tsdataleaks: An R Package to Detect Potential Data Leaks in Forecasting Competitions

26 December 2023

Summary

Forecasting competitions are of increasing importance as a means to learn best practices and gain knowledge. Data leakage is one of the most common issues that can often be found in competitions. Data leaks can happen when the training data contains information about the test data. There are a variety of different ways that data leaks can occur with time series data. For example: i) randomly chosen blocks of time series are concatenated to form a new time series; ii) scale-shifts; iii) repeating patterns in time series; iv) white noise is added to the original time series to form a new time series, etc. This work introduces a novel tool to detect these data leaks. The tsdataleaks package provides a simple and computationally efficient algorithm to exploit data leaks in time series data. This paper demonstrates the package design and its power to detect data leakages with an application to forecasting competition data.

Statement of Need

Time series forecasting competitions have played a significant role in the advancement of forecasting practices. Typically, in forecasting competitions, a collection of time series is given to the participants, and then the participants submit the forecasts for the required test period for each time series. During the competition period, only the training set for each time series is given to the public, and the test set is kept private from the public. Finally, competition organizers evaluate the forecast accuracy by comparing each participants submitted forecast values against the actual test period values. Participating in forecasting competitions not only aids in the identification of novel methods and facilitates their performance comparison against existing state-of-the-art forecasting techniques, as highlighted by Hyndman (2020), but also provides empirical evidence crucial for enhancing forecasting performance and advancing the theory and practice of forecasting (Makridakis, Spiliotis, and Assimakopoulos 2022).

Data leakage occurs when the training period of the time series includes test period data before officially releasing the test period of the time series. This idea is illustrated in Figure 1. A and B are two time series. The latter segment of the training set and the subsequent test set within the (B) series are the same as the red segment highlighted in the training segment inherent to series (A). This type of data leak could occur when randomly chosen blocks of time series are concatenated to form a new time series.

Competitions with data leaks will not be able to reach their original purpose. By exploiting data leakage, competitors can obtain a top rank in the leaderboard. Such models look highly accurate within the competition environment but become inaccurate when applied to a data set outside the competition environment. Hence, there is an increasing need to examine the potential data leaks in time series before the release of data to the public. The tsdataleaks package is designed to identify data leaks in time series.

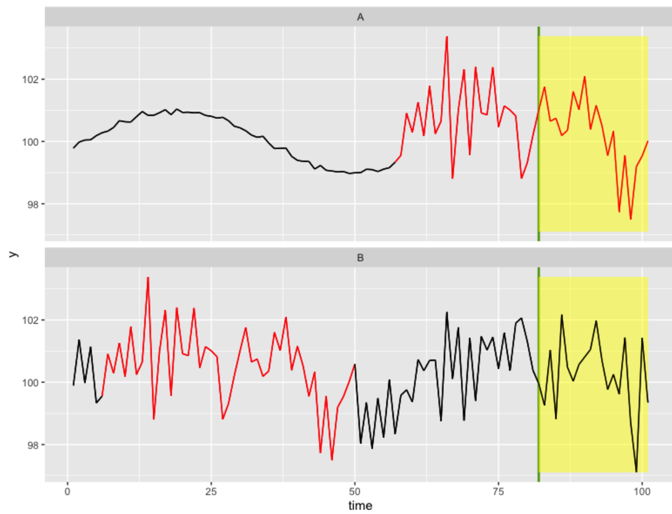

Figure 1: An example of a time series data leak. “A” and “B”” are two time series. The green verticle line and yellow background separates the training and test parts of the series. A training segment of series A (red colour segment) is the source of the latter segment of the training set and test set of the B series.

State of the Field in R

As of the latest information available on the Comprehensive R Archive Network (CRAN) Task View: Time Series Analysis (Rob J Hyndman 2023), there is no package available for detecting data leakages.

Algorithm

The algorithm operates as follows: it selects the final segment of the training portion from each time series in the collection, moves through all of the time series by one lag, and calculates the Pearson’s correlation coefficient. Hence, the input to the algorithm are: i) the time series collection, ii) segment length, and iii) cut of value for the correlation coefficient serve as the algorithm’s inputs. The algorithm returns the starting and end index of the segments that match each time series’ training part of the last segment.

