Schriftenreihe CIplus, Band 5/2013

Herausgeber: T. Bartz-Beielstein, W. Konen, H. Stenzel, B. Naujoks

UniFIeD
Univariate Frequency-based Imputation for Time Series Data

Martina Friese, Jörg Stork, Ricardo Ramos Guerra, Thomas Bartz-Beielstein, Soham Thaker, Oliver Flasch, Martin Zaefferer

Fachhochschule Köln
Cologne University of Applied Sciences

UniFIeD Univariate Frequency-based Imputation for Time Series Data

Martina Friese, Jorg Stork, Ricardo Ramos Guerra,
Thomas Bartz-Beielstein, Soham Thaker, Oliver Flasch, Martin Zaefferer

Faculty for Computer and Engineering Sciences
Cologne University of Applied Sciences, 51643 Gummersbach, Germany
firstname.lastname@fh-koeln.de

Abstract. This paper introduces UniFIeD, a new data preprocessing method for time series. UniFIeD can cope with large intervals of missing data. A scalable test function generator, which allows the simulation of time series with different gap sizes, is presented additionally. An experimental study demonstrates that (i) UniFIeD shows a significant better performance than simple imputation methods and (ii) UniFIeD is able to handle situations, where advanced imputation methods fail. The results are independent from the underlying error measurements.

1 Introduction

Missing data is a well-known problem in nearly every real-world time series. Sensors may fail, data might get lost during transfer, or measurements are simply missing. Although this problem is well-known, many standard time-series prediction and analysis methods rely on complete data. During the last decades, several ways have been developed to tackle missing data. Several of these methods are applicable to small gap sizes only.

A common suggestion, which is available in several software packages, is the imputation of mean values. This approach can destroy inherent data structures and may worsen the statistical modeling, resulting in large prediction errors [4]. Imputing missing elements from regression or *analysis of variance* (ANOVA) are usually better. More advanced methods, also from computational intelligence, for data imputation were developed in the context of univariate (linear, spline, and nearest neighbor interpolation), multivariate (regression-based imputation, nearest neighbor, self-organizing map, multi-layer perceptron), and hybrid methods of the previous by using simulated missing data patterns. A small study, which discussed the applicability of these methods to air quality data sets, was performed by Junninnen et al. [7]. Single imputation methods, i.e., filling in precisely one value for each missing one, can be distinguished from multiple imputation methods. The latter generate multiple simulated values for each missing value.

Our study was initialized by an real-world task and focuses on the applicability of univariate methods. It was motivated by a real-world problem, because we received time-series data with large gaps from one of our industrial partners. These data should be used for time-series predictions, where the methods

of choice require complete data. Since simple imputation methods failed completely in our setting and the advanced methods did not show the expected performance, we decided to develop a new imputation method.

The new method, entitled *univariate frequency-based imputation for time series data* (UniFIeD) outperformed the existing methods. The success in field settings right from the start motivated a first analysis and gave reason for performing an experimental study. Focussing on methods for large intervals of missing data, seasonal data, especially time series data, we consider the following scientific *goals*:

(G-1) Which method generates the smallest imputation error?
(G-2) What is the influence of data pre-processing methods on the performance of forecast methods?

Based on these goals, we are interested in developing an automated and robust procedure for data pre-processing, which can be implemented easily.

To generate scientifically significant results, we will proceed as follows. First we generate instances based on the real-world data. Then we run the imputation methods. Next, the errors based on different error measurements are determined. Finally, their prediction errors are reported and compared on different error measurements. As a future step to increase the plausibility of our findings, we are planning to perform predictions with different state-of-the-art methods.

This paper is structured as follows: First, the real-world data is described in Sec. 2. Pre-processing methods and the univariate frequency-based imputation for time series data (UniFIeD) are introduced in Sec. 3. The prediction models, which were used in the final comparison, are described in Sec. 4. Error measures, which play a crucial role in our study, are presented in Sec. 5, the two different experiments are introduced in Sec. 6. Results are presented in Sec. 7, our findings are discussed in Sec. 8. The paper concludes with a summary and an outlook in Sec. 9. An R version of the program code used in this study, is freely available for download and will be compiled as an R package [8].

2 Data

2.1 Missing Data

We consider three types of data: y^* denotes the underlying (latent and complete) data, y is the observed data, and $\hat{y}$ is the imputed data.

To evaluate the performance of an imputation method, criteria have to be defined. The imputation performance depends (at least) on two characteristics: (a) the structure of missing data pattern and (b) the amount of missing data. If the probability of missing data does not depend upon the observed or the unobserved data, then these data are called *missing completely at random* (MCAR)[9]. There is no predictive power in the observed values y, if the missing value process is MCAR. In general, the structure of missing data in our projects is MCAR. The simulation of missing data pattern randomly will be described in Sec. 2.3.

