Web-based Benchmarks for Forecasting Systems - The ECAST Platform

Robert Ulbricht
Robotron Datenbank-Software GmbH
Dresden, Germany
robert.ulbricht@robotron.de

Claudio Hartmann
Technische Universität Dresden
Dresden, Germany
claudio.hartmann@tu-dresden.de

Martin Hahmann
Technische Universität Dresden
Dresden, Germany
martin.hahmann@tu-dresden.de

Hilko Donker
Robotron Datenbank-Software GmbH
Dresden, Germany
hilko.donker@robotron.de

Wolfgang Lehner
Technische Universität Dresden
Dresden, Germany
wolfgang.lehner@tu-dresden.de

ABSTRACT

The role of precise forecasts in the energy domain has changed dramatically. New supply forecasting methods are developed to better address this challenge, but meaningful benchmarks are rare and time-intensive. We propose the ECAST online platform in order to solve that problem. The system's capability is demonstrated on a real-world use case by comparing the performance of different prediction tools.

CCS Concepts

•Information systems → Expert systems; •Applied computing → Forecasting;

Keywords

Time Series Forecasting; Benchmark; Transparency

1. INTRODUCTION

Forecasting time series is traditionally an important issue for any industry. It has always been the case that in corporate areas like production, distribution or pricing, many decisions have to be made based on uncertain data. This explains why countless methods and solutions have been published by the multitude of different communities active in that field of research. For example, considering only top-ranked energy journals[1], since 2005 more than 200 articles have been published related to the forecasting of renewable energy supply from fluctuating sources like solar and wind power. But choosing the optimal solution for a specific forecasting problem remains a formidable and work-intensive task, mainly caused by their poor qualitative comparability: First of all, evaluations are almost always conducted on dissimilar data sets, which means that they vary in aspects like the measurement quality of individual observation points, the amount of used time series and their completeness, individual characteristics like patterns, aggregation or normalization, or their geographical origin. Another source of bias emerges from the experimental setup, as different values are used for environmental parameters like forecasting horizon, the length of training histories or evaluation periods. Next, different statistical error metrics are applied to measure the accuracy of forecasts, which is a separate research topic as shown by e.g. Chen and Yang [1] or Hyndman and Koehler [3] to mention a few. Finally, conclusions can also suffer from the authors somehow 'personal' point of view. Experimental results may be optimized towards the desired findings by using only very few and specific test cases or compared against too simple baselines, so incredibly low errors or high improvement factors are achieved but are hardly reproducible in practice.

Besides literature, methods can be compared in forecasting competitions. Although they present a more practical alternative, they do have some drawbacks: Intensive efforts are required for preparing and participating alike, thus making them rather rare events. For the famous *M-Competitions* founded by Makridakis [6], the latest activities date back to the year 2005. A much more recent approach is the *Global Energy Forecasting Competition (GEFCom)*, started in 2012 and running every two years since then. The insights published by Hong et al. [2] show that such approaches have difficulties with the simulation of real-world situations where forecasts have to be provided on a rolling basis for intra-day or day-ahead periods. This results in a shift of the forecast origin with newly arriving observation data, thus leading to multiple time-intensive evaluation phases. Although repeatedly requested in the past, e.g. for the energy domain by Madsen et al. [5] or Kostylev and Pavlovski [4], there is still

[1] based on SCImago Journal & Country rank

SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-3531-7/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2882903.2899399

no reliable and internationally admitted reference model nor industry standard available for such benchmarks. For example something like the well-established TCP-H standard, the state-of-the-art benchmark for database performance evaluation [7], remains an unsolved research gap for the forecasting community. For the above mentioned reasons, it becomes obvious that interested parties like software providers or application users can hardly compare the quality of forecasting methods without implementing them, collecting interesting use cases, and conducting intensive evaluations. This, besides being a resource-intensive task, can only be successful if the descriptions provided for the proposed solutions turn out to be clear and detailed enough. The constantly high number of new articles and reviews proves the unbroken public interest in that field, at the same time it underlines the strong need for consolidated results. We believe that following a system integration approach can solve that problem, as this is a straightforward way to tackle one of the major challenges: To provide transparency by bringing together representative use cases, homogenous evaluation criteria, and competing tools or algorithms in one public place, anytime available for everybody.

