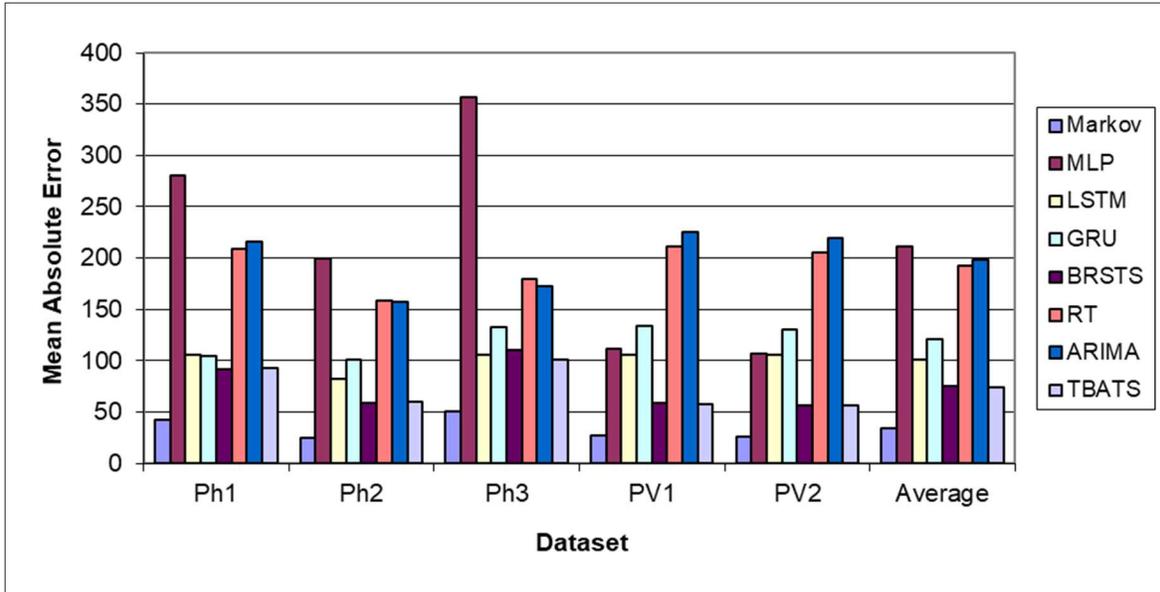


Fig. 6. Plot of the MAE from different types of models

Finally, Fig. 6 shows a comparison in terms of MAE between the Markov model used in [4] and [29], the MLP used in [10], the LSTM used in [14], the Gated Recurrent Unit (GRU) used in [30], Bayesian Regression Structural Time Series (BRSTS) model, Regression Tree (RT) and the best ARIMA and TBATS models identified in this work. It also presents the average of the MAE among all the five time series. On average, the ARIMA model performs better than the MLP, it is comparable with the RT, but far worse than the Markov, LSTM, GRU and BRSTS models. Apparently, the magnitude of the errors for the MLP, RT and the ARIMA methods is similar, while the Markov, BRSTS and TBATS models provide exceptionally good results, since their errors are one order of magnitude lower.

V. DISCUSSION

To provide an example of how the results of the proposed forecasting methods may be actually applied to a household-level smart system, a simple binary decision approach is described and evaluated. In this context, the photovoltaic panels are assumed to be coupled with a storage subsystem capable to store energy and supplement the grid supply with it at a later time. Excess energy can thus be stored during time periods when prices are low and retrieved when demand and prices are high. The residential energy management system

makes the decision whether to store energy.

The energy supplied from the grid is assumed to be priced according to two tariffs: R_1 and R_2 , (with $R_1 < R_2$) both of which are expressed in monetary units per unit of energy (e.g. €/Whatt). Tariff R_2 is assumed to apply from 8:00 to 18:00 every day excluding Sundays. Batteries are supposed to be adequately dimensioned, so they are capable of storing the energy generated by photovoltaic panels in one day. The cost of procurement and installation C , as well as the slight performance degradation which occurs at every charge/discharge cycle, can be modeled by making the hypothesis that the storage system has a lifetime of L hours. The upfront cost can be modeled as an hourly cost $R_0 = C/L$ of the energy produced by the photovoltaics. Next, considering that what really matters are the differences $R_2 - R_0$ and $R_1 - R_0$, it will be assumed that both R_1 and R_2 incorporate R_0 , so they will be thought of as incremental rates relative to R_0 .

