Comparing Ensemble-based Forecasting Methods for Smart-Metering Data*

Oliver Flasch, Martina Friese, Katya Vladislavleva, Thomas Bartz-Beielstein, Olaf Mersmann, Boris Naujoks, Jörg Stork, Martin Zaefferer

Fakultät für Informatik und Ingenieurwissenschaften, FH Köln
Steinmüllerallee 1, 51643 Gummersbach
Tel.: +49 2261 8196-0
Fax: +49 2261 8196-15
Email: {First Name}.{Last Name}@fh-koeln.de,
Katya Vladislavleva: katya@evolved-analytics.com

Abstract. This work provides a preliminary study on applying state-of-the-art time-series forecasting methods to electrical energy consumption data recorded by smart metering equipment. We compare a custom-build commercial baseline method to modern ensemble-based methods from statistical time-series analysis and to a modern commercial GP system. Our preliminary results indicate that that modern ensemble-based methods, as well as GP, are an attractive alternative to custom-built approaches for electrical energy consumption forecasting.

1 Introduction

Smart metering equipment records electrical energy consumption data in regular intervals multiple times per hour, streaming this data to a central system, usually located at a local public utility company. Here, consumption data can be correlated and analyzed to detect anomalies such as unusual high consumption and to provide energy consumption forecasts.

This paper describes results from an ongoing project with GreenPocket GmbH (http://greenpocket.de), a software provider for smart metering infrastructure. GreenPocket's software aims at giving consumers insight into their consumption habits and provide them with accurate forecasts of their future energy consumption. Among GreenPocket's customers are electric utility companies who collect and analyze smart metering data of thousands of customers. This means that the energy consumption forecasting methods employed have to be scalable and efficient. Since the forecast models applied by GreenPocket are relatively simple, it is of great interest to compare them with more sophisticated modeling approaches, including symbolic regression (SR) and ensemble-based model selection. From our experience, both of these approaches provide very good accuracy and scalability. We present a preliminary experimental study based on real-world data to analyze the applicability of these approaches in a real-world setting. At this stage we focus on comparing modern existing approaches.

Research goals of this study will be presented in Section 2, while Section 3 describes data and experiments. The different prediction methods are introduced in Section 4 and Section 5 discusses the results. The paper concludes with a short outlook in Section 6.

* The original publication is available at www.springerlink.com.

2 Research Goals

The first goal of our study is to analyze benefits of ensemble-based approaches, i.e., approaches that evaluate several time-series models in parallel and allow an adaptation or even change of the current model based on new features and trends in the data. In theory, model adaptation improves the prediction accuracy. The second goal of our study is to analyze the performance of symbolic regression, a generic modeling approach, on the same real-world problem of electrical energy consumption forecasting. This application is based on genetic programming (GP) and not limited to time-series forecasting.
This study will investigate the following research questions:

- **Q1** Can modern ensemble-based approaches, without further domain knowledge about the data used, compete with custom-built approaches in a real-world energy consumption time-series prediction scenario?
- **Q2** Is symbolic regression, as a generic method, able to create time-series prediction models as accurate as custom-built approaches in the same scenario?

3 Data and Experiments

All models are trained on the same data set and are then used to predict the same time interval. The experiments are run on energy consumption time-series data supplied by GreenPocket. The data was recorded by two independent smart metering devices, installed at a commercial customer. Some data points are missing due to measurement or transmission issues, which is a common situation in real-world settings.

