

Creating Unbiased Public Benchmark Datasets with Data Leakage Prevention for Predictive Process Monitoring

Hans Weytjens and Jochen De Weerd

Research Centre for Information Systems Engineering (LIRIS),
KU Leuven, Leuven, Belgium

{hans.weytjens,jochen.deweerd}@kuleuven.be

Abstract. Advances in AI, and especially machine learning, are increasingly drawing research interest and efforts towards predictive process monitoring, the subfield of process mining (PM) that concerns predicting next events, process outcomes and remaining execution times. Unfortunately, researchers use a variety of datasets and ways to split them into training and test sets. The documentation of these preprocessing steps is not always complete. Consequently, research results are hard or even impossible to reproduce and to compare between papers. At times, the use of non-public domain knowledge further hampers the fair competition of ideas. Often the training and test sets are not completely separated, a data leakage problem particular to predictive process monitoring. Moreover, test sets usually suffer from bias in terms of both the mix of case durations and the number of running cases. These obstacles pose a challenge to the field’s progress. The contribution of this paper is to identify and demonstrate the importance of these obstacles and to propose preprocessing steps to arrive at unbiased benchmark datasets in a principled way, thus creating representative test sets without data leakage with the aim of levelling the playing field, promoting open science and contributing to more rapid progress in predictive process monitoring.

Keywords: Predictive Process Monitoring · Remaining Time Prediction · Bias · Benchmarking · Reproducibility · Datasets · Preprocessing

1 Introduction

Process mining analyzes event data logged by information systems with the goal of process discovery, process conformance checking and process enhancement. Predictive process monitoring is an important sub-field of process mining and concerns predicting next events, process outcomes and remaining execution times. The field is enjoying increased interest and is progressing thanks to opportunities offered by developments in AI and machine learning and the availability of data. Consequently, an increasing number of papers concerning predictive process monitoring are being published. Studying them, however, we identified three major obstacles to further progress in the field.

The first obstacle involves the use of different datasets that complicates or even impedes the comparison of results between papers. Even though many researchers use publicly available datasets, differences in their pruning and training/test set splits lead to significantly diverging results. Somewhat unfairly, some of these preprocessing decisions may have been guided by domain knowledge unavailable to all researchers. The second obstacle concerns the proper splitting of training and test sets. When using standard temporal splitting or cross-validation techniques, a number of cases will have prefixes in both the training and test set which is a form of data leakage and affects prediction performance. Most papers do not remedy this issue. For correct science, however, predictive process monitoring problems require a specific data split technique. The third obstacle relates to two forms of bias that are often ignored: when crudely determining start and end times for training and test sets, the number and average length of running cases at any given time may be greatly impacted at the chronological beginning and ending of these training and test sets.

To overcome these three obstacles, we believe predictive process monitoring needs benchmark datasets with predefined targets, training and test sets. With such datasets available, researchers could process the training sets at will while having to test their results on the given test sets. Benchmark datasets have always played an important role in major machine learning areas such as natural language¹, reinforcement learning², image recognition, etc. They allow the research community to compare different proposed methods and spur the development of new state-of-the-art ideas, often in the form of competitions. Well-known examples in the field of image recognition include –sorted by increasing size and complexity– MNIST³ (60,000 black-white handwritten digit images), CIFAR-10⁴ (60,000 color images in 10 classes) and ImageNet⁵ (14 million images in 20,000 classes), progressively used as the field matured.

The contribution of this paper is twofold. First, we want to expose the three obstacles mentioned above by investigating nine of the most commonly used public process mining datasets. Second, based on this analysis, we derive a set of proposals on how to design benchmark datasets for predictive process monitoring. These will ensure fairness, enable reproducibility and facilitate progress in the field. The code to implement our proposals is publicly available⁶.

The next section refers to influential process outcome prediction research and relates it to the three obstacles. We then detail our approach in Section 3 and describe our experimental setup in Section 4. We derive preprocessing steps that convert public datasets into unbiased benchmark datasets in Section 5. Using a simple convolutional neural network (CNN), Section 6 exposes the impact of dataset design decisions before concluding in Section 7.

