Automated energy consumption forecasting with EnForce

Mary Karatzoglidi
National Tech. Univ. of Athens
marykaratzoglidi@gmail.com

Paraskevas Kerasiotis
National Tech. Univ. of Athens
pkerasiotis@mail.ntua.gr

Verena Kantere
National Tech. Univ. of Athens
vkante@dblab.ece.ntua.gr

ABSTRACT

The need to reduce energy consumption on a global scale has been of high importance during the last years. Research has created methods to make highly accurate forecasts on the energy consumption of buildings and there have been efforts towards the provision of automated forecasting for time series prediction problems. EnForce is a novel system that provides fully automatic forecasting on time series data, referring to the energy consumption of buildings. It uses statistical techniques and deep learning methods to make predictions on univariate or multivariate time series data, so that exogenous factors, such as outside temperature, are taken into account. Moreover, the proposed system provides automatic data preprocessing and, therefore, handles noisy data, with missing values and outliers. EnForce includes full API support and can be used both by experts and non-experts. The proposed demonstration showcases the advantages and technical features of EnForce.

PVLDB Reference Format:
Mary Karatzoglidi, Paraskevas Kerasiotis, and Verena Kantere. Automated energy consumption forecasting with EnForce. PVLDB, 14(12): 2771 - 2774, 2021.
doi:10.14778/3476311.3476341

1 INTRODUCTION

Energy efficiency and management in commercial buildings constitute a determining factor for the global need to save energy. Forecasting electrical energy consumption, the major form of energy consumed in such buildings, becomes a key component in the process of energy management. The use of energy is a substantial part of the total operating cost of buildings, but is also an important factor of comfort of the people who work or visit this building [7]. Building models and tools for analyzing related data is of high importance for drawing conclusions regarding both its operation and the energy efficiency of the building.

Both statistical techniques and deep learning methods are used for energy consumption time series prediction problems. However, their suitability varies significantly and depends on various factors, such as the data size. Moreover, energy consumption forecasting constitutes a complex problem, as the consumption of commercial buildings is affected by heterogeneous factors, such as the weather temperature, the number of people inside the building, holidays etc, mixed up with seasonal variations. As a result, the energy consumption patterns for individual buildings can be extremely irregular [2]. Making predictions on energy consumption represents a difficult type of predictive modelling problem due to the existence of complex linear and non-linear patterns [3]. Overcoming these obstacles is crucial for achieving accurate and reliable forecasts.

Determining which forecasting method to use is not easy.Thus, evaluating and comparing the accuracy of various models is important, in order to test their suitability on a given dataset. Higher accuracy can be achieved if factors which affect energy consumption are considered. Therefore, forecasting methods which handle multivariate time series should be used and compared with models trained on univariate time series, which might still achieve higher accuracy for specific cases. Moreover, using raw data for prediction can affect negatively forecast accuracy. Hence, data pre-processing is necessary, as it ensures the quality of the dataset, so that forecasting is not affected by non-causal effects. Forecasting on time series becomes even more difficult, when analyzing small datasets, which are insufficient to produce an accurate prediction model. There is a growing demand for time series forecasting from organizations of all sizes; developing systems for automated forecasting is valuable for users with little or no technical experience. Making forecasts on energy consumption of buildings is a complex problem, and developing models to make predictions requires both advanced technical knowledge and experience on time series forecasting.

Several commercial forecasting packages and open source packages have been developed in order to achieve automated time series forecasting. Hyndman and Khandakar [1] describe two automatic forecasting algorithms, only for univariate time series, implemented in the forecast package for R and, therefore, programming experience is required. These methods do not analyze multivariate time series and cannot take into account factors which affect the time series values. In addition, both algorithms, i.e. ARIMA and Exponential smoothing, are statistical, and deep learning models are not included. Facebook's Core Data Science team developed Prophet [4], a procedure implemented in Python and R, for forecasting univariate time series data based on General Additive Models. Prophet fits one single model and handles missing data and outliers. However, it cannot handle correlations between different time series and therefore, Prophet cannot take into account exogenous variables, which affect the energy consumption of a building. The user needs to have programming experience, in order to use Facebook's Prophet. DataRobot [10] offers ways to extract understanding and predict future outcomes of time series. This tool uses methods like ARIMA, Facebook Prophet and other advanced time series models to achieve forecasts with high accuracy and includes full API support. However, Datarobot's solution is complex to use and does not perform well with small datasets. Anodot's Autonomous Forecast [9] uses deep learning to automatically optimize forecasts. Anodot's system can be used without requiring a data scientist to create, train, tune and deploy forecasting models. Similarly to Datarobot's solution, Autonomous Forecast does not perform equally well with small

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476341

datasets. Moreover, it works exclusively with deep learning models and does not use statistical methods.