3.1 Training- and Test-Datasets

The data provided by GreenPocket are series of timestamp and meter reading pairs taken quarter hourly, a standard frequency for modern smart metering equipment. Timestamps are given an ISO 8601 derived date/time format, meter readings are given in kilowatt hours (KWH). As the energy consumption time-series data was recorded by two independent smart metering devices (`meter1` and `meter2`), each split into two time intervals (`series1` and `series2`) this results in four data sets.
We focus on the first time-series data set, i.e., `meter1_series1`, which was split into a training and a test data set for our experiments. The training interval of this data set starts at `2010-12-06 23:15:00` and ends at `2011-03-06 00:00:00`, the test interval starts at `2011-03-06 00:15:00` and ends at `2011-04-04 21:45:00`. This amounts to a training data length of approximately 12 weeks and a test data length of 4 weeks. Figure 1 shows plots of this training data set. Note that there are missing data points in the training data set. Visual inspection reveals daily periods, while weekly periods are detectable, but not as clearly defined. To give a first assessment of the generalizability of our ensemble methods, we also conducted first experiments with `meter1_series2` data.

3.2 Prediction Quality Rating

For these experiments each approach was fitted on the training data introduced above and had to deliver a quarter-hourly prediction time series for the 4 weeks of test data given above. To evaluate the quality of a predicted electrical energy consumption time series we consider three different error measures.

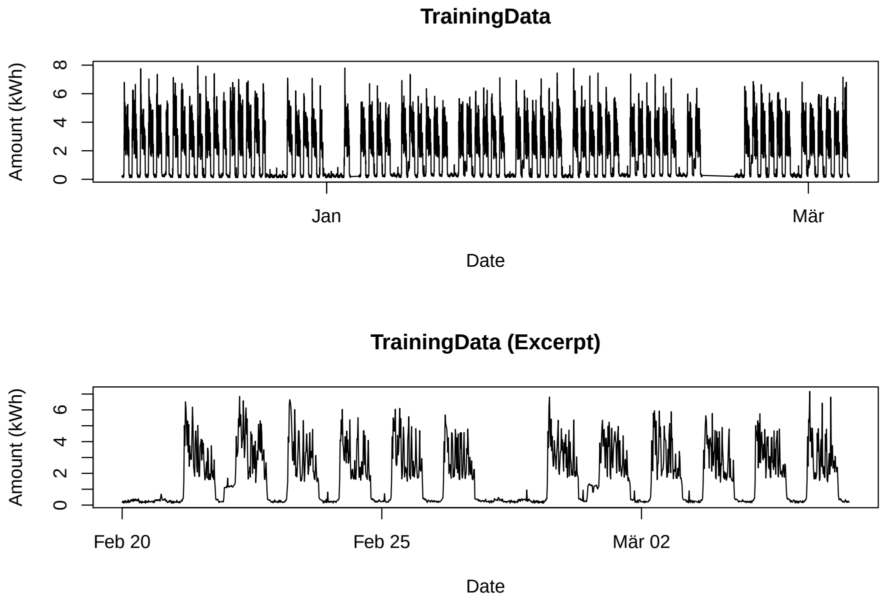

Fig. 1. Plots of the training time series `meter1_series1`. Energy consumption meter readings (measured in KWH) were recorded every fifteen minutes. The upper plot shows the complete data range, the lower plot the last two weeks.

MAE The *mean absolute error* (MAE) between the predicted time series $\hat{t}$ and the respective true energy consumption (test) time series t. Equation 1 defines the RMSE.

$$\text{MAE}(\hat{t}, t) := \frac{\sum_{i=1}^{n} |\hat{t}_i - t_i|}{n} \tag{1}$$

RMSE The *root mean square error* (RMSE) between the predicted time series $\hat{t}$ and the respective true energy consumption (test) time series t. Equation 2 defines the RMSE.

$$\text{RMSE}(\hat{t}, t) := \sqrt{\frac{\sum_{i=1}^{n} (\hat{t}_i - t_i)^2}{n}} \tag{2}$$

RMSElog The *root mean square error log* (RMSElog) is the RMSE between the logarithm of the predicted time series $\hat{t}$ and the logarithm of the respective true energy consumption (test) time series t:

$$\text{RMSElog}(\hat{t}, t) := \sqrt{\frac{\sum_{i=1}^{n} (\log(1 + \hat{t}_i) - \log(1 + t_i))^2}{n}}. \tag{3}$$

We apply a logarithmic transformation because energy consumption is always positive or zero and its distribution is highly skewed. The RMSE on the other hand is a symmetric loss function and therefore is best applicable if the error distribution is symmetric. By applying a log-transformation we hope to obtain a less skewed distribution. We also postulate that this loss is closer to what is important in a practical application.