The evaluation is performed following a time series cross validation approach. Data is considered as being acquired continuously and forecasts are made by using all of the available data up to the time of forecasting. An initial time window of 9 weeks is used for the initial calibration of the models, and an elongated window including all the observation up to the current time is considered for the forecasts.

At the beginning of a day, the forecasts for that day are generated and the smart energy management system will elaborate a plan for the day. As the forecasts get updated with the arrival of new data, the plan is adapted to better suit the current situation. At the end of each day, savings are evaluated and accrued.

We developed the following case study based on a full day energy production and consumption (the photovoltaics start producing at 7:50 in the morning and stop producing at 16:55).

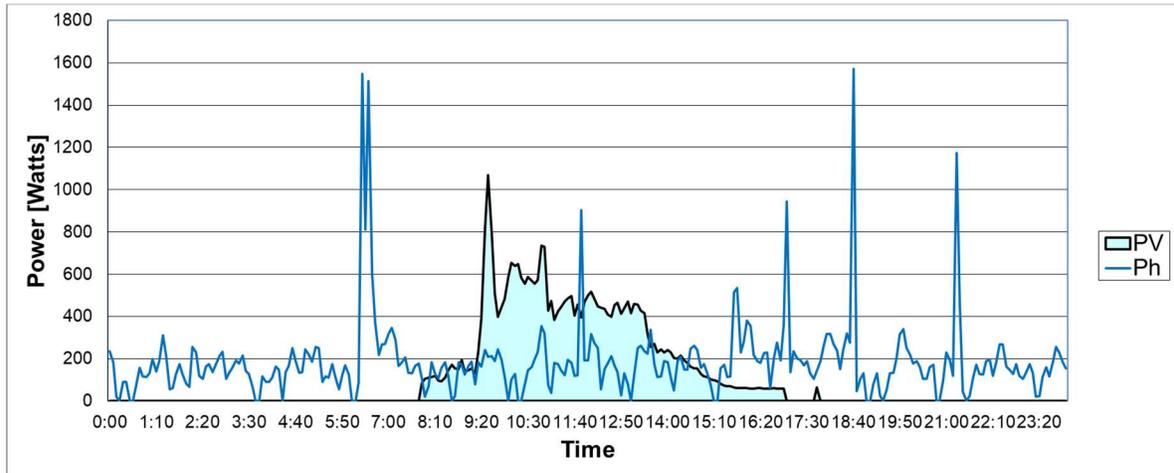


Fig. 7. Electricity consumption (Ph) and production (PV) aggregated for one day

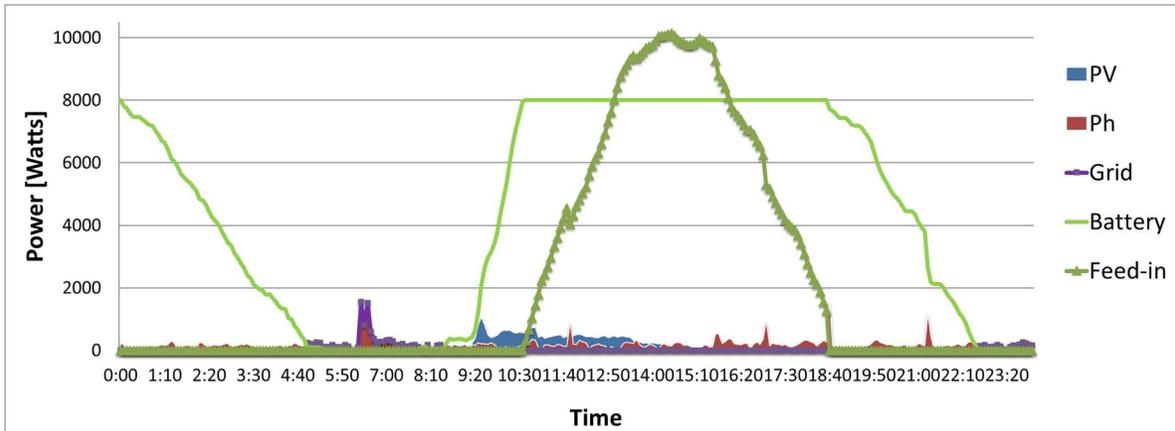


Fig. 8. Real situation using battery and grid for one day

We consider two scenarios: a social one and an economic one. The social scenario does not necessarily aim the financial benefits, mainly to reduce the personal / family costs for energy. The economic scenario targets to exploit every potential benefit by selling or buying energy depending on the price set by the energy market and house requirement. Remarks about changing contexts for a whole day:

- The analysis was performed in social context because we did not have access to the energy prices from the period when the data were collected (2015) and this could be a separate optimization study that would require much more time and more complex algorithms.
- The whole consumption (Ph) is realized from the battery from midnight until 5:20 in the morning when the battery is fully discharged. From that moment until 8:30 and in the evening after 22:35 the consumption was made from the grid. It should be mentioned that even the photovoltaics (PV) start to produce at 7:50, only after 8:30 will cover the consumption.
- From 8:30 till 10:40 the PV starts to recover the battery which will constantly charge until reaching its maximum value of 8000 Watts. In the economic scenario, the homeowner can choose to sell the energy and not charge the battery if this will bring him a financial benefit relying on the fact that later when he consumes, if his battery is discharged, the price of energy from the grid is smaller (and loses less).