¹ <https://gluebenchmark.com/>

² <https://arxiv.org/abs/2004.07219>

³ <http://yann.lecun.com/exdb/mnist/>

⁴ <https://www.cs.toronto.edu/~kriz/cifar.html>

⁵ <https://image-net.org/>

⁶ <https://github.com/hansweytjens/predictive-process-monitoring-benchmarks>

2 Related work

When exploring some of the most influential original papers and surveys on outcome [1, 2], next event [5, 6, 8] or remaining time [3, 5, 7] prediction, the first obstacle of poor comparability becomes immediately apparent. Even where researchers work with the same datasets, direct comparisons of their results prove impossible. For example, the classes for the BPIC 2011 and Traffic datasets are defined differently by Teinemaa et al. [1] and Kratsch et al. [2]. In next event prediction, Tax et al. [5] consider only events of type “complete” whilst Evermann et al. [6] create new features by concatenating the “activity” and “resource” features. In the remaining time prediction problem, Tax et al. [5] and Verenich et al. [3] use different ratios (67%/33% and 80%/20% respectively) when chronologically splitting training and test sets, whereas Polato et al. [7] use fivefold cross-validation. In their literature overview study, Neu et al. [4] acknowledge the latter problem and propose to “use a unified evaluation approach for further publications.” Other than Teinemaa et al. [1] largely solving and Verenich et al. [3] recognizing (but also accepting) the data leakage problem when naively splitting training and test sets, we found no mention of the second obstacle in these papers. To the best of our knowledge, no research diagnoses the third obstacle of bias.

3 Approach

To counter the first obstacle’s lack of comparability, we will formulate a standardized preprocessing approach to construct benchmark datasets in Section 5. Along the steps in this preprocessing procedure, we will observe and remedy both the second obstacle of data leakage between test and training sets and the third obstacle of bias as they appear. We will use nine datasets to analyze the problems and illustrate the preprocessing steps and their effects on prediction. Before doing so, we first define the four main guiding principles that steer this work:

- **A principled approach:** increases transparency and enables reproducibility to other datasets / problems,
- **Proximity to real-life:** creates datasets that resemble the original underlying business problems as much as possible,
- **Enabling of good science:** (only) where required, we deviate from the first two principles for the benefit of science,
- **Domain agnosticism:** as domain knowledge is often sparse and unevenly distributed amongst researchers, we refrain from using it altogether.

We demonstrate how different preprocessing impacts prediction results. It is rather unconventional to illustrate the importance of the problem towards the end of a paper; however, in our case, it offers the benefit of having clearly-defined preprocessing steps that each generate different variants of the original datasets on which we ran a simple convolutional network to predict remaining times.

4 Experimental setup

4.1 Preliminaries

In predictive process monitoring, datasets are event logs describing processes, often called *cases*. These cases consist of *events*. A number of *features* describe these cases and events. Every event is also associated with a *target* feature. A *prefix* is an ongoing case, with the *prefix length* its number of completed events. The learning problem is to train an algorithm on a training dataset containing events, described by their features and organized in prefixes labelled with targets, with the goal of predicting the targets of unseen prefixes.

We distinguish two phases in the preprocessing process. We will restrict the use of the term *preprocessing* to refer to the first phase steps that are only performed once and lead to the construction of unique benchmark training and test sets. The second phase involves further steps, such as feature selection and standardisation taken by individual researchers to prepare these fixed training and test sets to train and test their models. This latter phase is not subject of this paper and belongs to the area where researchers and ideas compete.

4.2 Datasets

We used five publicly available datasets from the 4TU.ResearchData repository. BPIC_2012⁷ contains logs of a loan application process at a Dutch bank. BPIC_2015⁸ is a collection of building permit applications in five Dutch municipalities, which we concatenated into one dataset. BPIC_2017⁹ is a richer, cleaner and larger set of logs of a Dutch bank loan application process. BPIC_2019¹⁰, while comparable in size, has much shorter cases and concerns a purchase order handling process. BPIC_2020¹¹ is a collection of five smaller datasets related to travel administration at a university. The five subsets are records of processes covering domestic declaration documents (Domestic Declarations), international declaration documents (Intl. Declarations), pre-paid travel costs and requests for payment (Payments), travel permits (Permits) and expense claims (Travel Costs). Our target for all of these datasets was defined as the fractional number of days until case completion. Table 1 provides some key statistics of these datasets.