4 Methods

The ensemble-based methods that we focus on in this work are implemented in the R forecast package and include Exponential smoothing state space (ETS) models and automated Autoregressive Integrated Moving Average (ARIMA) modeling [1]. Using ensembles the software chooses a single model from the large model class, which is used for prediction. In addition, we study the performance of symbolic regression via GP as a general method not limited to univariate time-series forecasting. For this we use Data Modeler, a commercial GP package based on Mathematica, as our implementation of symbolic regression [2]. All methods where applied with their default parameter settings.

4.1 Baseline (GreenPocket)

The baseline method provided by GreenPocket and applied in their production systems is an example of an extremely simple yet computationally highly efficient time-series prediction method. The prediction is the average energy consumption of the last 14 days at the same time of day as the prediction. This model is even simpler than a moving average as it has a fixed time horizon (14 days) after which an observation has no influence on a prediction.

4.2 Ensemble-based Methods

Classical time-series forecasting methods, including exponential smoothing state space models or ARIMA models, are widely and successfully applied to practical time-series forecasting problems much like the one discussed in this work. Both ETS and ARIMA models can capture a wide variety of different data generating processes. Consequently it is the burden of the user to choose a set of parameters for the model such that it adequately fits the data. Because selecting an appropriate model structure for a given forecasting problem is essential for obtaining good results, this selection process is often considered difficult by users not trained in statistical time-series analysis. Furthermore, manual model structure selection is a time-consuming and error prone task even for trained users.

To alleviate these difficulties and to enable automatic forecasting for time-series data of unknown structure, ensemble-based methods have been developed that automate the model selection process. In this work, we study the accuracy of two state-of-the art methods for automatic time-series forecasting: Automated ARIMA models and automated exponential smoothing state space models.

Both methods are limited to time-series with small to medium-sized seasonal frequencies. The automated ARIMA implementation we use in this study is able to support seasonal period lengths of up to $m = 350$ time units, while in practice, memory constraints of our implementation will limit this value to about $m = 200$. Yet in theory, the number of parameters to be estimated during ARIMA model fitting does not depend on m, so any m should be possible. Similarly, the automated ETS implementation we use restricts seasonality to a maximum period of 24 time units. This limitation stems from the fact that there are $m - 1$

parameters to be estimated for the initial seasonal states. As model parameters have to be estimated for many models structures, the automated ETS algorithm becomes computationally infeasible for large m.

As mentioned in Section 3, our quarter-hourly training data (96 measurements per day) indicates daily as well as weekly periods, warranting a period length of $m = 672$ to capture the weekly periodicity in the data. Unfortunately, this puts this problem clearly out of reach of our current implementations of both automated ARIMA and automated ETS. As a simple workaround, we therefore applied the STL decomposition to seasonally adjust the data, only then applied automated ETS or automated ARIMA to forecast the adjusted data, and finally added the seasonal component back into the forecasts. STL is a procedure that decomposes a seasonal time-series into trend, seasonal, and remainder components by a sequence of applications of the LOESS smoother [3]. We used the STL implementation from the R stats package [4].