2.2 The Datasets

Our study is based on real-world data. The experiments are based on energy consumption time-series data supplied by GreenPocket GmbH. The data was recorded by two independent smart metering devices, installed at a local commercial customer. Some data points are missing due to measurement or transmission issues, which is a common situation in real-world settings. The data provided by GreenPocket GmbHis a series of timestamp and meter reading pairs taken quarter hourly. Timestamps are given an ISO 8601 derived date/time format, meter readings are given in kilowatt hours (kWh). The energy consumption time series data was recorded at the same time interval by two independent smart metering devices resulting in the two data sets `series_meter1` and `series_meter2`. Both energy consumption time series datasets contain a total of 8548 entries starting at `2010-12-06 23:15:00` and ending at `2011-03-06 00:00:00` which makes a total time interval of more than 12 Weeks. The complete time series data set `series_meter1` is shown in Figure 1, whereas Figure 2 shows only the last two weeks of the same data set.

```
[1] "English_United States.1252"
```

```
[1] "German_Germany.1252"
```

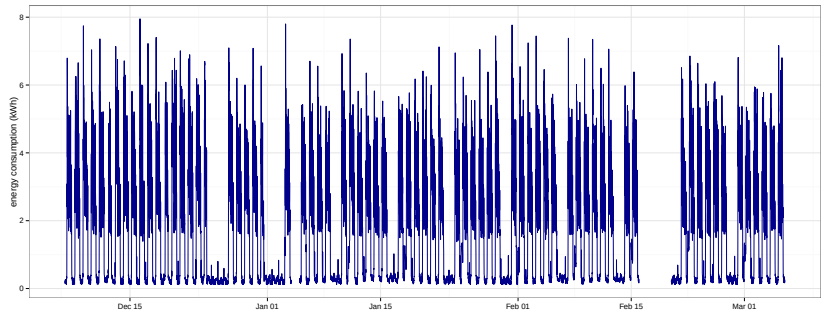

Fig. 1. plot of all the observations given in `series_meter1`

Visual inspection of the data reveals daily periods, while weekly periods are detectable, but not as clearly defined. Also time intervals with missing data can be clearly seen. Having a closer look at the missing data values, contained in both time series, reveals that there are altogether twelve gaps. Mostly smaller gaps of length one, but also larger gaps up to the size of 385 missing observations.

2.3 Test Instance Generation

Since missing data already occurs in the real-world test data, we are able to determine a realistic distribution of the gap sizes and frequencies. This includes

```
[1] "English_United States.1252"
```

```
[1] "German_Germany.1252"
```

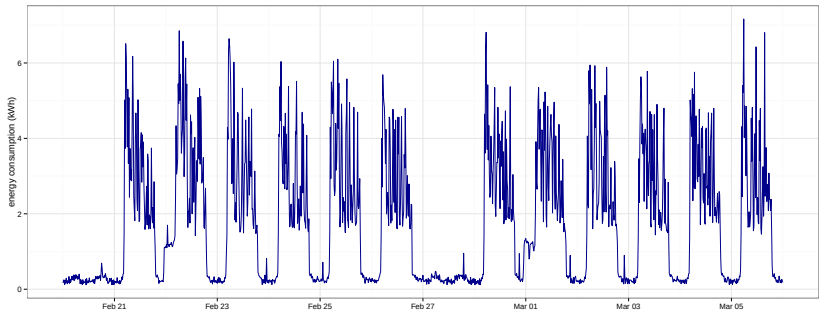

Fig. 2. excerpt of the observation from `series_meter1` showing only the last two weeks of the data set

the determination of two parameters: (a) distribution of the gap sizes `gapsize` and (b) the total amount of missing data, i.e., `gappercentage`. The `gapsize` distribution can be estimated from real-world data as follows. First, histogram plots were used for visual inspection of the gap sizes. Visual inspection suggests an exponential distribution of `gapsize` with smaller gaps appearing more frequently than larger gaps. The parameter λ from the density of the exponential distribution $f(t) = \lambda \exp(-\lambda t)$ is estimated from the real-word data. Now we are able to generate random gaps with reasonable sizes that are in correspondence with real-world data. In a second step, we determine the percentage of missing data, `gappercentage`. Here, we consider values between 5 and 30. The generation of a single test instance than works as outlined in algorithm 1.

3 Pre-processing Methods

3.1 Existing Methods

Existing imputation methods can be partitioned into two groups: the first group included basic methods [7], that do not use complex computations to determine the imputed values. A second group uses sophisticated techniques for imputations. We will present the basic methods first.