In this paper, we demonstrate the ECAST platform to cope with the problem of qualitative evaluation of competing forecasting methods as discussed earlier in this introduction. The remainder of the paper is organized as follows: In Section 2 we describe the system's core features and components before we demonstrate the functionality in Section 3. Finally, we summarize our work and show directions for future developments in Section 4. For additional material and direct access to the platform, we refer to the REEF project web site[2].

2. SYSTEM DESCRIPTION

In this section we explain the general idea behind ECAST and the system's architecture. Originally, ECAST was designed as a tool for energy forecasting automation and later expanded to allow for benchmarking forecasting practices in any domain. As a consequence, it combines functionality for task automation, data handling and result representation as shown in Figure 1. In order to define a new experiment, users can choose among the available use cases and predictors or upload their own time series and methods. All necessary parameters are defined via the web-interface. This includes the data to be used for training and forecasting and the experimental conditions, e.g. the forecasting horizon or the evaluation criteria. Hereby, the system can support the user by recommending the initial values. Once an experiment is defined, all corresponding tasks are generated and executed sequentially by a background batch process. Results are stored in the use case repository and immediately displayed in the interface. Further, users can compare the ranking of the obtained output compared to all experiments formerly conducted on that time series.

In short, ECAST helps making results transparent and reproducible. Any personal bias is eliminated by such a neutral tool. It prevents users from battling with large amounts of locally stored CSV-files. Finally, due to the consequent employment of web technology, ECAST is easy accessible, always available and does not require installation or configuration before usage.

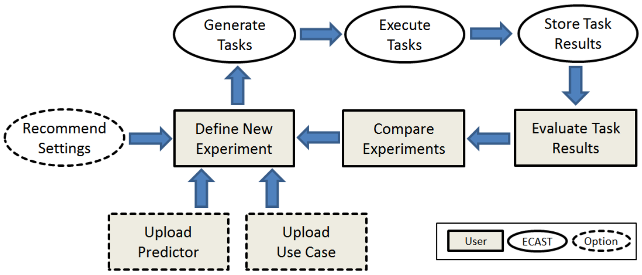


Figure 1: Task Automation Chain

The ECAST platform is composed of the four main components as displayed in Figure 2: (1) A *Use Case Repository (UCR)* as the central data storage unit, (2) the *Core Logic Component (CLC)* as a container to encapsulate all necessary functions, (3) the *Prediction-Interface (API)* as the connector to the forecasting methods and finally (4) the *Web User Interface (GUI)* used for interaction and result presentation.

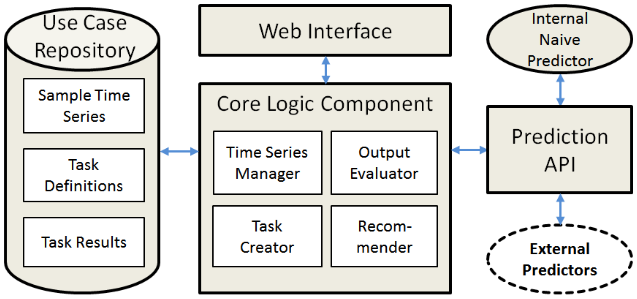


Figure 2: ECAST System Architecture

(1) Use Case Repository. The UCR is the central data storage unit. In its relational data structure we store the reference parameters used for system and experiment configuration, all time series and their associated context information, the generated tasks for the experiments and the output obtained from the predictors. The latter includes the forecasted values, the errors and the experiments' computation time. We implemented the data model in a PostgreSQL database.

(2) Core Logic Component. The CLC is the heart of the system. It contains all functions needed for configuration, handling input and output data and task automation procedures. The implementation is split into separate modules as displayed in Figure 2: First, the *Time Series Manager* is responsible for data preparation and transformation. All incoming time series are converted into the platform's internal format. Next, the *Recommender* queries the repository in order to find the optimal parameter settings for a new experiment. It compares the target time series to all stored use cases based on their structural similarity and the provided context information. Then, it selects the experi-

[2] https://wwwdb.inf.tu-dresden.de/research-projects/projects/reef/

ments with the lowest error values ever conducted on that data and offers the associated parameters as default setting for the user. The *Task Creator* converts all forecast queries belonging to an experiment definition into individual tasks. This means that the chosen settings and parameters are persisted and stored until their final execution. Depending on the definition, a single forecast query can lead to multiple tasks, for example by including various predictors, different forecasting horizons and a variable training history length. Finally, the *Output Evaluator* computes different statistical error metrics to evaluate the accuracy of obtained forecasts.