Automated ARIMA *Models* By using STL to seasonally adjust the input data, we are able to apply a non-seasonal ARIMA(p, d, q) process of the form

$$\phi(B)(1 - B)^d y_t = c + \theta(B)\epsilon_t. \tag{4}$$

Here, $\{\epsilon_t\}$ is a white noise process with mean zero and variance σ^2. B is the backshift operator, and $\phi(z)$ and $\theta(z)$ are polynomials of order p and q. For $c \neq 0$, the implied polynomial in the forecast function has order d. Automated ARIMA forecasting then consists of selecting appropriate values p, q and d, i.e. an appropriate model order. We do this using Akaike's Information Criterion (AIC).

$$\text{AIC} := -2\log(L) + 2(p + q + k), \tag{5}$$

where $k := 1$ if $c \neq 0$ and 0 otherwise. L is the likelihood of the model when fit to the differenced data $(1 - B)^d y_t$. As the likelihood of the full model for y_t is not actually defined and therefore AIC values for different levels of differencing are not comparable, d is chosen via unit-root tests based on a null hypothesis of no unit-root. ARIMA(p, d, q) models where d is selected based on successive KPSS unit-root tests are considered as models. The procedure successively tests higher order differences of the data for a unit root until a non-significant p value is observed.

As there are too many parameter combinations of p, q, and d for an exhaustive search for the model with globally best AIC, a step-wise algorithm is applied. First, the four models ARIMA$(2, d, 2)$, ARIMA$(0, d, 0)$, ARIMA$(1, d, 0)$, and ARIMA$(0, d, 1)$ are fitted with $c \neq 0$ if $d \leq 1$ or with $c = 0$ otherwise. The model with the best AIC is designated as the *current* model. Second, variations of the current model are considered by varying the model parameters by ±1 in an iterative process, respecting several constraints on the fitted models. When a model of better AIC is discovered, it becomes the new current model, until no variation of the current model with lower AIC is found. The then current model is used to produce forecasts. [1]

Automated ETS *Models* As shown in [1], exponential smoothing methods are equivalent to optimal forecasts from innovations state space models. We thus consider the class of all innovations state space models as the pool for model selection in automated ETS modelling. To distinguish model structures, the notation ETS(error, trend, seasonality) is employed, where the error component can be either additive or multiplicative, the trend component can be either missing, additive, additive damped, multiplicative, or multiplicative damped, and the seasonality component can be either missing, additive, or multiplicative. Considering all combinations, there are 30 model structures. In our case, as our input data is not strictly positive and already seasonally adjusted, multiplicative error models are not applicable, and the seasonality component may be disabled (missing), reducing the pool to only 5 model structures.

All 30 (in our case only 5) innovations state space model structures share a general layout consisting of a state vector $\boldsymbol{x}_t := (l_t, b_t, s_t, s_{t-1}, \ldots, s_{t-m+1})'$ and the state space equations

$$y_t = w(\boldsymbol{x}_{t-1}) + r(\boldsymbol{x}_{t-1})\epsilon_t \tag{6}$$

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}) + g(\boldsymbol{x}_{t-1})\epsilon_t. \tag{7}$$

In these equations, $\{\epsilon_t\}$ is a Gaussian white noise process with mean zero and variance σ^2, and $\mu_t = w(x_{t-1})$. In the model with additive errors, $r(x_{t-1}) = 1$ holds, so that $y_t = \mu_t + \epsilon_t$. To calculate point forecasts until horizon h, these equations can be iteratively applied for $t = n+1, n+2, \ldots, n+h$, while setting $\epsilon_{n+j} = 0$ for $j = 1, \ldots, h$.

Parameters for these innovations state space models are obtained by maximum likelihood estimation. The model structure most appropriate for the input data at hand can then be selected based on AIC, leading to the automated ETS forecasting algorithm of [1]:

1. Apply all model structures to forecast the input time series, choosing model parameters by maximum likelihood estimation.
2. Select the model with the best AIC.
3. Use this model to produce h point forecasts.

4.3 Data Modeler

In addition to GreenPocket's approach and our ensemble approach a state of the art modeling approach based on Genetic Programming (GP), namely Evolved Analytics' Data Modeler, will be included in our study as the third modeling tool. [5]

We would argue that the challenge of predicting one month of energy consumption based on three months worth of data is not a conventional problem for symbolic regression (SR) modeling with GP. Symbolic regression is a methodology for finding global relationships in data and for global approximation. SR via GP uses stochastic iterative search process (evolution) to evolve appropriate model forms using supplied set of input variables, functional primitives, and constants. Models are developed to achieve optimal trade-offs between prediction accuracy on provided input-response data and model complexity and numerical stability. SR is particularly useful for finding laconic expressions for smooth albeit high-dimensional response surfaces.