4.3 Models

To illustrate the relevance of the preprocessing choices to construct datasets at the hand of the remaining time prediction problem, we calculated the mean absolute error (MAE) of a CNN model on various preprocessing versions of the datasets. For each dataset, we included one categorical (“activity”) and one range (“elapsed”: time since case start) feature. We used embedding to feed the

⁷ https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204

⁸ https://data.4tu.nl/collections/BPI_Challenge_2015/5065424

⁹ https://data.4tu.nl/articles/dataset/BPI_Challenge_2017/12696884

¹⁰ https://data.4tu.nl/articles/dataset/BPI_Challenge_2019/12715853

¹¹ https://data.4tu.nl/collections/BPI_Challenge_2020/5065541

Table 1. Statistics of the used datasets*.

Dataset	Nr. cases	Nr. events	Median nr. events	Avg. nr. events	Median nr. days	Mean nr. days	Max nr. days	Dataset nr. days
BPIC_2012	12,183	228,873	9	18.8	0.5	7.8	137.2	152
BPIC_2015	5,641	262,328	45	46.5	68.4	100.8	1,512	1,617
BPIC_2017	31,497	1,201,390	35	38.1	19.1	21.9	286.1	397
BPIC_2019	251,465	1,587,810	5	6.3	64.0	69.5	380.0	830
BPIC_2020:								
Domestic. Declarations	6,087	56,359	5	9.3	14.3	101.7	735.2	773
Intl. Declarations	1,440	71,735	13	49.8	416.1	382.9	781.7	1,010
Payments	6,593	35,046	5	5.3	7.9	11.1	1,238	711
Permits	7,063	86,560	11	12.3	71.7	87.1	502.4	1121
Travel Costs	2,097	18,238	8	8.7	24.0	36.7	325.0	749

*Outliers already removed as per Subsection 5.2.

categorical feature in the models and reduced its dimensionality to the square root. Two convolutional layers followed the embedding layer, each with 40 kernels of size three and a max-pooling operation with a window of size three. The stride size for both the kernels and pooling was one. Two dense layers of sizes 100 and ten and with a ReLu activation completed the architecture. We reduced overfitting with early stopping using a validation set of the 20% latest-starting cases in the training sets and a patience of 30. The sequence length was ten, the batch size 2048.

5 Constructing benchmark datasets

This section applies to remaining time prediction and classification problems. There are some important differences with next event predictions that will be covered in subsection 5.8.

5.1 Target definition

Remaining time targets are defined as the remaining time between an event’s time stamp and the case’s last time stamp. For classification tasks, however, determining targets requires weakening the domain agnosticism principle and recourse to an authority for a final decision.

5.2 Elimination of chronological outliers and duplicates

The BPIC_2015 dataset contains cases starting in the period 2009-2015. As is shown in the top row in Fig. 1, the cases in the year 2009 appear aberrant as they are few in number and last much longer than cases in the following years. When zooming in on cases starting before 2011-01-31st (bottom row), we see that the log starts behaving “normally” as of October 2010. In the following, we only included cases starting after Oct 1st, 2010. We found similar outliers in BPIC_2019.

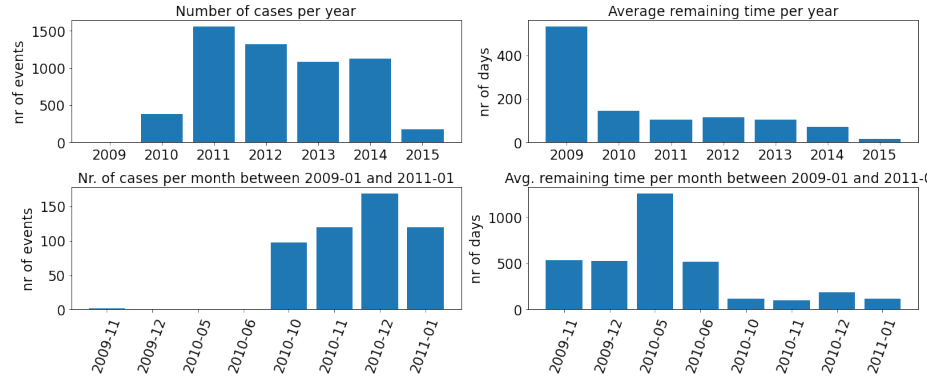


Fig. 1. BPIC_2015 contains remote, faulty outliers at the beginning of the dataset.