Algorithm: Time Series Matching

Input:
1. *lst*: A collection of time series as a list.
2. *h*: Length of the segment to be considered.
3. *cutoff*: Cut-off value for the absolute value of the Pearson’s correlation coefficient.

Output:
A list containing starting and ending indices of segments that match each time series’ training part of the last segment.

Steps:
1. *Initialize* an empty list: *matching_segments*.
2. *Loop through each time series* in the *lst*:
a. Extract the final segment of the training portion with length *h*.
b. *Loop through the time series* with a step of one time point, considering each segment:
- Calculate the Pearson’s correlation coefficient between the extracted segment and the current segment.

Algorithm: Time Series Matching
- If the correlation coefficient is above the *cutoff*:
- Return the matching segments list with the starting and ending indices of the matching segments.
3. *Return* the matching segments list as the output.

Figure 2 illustrates the first iteration of the algorithm.

Figure 3 visualize the second iteration of the algorithm. At the second iteration correlation between the observation 2-7 and the purple segment is measured. Figure 3 illustrates an intermediate step of the algorithm.

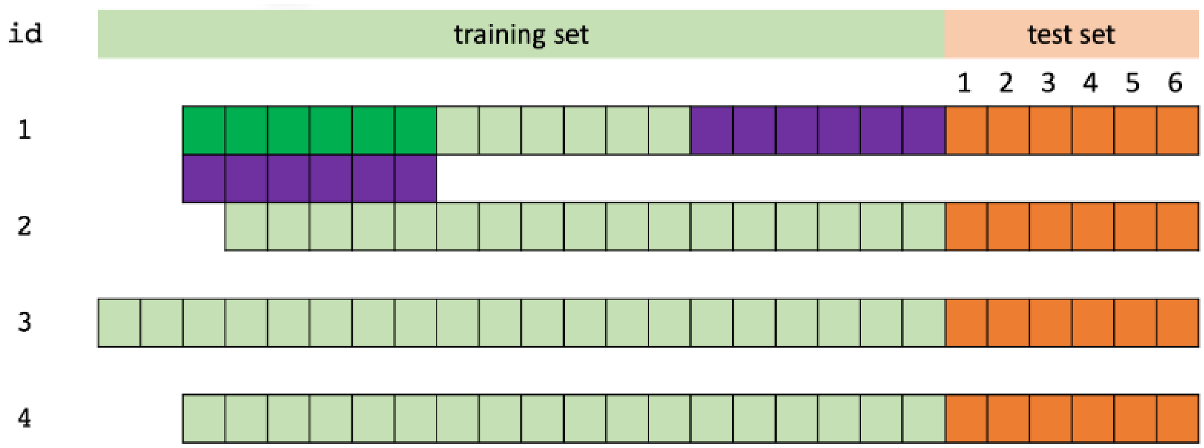

Figure 2: Visualization of the first iteration of the algorithm. The last segment of the training part of the first series is coloured in purple. As the first step of the algorithm it computes the Pearson's correlation coefficient between the observations 1-6 and the purple segment.

Figure 3: Visualization of the first iteration of the algorithm. The last segment of the training part of the first series is coloured in purple. As the first step of the algorithm it computes the Pearson's correlation coefficient between the observations 1-6 and the purple segment.

Figure 4: Intermediate step of the algorithm: Identification of potential data leak. Light purple colour section of the fourth series perfectly correlates with the last segment of the first series. Hence, red colour section of the fourth series could be the test part of the first series.

Usage

Installation

The package tsdataleaks is available on GitHub and can be installed and loaded into the R session using:

```
devtools::install_github("thiyangt/tsdataleaks")
library(tsdataleaks)
```

Functionality

There are three functions in the package: i) `find_dataleaks`, ii) `viz_dataleaks` and iii) `reason_dataleaks`. To demonstrate the package functions, I created a small data set with 4 time series.

```
set.seed(2024)
x <- rnorm(15)
lst <- list(
  x = x,
  y = c(rnorm(10), x[1:5]),
  z = c(rnorm(10), x[10:15]))
```

Following are the steps in detecting data leakages and visualize the results.