Main goal of applying ensemble-based symbolic regression to the non-smooth data was to see whether this flexible methodology is capable to appropriately identifying the time lags and combine them together into acceptable dynamic models. We used ensemble-based symbolic regression implemented in Data Modeler [2], because it is best suited for modeling very high dimensional data with a only small fraction of input variables significantly related to the response. Due to space limitations we only report the results of GP experiments which used variations of lagged consumption as input variables. While predicting energy consumption one day ahead did not pose real challenges for GP, achieving success for predictions one month ahead has hit a wall of decreased prediction accuracy on both training and test data. To predict four weeks ahead we were forced to set the considered time-lags considered to 28 days (2688 quarter intervals) and earlier.

In described experiments we used the quarter-hourly time lags between 28 and 35 days from the moment of prediction (672 time-lags, i.e. input variables), and we also constructed a reduced set of inputs of all quarter-hour intervals between 28 and 29 days ago, lags between exactly 28 and 35 days ago, and lags between 28 days and one quarter-hours and 35 days and one quarter-hour ago – 110 variables in total. The variables used in the experiment are:

d_{2688}–$d_{2784}, d_{2880}, d_{2976}, d_{3072}, d_{3168}, d_{3264}, d_{3360}, d_{2785}, d_{2882}, d_{2979}, d_{3076}$, d_{3173}, d_{3270}, d_{3367}, where $d_i(t) = c(t - i)$, for $c(t)$ being a quarter-hourly energy consumption at time moment t. Because of the large backshift we could only use 5093 records from the given training data (63%). All GP runs used a arithmetic operators and *square*, *minus*, *inverse* as functional primitives, random constants sampled from the interval $[-5, 5]$, and time-limited evolutions of 20 minutes each (for other default GP settings see [2]). In the first evolutions a consistent set of driving variables was discovered (see Figure 2). The top ten driving variables were further used for new runs in a reduced space with all other settings the same.

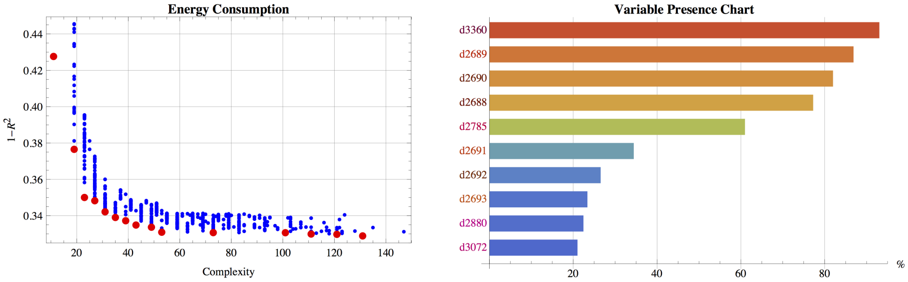

Fig. 2. GP results of the first experiments: Selected Model Set and Driving Variables. Left plot presents the selected set of models plotted in the space of model complexity (sum of nodes of all subtrees corresponding to the parse-tree defining the model) and model accuracy (1-R^2). Variable presence in the right plot stands for the fraction of models in the selected set containing variable in question.

The runs in the second batch generated models of higher prediction accuracy and smaller complexity. From these models we selected an ensemble of models using the SelectModelEnsemble. DataModeler defines final model prediction at a certain point as an average of five predictions closest to the median of predictions of ensemble models at this point. The model closest to the median prediction on test data was the model from the knee of the Pareto Front of the ensemble models:

$$t(i) = 8.94 - \frac{253.62}{28.15 + t(i - 2688) + t(i - 2689) + t(i - 2690) + t(i - 2785) + t(i - 3360)}.$$

Even without constant optimization the model alone produces the prediction error on the test set of RMSE $= 1.03$. This accuracy is comparable with an accuracy of the RMSE $= 0.97$ of the golden batch week prediction constructed by averaging available consumption per day of the week (from Monday to Sunday) and predicting test data only using day of the week and a quarter-hour time moments.