Basic Imputation Methods Mean imputation is an often used method because of its simplicity. The missing data is replaced by the mean of the non-missing observed data.

$$\hat{y} = \overline{y}, \tag{1}$$

where $\hat{y}$ is the imputed value, y_t are the observed values, and $\overline{y}$ denotes the sample mean. Linear interpolation uses the start and end point of a gap to

input : Time Series t
input : `gappercentage`
countData = number of observations ! = NA in t
countDrop = *countData* × `gappercentage`/100;
repeat
| draw random `gapsize` from exponential distribution;
until *countDrop reached*;
countGaps = number of gaps drawn;
remainingData = *countData* - *countDrop*;
generate random *partition* of *countGaps*+1 parts summing up to the size of *remainingData*;*(assuring that the generated time series neither starts nor ends with a gap and has at least one data point between two gaps.)*
$t^* = t$;
for $i = 1 \rightarrow$ *countGaps* **do**
| *position* = sum(partitions[1:i] + sum(`gapsize` [1:i-1]));
| remove data from s^* from *position* to *position*+`gapsize` [i];
end
output: t^*

Algorithm 1: Generation of a test instance

construct a straight line.

$$\hat{y} = y_{t1} + k \times (t - t_1) \; with \; k = \frac{y_{t_2} - y_{t_1}}{t_2 - t_1} \tag{2}$$

y_{t_1} and y_{t_2} are the start and end values of the gap, while t_1 and t_2 are the start and end time values. x is the current time value. Nearest neighbors uses the start and end points of a gap as estimates for the imputation.

$$\begin{aligned} \hat{y} &= y_{t_1} \; if \; t < t_1 + \frac{t_2 - t_1}{2} \\ \hat{y} &= y_{t_2} \; if \; t > t_1 + \frac{t_2 - t_1}{2} \end{aligned} \tag{3}$$

Advanced Imputation Methods The second group of imputation methods uses advanced techniques. The mice (Multivariate Imputation by Chained Equations) package specializes on multiple imputation methods [1]. The methods works best on multivariate data and no method applicable to the smart metering data was found to deliver good results. The zoo packages provides a methods `na.structTS` uses a generic function for filling NA values using seasonal Kalman filter [10]. Finally the Amelia II package can be mentioned here [3]. It was not able to find suitable values for the time interval from the Smart Metering data set. These packages obtain very good results, if multivariate time-series data were available.

3.2 Univariate Frequency-based Imputation

The UniFIeD method proposed in this work relies on an automated estimation of time-series frequencies.

Estimating time-series frequencies automatically For the estimation of the frequencies contained in the data, the auto correlation function (acf) is used. The algorithm works as described in Algorithm 2.

input : Time Series t
determine acf values via auto correlation function on t;
remember *autocorrelation* values from acf;
remember related *lags* from acf;
repeat
  reduce *autocorrelation* values to its peaks
  remember related *lags*;
  determine $frequency$ from lags; *via the frequency of the distances from one peak-lag to another*
until *no new* $frequency$ *found*;
output: all $frequencies$ found

Algorithm 2: Estimation of an underlying frequency using the auto correlation function.

For better illustration, figure 3 shows the auto correlation function for `series_meter1`. Both, daily and weekly periods are clearly recognizable. The set of peaks that are considered as indicators for the data's underlying frequency, are marked with small circles. In the second iteration of the algorithm, this set of peaks is reduced to a smaller set, which is marked with filled circles. After two iterations, the algorithm stops since there is no lower frequency in the data.

How UniFIeD works UniFIeD is valuable for univariate time series data that presens a pattern or a seasonal effect. The algorithm takes advantage of this seasonal effect to find correlated patterns of the missing window to develop the imputation. Using the frequency estimated with Algorithm 2, we proceed to impute the missing values from the time series data set t.

UniFIeD was developed not only to consider single missing points $\hat{y}$ but full missing time windows of any size. The basic idea is to iteratively look for the next missing point $\hat{y}$ of the time series t at the time moment t_m and, using the frequency f and the number of similar windows k to search in, gather the amount of $2k$ non missing points y that correlate to the time moment t_m of the found missing point and form a vector t_s. Once this vector t_s is formed, and depending on the user's request, the appropriate method to calculate the value to impute into the time series t is selected, where the available options to determine the new value are the mean, median, maximum, or minimum values of vector t_s. Figure 4 shows an example of a missing value $\hat{y}$, the selected correlated values in the vector t_s and the different methods used to impute the new value of a randomly selected missing window for the purposes of illustration.

