CIplus
Band 7/2016

Data Preprocessing: A New Algorithm for Univariate Imputation Designed Specifically for Industrial Needs

**Sowmya Chandrasekaran, Martin Zaefferer, Steffen Moritz,
Jörg Stork, Martina Friese, Andreas Fischbach,
Thomas Bartz-Beielstein**

Technology
Arts Sciences
TH Köln

Data Preprocessing: A New Algorithm for Univariate Imputation Designed Specifically for Industrial Needs

Sowmya Chandrasekaran, Martin Zaefferer, Steffen Moritz,
Jörg Stork, Martina Friese, Andreas Fischbach,
Thomas Bartz-Beielstein

SPOTSeven Lab, TH Köln
Steinmüllerallee 1, 51643 Gummersbach
E-Mail: {sowmya.chandrasekaran, martin.zaefferer, steffen.moritz, joerg.stork,
martina.friese, andreas.fischbach, thomas.bartz-beielstein}@th-koeln.de

Abstract

Data pre-processing is a key research topic in data mining because it plays a crucial role in improving the accuracy of any data mining algorithm. In most real world cases, a significant amount of the recorded data is found missing due to most diverse errors. This loss of data is nearly always unavoidable. Recovery of missing data plays a vital role in avoiding inaccurate data mining decisions. Most multivariate imputation methods are not compatible to univariate datasets and the traditional univariate imputation techniques become highly biased as the missing data gap increases. With the current technological advancements abundant data is being captured every second. Hence, we intend to develop a new algorithm that enables maximum utilization of the available big datasets for imputation. In this paper, we present a Seasonal and Trend decomposition using Loess (STL) based Seasonal Moving Window Algorithm, which is capable of handling patterns with trend as well as cyclic characteristics. We show that the algorithm is highly suitable for pre-processing of large datasets.

1 Introduction

Data pre-processing involves removal of noise and outliers from a dataset, handling of missing values, data redundancy and data inconsistency. One of the most challenging task among them is to impute the missing values with entries that reasonably complete the datasets. The loss of data may be due to sensor errors, transmission errors, errors of the operator and other errors. Recovery of these missing values heavily affects the performance of the data mining models. The accuracy of forecasting, classification, estimation, and pattern detection of any data mining algorithm depends significantly on the accuracy of data used in modeling. Thus, inaccurate training and testing data may introduce bias into the models and provide misleading conclusions [14]. In reality, datasets are commonly univariate. Moreover, multivariate datasets may lack correlation. The need for univariate time series analysis is prevalent in many fields, for example, online data monitoring and pattern detection in intensive care units [1], forecasting in hydrology and environmental management fields [2], functional magnetic resonance imaging statistical analysis [3], forecasting intra day arrivals at a call center [4], forecasting electricity spot-prices [5], forecasting macroeconomic time series [6].

With univariate time series data, the complexity of replacing these missing value increases as no correlated variables are available. Almost all of the well known standard techniques fail to handle univariate time series data as their scheme is based on the inter-attribute correlations in estimating the values for the missing data. Also, the existing multivariate algorithms either cannot be applied or perform poorly. Furthermore, some traditional imputation techniques perform well with trend datasets, while some techniques perform only well if the dataset is seasonal. There exists no single univariate imputation technique that is suitable for all types of data patterns [7]. This is owing to the reason that most of the existing algorithms are designed extensively to handle either seasonality or trend and not both.

The major motivation for this work is the GECCO Industrial Challenge 2015. The task was to recover the missing data in heating systems (`http://www.spotseven.de/gecco/gecco-challenge/gecco-challenge-2015/`). The data contains 606,837 observations of four parameters sampled every minute from real industrial heating systems. The most challenging aspect of this challenge is to impute the missing data for a large interval with the all data missing. This is a commonly occurring real world scenario, when there is some data transmission failure, file over writing or data saving issue requiring univariate imputation. Also, such scenarios lead to large intervals of missing data for which the standard imputation techniques perform poorly.

The remainder of this paper is organized as follows. Section 2 focuses on the proposed imputation technique. Section 3 illustrates the experimental study and performance comparisons among various imputation schemes. Section 4 presents our concluding remarks.

2 Proposed Algorithm

The new algorithm entitled *Seasonal Moving Window Algorithm* (SMWA) is proposed mainly for large intervals of missing data, especially seasonal and cyclic data. The key aspect of our algorithm is that strong seasonality exists in almost all practical applications. Moreover, this seasonal behavior has to be considered as cyclic. There is no guarantee that the behavior of a system at a specific time is identical on two different days. Although it is very likely that the system behavior will be similar, regardless of the exact time. SMWA utilizes Seasonal and Trend decomposition using Loess (STL) [10] to decompose the data. Our proposed approach differs from other existing STL based technique [8] in how we handle imputation after performing STL decomposition: The decomposed trend component is linearly interpolated. The seasonal and remainder component is fitted with best pattern identified from the past available data. Then, the imputed decomposed data is recomposed to form the complete dataset.

SMWA initially identifies the missing interval as *missing data*. Then it selects a finite set of data before the missing interval as *head* and in a similar fashion chooses a finite set of data after the missing interval as *tail*. The combination of *head*, *missing data*, and *tail* forms the *window* as shown in Figure 1. This *window* then slides through the past data and the best matching window with the minimum root mean square error (RMSE) is imputed in the *missing data*.

The data preparation for SMWA imputation is explained in Algorithm 1. A univariate time series *ts* is given as the input dataset. The input time series is first validated for the presence of missing values. Then, the indexes of the missing data are identified.