Our results indicate that ensemble-based symbolic regression while used for constructing global approximation of high-dimensional data is capable of identifying appropriate time lags and creating small and very interpretable dynamic models in such a challenging problem like energy consumption.

5 Results and Discussion

The predicted energy consumption for all four models is shown in Figure 3. From that figure it is easy to see that the GreenPocket prediction for each day is the same which is explained by the fact that their model has no covariates and only carries the average daily consumption profile of the last 14 days of the training data forward. Consequently it fails to predict the days of low consumption (Sundays) in the test data. Both the ETS as well as the ARIMA prediction are dominated by the seasonal effect which is estimated by the STL procedure. Therefore their predictions differ only slightly. If we check the chosen ARIMA model, we see that it has, as expected, no lag and the ETS model is a simple additive model without any periodicity. The prediction from Data Modeler, the GP system in our comparison, appears quite different. It has a lower intra-day variance and mispredicts the consumption on “off” days by a small but noticeable offset.

It is unclear which of the three models is the “best”. In practice there are likely no measurable differences between the ARIMA and ETS model so one would choose the model with the lower computational burden. This is confirmed by the three error measures depicted in Table 1. The results of the ensemble-based approaches pass a preliminary test of generalizability, as the result ranking of `meter1_series2` matches the result ranking of `meter1_series1`.

	Series 1			Series 2		
Method	RMSE	MAE	RMSElog	RMSE	MAE	RMSElog
Automated ARIMA	1.157	0.710	0.352	**0.962**	**0.596**	**0.235**
Automated ETS	1.157	0.710	0.352	0.967	0.605	0.239
Data Modeler	**1.030**	**0.699**	**0.328**			
GreenPocket	1.151	0.741	0.382	1.196	0.763	0.360

Table 1. Experiment results for time series 1 and 2. Error measure values were truncated to 3 decimal digits. Best error values are shown in bold font. Data Modeler results for time series 2 where not available at the time of publication.

According to those error measures, the predictions by Data Modeler are always slightly better, regardless of the error measure used. This is not all that surprising given that Data Modeler can choose its model from a much broader and richer class of models.

Overall it is entirely not clear which error measure is the most appropriate for this problem. All of the models discussed still have weaknesses which are not necessarily captured by the error measures used. In the future