The UniFIeD algorithm works as follows:

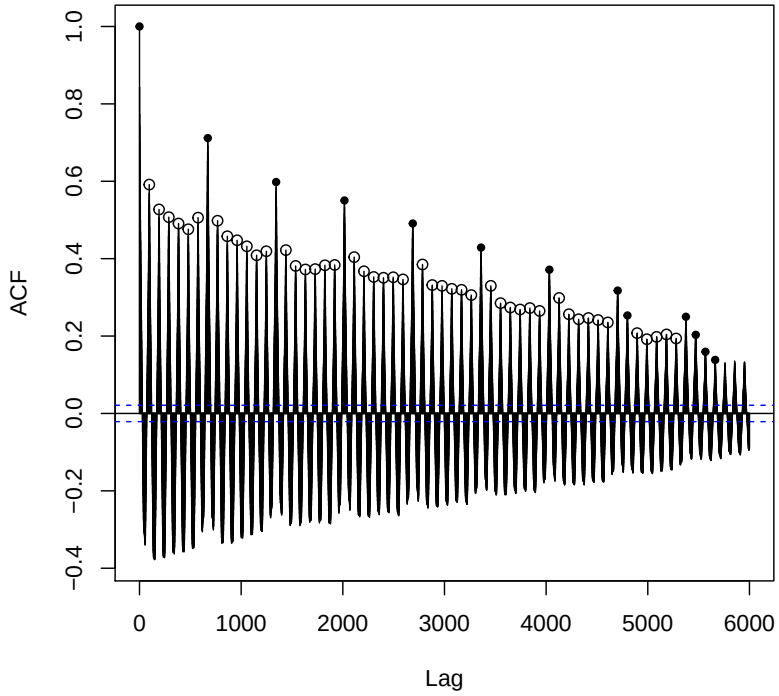

Fig. 3. Plot of the auto correlation function output on `meter_series1`. It shows the auto correlation value for each lag. Values, considered as peaks indicating the frequency, are marked with small circles.

Note that the final step of Algorithm 3 is to indicate whether any NA were imputed into time series t. If this is fact, the algorithm will look for the left neighbor and impute it's value into $\hat{y}$.

4 Prediction Models

4.1 Holt-Winter's

Holt-Winter's algorithm [2] is used to forecast time series with trends and seasonal effects. In R, Holt-Winter's algorithm is implemented in the stats package.

input : Univariate Time Series t
input : Frequency f of t
input : Constant k of similar windows to look for correlated values
input : Method M to calculate value to impute: mean, median, max or min
output: Time Series $\hat{t}$
initialization; define a vector of window numbers to look for, using k
$win \leftarrow \{-k, -k+1, \ldots, -1, 1, \ldots, k-1, k\}$
for $i \leftarrow$ *initial point of data* **to** *end of data* **do**
  if found y point *is NA* **then**
    look for correlated values to y and form vector t_s by using win vector and freq. f as follows:
    $t_s \leftarrow \{y_{(t_m-kf)}, y_{(t_m-(k-1)f)}, \ldots, y_{(t_m-f)}, y_{(t_m+f)}, \ldots, y_{(t_m+(k-1)f)}, y_{(t_m+kf)}\}$
  end
  switch method M chosen, calculate **do**
    mean: $\hat{y} \leftarrow mean(t_s)$
    median: $\hat{y} \leftarrow median(t_s)$
    max: $\hat{y} \leftarrow max(t_s)$
    min: $\hat{y} \leftarrow min(t_s)$
  endsw
end
check if some NA values were imputed into time series $\hat{t}$, and if there are, fix by using the left neighbor value.

Algorithm 3: Imputation Algorithm.

The method works with three exponential smoothing equations:

$$\textbf{Level:}\quad \ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{4}$$

$$\textbf{Trend:}\quad b_t = \beta^*(\ell_t - l_{t-1} + (1-\beta^*)b_{t-1} \tag{5}$$

$$\textbf{Seasonal:}\quad s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{6}$$

The forecasting equation is:

$$\textbf{Forecast:}\quad \hat{y}_{t+h|t} = \ell_t + hb_t + s_{t-m+h_m^+} \tag{7}$$

with $h_m^+ = \lfloor (h-1) \mod m \rfloor + 1$. α, β^* and γ are so-called smoothing parameters. The level equation is a weighted average of the seasonally adjusted observation $(y_t - s_{t-m})$ and the non-seasonal forecast $(\ell_{t-1} + b_{t-1})$ for time t. The trend equation shows that b_t is a weighted average of the estimated trend at time t based on $\ell_t - \ell_{t-1}$ and b_{t-1}, the previous estimate of the trend. The seasonal equation is a weighted average of the current seasonal index, $(y_t - \ell_{t-1} - b_{t-1})$ and the seasonal index of the same season last term. Initial values for the level, trend, and seasonal indices are calculated using a simple decomposition and regression on the first two seasons of the given time series. The smoothing parameters are, if not given manually, fitted by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method with the sum of squared errors of prediction (SSE).