Similar patterns arose for the case’s ending times at the end of all datasets involved. To remove those outliers, we imposed new end dates. An overview can be found in Table 2.

Table 2. Start and end dates before and after eliminating outliers.

Dataset	Original dataset		After removing outliers	
	First event	Last event	Start in/after	End before/in
BPIC_2012	2011-10	2012-03	-	2012-02
BPIC_2015	2009-11	2015-07	2010-10	-
BPIC_2017	2016-01	2017-02	-	2017-01
BPIC_2019	1948-01	2020-04	2018-01	2019-02
BPIC_2020	-	-	-	-
Domestic Declarations	2017-01	2019-06	-	2019-02
Intl. Declarations	2016-10	2020-05	-	2019-07
Payments	2017-01	2019-08	-	2018-12
Permits	2016-10	2021-08	-	2019-10
Travel Costs	2017-01	2019-02	-	2019-01

A dataset may also contain faulty outliers. These are not detectable without domain knowledge and therefore we did not attempt to remove them. The elimination of duplicates requires no further discussion. None were found in our datasets. For clarity, all tables and graphs (outside of this subsection) in this paper, including the preceding Table 1, rely on datasets without chronological outliers.

5.3 Debiasing the end of the dataset

In order to know their targets, a dataset should only contain completed cases. Without domain knowledge, however, we cannot be 100% certain which cases are completed. We therefore drop the potentially incomplete cases that start in the yellow zone (both light and dark maroon) in the artificial example of

Fig. 2. To be on the safe side, the zone’s width equals the duration of the dataset’s longest case. This introduces two forms of bias towards the end of the dataset (third obstacle). First, the black-grey cases that end in the yellow zone have a longer average duration than the grey cases ending before the yellow zone. Moreover, the number of running cases in the yellow zone (black prefixes) no longer reflects the underlying reality, which is the second form of bias. It prohibits the use of inter-case variables to compute for example the load of a resource dealing with the cases. To debias the ending of the dataset, we must reject the (black) *prefixes* ending in the yellow zone. *Cases* for which we removed the longer (black) prefixes, still remain in the dataset with their shorter (grey) prefixes whose targets are known.

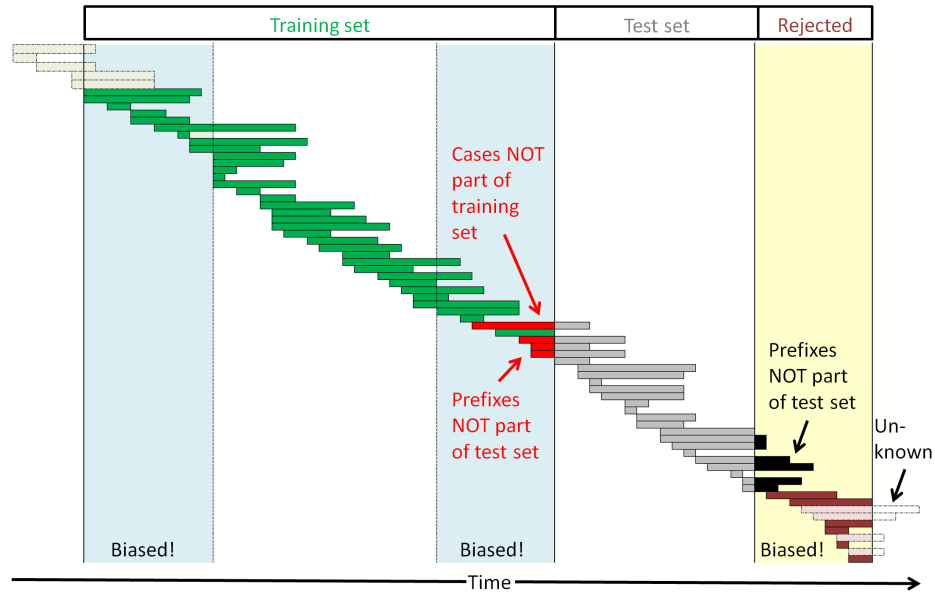


Fig. 2. An artificial dataset with cases ranked by starting time.