In summary, most of the existing tools work with a single family of models and do not combine statistical and deep learning models. In addition, they make predictions are made on univariate time series data and do not take exogenous factors into consideration. Moreover, most of them do not include automatic data preprocessing and they cannot be used by non-experts, with little or no programming experience. Finally, they do not work well with small datasets, and, thus, high accuracy is not guaranteed.

We demonstrate EnForce, a novel system for producing forecasts on the energy consumption of buildings, based on historical time series data. In more detail, our system:
• performs automatic data preprocessing, i.e. data are denoised, outliers are detected and removed, missing values can be filled in
• performs time series analysis and forecasting, using both state-of-art statistical and machine learning models
• automatically chooses the most accurate forecasting method for the analyzed data, according to size, type, linearity, number of features of the input and the accuracy metrics of the methods
• takes into account exogenous variables, like temperature data, while predicting future values of energy consumption
• enables the creation of pre-trained models based on forecasts of large datasets, and employs them to achieve very high accuracy for forecasting on similar small datasets
• can be used from both experts and non-experts
• includes full API support

EnForce can be used as a black box, which receives input from the user, i.e. the data, and produces the forecasts of the method with the highest accuracy. First, time series data are preprocessed, missing values are handled, outliers are detected and removed and the data are denoised. Forecasting models are then developed, using both statistical and deep learning methods, such as ARIMA, LSTM, CNN-LSTM, XGBoost and procedures like Facebook Prophet [4], to predict future values of energy consumption, based on historical data. The best-suited methods are used, according to the size, type, linearity and the number of features of the input. These methods are evaluated according to the accuracy of the forecasts. The results of the best, most accurate, method are presented to the user.

2 SYSTEM ARCHITECTURE

Figure 1 depicts the architecture of EnForce and the interaction between the different parts of the system.

2.1 User Interface

Input The input is one or more files, CSV, EXCEL or TxT preformatted, which contain time series data referring to the energy consumption of a building, with the timestamp and the value. Optionally, additional time series of factors that affect the energy consumption, such as outside temperature, can be input. EnForce automatically creates multivariate time series from multiple univariate time series data that are input, based on timestamps, as shown in Figure 4a. These files are provided by the user, with optional metadata information about the type of each of the time series (e.g. retail, hospital, super-market etc.). Also, the user can choose whether or not the systems should fill in the missing values of the dataset, and adjust the forecasting frequency. Nevertheless, there are default choices.

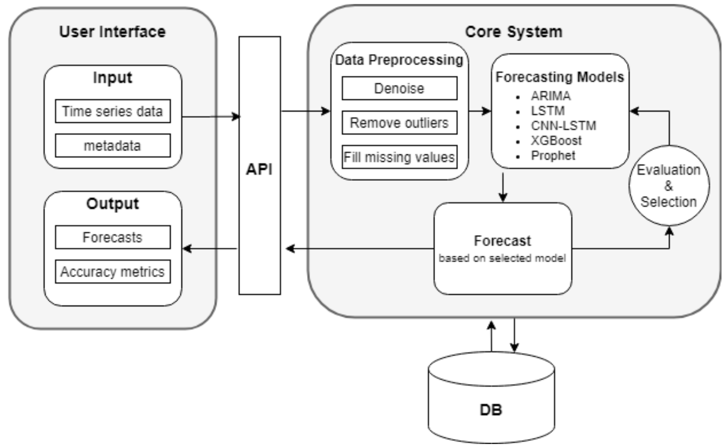

Figure 1: System Architecture Overview

Also, if the forecasting frequency is not determined by the user, it is considered equal to the frequency of the input time series data. In addition, if the percentage of the missing values is higher than 25%, they will be filled in automatically, by default. An expert user can also determine the size of the window, concerning the creation of a windowed dataset during data preprocessing. Non-expert users let the system determine automatically the optional size of the window. Moreover, the user optionally defines the number of the predictions to be made. By default, EnForce produces predictions for a number of dates equal to the 5% of the size of the input dataset. Users can give their email address, to get a notification when the forecasting process is completed. The input data are uploaded to the system and no further configuration is needed to extract the final results.
Output The output includes an interactive image (see Section 3) with the forecast values of energy consumption, along with the accuracy metrics of the selected method, as shown in Figure 4b.