- Since 10:40 when the battery is fully charged and PV continues to produce and exceed consumption, the energy is sold (Feed-in). This process ends at 18:40 when the consumption is relatively high and the PV does not produce at all.
- After the energy is no longer sold to the network (Feed-in becomes 0), the house uses the battery for own consumption until the evening at 22:35 after which the grid is used. But, if the next night the battery remains discharged, it will recharge only the next day (most likely after the same time of 7:50) and overnight will use the grid for internal consumption.
- Our algorithm can set time horizons in both scenarios in which the owner can decide from where to consume energy and whether or not to sell in the network. The moment when the consumption is changed from the battery to the grid or conversely and also

the degree of charge of the battery in the evening for the next day may vary depending on the day or on the season (production and consumption being influenced by several environmental factors) [3]. For this reason, the analysis must be extended to a long period, but our study wanted to show the major decision stages that our predictive system can provide.

- An extensive study that can be done in the economic scenario must follow in parallel with the prediction system developed by us, the price of energy on the market for the next day¹ (MND). This is a component of the wholesale electricity market on which firm transactions with electricity for delivery on the next day are hourly realized based on the offers submitted by the MND participants. This study is left for further research. Since 17.06.2021, the MND in Romania started operating in a coupled mechanism at European level with the implementation of the DE-AT-PL-4M MC project², also known as Interim Coupling.

Ensuring sustainability of cities and society regarding energy requires significant shifts in both production and consumption patterns. The backcasting techniques could help but, unfortunately, we did not collect data about each household behavior or each power consumer like home heating, personal washing, heating the food, ironing, cleaning etc., neither weather conditions from that period.