To implement the algorithm, we first perform seasonal decomposition with `stl` [10]. As we require a complete dataset for `stl`, linear interpolation is performed as described in [11]. After the data is decomposed into seasonal, trend, and irregular components, the trend component is separated. Then,

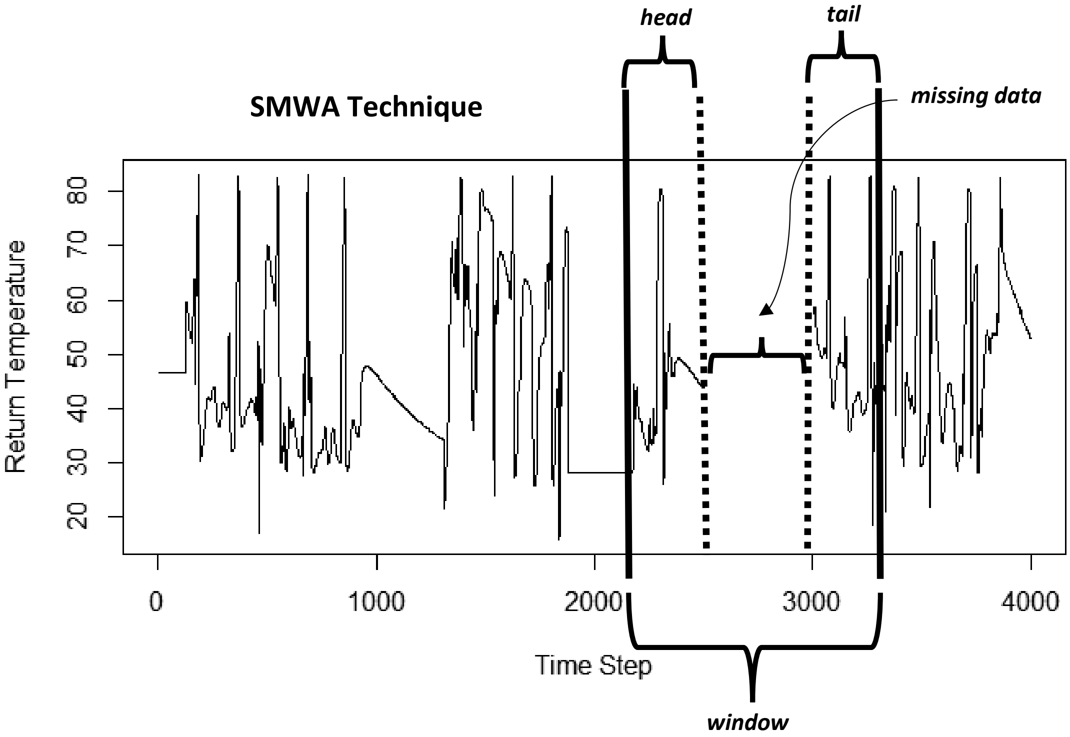

Figure 1: Pictorial representation of the SMWA Technique

SMWA is applied to the remaining component (i.e., seasonal and irregular) denoted as *SMWAInput*. In the separated trend component, missing values are filled in using linear interpolation as in [11]. Since the trend represents the long term rise or drop in data, simple interpolation is sufficient to fit the missing data in the trend component.

The SMWA imputation is explained in Algorithm 2. It uses *SMWAInput* obtained from Algorithm 1. Based on the length of data and the missing values a value for l (l=*head*=*tail*), which represents the common size of the dataset for both *head* and *tail*, has to be provided. The minimum length of the number of missing data samples g, for which the imputation has to be done, has to be provided. For smaller missing intervals, i.e., less than g missing samples, linear interpolation is performed. The user is also free to choose the maximum length of past window w to be considered for the evaluation. The default suggestions based on preliminary experiments are past window size w of $\frac{n}{3}$ and l of $\frac{w}{12}$ for smaller datasets, where n represents the length of the dataset.

Let $\text{freq}(ts)$ denote the frequency of the dataset. For larger datasets, e.g, $n > 100,000$, default suggestions are the past window w of size $\frac{n}{30}$ and l of $\frac{n}{(\text{freq}(ts))}$ for minutely data, for hourly data l of $\frac{n}{(60\times\text{freq}(ts))}$, and for other

Algorithm 1.: Data preparation for the Seasonal Moving Window Algorithm (SMWA)

```
Input: univariate Time Series $ts$

Validate the input data
Determine the indexes of missing samples: $index$
Perform linear interpolation on $ts$
Decompose interpolated $ts$ into $Seas$, $Trend$, $Irreg$ components with $stl$
Separate the $Trend$, fill in NA in $index$, perform linear interpolation
Add $Seas$, $Irreg$ as $SMWAInput$
Output: $SMWAInput$, $Trend$,
```

Algorithm 2.: Seasonal Moving Window Algorithm (SMWA)

```
Input:
    univariate Time Series $ts$
    $l$                  ▷ size of dataset for head and tail
    $g$                  ▷ minimum length of missing gap to be imputed
    $w$                  ▷ maximum length of past window
    $option$             ▷ accepts string head, tail, both
    $SMWAInput$ and $Trend$ from Algorithm 1

Calculate $n$ as length of $ts$
for $i$ in 1:$n$ do
    Identify the indices with missing intervals $\geq g$ in $ts$
for eachindex in indices do
    Identify the actual missing data in $ts$
    Formulate window as head + missing data + tail ▷ as per specifications in option
    for $j$ in 1:$w$ do
        Try{ ▷ try-catch to ensure if past window of specified length '$w$' exist before missing data
        Slide the window in the past by $j$ in $ts$ and calculate the RMSE for head and tail}
        Catch{
        Notify error }
        Find the best fitting past window with least RMSE for head and tail
    return The best fitting window
    $SMWAImputed \leftarrow$ Impute the values from the best fitting window in the missing data in $ts$
Output: $SMWAResult \leftarrow Trend + SMWAImputed$    ▷ Final SMWA imputed time series
```

frequencies l of $\frac{w}{10}$. Considering the computational time, we recommend this algorithm for fairly large g. This algorithm can be computed with either *head* only or *tail* only, or *both*. The performance of the method depends on the choice of tuning parameters l, g and w. The value of the input parameters depends on the length of the dataset and the percentage of the missing values and hence it might vary for each dataset. The window is formed with *head*, *missing data*, and *tail* for each of the missing intervals greater than g. Then we slide this window through past data, but no earlier than w steps before the window. The best matching past window with the smallest *root mean squared error* (RMSE) is identified and the values of this best match are imputed into the gap. Finally, the trend component is added back into the imputed dataset.