2.2 Core System

Automatic Data Preprocessing Data preprocessing ensures the quality of the time series in a way that forecasting is not affected by non-causal effects. Most forecasting methods do not work well in the presence of outliers, which may suggest experimental errors, variability in a measurement or anomaly. Hence, they are detected, using the Interquartile Range IQR [5] and removed. A digital filter is used to denoise the data and, optionally, a decision tree is used to fill in missing values. Concerning machine learning methods, a windowed dataset is derived from the original data, to achieve better performance of the models. If the window size is not given as an input, our system tests three values of the window size, which are equal to 5%, 10% and 15% of the size of the dataset, and picks the window size that minimises the prediction error on the test set.

Forecasting models The dataset is divided into a training (80% of the dataset) and a test set (the rest 20%). The training set is used to train the models and the test set is used to evaluate the accuracy of the model. We selected the forecasting methods to be included in Enforce based on the following requirements:
• process datasets of various sizes
• detect short-term and long-term dependencies of historical data
• extract seasonal and linear or non-linear patterns of time series.
EnForce implements the following methods. Figure 2 shows the

Requirements	Method
Small dataset	Pretrained models
Short-term dependencies	ARIMA
Long-term dependencies	LSTM, CNN-LSTM
Seasonality	Prophet
Linear patterns	ARIMA
Non-linear patterns	Machine learning methods

Figure 2: Suitability of methods for various requirements.

suitability of selected methods with respect to the requirements.
• **ARIMA** (autoregressive integrated moving average): a widely used statistical model and a popular traditional approach used for time series prediction problems.
• **Bidirectional LSTM**: Long Short-Term Memory (LSTM) is a specialized Recurrent Neural Network (RNN) able to learn long-term dependencies using a mechanism called gates. In a Bidirectional LSTM (BiSTM) information from both past and future is preserved.
• **CNN-LSTM**: the combination of a convolutional neural network (CNN) layer with the long short-term memory (LSTM) model is used to effectively predict the energy consumption of buildings. This method is able to extract complex features of energy consumption. The CNN layer is used to extract the features from the input data, which are then fed to the LSTM, which is used for modeling temporal information of irregular trends in time series components.
• **XGBoost Regressor**: a powerful and popular machine learning algorithm, which is used to model complex processes. XGBoost (Extreme Gradient Boosted trees) has high predictive power, is almost 10 times faster than the other gradient boosting techniques and can be used for both regression and classification.
• **Prophet**: a procedure for forecasting time series data based on an additive model, in which non-linear trends are fit with yearly, weekly, and daily seasonality. It works best with time series that have strong seasonal effects and several seasons of historical data[4].
Along with the models described above, our system enables the creation of pre-trained models (see Section 3).

Automatic Evaluation and Model Selection - Forecasting EnForce chooses the best-suited methods to the input, through the analysis of the dataset, as shown in Figure 3. If the input contains a small dataset (less than 500 rows) and the resource type is specified, both pre-trained models and methods described in Section 2 are examined. Statistical and deep learning methods are chosen according to the size, type, linearity and the number of features of the dataset. We use random forest, a classification algorithm, to decide based on previous datasets and their characteristics, which are the most suitable methods for the input. For example, the ARIMA model is chosen for datasets with linear patterns, with little or no missing values and datasets of a small size. In contrast, Prophet is chosen for big datasets, with missing values and strong seasonality, whereas LSTM is used for multivariate time etc. The selected methods are trained and their performance is evaluated. The mean absolute error (MAE), the root mean squared error (RMSE) and the coefficient of determination (R-squared) are calculated for each model, using the test set to evaluate their accuracy. Pre-trained models with similar characteristics to the dataset (i.e. type) are queried and, if found, their performance is compared to that of the methods described above. The method that performs better is used to produce the