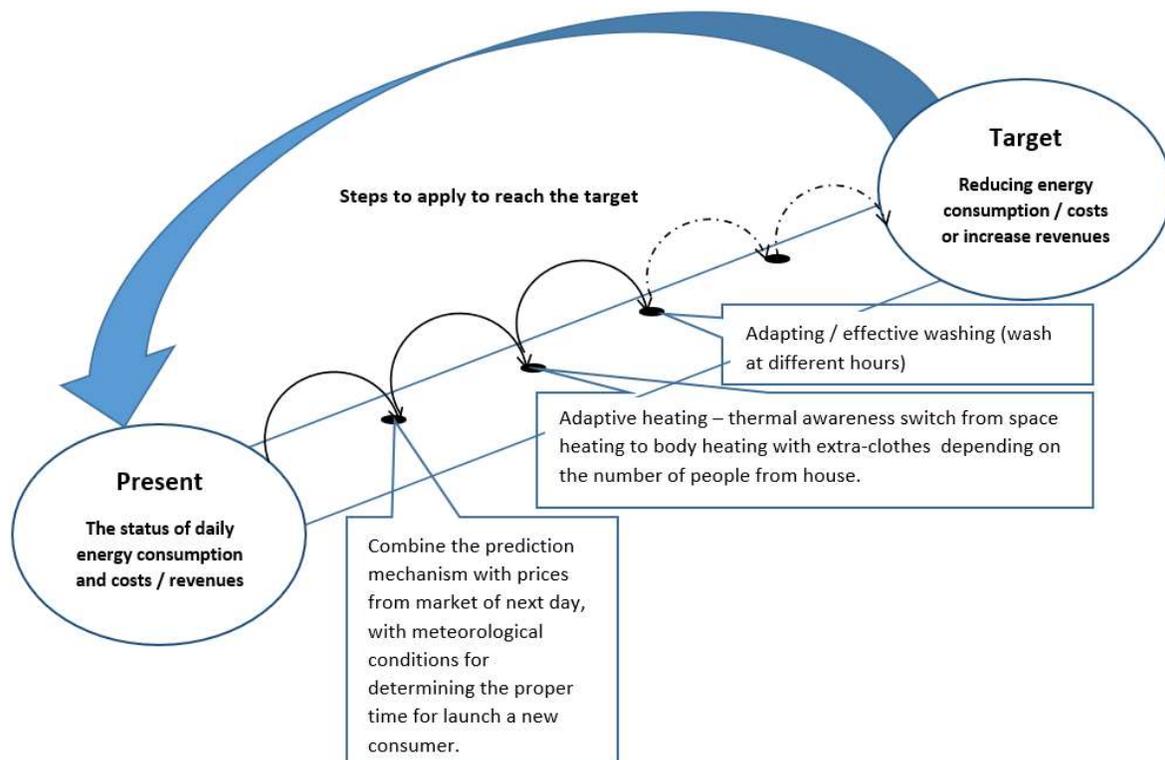


Fig. 9. Backcasting

¹ https://www.opcom.ro/pp/grafice_ip/raportPIPsiVolumTranzactionat.php?lang=ro#url

² <https://www.ote-cr.cz/en/short-term-markets/market-coupling-day-ahead-market/de-at-pl-4m-mc>

One example that could be implemented in the future is based on shifting of some household activities that consume high power in the night when the consumption decrease even to 0. For example, the clothes washing machine, dishwasher or electric bread oven could be scheduled to start at night, or very early in morning when the rest of the consumers are idle or stopped. The next scheme presents our tentative of backcasting strategy to improve the house energy efficiency and maybe to increase the revenues.

VI. CONCLUSIONS AND FURTHER WORK

The energy consumption and photovoltaic production curves can show different patterns. While the photovoltaic curves are typically smooth, showing peaks around noon, electricity consumption is highly dependent on individual parameters. Thus, households often show peaks in the morning and evening. Therefore, the final target is an energy management by matching the production and consumption curves.

In this work the goal was forecasting the electricity consumption and production of a house over a period of time. Two statistical prediction methods were implemented to find their performance comparing with several other models to determine the best method, such that its optimal configuration can be integrated into an embedded smart energy management system. For this, we have used the ARIMA and the TBATS models. The usage of the sampled data acquired at every 5 minutes provided the best prediction results on the TBATS model. The TBATS model had a prediction error lower with 11.15% at average on the five datasets when we used the SQR data with respect to the original data. The evaluation results have shown a mean absolute error of 73.62 Watts with the TBATS model using a history of 750 data points, which is far better than the ARIMA (198.27 Watts), RT (192.71 Watts), MLP (211.07 Watts), LSTM (100.77 Watts) and GRU (120.64 Watts) and even than the BRSTS model (75.38 Watts).