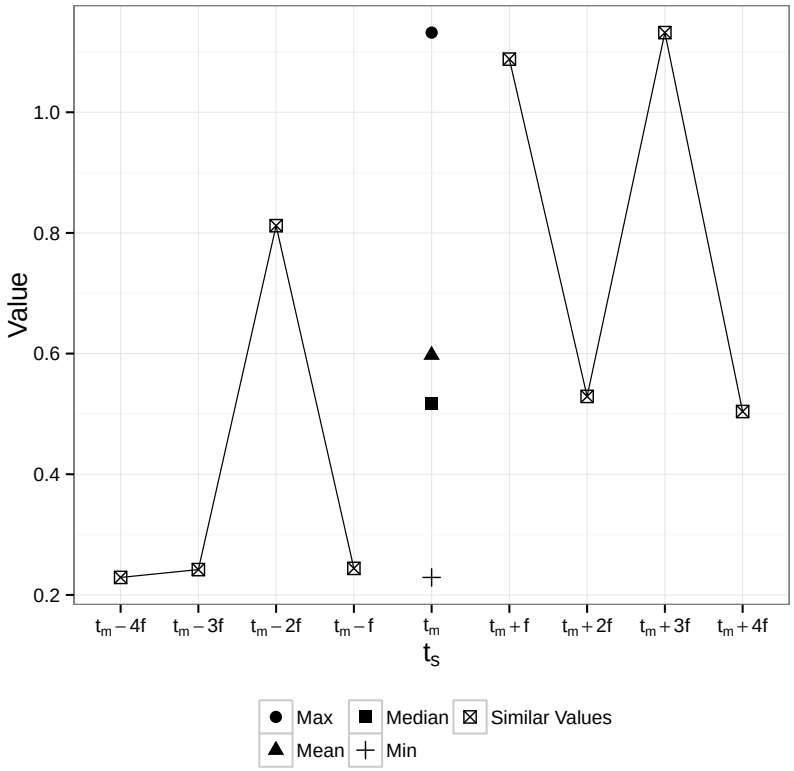

Fig. 4. The plot shows the vector t_s of correlated values to the missing y value at t_m and the new calculated values $\hat{y}$ to impute based on t_s, median, mean, maximum, and minimum. The lines are only for the purposes of illustration, the points are not continuous in time.

4.2 ETS and ARIMA

Exponential smoothing state space (ETS) models and autoregressive integrated moving average (ARIMA) models are state-of-the art methods for time-series forecasting. Both types of models support a high variety of different data structures. In this work, an implementation provided by the R package forecast [5] is used. This implementation features ensemble-based methods with an automated model selection process. The automated ARIMA and automated ETS models are limitied in terms of seasonal period length to $m = 13$, so for our data with $m = 672$, preprocessing is done by an STL decomposition. The seasonal component is hereby extracted, then the automated ARIMA or ETS are applied to forecast the adjusted data. After this, the seasonal component is added to the forecasted values.

5 Error Measures

In our experiments, every method had to impute each of the generated test instances. For the evaluation of the quality of the imputation method, only the imputed parts of the time series have been taken into consideration. We considered the following error measures:

MAE The *mean absolute error* (MAE) between the predicted time series $\hat{y}$ and the respective true energy consumption (test) time series $y.$, i.e.,

$$\mathrm{MAE}(\hat{y}, y) := \frac{\sum_{t=1}^{n} |\hat{y}_t - y_t|}{n}$$

RMSE The *root mean square error* (RMSE) between the imputed time series $\hat{y}$ and the respective true energy consumption (test) time series y, i.e.,

$$\mathrm{RMSE}(\hat{y}, y) := \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}}$$

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

$$\mathrm{RMSElog}(\hat{y}, y) := \sqrt{\frac{\sum_{t=1}^{n} (\log(1 + \hat{y}_t) - \log(1 + y_t))^2}{n}}.$$

The reason for applying a logarithmic transformation to the time series is that energy consumption is always positive or zero and its distribution is highly skewed. By applying a log transformation we aim to obtain a less skewed distribution. The RMSE on the other hand is a symmetric loss function and therefore, it is best applicable if the error distribution is symmetric. We also postulate that this loss features larger relevance in practical applications.

6 Experiments

6.1 Imputation Experiments

To get a deeper insight in the benefits of the proposed imputation method and to have a fair comparison to established methods, we decided to define two sets of experiments applied to both `series_meter1` and `series_meter2`.