3 Experimental Study

In all our datasets, as the probability of missing data does not depend upon the observed or the unobserved data these are classified as *missing completely at random* (MCAR) [9]. As this algorithm is proposed mainly for large intervals of missing data, we examine this feature by removing relatively large data intervals. As the performance of the algorithms may depend on the position of missing values, the missing intervals were chosen randomly each time based on 30 different random seeds. The performance of the algorithm is evaluated with various test scenarios. For each test scenario the following steps are performed:

1. Load a complete time series *tsComplete*
2. Randomly remove values in *tsComplete* as per each scenario requirements and obtain *tswithNAs*
3. Apply an imputation algorithm to *tswithNAs* to get *tsImputed*
4. Compare *tsComplete* and *tsImputed* by using a suitable accuracy or error measure

3.1 Comparison Framework

To analyze the efficiency of the SMWA algorithm, its performance is compared with several state of the art imputation techniques which were implemented in the statistical programming language R. The RMSE was chosen as a performance measure, i.e.,

$$RMSE(z, z_{\text{imp}}) = \sqrt{\frac{\sum_{t=1}^{n}(z - z_{\text{imp}})^2}{n}} \tag{1}$$

where z_{imp} is the imputed value and z denotes the actual value of the time series. Mainly, methods from imputeTS [12], zoo [11], and forecast [8] packages in R are used for experiments as the described below:

- **Spline interpolation**: This method from the imputeTS package uses `na.interpolation` to implement the spline interpolation of missing values.
- **Seasonal decomposition**: This method is also from the imputeTS package. It initially separates the seasonal component from the time series, then performs imputation on the trend and irregular components and finally adds the seasonal component again. The method used is `na.seadec`. The algorithm internally uses mean imputation for nonseasonal series.
- **Seasonal split**: This is the third method from the imputeTS package. It splits the times series into seasons and then performs imputation separately for each of the season. The algorithm used is `na.seasplit`. The algorithm internally uses mean imputation for non seasonal series.
- **LOCF**: LOCF stands for last observation carried forward. It is a method from the zoo package. It replaces each missing value with the most recent non missing value prior to it. The algorithm used is `na.locf`.
- **Mean Imputation**: This method is from the zoo package. It fills the missing values with mean value of a time series using `na.aggregate`.
- **Linear interpolation**: It is a method from the zoo package. It replaces the missing values with interpolated values using `na.approx`.
- **Structural time series model**: It is a method from the zoo package. It fills missing values using seasonal Kalman filter using `na.StructTS`.
- **STL based interpolation**: It is a method from the forecast package, which uses linear interpolation for non-seasonal series and a periodic STL decomposition with seasonal series. The method used is `na.interp`.

- **Kalman smoothing**: This method is also from the imputeTS package. It performs Kalman smoothing using the state space representation of an ARIMA model for imputation. The method used is `na.kalman` with auto.arima model.

3.2 Case I: Large real-world data set

The proposed SMWA technique was evaluated for the *Return Temperature* dataset from GECCO Industrial Challenge 2015 which consists of 606,837 observations. The challenge provides separate missing and complete datasets. The algorithm is implemented for a minimum gap of $g = 200$. For the rest of the missing data linear interpolation is performed. The preceding 400 values form the *head* and after the missing gap and succeeding 400 values form the *tail* ($l = 400$). This *window*, formed with $head + missing\,data + tail$, is moved through the previous $w = 50,000$ values and the window with the best match according to RMSE is chosen.

The results for various algorithms tested are described in Table 1. A RMSE of 5.56 was achieved by SMWA when compared to the original test dataset. Although little improvement in RMSE was obtained for the proposed scheme when compared with linear interpolation, a closer look into the imputed intervals reveals the improvement made.

The best performing algorithms are visualized for some large missing intervals in Figures 2 and 3. They show the improvement achieved with SMWA imputation. From the plots it can be seen that SMWA outperforms other competing algorithms for large intervals of missing data. It also reveals the ability of SMWA in determining the best similar pattern as in the original underlying test dataset. By visual analysis, it can be seen that SMWA is often (but not always) able to reproduce the more complex patterns observed in the data. The competing methods mostly fail to do so. Still, such an imputed pattern will often be shifted forward or backward. Thus, the evaluation by RMSE may fail to give proper credit to the SMWA. In practice, the pattern is more desirable than a good value of RMSE, we therefore propose to use a more adequate error measure. For instance, an alignment-based error measure like the *Dynamic Time Warping distance* might be more adequate in these situtions [15].

Table 1 also shows that conventional imputation schemes like mean or locf imputation require less time for computation, but their RMSE is very high. The Kalman smoothing achieved the second best algorithm. But

Table 1: Comparison of RMSE and computation time. Results on the return temperature (GECCO challenge) dataset for various imputation techniques are shown. Smaller values are better. The structural time series model algorithm failed to complete the imputation process. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	15.59	**0.33**
Seasonal split	13.83	16.69
Seasonal decomposition	13.32	576.98
Spline interpolation	13.25	561.27
LOCF	8.81	11.28
STL based interpolation	6.74	72.02
Linear interpolation	5.60	11.55
SMWA imputation	**5.56**	143.89
Kalman smoothing	5.58	9134.15
Structural time series model	NA	NA

Kalman smoothing is computationally demanding (a factor of 60 compared to SMWA imputation). The structural time series model algorithm failed to complete the imputation process for the given data for more than 12 hours and hence the scheme was suspended and the result using this scheme is not discussed.

3.3 Case II: Selected smaller data sets

The algorithm is tested for its adaptability to various kinds of data patterns. Three different datasets that exhibit specific data patterns are considered. The SP dataset [13] (with trend but no seasonality), the Beersales dataset [13] (no trend but with seasonality), and the Air Passengers dataset [13] (with trend and seasonality).

Since the size of these datasets is very small, 10% of the datasets is removed in a single interval. This missing interval is determined randomly for 30 repeated experiments. The SMWA parameters l and w for three datasets are chosen as per default suggestions for smaller datasets.