Producing usable forecasts from electrical consumption and production data is possible and the presented forecasting principles could be used in the future for smart devices and smart homes to optimize the consumption of electricity [31], or any other resource for that matter, to ensure a better, greener, healthier environment.

One possible reason for the superiority of TBATS over ARIMA consists in the fact that the TBATS is highly suitable for forecasting time series rich in complex seasonal patterns. Moreover, TBATS allows automated model selection and therefore can easily find a more suitable configuration. The ARIMA models give good accuracy in forecasting relatively stationary time series. Regarding the Markov model, its superiority could be assured by the fact that the electricity demand profile of a certain household usually remains constant for long time. Other forecasting methods, like neural networks (which are nonlinear methods), provide less accuracy if they are not embedded into a framework that adapts their use to individual household attributes. The Markov and TBATS models can faster adapt to changes in the data than neural networks.

A future research direction consists in the analysis of large datasets for different places of consumption from domestic customers connected to the regional/national electricity network to establish the performance of ARIMA and TBATS predictors. We will also try to establish a residence profile by recognizing household appliances because the electricity consumption of a household changes over time based on the operation of individual appliances used

differently by each family. The identification of house appliances may be an additional feature used for more accurate energy consumption forecasts. Forecasts would feed a dedicated optimization system, leveraging on the flexibility of nature-inspired algorithms and the potential of hybridization, as recently applied with success in cognate fields [32]. In a wider perspective, forecasts obtained could be integrated with a stream of energy prices and the ability to charge one or more batteries to store temporarily the surplus energy from panels, resulting in the minimization of household expenditure to use power from the grid or internally. The way prices evolve and the characteristics of batteries (capacity, charging time, working life in charge cycles) would involve simulations to generate realistic scenarios to work with. In addition, further refinements can be achieved by integrating other external regressors, such as weather conditions, temperature, and the number of occupants in a building, all of which can be readily obtained by installing sensors. An extensive study that can be done in the economic scenario and the use of backcasting techniques based on additionally collected data are other further work directions.

ACKNOWLEDGEMENT

Project financed by Lucian Blaga University of Sibiu & Hasso Plattner Foundation research grants LBUS-IRG-2019-05. We express our thanks to our M.Sc. students Paul-Gheorghe Barbu and Ana-Maria Oros for their help.

REFERENCES

- [1] M. Duranton, K.D. Bosschere, C. Gamrat, J. Maebe, H. Munk, O. Zendra. (2017). The HIPEAC Vision 2017, <https://www.hipeac.net/vision/2017/> , accessed 27 February 2019.
- [2] X. Cao, X. Dai, J. Liu. (2016). Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and Buildings*, 128, 198-213.
- [3] J.A. Oliveira-Lima, R. Morais, J.F. Martins, A. Florea, C. Lima. (2016). Load forecast on intelligent buildings based on temporary occupancy monitoring. *Energy and Buildings*, 116, 512-521.
- [4] A. Gellert, A. Florea, U. Fiore, F. Palmieri, and P. Zanetti. (2019) A study on forecasting electricity production and consumption in smart cities and factories. *International Journal of Information Management*, 49, 546–556.
- [5] T. Ahmad, H. Zhang, B. Yan. (2020). A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. *Sustainable Cities and Society*, 55, 102052.
- [6] M.K. Kim, Y. S. Kim, J. Srebric. (2020). Predictions of electricity consumption in a campus building using occupant rates and weather elements with sensitivity analysis: Artificial neural network vs. linear regression. *Sustainable Cities and Society*, 62, 102385.
- [7] M. Ilbeigi, M. Ghomeishi, A. Dehghanbanadaki. (2020). Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. *Sustainable Cities and Society*, 61, 102325.