Step 1: The main function in the package is `find_dataleaks`. It exploits the data leakages according to the algorithm. The inputs to the function are list of time series collection (lst), length of the segment to be considered (h), and cutoff value for absolute value of the Pearson's correlation coefficient (cutoff).

```
f1 <- find_dataleaks(lstx = lst, h=5, cutoff=1)
```

Step 2: `viz_dataleaks` function visualize the results obtained in `find_dataleaks` for easy understanding as shown in **??**

```r
viz_dataleaks(f1)
```

Step 3: `reason_dataleaks` displays the reasons for data leaks and evaluate usefulness of data leaks towards the winning of the competition. The inputs to the function are list of time series collection (lst), length of the segment to be considered (h), output of the find_dataleaks function (finddataleaksout).

```r
reason_dataleaks(lstx = lst, finddataleaksout = f1, h=5)
```

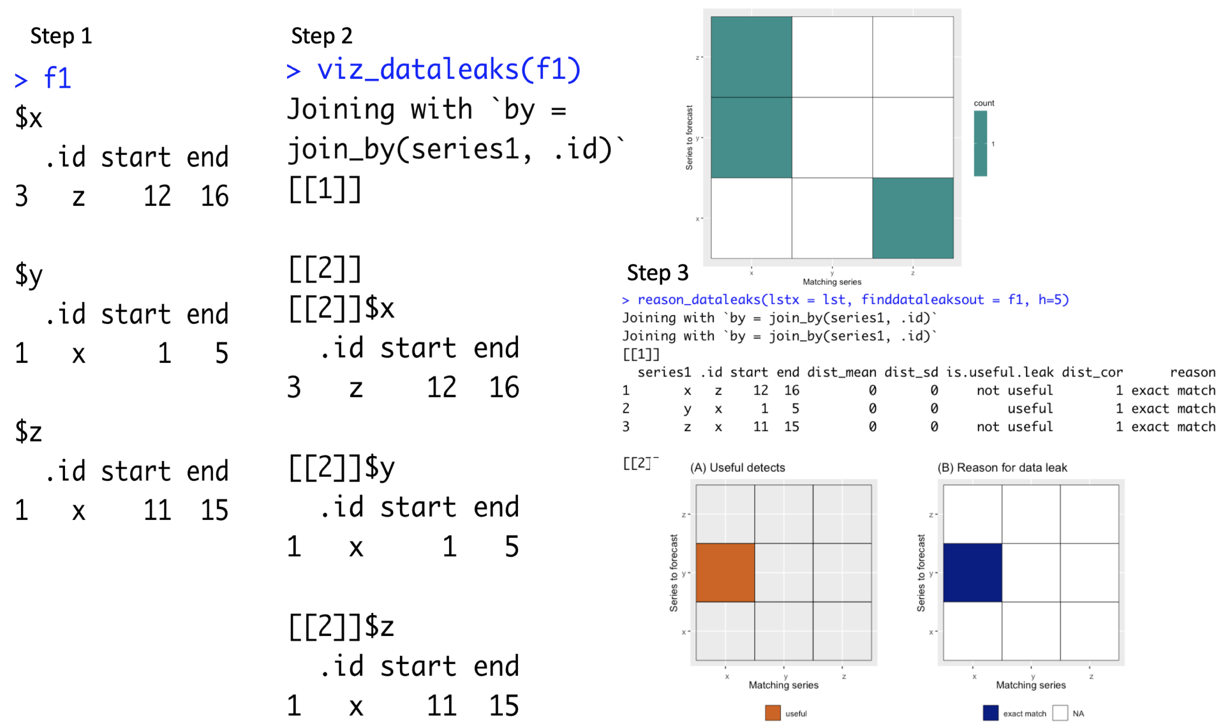

Figure 5: The text output of viz_dataleaks

For example, according to the 2nd row in the output, series b last part correlates with series a index 2 to 6. Hence, series `a` segment indices 7-12 can be the series b remaining part. Hence, this identification is an useful identification. Furthermore, according to the fourth row of the same output series b last part correlates with series c segment with indices 11-15. However, we do not have observations from 16 on wards for the series c. Hence, it is not a useful identification in winning the forecasting competition.