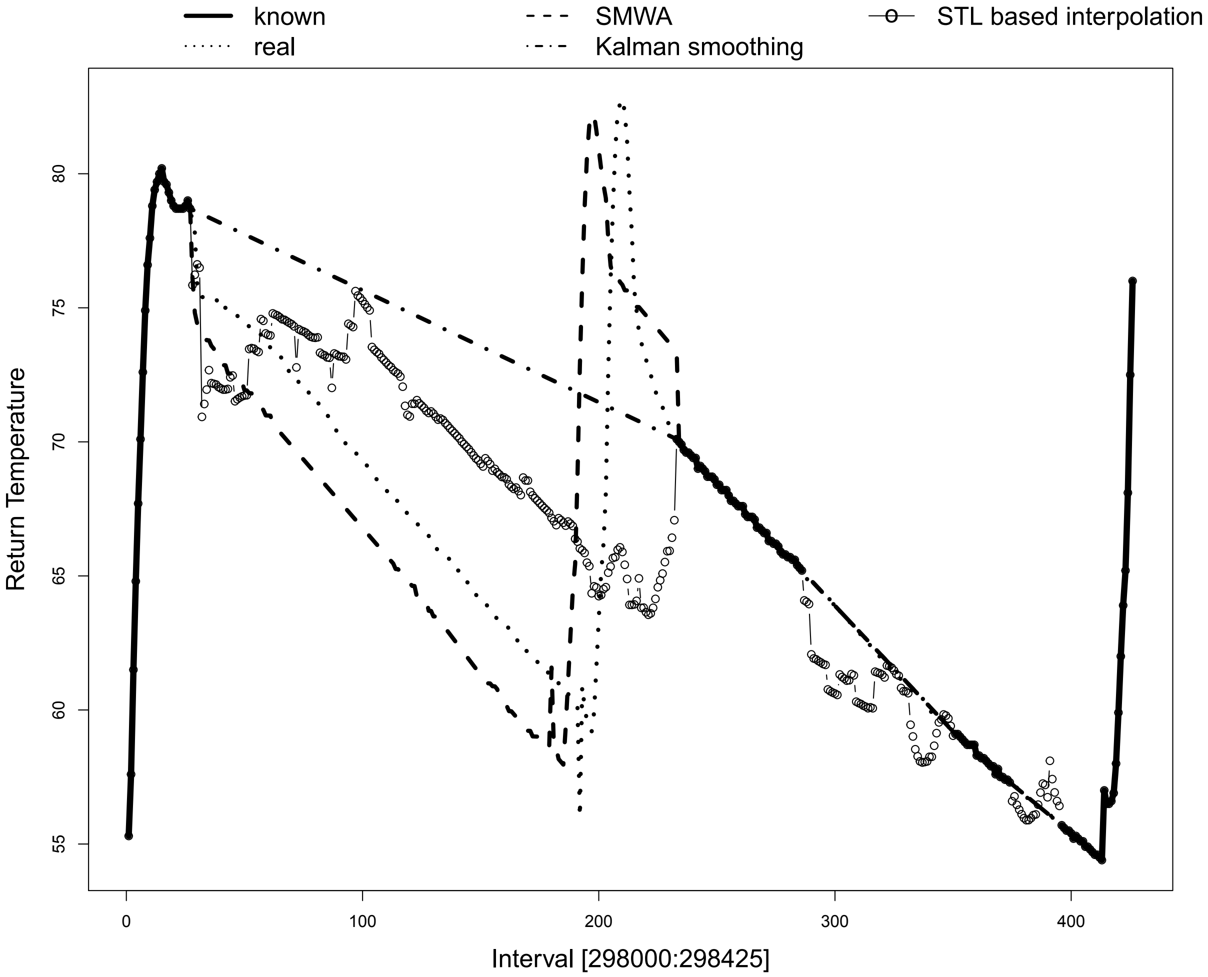

Figure 2: Comparison of the best performing algorithms for the Return Temperature dataset. The RMSE[298000:298425] for SMWA is 3.64, Kalman smoothing is 5.17, STL based interpolation is 4.12

3.3.1 Air Passengers dataset

The Air Passengers dataset contains 144 observations with the frequency of time series as 12. Table 2 reveals the mean performance metrics and computational time for the Air Passengers dataset. The box plot of RMSE obtained with 30 repeated experiments for various imputation techniques is shown in Figure 4. In the box plot each algorithm is represented with their abbreviated method names as explained in Section 3.1.

The algorithm Kalman smoothing turns out to be the best followed by STL based interpolation and SMWA. Though SMWA ranks in the third position, it is interesting to note that variance in mean RMSE as shown in the box plot in Figure 4 is less than for the STL based interpolation and