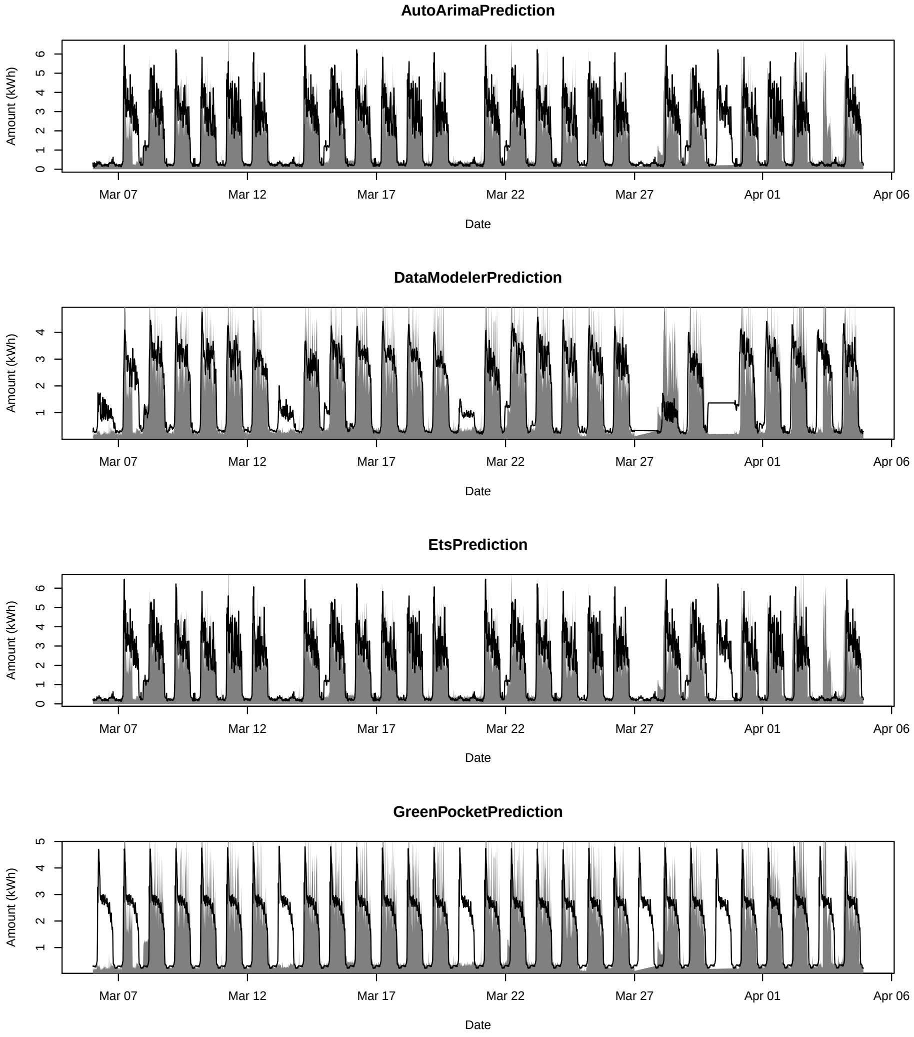

Fig. 3. Time series plots of the test data range predictions generated by the methods studied in this paper. From top to bottom: Ensemble-based ARIMA (AutoArimaPrediction), symbolic regression (DataModelerPrediction), ensemble-based exponential smoothing (EtsPrediction), GreenPocket baseline (GreenPocketPrediction). The true energy consumption time series is shown as a gray backdrop in each plot.

we want to investigate look into loss functions which model the actual business case—Euros saved or spent based on the (mis)prediction of the model. This would require yet another time series, the energy prices per quarter hour, and its prediction to evaluate a joint model for a purchasing strategy.

6 Conclusions and Outlook

The two ensemble-based time-series prediction methods are easily applicable and are able to provide precise forecasts. Regarding research question **Q1** posed in Section 2 of this work, our experiments show that modern ensemble-based approaches, without further domain knowledge about the data used, are able to compete with custom-built approaches in our real-world energy consumption time-series prediction scenario.

GP and symbolic regression in particular is a promising model generation strategy, because it can find structure as well as driving variables at the same time. Given such a model, the constants can then be adapted to fit different data from the same domain. It should also be applicable to other consumption problems. Regarding research question **Q2**, our results show that symbolic regression, as a generic method, is able to create time-series prediction models that are as accurate as custom-built approaches, at least in our application scenario. All models studied in this work are efficiently executable, making them applicable to the large data volumes common in smart metering.

In future work we will investigate whether the models constructed for one particular customer can be used to predict energy consumption of other customers by merely re-fitting model parameters.

References

1. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting. the forecast package for r. Journal of Statistical Software **27**(3) (7 2008) 1–22
2. Evolved Analytics LLC: DataModeler Release 8.0 Documentation. Evolved Analytics LLC (2011)
3. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics **6**(1) (1990) 3–33
4. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2012) ISBN 3-900051-07-0.
5. Kordon, A.K., Smits, G., Kotanchek, M.E.: Industrial evolutionary computing. In Thierens, D., ed.: GECCO (Companion), ACM (2007) 3297–3322