- [8] M. Shao M., X. Wang, Z. Bu, X. Chen, Y. Wang. (2020). Prediction of energy consumption in hotel buildings via support vector machines. *Sustainable Cities and Society*, 57, 102128.
- [9] F. Falaki, A. Merabtine, D. Martouzet. (2021). A Spatio-Temporal Analysis of electric appliance end-use demand in the residential sector: Case study of Tours (France). *Sustainable Cities and Society*, 65, 102635.
- [10] S. Feilmeier. (2015) Loads management based on Photovoltaic and Energy Storage System, M.S. thesis, Lucian Blaga University of Sibiu.
- [11] N. Fumo, M.A.R. Biswas. (2015). Regression analysis for prediction of residential energy consumption. *Renewable and Sustainable Energy Reviews*, 47, 332-343.
- [12] T.-Y. Kim, S.-B. Cho. (2019). Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182, 72-81.
- [13] T. Le, M.T. Vo, B. Vo, E. Hwang, S. Rho, S.W. Baik. (2019). Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM. *Applied Sciences*, 9(20).
- [14] M.-A. Bachici, A. Gellert. (2020). Modeling Electricity Consumption and Production in Smart Homes using LSTM Networks. *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences*, 10(1), 80-90.
- [15] M. Cai, M. Pipattanasomporn, S. Rahman. (2019). Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Applied Energy*, 236, 1078-1088.
- [16] G. Bedi, G.K. Venayagamoorthy, R. Singh. (2020). Development of an IoT-Driven Building Environment for Prediction of Electric Energy Consumption. *IEEE Internet of Things Journal*, 7(6), 4912-4921.
- [17] G.E.P Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung. (2015). Time Series Analysis: Forecasting and Control, 5th ed., Wiley.
- [18] R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, 2nd ed.; OTexts: Melbourne, Australia. Available: <https://otexts.com/fpp2>
- [19] M. Bouzardoum, A. Mellit, and A. M. Pavan. (2013). A hybrid model (SARIMA–SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant. *Solar Energy*, 98, 226–235.
- [20] H. T. C. Pedro, and C. F. M. Coimbra. (2012). Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*, 86, 2017–2028.
- [21] M. Ding, L. Wang, and R. Bi. (2011). An ANN-based approach for forecasting the power output of photovoltaic system. *Procedia Environmental Sciences*, 11, 1308–1315.
- [22] R. J. Hyndman, Y. Khandakar. (2008). Automatic time series forecasting: the forecast package for R, *Journal of Statistical Software*, 26, 1–22.
- [23] A. Fernandez, SARIMA models, Available: https://www.stat.berkeley.edu/~arturof/Teaching/STAT248/lab07_part1.html
- [24] R. Nau, ARIMA models with regressors, Available: <https://people.duke.edu/~rnau/arimreg.htm>
- [25] R.J. Hyndman, The ARIMAX model, Available: <http://robjhyndman.com/hyndsight/arimax/>

- [26] A.M. De Livera, R. J. Hyndman, and R. D. Snyder. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496), 1513–1527.
- [27] G.E.P. Box, R.D. Meyer. (1986). An analysis for unreplicated fractional factorials. *Technometrics*, 28(1), 11–18.
- [28] R.J. Hyndman, A.B. Koehler. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688.
- [29] V.C. Antonescu. (2017). Predictia consumului si a productiei de energie folosind metode contextuale. B.Sc. Thesis, Lucian Blaga University of Sibiu.
- [30] G.A. Le Roy. (2021). Forecasting Electricity Consumption and Production using GRU Networks. M.Sc. Thesis, Lucian Blaga University of Sibiu
- [31] D. Sembroiz, D. Careglio, S. Ricciardi, U. Fiore. (2019). Planning and operational energy optimization solutions for smart buildings. *Information Sciences*, 476, 439–452.
- [32] P. Vasant, J. A. Marmolejo, I. Litvinchev, R. R. Aguilar. (2020). Nature-inspired meta-heuristics approaches for charging plug-in hybrid electric vehicle. *Wireless Networks*, 26(7), 4753-4766.