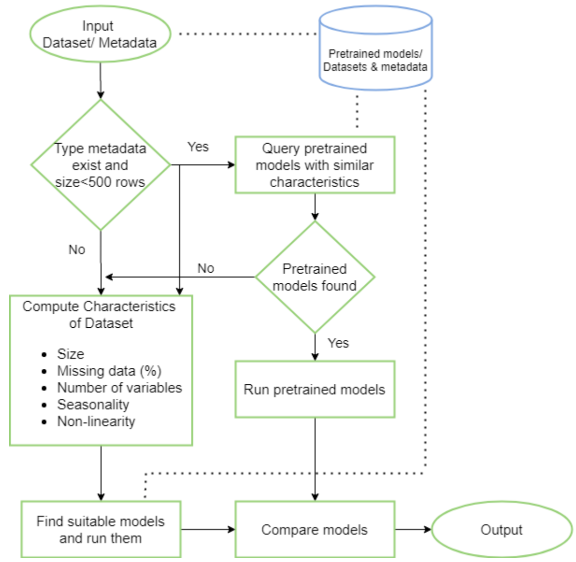

Figure 3: Automatic Evaluation and Model Selection.

final forecasts. If the input is big enough or the resource type is not specified, then the suitability of pre-trained models is not examined.

2.3 Database

EnForce handles large data. We use Apache Cassandra [8], a highly scalable, high-performance distributed database. This is where the input time series data and metadata are stored, until the forecasting process is completed and the final result are extracted. The pre-trained models are also stored in the database and are accessible to the Core System, in order to be used for the forecasting process.

3 SYSTEM FUNCTIONALITIES

EnForce improves predictive accuracy, while automatically performing time series forecasting, through a series of functionalities.
Forecasting on small datasets Except from the forecasting models described in Section 2, pre-trained models are also included. These are models trained on energy consumption time series and used to achieve high predictive accuracy for similar inputs, according to the resource type of the data. EnForce saves a model if it achieves accuracy more than 90%, for big datasets with no missing values. Most importantly, pre-trained models enable the system to handle the problem of making predictions on small datasets and performs equally well with datasets of bigger size.
Correlation of multiple time series The system input contains one or more time series data. Along with the energy consumption time series, the user can provide data that refer to factors that affect the consumption of a building. Forecasting the energy consumption, while also considering exogenous factors, helps achieve higher predictive accuracy. If the input contains more than one factors, EnForce finds the one with the strongest correlation to the energy consumption and produces forecasts based on this factor and the energy time series. However, methods like Prophet and ARIMA do not take into account multiple time series, in contrast to models like LSTM. The performance of all of these methods is evaluated and compared, as models which are trained exclusively on the energy data can still achieve higher accuracy, if there is not a strong

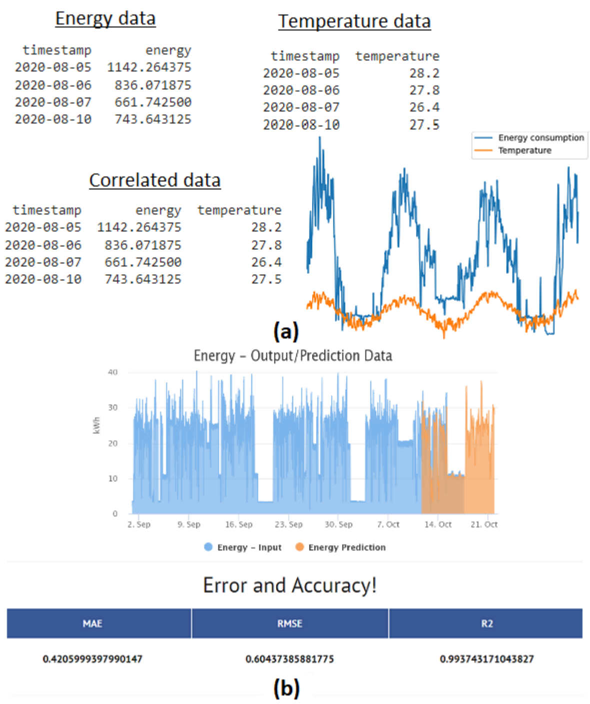

Figure 4: (a) Data correlation (b) User Interface - Input.

correlation between the different time series.