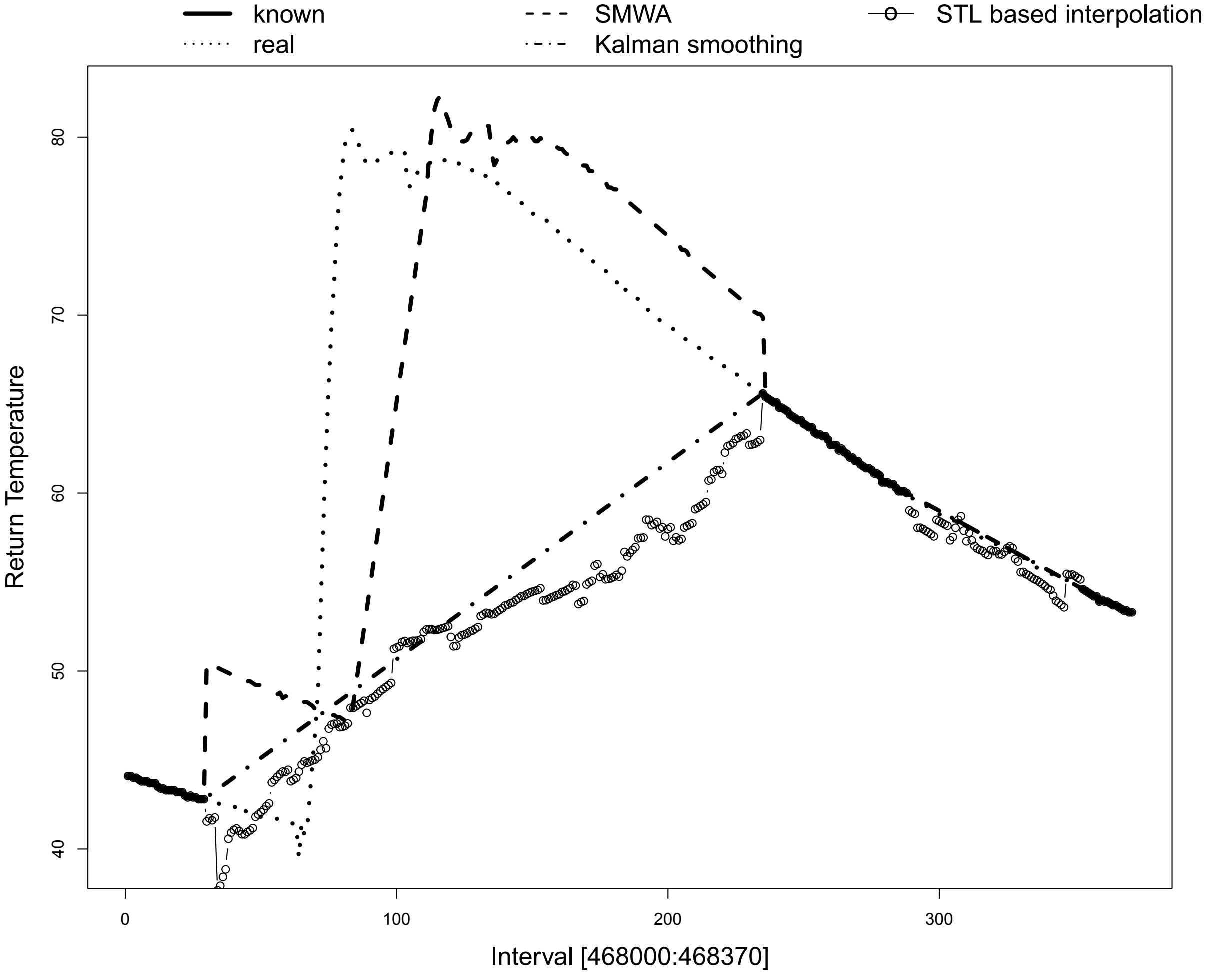

Figure 3: Comparison of the best performing algorithms for the Return Temperature dataset. The RMSE[468000:468370] for SMWA is 7.63, Kalman smoothing is 12.97, STL based interpolation is 13.95

Kalman smoothing. Though the dataset exhibits both trend and seasonality patterns, seasonal decomposition and seasonal split algorithms were able to secure fourth and fifth positions only. The remaining algorithms fail to perform well with relatively continuous missing data.

The Spline interpolation performs badly with very high mean RMSE and also with a very large outlier. The fastest running algorithm is simple the mean imputation and the Kalman smoothing takes the highest computational time.

Table 2: Comparison of RMSE (mean) & Computation time (mean) - Air Passengers dataset for various imputation techniques for 10% of missing data at random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	52.66	**0.002**
Seasonal split	40.50	0.01
Seasonal decomposition	39.59	0.02
Spline interpolation	99.18	0.008
LOCF	54.38	0.007
STL based interpolation	13.24	0.03
Linear interpolation	47.09	0.03
SMWA imputation	19.11	0.32
Structural time series model	60.82	0.55
Kalman smoothing	**9.75**	2.42

3.3.2 Beersales dataset

The Beersales dataset contains 192 observations with a time series of frequency 12. Since the Beersales dataset possesses very high seasonality, it is obvious that seasonal based algorithms like STL based interpolation, seasonal split, and seasonal decomposition would dominate the performance comparisons.

Table 3 shows the mean performance metrics and computational time for the Beersales dataset. The Kalman smoothing and STL based interpolation algorithm both perform the best, followed by seasonal split and seasonal decomposition. Though SMWA ranks in the middle, the performance is very close to the best algorithms. The box plot in Figure 5 shows that all these five algorithms perform better. The Spline interpolation again turns out to be the bad performing one with very high mean RMSE and Kalman smoothing takes the highest computational time.

3.3.3 SP dataset

The SP dataset contains 168 quarterly observations. For the SP dataset, which is the series with just trend and no seasonality, the linear interpolation

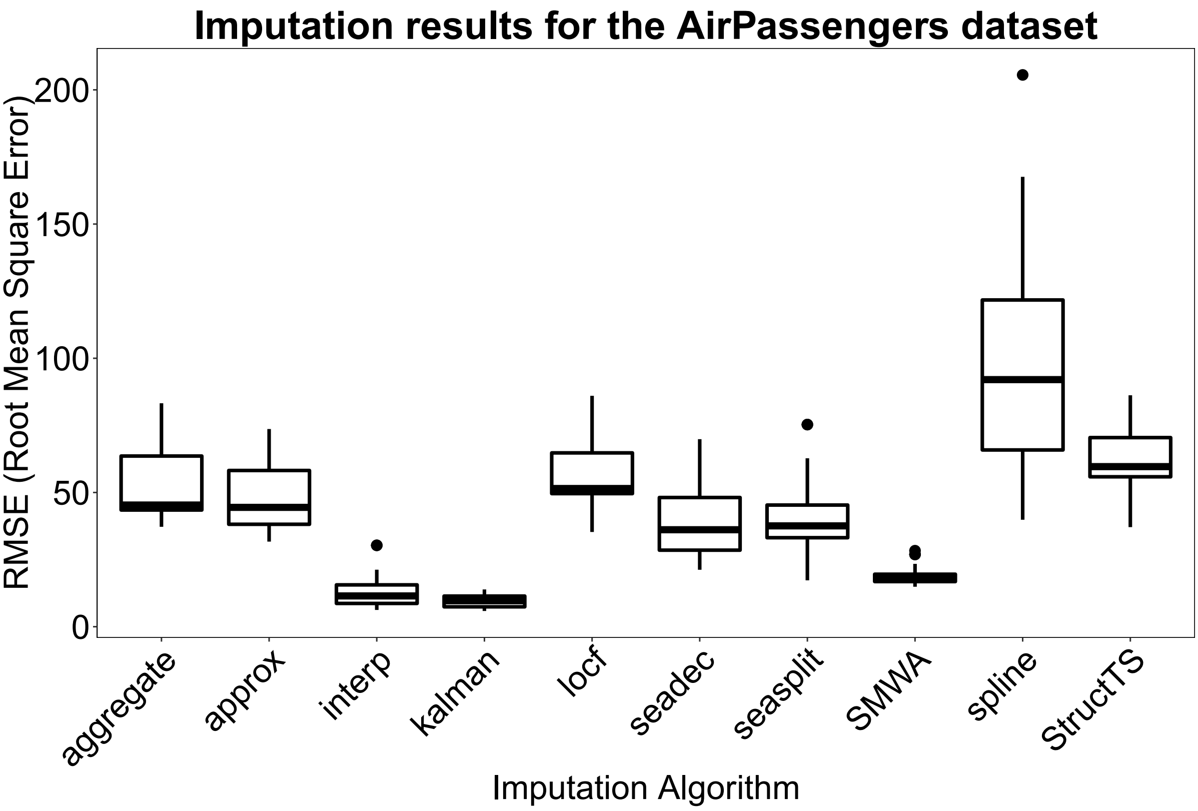

Figure 4: Imputation result for the Air Passengers dataset with 10% of missing data at random locations based on 30 different seeds