Note that one (possibly faulty) long case in the dataset could suffice to severely reduce the training set size or even annihilate it when debiasing the dataset’s end. This problem will be tackled below in Subsection 5.7.

5.4 Choice of test set

Some authors (e.g. [6, 7]) use cross-validation for predictions (not just whilst training) to make their models robust against concept drift. However, some of that concept drift can be attributed to the end-of-dataset bias we described and remedied above. Moreover, concept drift happens in real-life situations too. It may be even learnable to some extent. As the strict separation of training and test sets (second obstacle) is also more difficult to implement in a cross-validation setting, we recommend using a classic temporal training/test split for

all of these reasons. Larger test sets are more likely to be representative for the whole dataset, but will reduce the size of the corresponding training set and carry a higher risk of real concept drift. As a compromise, we propose that test sets consist of the last 20% of the datasets’ cases chronologically.

5.5 Temporal splitting

To isolate a test set, many authors rank cases by the timestamp of their first event and determine a separation time, with the $x\%$ last cases starting after that time going to the test set and the remainder constituting the training set. From a real-life perspective, the logical flaw is that the outcome (remaining time or class) of incomplete cases in the training set (red/grey in Fig. 2) is not known at that separation time. Moreover, some cases from the training set and test will be running concurrently and influencing each other. This is the second obstacle mentioned in the introduction. To counter it, we recommend to always apply “strict temporal splitting” by only retaining in the training set those cases that are completed before the separation time. These are colored in green in Fig. 2. As a result, the training set will contain two biased zones (light blue), at its beginning and end. It is left to researchers to debias these zones as they see fit. A compelling reason to make this imperative is not present. Depending on the duration of the cases in relation to the time span covered by the dataset, the difference between strict and regular temporal splitting can be rather significant, as demonstrated in Fig. 3.

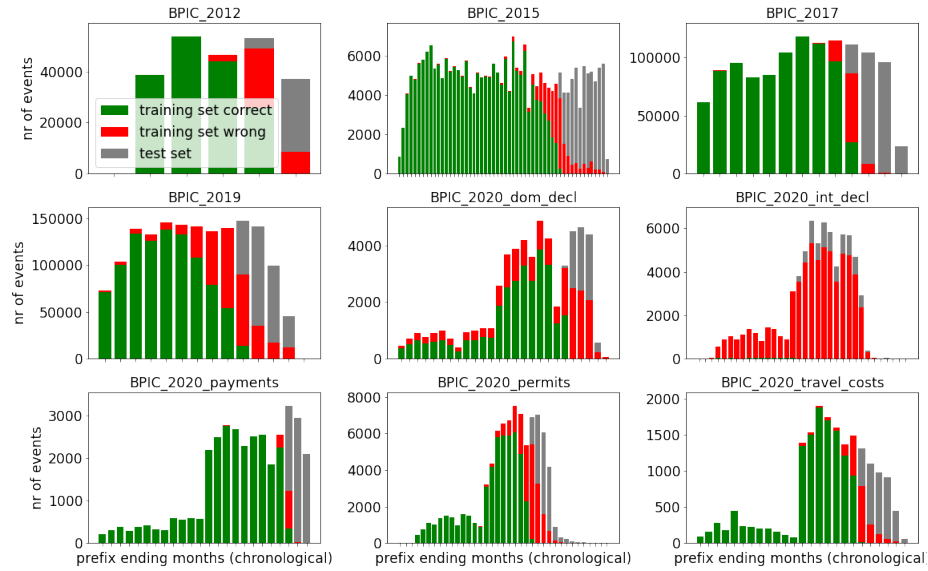


Fig. 3. Grey bars represent test set events (20%). Green bars are events in training set cases ending before the separation time (strict temporal splitting), red ones belong to cases ending after that separation time (regular temporal splitting). As debiasing was not done to retain sufficient samples, the bias at the datasets’ ends is clearly visible.