Energy-temperature correlation As shown in Figure 4a, the outside temperature has a high effect on the heating and cooling needs of a building, which play a significant role in the total building energy use. A key notion for the incorporation of weather conditions in the analysis of building energy consumption, is *Degree Days* [12]. Based on both energy and temperature time series, EnForce calculates the base temperature [11] of the building and the degree days (heating degree days - HDD, cooling degree days - CDD) [12]. A degree-day-based analysis and forecasting is performed, and the predicted values are based on the heating and cooling needs of the building, according to the outside temperature.

Data pre-processing While providing the input data, the user specifies whether or not data pre-processing should include the task of filling in missing values. Handling missing data can affect the accuracy of the forecasts. The simplest approach to deal with missing data is to ignore them. However, missing values are filled in using a Decision Tree, which is a commonly used algorithm for this task [6]. If the percentage of missing values of the dataset is higher than 25%, they will be filled in automatically.

Customizable forecasting frequency The input is hourly, daily or weekly time series data. EnForce automatically performs forecasting on the same (hourly, daily or weekly) level as the input, by default. Moreover, EnForce allows the user to specify the forecast frequency. It performs automatically data aggregation and trains the methods both on the original and the aggregated data. It then compares the performance for both datasets and informs the user on the accuracy of the forecasts, i.e. whether or not higher accuracy is achieved with the aggregated data, on the output of the system.

Visualization of forecasts EnForce outputs a general overview of the forecasts as an interactive image, with a fully functional zoom in & out component. The image is a chart, which contains the values of the input time series data, for the most recent period of time, along with the predicted values of the energy consumption. In addition, when viewers mouse over a chart element, they can see the value of the chart element and its timestamp, as shown in Figure 4b. The accuracy metrics of the method are also shown.

4 DEMONSTRATION

We show the system through three scenarios, which employ proprietary time series data from very large commercial buildings.

Non-expert user scenario: The purpose of this scenario is to demonstrate the use of pre-trained models to achieve high predictive performance on small datasets, with less than 500 rows. The audience will have the opportunity to load two big datasets of a different resource type, i.e. a supermarket and a company building, and create two pre-trained models. The goal is to show that according to the criteria described in Section 2, the most suitable of the models is used to make predictions on the smaller dataset.

Expert-user scenario: With this scenario, we demonstrate how an expert can use EnForce to produce forecasts on univariate time series for a predefined horizon. The input contains energy consumption values of a building, per 15 min, with a total of 5000 rows. The goal is to show how an expert user can affect the final forecasts, by differentiating the input metadata, such as adjusting the forecasting frequency, determining the window size and the length of the forecasts, as described in Section 2.1.

Energy-temperature correlation scenario: This scenario focuses on making forecasts on bivariate time series. The correlation between the two variables, energy consumption and outside temperature, is examined. EnForce makes a Degree Day analysis, as explained in Section 3. The goal is to highlight the difference between the accuracy of forecasting models, which take, or not, into consideration exogenous factors. We will show cases with models trained on bivariate series have better accuracy, but, also, cases for which models trained on univariate time series achieve higher accuracy, than models trained on bivariate time series, due to weak correlation between the data. For this purpose, data with daily energy and temperature data of a commercial building are used.

REFERENCES

[1] Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. Journal of Statistical Software
[2] K. Yan, W. Li, Z. Ji, M. Qi and Y. Du, "A Hybrid LSTM Neural Network for Energy Consumption Forecasting of Individual Households," in IEEE Access, vol. 7, pp. 157633-157642, 2019.
[3] C. Nichiforov, I. Stamatescu, I. Făgărăşan and G. Stamatescu, "Energy consumption forecasting using ARIMA and neural network models," in International Symposium on Electrical and Electronics Engineering (ISEEE), pp. 1-4, 2017.
[4] Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2
[5] Tukey, John W (1977). Exploratory Data Analysis. Addison-Wesley. ISBN 978-0-201-07616-5. OCLC 3058187.
[6] S. Gavankar and S. Sawarkar, "Decision Tree: Review of Techniques for Missing Values at Training, Testing and Compatibility," in International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), pp. 122-126, 2015.
[7] A. Naug and G. Biswas, "A Data Driven Method for Prediction of Energy Demand in Commercial Buildings,"in IEEE International Conference on Automation Science and Engineering (CASE), pp. 335-340, 2018.
[8] https://cassandra.apache.org/
[9] https://www.anodot.com/autonomous-forecast/
[10] https://www.datarobot.com/platform/automated-time-series/
[11] https://www.degreedays.net/base-temperature
[12] https://www.degreedays.net/calculation