algorithm would do a good job. Table 4 lists the mean performance metrics and computational time for the SP dataset. The algorithms Kalman smoothing, STL based interpolation, linear interpolation, SMWA, and structural time series model turn out to be the best performing ones.

It is interesting to note that variance in mean RMSE for the SMWA algorithm is found to be the lowest among other algorithms as shown in the box plot in Figure 6. Since there is no seasonality in SP dataset, the seasonal split and seasonal decomposition algorithms perform just the mean imputation. All other algorithms fail to perform well with relatively continuous missing data. The locf imputation performs badly with very high mean RMSE and Kalman smoothing takes the highest computational time.

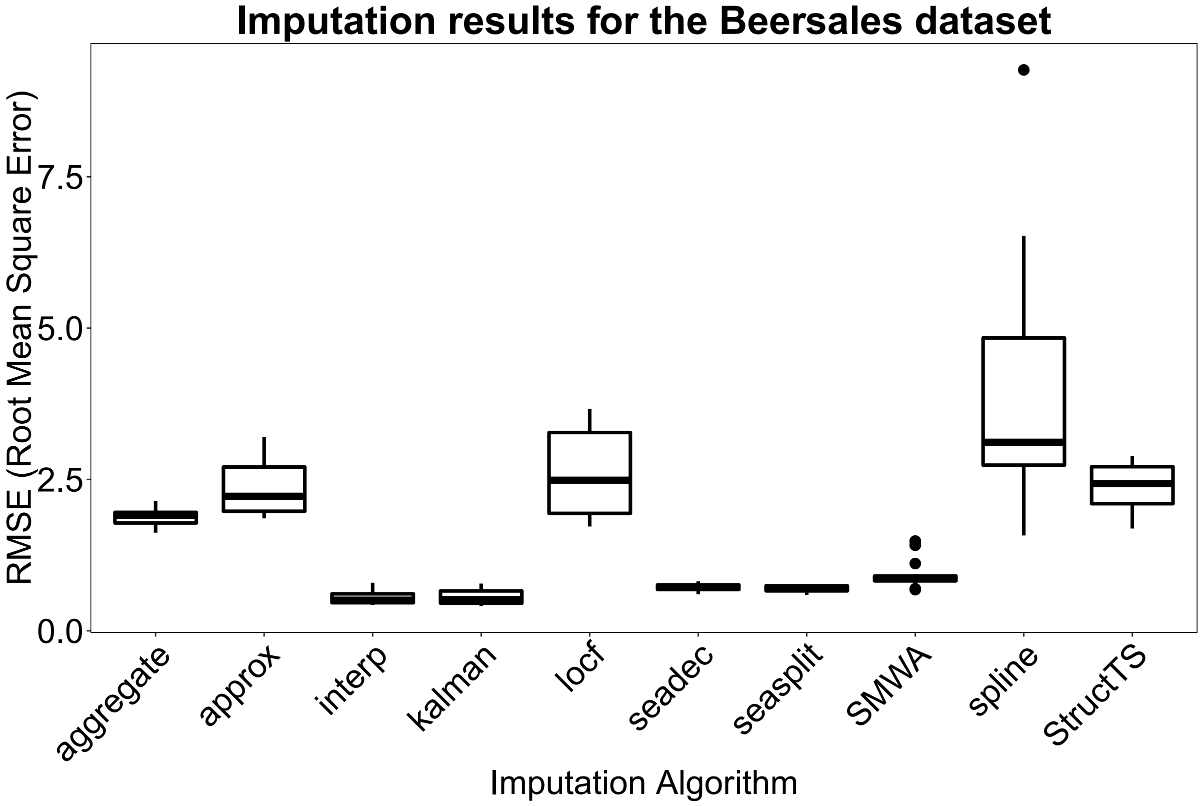

Figure 5: Imputation result for the Beersales dataset with 10% of missing data at random locations based on 30 different seeds

Table 3: Comparison of RMSE (mean) & Computation time (mean)- Beersales dataset for various imputation techniques for 10% of missing data at random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	1.88	**0.001**
Seasonal split	0.69	0.01
Seasonal decomposition	0.72	0.03
Spline interpolation	3.85	0.009
LOCF	2.61	0.008
STL based interpolation	**0.55**	0.05
Linear interpolation	2.33	0.012
SMWA imputation	0.93	0.42
Structural time series model	2.37	1.17
Kalman smoothing	**0.55**	4.61