(3) Prediction API. The API realizes the connection to the predictors. This includes commercial or scientific standalone forecasting tools as well as algorithms scripted in the statistical language *R* In both cases, the API calls the forecasting method, delivers training and evaluation data and sets the values for all available parameters, if any. With this input data, ECAST is able to externally set the configuration for the supported predictors, execute the training of the model and finally the calculation of the forecast. Subsequently, the API returns the forecasted values and the total computation time consumed by the called predictor.

(4) Web User Interface. The GUI is primarily designed to facilitate experiment definition and for result presentation. Consequently, it supports the user during the selection of the uses case, the predictors to be assessed, their parameters and the experimental conditions. In the post-experimental phase, the interface plots the original and the forecasted time series. Visual inspection of time series is common and hard to replace by analytic algorithms, as skilled users often posses expert knowledge which helps to reveal unusual data points. The GUI also displays accuracy values and all parameters and features used in each experiment.

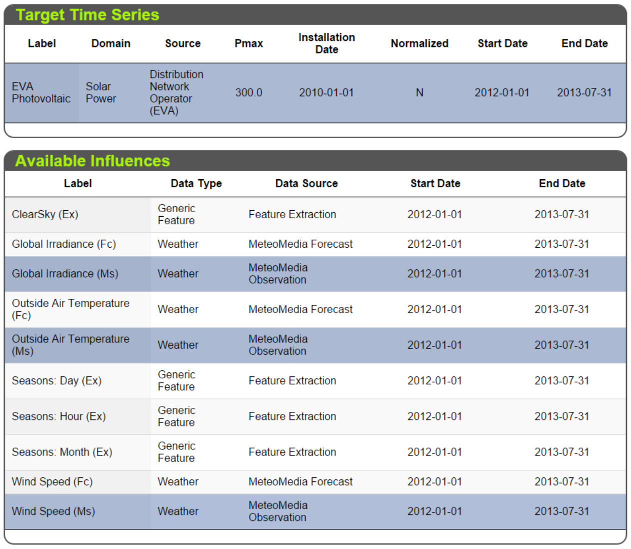
3. DEMONSTRATION

In this section we describe the platform's core features and give a brief walk-through, based on an exemplary energy forecasting scenario.

3.1 Experiment Definition

First, we start by searching the UCR for publicly available data sets we may access. Using the filter functions, we find a data set from the energy production domain having a history length of more than one year and set up a new experiment. We select the external influences which we think will have a positive effect on the forecasting result. As shown in Figure 3, ECAST lists all recorded external influences related to the target time series, such that we can see if the desired information is available or not. In this example we decide to include the observation values for global irradiation, outside air temperature, and wind speed.

In the next step we set the parameters for the experiment (compare Figure 4). We begin with the start and the end dates for the evaluation period, and the horizon which determines how many values we want to predict. Afterwards we choose the error measures we are interested in and decide for MAPE and RMSE, since these are widely used in the energy domain. Finally, we choose the forecasting algorithms we want to evaluate, in our example the MARS algorithm (implemented in the Earth-Package for *R*) and the naive predictor, which is used as base line. As we have included multiple algorithms, they will be executed one after another in separate tasks but using identical parameters. On top of that, ECAST will show all experiments that were already conducted on the currently used data set. This helps to find available experimental results and setting up new experiments which are directly comparable to the previous one by using identical parameters but different forecasting methods. During the set-up process the system supports the user by recommending a pre-selection of forecasting algorithms which performed well on similar data sets based on the results already present in the UCR. The user can modify these recommendations at will. The completely parametrized experiment is subsequently executed by ECAST and the results are stored in the UCR In case we had started by uploading a new data set via a csv-file, the only limitation would be that ECAST would not be able to compare the results with previous experiments.