- The first set is based on the data as is without any changes or additional constraints.
- The established simple methods that are compared to our methods work better on smaller gap sizes. Thus, for the second set of experiments we added a constraint allowing a maximum gap size of 5.

We considered 5%, 10%, 20% and 30% of total missing data. For each combination of experiment set, data set, and percentage of missing data, 10 random test instances were generated, thus, we get a total of $4 \cdot 6 \cdot 10 = 240$ test instances.

6.2 Forecast Experiments

The new imputation method is expected to improve the results of the forecast. Therefore, in an additional experiment, the accuracy of the forecast models on the imputed time series `series_meter1` are tested. The last four weeks of `series_meter1` were extracted and used for comparison.

We will use Holt-Winter's algorithm (cf. Sec. 4.1) as a classical forecasting method, which will be complemented by an automatic forecasting method proposed by Hyndman [6].

7 Results

7.1 Imputation Results

Table 1 shows the results for the simulated smaller `gapsize` of missing data on `series_meter1`. The mean of the MAE and RMSE errors shows that the UniFIeD methods, using the mean and median options, presents similar results and better mean distribution over all the runs than the linear and nearest neighbors methods. Figure 5 shows the plot for the mean RMSElog error of 10 experiments for each of the four `gappercentage` , grouped by method, on the x axis. We can see that the UniFIeD method with the mean and median options can compete with the expected good results of the linear and nearest neighbors, and with the minimum and maximum options, perform outside the best methods on this data.

Table 1. Mean errors for 40 runs of each method with smaller gapsize.

Method	RMSE	RMSElog	MAE
impmax	1.51 (±0.15)	0.40 (±0.07)	0.98 (±0.11)
impmean	**0.93 (±0.12)**	0.26 (±0.05)	**0.58 (±0.08)**
impmedian	0.98 (±0.13)	0.27 (±0.06)	0.59 (±0.09)
impmin	1.40 (±0.13)	0.44 (±0.06)	0.84 (±0.10)
linear	1.01 (±0.05)	**0.25 (±0.01)**	0.61 (±0.05)
mean	1.75 (±0.07)	0.64 (±0.02)	1.50 (±0.05)
nn	1.07 (±0.05)	0.27 (±0.01)	0.65 (±0.05)

Table 2 shows the results with simulated larger `gapsize` of missing data on `series_meter1`. The mean of RMSE and MAE shows, as before, that the UniFIeD methods with mean and median options performs better than others, having a RMSE of **0.93** with a standard deviation of ±0.10 and **0.98** with a standard deviation of ±0.12, respectively. In this case, the maximum and minimum options perform better than the basic imputation methods but not as good as their partner options from UniFIeD. Figure 6 plots the mean RMSElog errors over 10 experiments for each simulation of `gappercentage` and larger `gapsize`. Note that for these experiments, the linear and nearest neighbors methods did

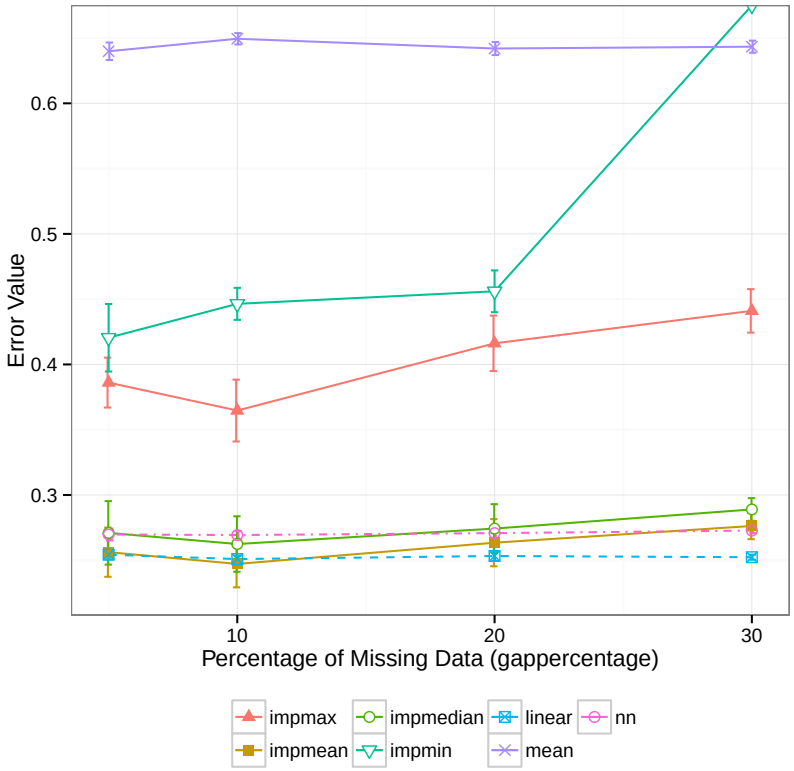

Fig. 5. Mean RMSElog error of the results for smaller `gapsize` simulated on `series_meter1`. On the x axis, the `gappercentage` is plotted vs. the error value over the 40 experimental runs per method on the y axis. The labels with "imp" prefix refer to the UniFIeD method with different options.

not perform as well as in the previous experiments, which was expected. Additionally, for larger `gapsize` of missing data, the UniFIeD method still performs well, being the only one delivering best results.