5.6 Debiasing the beginning of the test set

We now face both obstacle three biases at the beginning of the test set. Of the (red-grey) cases that start before the separation time but end after it, we also include in the test set those (grey) *prefixes* that end after the separation time. This restores balance both in terms of the case lengths and the number of running cases. It also better resembles real-life settings where a maximum of available data at any given time would be considered.

5.7 Removal of long cases

Applying our third guideline, we suggest discarding extremely long cases from the dataset in the interest of the dataset’s survival when removing biases at the end of the dataset as described in Subsection 5.3. This clearly changes the nature of the original dataset, but we believe that leaving the biases in would have a more negative impact. Fig. 4 plots the size of the training and test sets (in number of events) constructed as described above for different maximum case durations. Keeping the longest cases would drastically, often fatally shrink the datasets as can be seen on the right side of the graphs. Some degree of pragmatism is called for: we propose removing up to, but no more than, 5% of the longest-lasting cases to find the case duration within the window that corresponds with the largest training set (in number of cases) as shown by the red lines in Fig. 4. We, thus, obtain the training and test sets in Tables 3 and 4. The (quasi) disappearance of the training sets for BPIC_2020 Domestic and Intl. Declarations disqualifies them for serious research. Of course, steps 5.3 - 5.6 have to be carried out after removing the longest cases. We only discussed the removal last, as it assumes an understanding of these same steps 5.3 - 5.6.

5.8 Next event prediction

As the targets in the next event prediction problem do not depend on the knowledge of cases’ final events, subsections 5.3 - 5.7 no longer apply: we can work with prefixes rather than completed cases. After removing the chronological outliers, it suffices to separate training and test sets by chronologically sorting all remaining events in the log and determining the 80%/20% separation time. All prefixes (not cases) ending before that separation time belong in the training set, and those ending after it constitute the test set. Viewed on Fig. 2, this means we retain all cases in the yellow zone and keep the red prefixes in the training set.

6 Model predictions

With the example of the remaining time prediction problem, Fig. 5 visualizes the impact of the preprocessing decisions on predictions. For every dataset, we ran seven simulations. The first simulation concerned a base case with a train/test

split of the first 90% and last 10% starting cases. In the following six simulations, we successively introduced the measures from Section 5. The maximal durations used in the last simulation correspond to the red vertical lines in Fig. 4. The resulting quality of the predictions from the CNN model, expressed as MAE, is represented by the black lines (left vertical axis). The obtained sizes of training (green bars) and test set (grey bars) are calibrated to the right vertical axis.