APPENDIX

A. Stationarity and transformations

The ARIMA model needs to be fed only with stationary data to yield good results. Data are said to be stationary when the statistical properties do not change regardless of the time window. A statistical test that can determine whether a data set is stationary or not is the augmented Dickey-Fuller test (for short: ADF). Seasonal time series are not stationary, their properties changing in function on the time they are observed at. Transformations can be applied to them to make them stationary. One such transformation is differencing [17], where the difference $y'_t = y_t - y_{t-1}$ from an observation and the previous one is computed, absorbing a linear trend. Higher-order differences can be computed if needed. In a similar way, seasonal differencing $y'_t = y_t - y_{t-m}$, applies to seasonal data with a seasonal period of m . As noted in the description of SARIMA, the order of seasonal differences will be denoted by the D parameter for the model. Other examples of data transformations that make the series stationary include calendar transformations, logarithm transformations, power transformations, and the Box-Cox transformations that encompasses the preceding two [18].

B. Autocorrelation and partial autocorrelation

The autocorrelation function (ACF) is the correlation between the currently observed sequence with a previously observed sequence of the time series (see [18]):

$$r_k = \frac{\sum_{t=k+1}^n (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^n (y_t - \bar{y})^2}$$

where r_k is the autocorrelation at lag k . The plot of r_k versus k is known as a correlogram, its examination being one of the ways for identifying stationary or non-stationary time series. The autocorrelation will drop to zero quite quickly for a stationary time series, while on the other hand the ACF of a non-stationary time series will decrease very slowly [18]. The partial

autocorrelation function (PACF) [17] gives the amount of correlation of a stationary time series with its own lags, without the influence of the intermediate lags. For example, it eliminates the possibility of y_t and y_{t-2} being correlated only due to their association with y_{t-1} .

C. Choosing the AR(p) and MA(q) terms

The ACF and the PACF will guide the choice of the p and q parameters of ARIMA (and, of course, their seasonal counterparts P and Q). The auto-regressive part of the ARIMA equation (2) uses lagged values up to p , so p should be chosen so that the PACF has significant lags up to and including lag p , but none beyond it. At the same time, positive values for the first few ACF values is also an indication of the need to include AR terms. For choosing the MA(q) part of the model, one has to apply a similar reasoning, but swapped for the two plots [18]. The autocorrelation function should have significant lags only up to lag q and a negative ACF is also an indication of MA terms.

D. Model diagnosis

Once the parameters for the model are chosen and the model is fitted on the data, one has to examine the residuals $y_t - \hat{y}_t$, with \hat{y}_t the fitted values from the model (the forecasts of the model for the training data). Residuals should be zero-meaned, homoscedastic, not autocorrelated and ideally normally distributed [18]. The Ljung-Box test has been specifically developed for use with residuals generated by ARIMA models:

$$Q^* = n(n + 2) \sum_{k=1}^h \frac{r_k^2}{n-k}$$

where n is the same size (total number of residuals), r_k is the sample autocorrelation at lag k , whereas h is the number of residual lags being tested. From Ljung-Box's formula we conclude that a higher Q^* will mean a more correlated set of residuals, hence we expect that the ARIMA model might be improved. This test statistic becomes helpful when comparing two ARIMA models by their residuals, a lower Q^* will mean less correlated residuals, hence a better model that extracts more information from the data.

E. Dynamic harmonic regression

Dynamic harmonic regression might be employed to model longer seasonality periods. This means that the seasonal pattern is modeled using Fourier series and the residuals are handled using ARIMA. The dynamic harmonic regression is included in the ARIMAX model, ARIMA with exogenous regressors. To determine how many Fourier terms to include, we have computationally varied the number of Fourier terms and selected the best performing model according to the Mean Absolute Error (MAE).

F. Forecasting with dummies

One-time events, for example, public holidays, concerts or sport events that may last several days and influence electrical energy consumption can be modeled with indicator variables conventionally denoted as dummies. In the original ARIMAX equation, we have used only one external variable: x_t . If one needs to model the seven days of the week, then one needs to use six dummy variables, that is: $x_{1...6,t}$. The general rule is that $n-1$ dummy variables are needed if there are n separate events to be modeled [18]. Using these variables,

the ARIMAX equation would become:

$$y'_t = c + \sum_k^6 \beta_k x_{k,t} + \sum_h^p \varphi_h y'_{t-h} + \sum_j^q \theta_j \varepsilon_{t-j} + \varepsilon_t$$

giving weights (in the form of $\beta_{1...6}$) to each weekday, whenever it is the case.