On the performance, Table 3 presents the means of the required time in seconds for each imputation of every method. The linear and nearest neighbors methods stand together as the fastest imputation methods and all the options from UniFIeD take more time in comparison with these. However, in our opinion, it still does not require too much time to achieve the previous accuracy results.

7.2 Forecast Results

Once the imputation experiments performed, we proceeded to forecast on this data and obtained the results presented in Table 4. These show that, considering all three error measurements, the UniFIeD method with the mean option is the

Table 2. Mean errors for 40 runs of each method with larger gapsize.

Method	RMSE	RMSElog	MAE
impmax	1.47 (±0.17)	0.39 (±0.08)	0.94 (±0.12)
impmean	**0.93 (±0.10)**	**0.26 (±0.05)**	**0.57 (±0.07)**
impmedian	0.98 (±0.12)	0.28 (±0.06)	0.57 (±0.07)
impmin	1.40 (±0.14)	0.44 (±0.08)	0.83 (±0.10)
linear	2.00 (±0.20)	0.70 (±0.07)	1.49 (±0.20)
mean	1.75 (±0.06)	0.65 (±0.02)	1.51 (±0.05)
nn	2.22 (±0.26)	0.77 (±0.08)	1.60 (±0.24)

Table 3. Mean time (seconds) consumption for single imputations of each method with larger gapsize.

Method	Time
impmax	1.52 (±0.50)
impmean	1.54 (±0.50)
impmedian	1.60 (±0.52)
impmin	1.52 (±0.48)
linear	0.34 (±0.06)
mean	0.73 (±0.22)
nn	**0.34 (±0.05)**

only method with a better outcome than the normal mean and nearest neighbors methods.

Table 4. Forecast mean errors for 40 runs of each method with larger gapsize.

Method	RMSE	RMSElog	MAE
impmean.Arima	0.82 (±0.01)	0.23 (±0.01)	0.54 (±0.02)
impmean.ETS	**0.82 (±0.01)**	**0.23 (±0.01)**	**0.53 (±0.01)**
impmean.HW	1.08 (±0.11)	0.40 (±0.08)	0.81 (±0.12)
mean.Arima	0.91 (±0.16)	0.29 (±0.09)	0.67 (±0.20)
mean.ETS	1.00 (±0.23)	0.32 (±0.13)	0.73 (±0.29)
mean.HW	1.41 (±0.33)	0.49 (±0.13)	1.09 (±0.33)
nn.Arima	0.94 (±0.11)	0.31 (±0.07)	0.69 (±0.13)
nn.ETS	0.91 (±0.08)	0.29 (±0.04)	0.64 (±0.08)
nn.HW	1.75 (±0.83)	0.53 (±0.16)	1.37 (±0.76)

8 Discussion

The experiments result indicate that for small gap sizes the nearest neighbours, linear imputation, and the different options of the UniFIeD method work best,