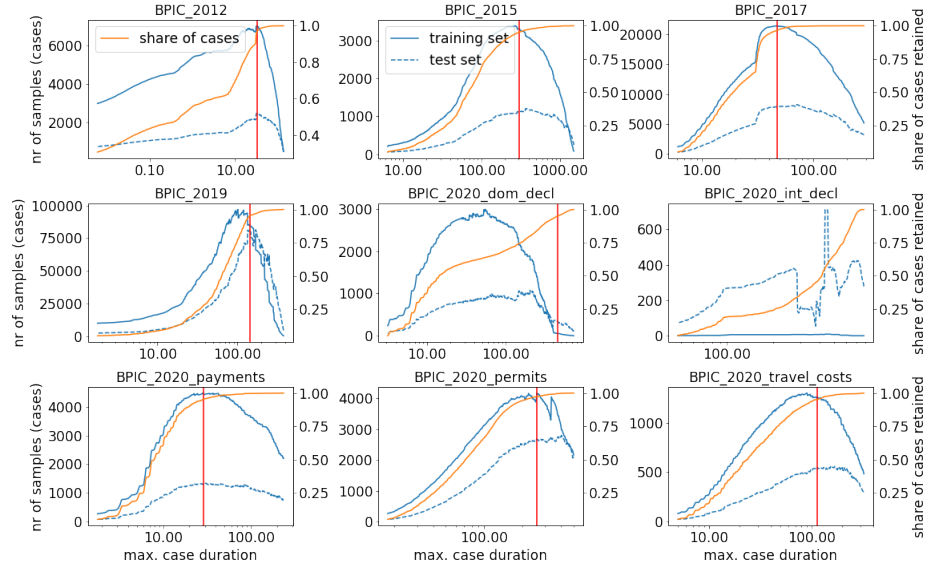


Fig. 4. Size of training and test sets in function of maximal case lengths. Debiasing and strict temporal splitting applied. Vertical red lines show our proposals.

Table 3. Proposed benchmark datasets* (test set 20%): Timing.

Dataset	Original datasets*				Proposed datasets			
	Max. duration cases	Data-set nr. days	Data-set start	Data-set end	Max. duration cases	Data-set nr. days	Test set start	Test (= Data-) set end
BPIC_2012	137.2	152	2011-10	2012-02	32.3	119.7	2012-01-05	2012-01-28
BPIC_2015	1,512	1,617	2010-10	2015-03	302.8	1,314.2	2013-06-10	2014-05-10
BPIC_2017	286.1	397	2016-01	2017-01	47.8	348.5	2016-10-10	2016-12-14
BPIC_2019	380.0	830	2018-01	2019-02	143.3	253.6	2018-07-19	2018-09-11
BPIC_2020:								
Payments	1,238	711	2017-01	2018-12	28.9	682.2	2018-09-14	2018-11-22
Permits	502.4	1,121	2016-10	2019-10	258.8	806.2	2018-09-07	2018-12-20
Travel Costs	325.0	749	2017-01	2019-01	114.3	634.0	2018-07-06	2018-10-06

*Outliers removed as per Subsection 5.2.

Table 4. Proposed benchmark datasets* (test set 20%). For the test sets, the total number of cases are listed first, followed by cases at the beginning of the test set missing some of their shorter prefixes (red/grey in Fig. 2), cases at the end of the test set missing their longer prefixes (grey/black in Fig. 2) and cases with all prefixes present in the test sets (grey in Fig. 2). The last column is the total number of cases in the dataset, but with the average of both kinds of incomplete test set cases rather than their sum.

Dataset	Number of cases							
	Original dataset	Training set	Test set				Total dataset	
			all	missing short prefixes	missing long prefixes	complete	all	full case equivalent
BPIC_2012	12,183	7,019	2,468	570	621	1,340	9,487	8,955
BPIC_2015	5,641	3,311	1,116	229	221	666	4,427	4,202
BPIC_2017	31,497	21,404	7,902	2,040	1,754	4,108	29,306	27,409
BPIC_2019	251,465	84,233	86,905	52,676	46,423	5,327	171,138	139,110
BPIC_2020:								
Payments	6,593	4,494	1,293	135	171	987	5,787	5,634
Permits	7,063	4,168	2,679	1,309	831	667	6,847	5,905
Travel Costs	2,097	1,265	533	173	100	260	1,798	1,662

*Outliers removed as per Subsection 5.2.