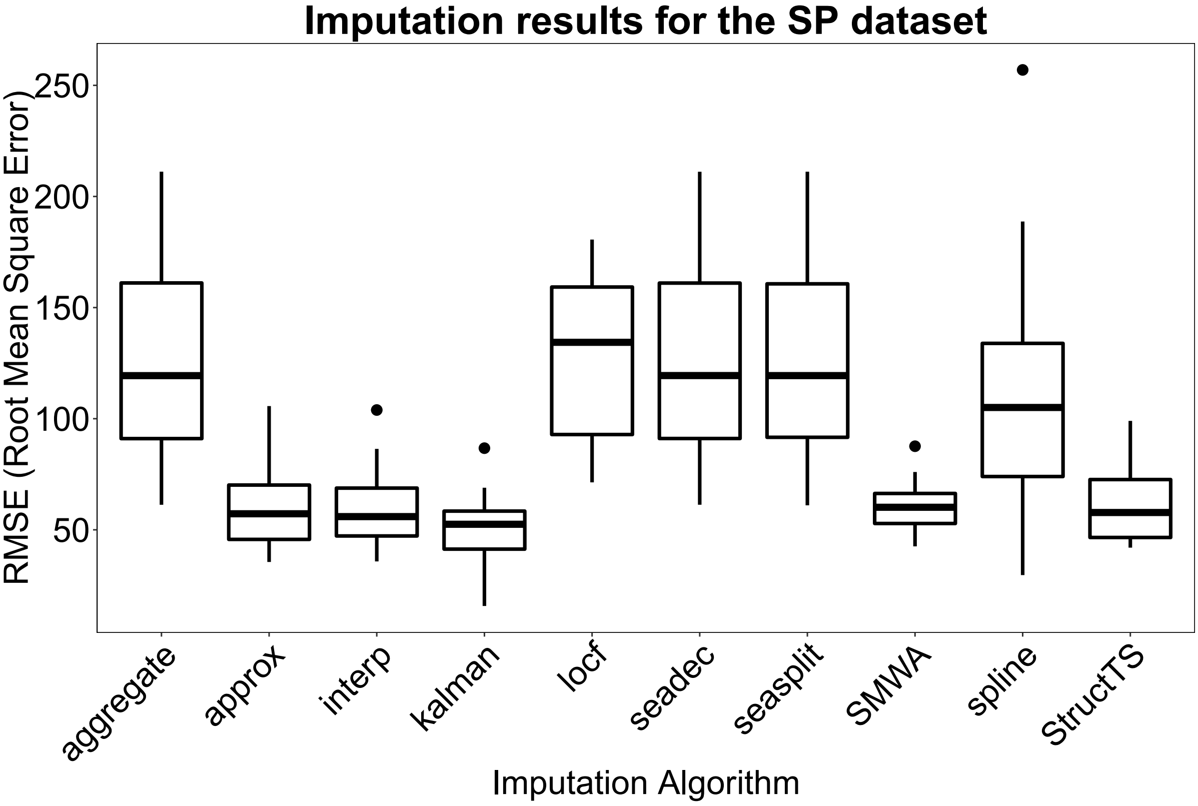

Figure 6: Imputation result for the SP dataset with 10% of missing data at random locations based on 30 different seeds

Table 4: Comparison of RMSE (mean) & Computation time (mean) - SP dataset for various imputation techniques for 10% of missing data at random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	124.16	**0.001**
Seasonal split	124.36	**0.001**
Seasonal decomposition	124.17	0.02
Spline interpolation	106.45	0.008
LOCF	129.55	0.009
STL based interpolation	59.76	0.03
Linear interpolation	59.89	0.01
SMWA imputation	60.40	0.34
Structural time series model	60.42	0.22
Kalman smoothing	**49.78**	1.01

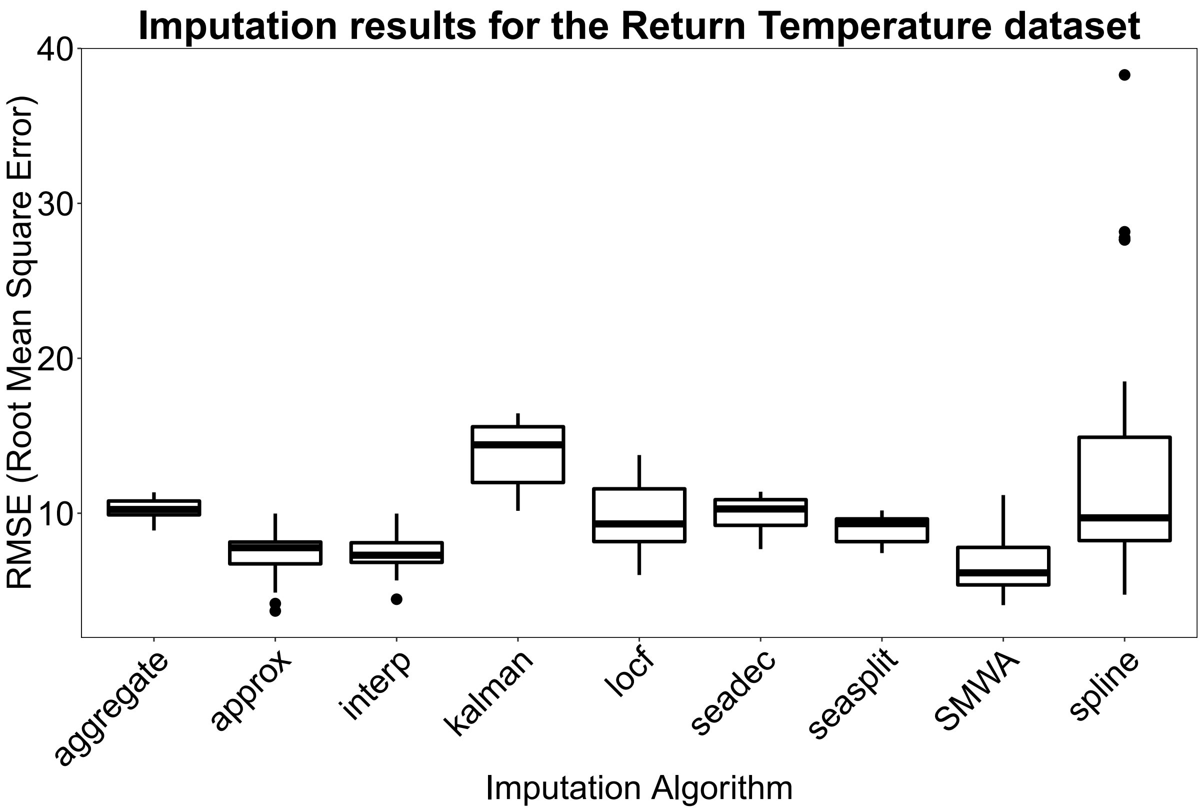

Figure 7: Imputation result for Return Temperature dataset with missing gap of size 100 at 10 random locations based on 30 different seeds