Appication to the M1 competition yearly time series data

Before applying find_dataleaks function all of the training sets of yearly series are stored into a list called `M1Y_x`. In the M1 competition, length of the test period for yearly series is 6. Hence, `h` value is selected as 6. The cutoff value for the absolute value of Pearson's correlation coefficient is 1.

```r
library(Mcomp)
data("M1")
M1Y <- subset(M1, "yearly")
M1Y_x <- lapply(M1Y, function(temp){temp$x})
m1y_f1 <- find_dataleaks(M1Y_x, h=6, cutoff = 1)
```

```r
viz_dataleaks(m1y_f1)
reason_dataleaks(M1Y_x, m1y_f1, h=6, ang=90)
```

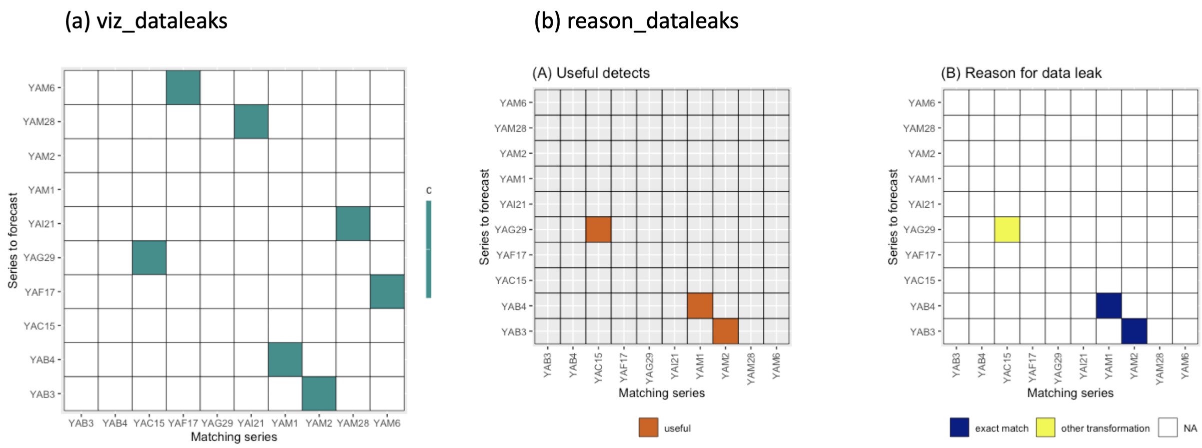

Figure 6: The text output of viz_dataleaks

Documentation and Examples

The outputs of the above code and application of other functionalities are available at package readme file at https://github.com/thiyangt/tsdataleaks

Conclusion

The new open source R package described in this paper enable, i) exploit data leakages, ii) identify the reasons for data leakage as exact match or add a constant, iii) determining whether the data leakages identified are useful in winning the forecast competition and iv) visualize the results. tsdataleaks is a valuable tool for competition Organizers to avoid data leakages , Competitors to detect data leakages, and participants alike, entire forecasting research community to evaluate quality of data.

Reproducibility

Codes to generate this manuscript is available at https://github.com/thiyangt/tsdataleaks

References

Hyndman, Rob J. 2020. "A Brief History of Forecasting Competitions." *International Journal of Forecasting* 36 (1): 7–14.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2022. "M5 Accuracy Competition: Results, Findings, and Conclusions." *International Journal of Forecasting* 38 (4): 1346–64.

Rob J Hyndman, Rebecca Killick. 2023. "CRAN Task View: Time Series Analysis." https://cran.r-project.org/web/views/TimeSeries.html.