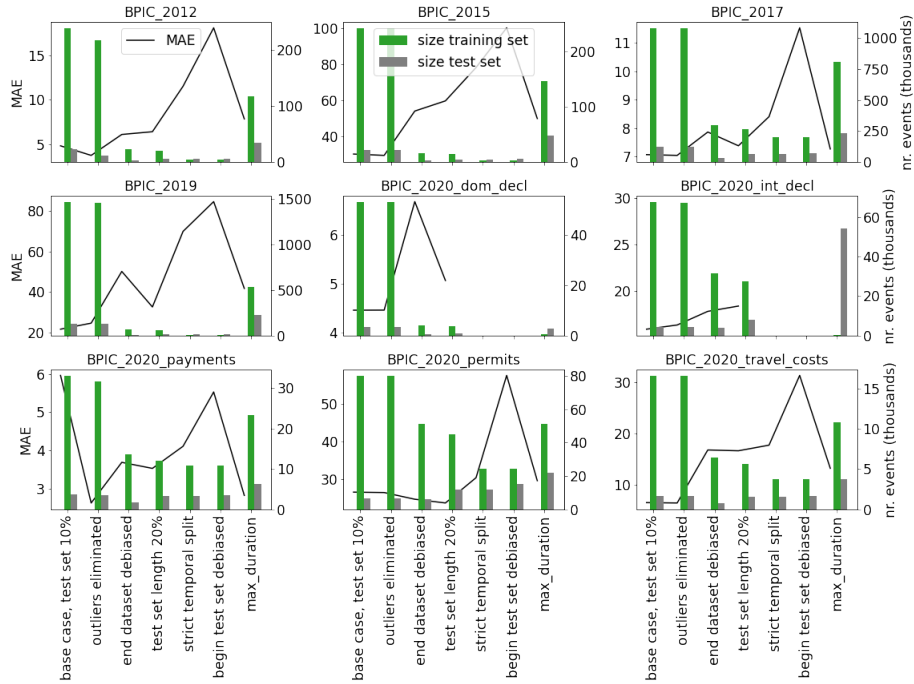


Fig. 5. Prediction results (MAE, lower is better) vary significantly in function of the chosen training and test data sets, calling for an agreement on proper benchmarks.

7 Conclusion and future work

In this paper, we demonstrated the impact of dataset preprocessing choices and pointed out the problem of overlapping training and test sets and at two sources of bias in test sets. Left unaddressed, these three obstacles risk slowing down progress in the field of predictive process monitoring. We argued for the use of unbiased benchmark datasets with strict temporal splitting of training and test sets. Using the example of nine BPIC datasets and the remaining time prediction problem, we established preprocessing steps to construct such benchmark datasets. The scripts for our examples are publicly available¹². It is our hope that researchers will use these benchmarks and thus facilitate a more rapid advancement of predictive process monitoring as the comparability and quality (no data leakage, no bias) of their results improves. In the future, perhaps recommendations for metrics and statistical analysis of results could be formulated to further enhance comparability and interpretability. We also hope that other publicly available datasets will be constructed using our proposed transformation steps. The scientific marketplace will then decide which of these datasets will become predictive process monitoring's next MNIST, CIFAR-10 or ImageNet.

References

1. Teinemaa I., Dumas M., La Rosa M., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. In: ACM Transactions on Knowledge Discovery from Data (TKDD) 13(2), Article No. 17 (2019)
2. Kratsch W., Manderscheid J., Roeglinger M., Seyfried J.: Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction. In: Business & Information Systems Engineering (2020). <https://link.springer.com/article/10.1007%2Fs12599-020-00645-0>
3. Verenich I., Dumas M., La Rosa M., Maggi F.M., Teinemaa I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. In: ACM Transactions on Intelligent Systems and Technology (TIST) 10(4): 1-34 (2019)
4. Neu D. A., Lahann J., Fettke P.: A systematic literature review on state-of-the-art deep learning methods for process prediction. In: Artificial Intelligence Review (2021). <https://doi.org/10.1007/s10462-021-09960-8>
5. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive Business Process Monitoring with LSTM Neural Networks. In: Dubois, E., Pohl, K. (eds) Advanced Information Systems Engineering. CAiSE 2017. Lecture Notes in Computer Science, vol 10253. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
6. Evermann, J., Rehse, J.-R., Fettke, P.: Predicting process behaviour using deep learning. Decision Support Systems 100, 129–140 (2017)
7. Polato M., Sperduti A., Burattin A., de Leoni M.: Data-aware remaining time prediction of business process Instances. In Proceedings of the International Joint Conference on Neural Networks, pp. 816-823 (2014)
8. Camargo, M., Dumas, M., González-Rojas, O.: Learning Accurate LSTM Models of Business Processes. In: Business Process Management 2019, LNCS 11675, pp. 286–302. Springer Nature Switzerland AG, Cham (2019)

¹² <https://github.com/hansweytjens/predictive-process-monitoring-benchmarks>