3.4 Case III: Large real-world data set with increased data gaps

In this test case, the complete *Return Temperature* dataset from the GECCO Industrial challenge is taken and 100 continuous data are removed randomly at 10 different locations for 30 runs. This particular scenario occurs in industries when data transmission or a sensor fails. The parameter settings for SMWA are $l = 100$, $g = 100$, and $w = 20,000$. Then SMWA is used to impute these large missing gaps along with other algorithms. The metrics in Table 5 clearly show the importance and performance of SMWA for very large industrial datasets. The SMWA bags the first position with a very low mean RMSE of 6.8 as in Figure 7. The Kalman Smoothing which served best in smaller data sets cannot perform well with larger intervals of missing data although it took very high computational time. Also, STL based interpolation, which showed better performance in smaller datasets ranks in second position following SMWA. It is to be noted that structural time series model is not evaluated with such large datasets for multiple runs as it takes very large processing time. In general, the performance of linear interpolation, STL based interpolation and SMWA are highly convincing

Table 5: Comparison of RMSE (mean) & Computation time (mean) - Return Temperature dataset for various imputation techniques with missing gap of size 100 at 10 random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	10.25	**0.02**
Seasonal split	9.02	1.45
Seasonal decomposition	9.97	3.25
Spline interpolation	13.47	1.49
LOCF	9.73	0.45
STL based interpolation	7.53	3.63
Linear interpolation	7.35	0.46
SMWA imputation	**6.8**	68.28
Kalman smoothing	13.87	4110.01
Structural time series model	NA	NA

in terms of RMSE with such large intervals of missing data. Though linear interpolation gives similar RMSE values compared to SMWA, they replace long gaps with a straight line, while SMWA tries to reproduce the similar complex pattern as in underlying real data. This pattern recovery combined with better accuracy and relatively less computational time makes the SMWA technique suitable for real time industrial data.

4 Conclusion

The SMWA is specially proposed for large intervals of missing data, which is a frequently occurring scenario in real industrial applications. The proposed SMWA combines good imputation accuracy with quick computational time. It focuses on extracting the best possible patterns from the available past data and utilizing it filling in the missing interval. The additional positive effect of using SMWA technique is for the ability of the algorithm to preserve the underlying real patterns better than other techniques.

It is also shown that the SMWA algorithm is well suited to work with various kinds of univariate datasets. Although algorithms like Kalman smoothing are highly robust for smaller datasets, they fail to perform well with large intervals of missing data. Also, due to very expensive computational time, Kalman smoothing cannot be used in practice for large industrial datasets.

Furthermore, the SMWA technique can be utilized to fit the missing data even from the available future time series. This can be easily done by running the algorithm with a reversed time series. Further improvement could be gained by approximating the results of both, past and future data. This algorithm can be easily used on top of any other imputation algorithm, especially for large missing intervals.

5 Acknowlegment

Parts of this work have been developed in the project '*IMProvT: Intelligente Messverfahren zur Prozessoptimierung von Trinkwasserbereitstellung und -verteilung*' (reference number: 03ET1387A). Kindly supported by the Federal Ministry of Economic Affairs and Energy of the Federal Republic of Germany.

References

[1] Imhoff, M., Bauer, M., Gather, U. and Löhlein, D. "Statistical pattern detection in univariate time series of intensive care on-line monitoring data." Intensive care medicine 24(12), pp.1305-1314. 1998.

[2] Momani, P.E.N.M. and Naill, M. "Time series analysis model for rainfall data in Jordan: Case study for using time series analysis." American Journal of Environmental Sciences, 5(5), p.599. 2009.

[3] Woolrich, M.W., Ripley, B.D., Brady, M. and Smith, S.M. "Temporal autocorrelation in univariate linear modeling of FMRI data." Neuroimage, 14(6), pp.1370-1386. 2001.

[4] Taylor, J.W. "A comparison of univariate time series methods for forecasting intraday arrivals at a call center." Management Science, 54(2), pp.253-265. 2008.

[5] Cuaresma, J.C., Hlouskova, J., Kossmeier, S. and Obersteiner, M. “Forecasting electricity spot-prices using linear univariate time series models.” American Journal of Environmental Sciences, 5(5), p.599. 2004.

[6] Stock, J.H. and Watson, M.W. “A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series.” (No. w6607). National Bureau of Economic Research. 1998.

[7] Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M. and Stork, J. “Comparison of different Methods for univariate Time Series Imputation in R.” arXiv preprint arXiv:1510.03924. 2015.

[8] Hyndman, R.J. and Khandakar, Y. “Automatic time series for forecasting: the forecast package for R (No. 6/07).” Monash University, Department of Econometrics and Business Statistics. 2007.

[9] Rubin, D.B. “ Inference and missing data.” Biometrika 63(3), 581-592. 1976.

[10] 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), 3-73. 1990.

[11] Zeileis, A. and Grothendieck, G. “zoo: S3 infrastructure for regular and irregular time series.” Journal of Statistical Software 14(1). 2005.

[12] Moritz, S. “imputeTS: Time Series Missing Value Imputation.” R package version 0.4. 2015.

[13] Chan, K.S., Ripley, B. “Tsa: Time series analysis.” R package version 1. 2012.

[14] Brown, M.L. and Kros, J.F. “The impact of missing data on data mining.” Data mining: Opportunities and challenges, 1, pp.174-198. 2003.

[15] Ratanamahatana, C.A. and Keogh, E. “Everything you know about dynamic time warping is wrong.” Third Workshop on Mining Temporal and Sequential Data. 2004.

Kontakt/Impressum

Diese Veröffentlichungen erscheinen im Rahmen der Schriftenreihe "CIplus". Alle Veröffentlichungen dieser Reihe können unter
`https://cos.bibl.th-koeln.de/home`
abgerufen werden.

Köln, Januar 2012

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.
Datum der Veröffentlichung: 14.11.2016

Herausgeber / Editorship

Prof. Dr. Thomas Bartz-Beielstein,
Prof. Dr. Wolfgang Konen,
Prof. Dr. Boris Naujoks,
Prof. Dr. Horst Stenzel
Institute of Computer Science,
Faculty of Computer Science and Engineering Science,
TH Köln,
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,
TH Köln,
Steinmüllerallee 1, 51643 Gummersbach
phone: +49 2261 8196 6391
url: `http://www.spotseven.de`
eMail: thomas.bartz-beielstein@th-koeln.de

ISSN (online) 2194-2870

Technology
Arts Sciences
TH Köln

Supported by:

Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

Grant No. 03ET1387A

Technology
Arts Sciences
TH Köln