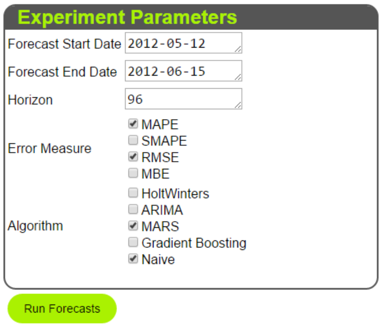
Target Time Series

Label	Domain	Source	Pmax	Installation Date	Normalized	Start Date	End Date
EVA Photovoltaic	Solar Power	Distribution Network Operator (EVA)	300.0	2010-01-01	N	2012-01-01	2013-07-31

Available Influences

Label	Data Type	Data Source	Start Date	End Date
ClearSky (Ex)	Generic Feature	Feature Extraction	2012-01-01	2013-07-31
Global Irradiance (Fc)	Weather	MeteoMedia Forecast	2012-01-01	2013-07-31
Global Irradiance (Ms)	Weather	MeteoMedia Observation	2012-01-01	2013-07-31
Outside Air Temperature (Fc)	Weather	MeteoMedia Forecast	2012-01-01	2013-07-31
Outside Air Temperature (Ms)	Weather	MeteoMedia Observation	2012-01-01	2013-07-31
Seasons: Day (Ex)	Generic Feature	Feature Extraction	2012-01-01	2013-07-31
Seasons: Hour (Ex)	Generic Feature	Feature Extraction	2012-01-01	2013-07-31
Seasons: Month (Ex)	Generic Feature	Feature Extraction	2012-01-01	2013-07-31
Wind Speed (Fc)	Weather	MeteoMedia Forecast	2012-01-01	2013-07-31
Wind Speed (Ms)	Weather	MeteoMedia Observation	2012-01-01	2013-07-31

Figure 3: Data Selection in ECAST



Experiment Parameters

Forecast Start Date 2012-05-12

Forecast End Date 2012-06-15

Horizon 96

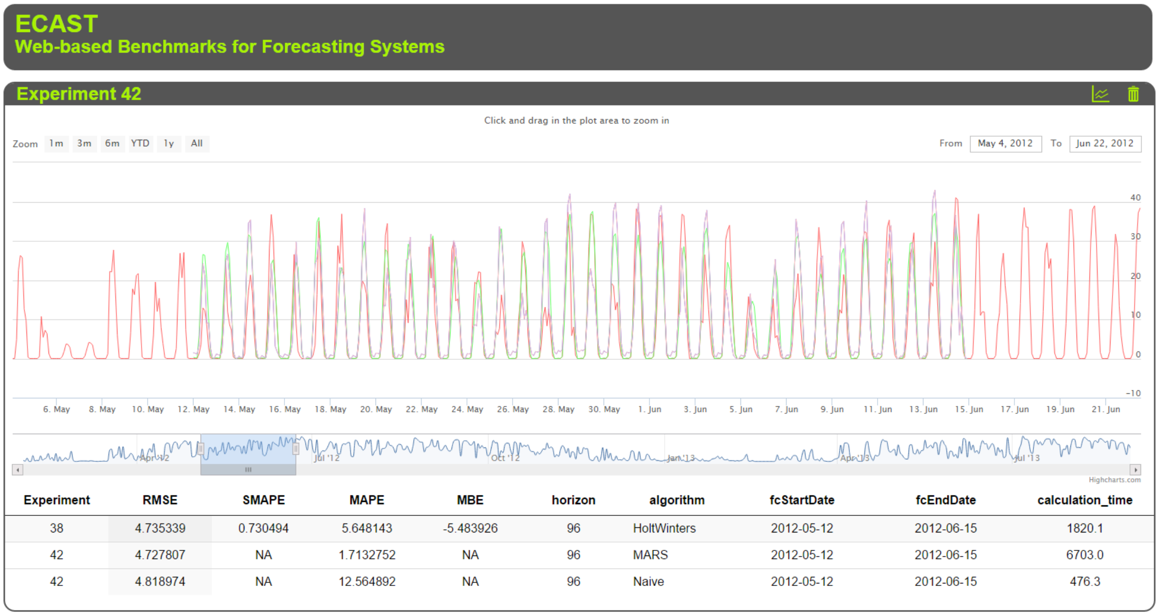
Error Measure
☑ MAPE
☐ SMAPE
☑ RMSE
☐ MBE