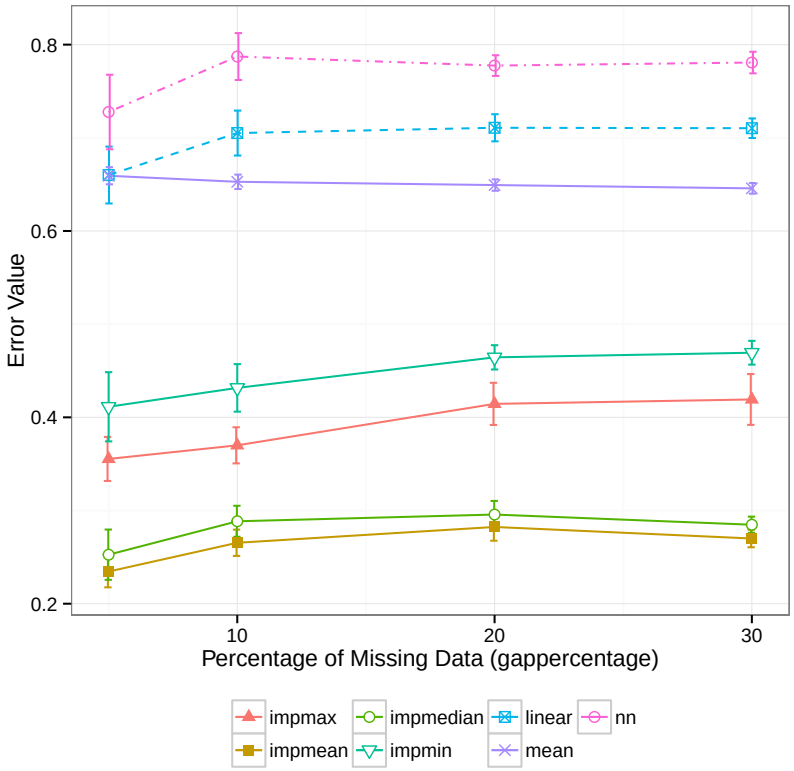

Fig. 6. Mean RMSElog error of the results for larger `gapsize` simulated on `series_meter1`. On the x axis, the `gappercentage` is plotted vs. the error value over the 40 experiments for each method on the y axis. The labels with "imp" prefix refer to the UniFIeD method with different options.

while the simple mean method delivers overall worse results. These results indicate that our method is better suited for large gap sizes. It is also competitive for small gap sizes, although the differences are not so apparent. Surprisingly, the difference between different percentages of missing data is smaller than expected. Implementing this method is easy and does only require a few dozen lines of code. The additional time requirements for executing UniFIeD is only marginal compared to the simple methods. UniFIeD itself is robust and improves the robustness of the whole forecasting process. Similar results were obtained with different error measurements. An extension of our studies is needed to demonstrate the applicability of UniFIeD in complex forecasting procedures. First experiments indicated promising results. But we are aware that these experiments require complex experimental setups.

9 Summary and Outlook

In this work, we introduced a scalable test problem generator for simulating missing data. We generated problem instances with small and large gaps, furthermore we introduced a new method called UniFIeD for handling missing data, which uses an automated frequency estimator using autocorrelation. We demonstrated that this method outperformes simple imputation methods. The results are independent from the underlying error measurements.

Future plans involve the following steps:

- extended experiments with additional data
- discover the limits of the method
- discover the maximum gap sizes
- statistical validate the method
- can it be used as a hybrid method
- experiments on mutlivariate time-series

We are also planning to provide UniFIeD as an R package on CRAN.

10 Acknowledgments

This work has been kindly supported by the German Federal Ministry of Education and Research (BMBF) under the grants MCIOP (FKZ 17N0311) and CIMO (FKZ 17002X11).

References

1. S. Buuren and K. Groothuis-Oudshoorn. Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3), 2011.
2. C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004.
3. J. Honaker, G. King, and M. Blackwell. Amelia ii: A program for missing data. *Journal of Statistical Software*, 45(7):1–47, 12 2011.
4. N. Horton and K. Kleinman. Much ado about nothing. *The American Statistician*, 61(1):79–90, 2007.
5. R. Hyndman and Y. Khandakar. Automatic time series for forecasting: The forecast package for r. Technical report, Monash University, Department of Econometrics and Business Statistics, 2007.
6. R. J. Hyndman and Y. Khandakar. Automatic Time Series Forecasting. The forecast Package for R. *Journal of Statistical Software*, 27(3):1–22, 7 2008.
7. H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen. Methods for imputation of missing values in air quality data sets. *Atmospheric Environment*, 38(18):2895 – 2907, 2004.
8. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
9. D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.
10. A. Zeileis and G. Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *arXiv preprint math/0505527*, 2005.

Kontakt/Impressum

Diese Veröffentlichungen erscheinen im Rahmen der Schriftenreihe "CIplus". Alle Veröffentlichungen dieser Reihe können unter
www.ciplus-research.de
oder unter
http://opus.bsz-bw.de/fhk/index.php?la=de
abgerufen werden.

Köln, Januar 2012

Herausgeber / Editorship

Prof. Dr. Thomas Bartz-Beielstein,
Prof. Dr. Wolfgang Konen,
Prof. Dr. Horst Stenzel,
Dr. Boris Naujoks
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
Cologne University of Applied Sciences,
Steinmüllerallee 1,
51643 Gummersbach
url: www.ciplus-research.de

Schriftleitung und Ansprechpartner/ Contact editor's office

Prof. Dr. Thomas Bartz-Beielstein,
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
Cologne University of Applied Sciences,
Steinmüllerallee 1, 51643 Gummersbach
phone: +49 2261 8196 6391
url: http://www.gm.fh-koeln.de/~bartz/
eMail: thomas.bartz-beielstein@fh-koeln.de

ISSN (online) 2194-2870