Algorithm
☐ HoltWinters
☐ ARIMA
☑ MARS
☐ Gradient Boosting
☑ Naive

Figure 4: Experiment Definition in ECAST

3.2 Result Presentation

The results of the experiment are presented as shown in Figure 5: The diagram section shows the forecasted values of the selected algorithms next to the original time series.



Experiment	RMSE	SMAPE	MAPE	MBE	horizon	algorithm	fcStartDate	fcEndDate	calculation_time
38	4.735339	0.730494	5.648143	-5.483926	96	HoltWinters	2012-05-12	2012-06-15	1820.1
42	4.727807	NA	1.7132752	NA	96	MARS	2012-05-12	2012-06-15	6703.0
42	4.818974	NA	12.564892	NA	96	Naive	2012-05-12	2012-06-15	476.3

Figure 5: Result Presentation in ECAST

This allows for visual exploration of the results to see where the forecast algorithms perform best. The table section in the lower part contains the error measures that were calculated during the experiments, which allows a quick first comparison of the evaluated algorithms. Additionally, older experiments conducted on this data set can be loaded into the table. In this case, we observe that experiment no. 38 using the HoltWinters algorithm obtained almost similar results to the current experiment (no. 42) in terms of RMSE, but under-performs in terms of MAPE. Any of these experiments can be chosen to be displayed next to the current one for a more detailed visual comparison in the diagram area.

Now, with this information we are able to decide whether we are satisfied with the performance of the chosen algorithm or not. For example, if the visual exploration shows that the deviation of the forecasted values from the original time series is too high for the time stamps with the highest energy production, although the average RMSE error is satisfying we might want to start another experiment. In this case we can start over using different algorithms or external influences. The ECAST system even allows it for experienced users to upload their own forecasting algorithms written in *R* and use them in the experiments.

4. SUMMARY AND OUTLOOK

In this demonstration we have shown that ECAST is an open and easy-to-use platform for time series forecasting. It supports the user with automated tasks, provides real-world use cases and offers sophisticated pre- and post-processing procedures, thus making complex benchmarks much more efficient. In the future, we plan to integrate additional logical modules like an Ensembler, to build flexible hybrid models by using appropriate combination criteria, or a Feature Optimizer to allow for an automatic creation and optimal pre-selection of external influences for the models to be built.

Acknowledgment

The work presented in this paper was funded by the European Regional Development Fund (EFRE) under co-financing by the Free State of Saxony and Robotron Datenbank-Software GmbH. We gratefully acknowledge the contributions of Lucas Bruenings and Johannes Wilke.

5. REFERENCES

[1] Z. Chen and Y. Yang. Assessing forecast accuracy measures. Technical Report 2004-2010, Iowa State University, Department of Statistics & Statistical Laboratory, 2004.

[2] T. Hong, P. Pinson, and S. Fan. Global Energy Forecasting Competition 2012. *International Journal of Forecasting*, 30(2):357–363, 2013.

[3] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[4] V. Kostylev and A. Pavlovski. Solar Power Forecasting Performance - Towards Industry Standards. In *1st Int. Workshop on the Integration of Solar Power into Power Systems*, Aarhus, Denmark, 2011.

[5] H. Madsen, G. Kariniotakis, H. Nielsen, T. Nielsen, and P. Pinson. A Protocol for Standardizing the Performance Evaluation of Short-Term Wind Power Prediction Models. Anemos project, European Commission, 2004.

[6] S. Makridakis and M. Hibon. The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, 16:451–476, 2000.

[7] R. Nambiar, M. Poess, A. Masland, H. R. Taheri, M. Emmerton, F. Carman, and M. Majdalany. TPC Benchmark Roadmap 2012. In *Selected Topics in Performance Evaluation and Benchmarking*, pages 1–20. Springer Berlin Heidelberg, 2013.