Replication of *Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction* by *Kratsch et al.*

Luiz Alberto Fonseca

Technische Universiteit Eindhoven, Eindhoven 5612 AZ, The Netherlands
l.a.fonseca@tue.student.nl

1 Introduction

The study replicated in this experiment aims to investigate the relationship between event logs' properties and the performance of deep learning algorithms on the task of outcome-prediction. In other words, they want to know which properties would facilitate the use of deep learning techniques.

To address that research problem they made a selection of 5 datasets that are purposely different in terms of properties such as number of variants and average events per case. For each dataset they created an outcome prediction task and trained 4 different algorithms, being 2 of them from classical machine learning (Random Forest and SVM) and the other 2 from deep learning (DNN and LSTM). In the evaluation they compared the performance of each technique on each dataset and related that performance with the properties of the datasets.

In this experiment I am replicating the original study mentioned previously aiming to identify if the original paper is detailed enough to allow a replication with a small number of assumptions, or preferably zero assumptions, made by the replicator. As a secondary objective this replication would be able to either promote the study's correctness or to point out possible threats to validity (threats to replication) of the original study.

To achieve that purpose, I first read carefully the paper to understand all the details and then I performed the same steps done in the original research using 4 out of the 5 datasets they used. I first computed the data properties form the datasets. Then, I pre-processed the data following all the steps mentioned in the original paper. After that, I trained the algorithms using and evaluated their performance using the same metrics they used. Lastly, I compare the results in the original study with the results I obtained in the replication both qualitatively and quantitatively.

In the results, I see that the values of the data properties extracted from the datasets are very different. In this part, I had to make assumptions since they do not specify in details how these properties were computed. For the algorithm's performance, they showed a mild alignment with the performance in the original results and the alignment is different depending on the metric and algorithm. The F1-Score was the metric that showed the best alignment.

The original study concludes that for accuracy and F1-score, deep learning has an overall better performance across event logs independently of data properties. This is also seen in the replication, however the difference between the approaches is smaller.

In section 2, I summarize the original study. In section 3, I describe the design of the replication, implementation details and point out the differences from the original study. In section 4, I present the results of the replication along with the results of the original result for comparison. In section 5, I interpret the results, discuss the threats to validity and state the conclusion of this experiment.

2 Background

2.1 Study Summary

The original paper investigates the following research question:

"Which event log properties facilitate the use of deep learning techniques for outcome-oriented predictive process monitoring?" [1]

Event logs have certain properties, such as number of activities and number of variants, that are connected with aspects of the process they refer to. In this study, the authors analyze if there is a relation between certain properties and the performance of deep learning techniques. In other words, they want to know if deep learning algorithms would perform better (or worse) depending on the properties of the log.

To achieve that purpose, a study was set up with 5 different datasets having different properties and 4 different algorithms, being two of them classical Machine Learning (ML) approaches and the other two Deep Learning (DL) approaches. The evaluation of the techniques was based on three different metrics, namely accuracy, F-score and ROC-AUC.

2.2 Data Description

The first step in the experiment was to create an outcome-oriented prediction task for each dataset. In the table below there's a summary of the tasks created for each event log.

Table 1. Overview of labeling rules [1]

Dataset	class 0	class 1
BPIC11	Normal Patients (875)	Urgent patients (268)
BPIC13	Resolved in lane 1 (4546)	Resolved in lane 2 or 3 (3008)
RTFM	No judge involved (149,815)	Judge involved (555)
PL	Above average rejects (50)	Up to average rejects (175)
RL	Accept (4932)	Reject (5068)

It's worth it to give a little description for each event log.

BPIC11 (BPI Challenge 2011) - Contains information about a process of cancer treatments for patients in a hospital.

BPIC13 (BPI Challenge 2013) - Refers to an incident management process with three support levels (lanes 1, 2 or 3).

PL (Production Log) - A log that registers events about a production process and the quality of the goods produced.

RL (Review Log) - A synthetic dataset about scientific papers reviews.

RTFM (Road Traffic Fine Management) - It's a dataset with information about a process for handling traffic regulation infringements.

Several properties were extracted from the datasets (see table below).

Table 2. Descriptive statistics of the event logs [1]

Event Log	# cases	# variants	# activities	# events	avg events per case	avg activities per case
BPIC11	1143	981	624	150,291	131	33
BPIC13	7554	1630	13	65,553	9	4
RTFM	150,370	231	11	561,470	4	4
PL	225	221	55	9086	40	12
RL	10,000	4118	14	236,360	24	15

Apart from the properties above, other properties were defined, such as:

activity-to-instance ratio - It's the ratio number of activities / number of instances. This property tells how early the attributes are available for prediction.

numeric-to-categorical ratio (of payload data) - It's the ratio number of categorical attributes / number of numeric attributes.

events-to-activity ratio - It's the ratio of average number of events / average number of activities per case. This property informs the number of loops and iterations inside a case.

variants-to-instances ratio - It's the ratio number of variants / number of instances. This property tells how unique the cases are inside a process.

Class imbalance - How imbalanced the labelling classes are in the dataset.

2.3 Models Implementation

The algorithms tested were Random Forest (RF) and Support Vector Machine (SVM) as representatives of classical machine learning for being approaches commonly used, and Deep Neural Network (DNN) and Long Short-Term Memory (LSTM) as representatives of the deep learning area for being entry level (DNN) and advanced (LSTM) approaches.

The beta parameter they used for the F-score metric was 1, which gives both precision and recall the same strength in the evaluation.

A preprocessing step was performed on the datasets to put them in the right format to serve as input of the algorithms. They used index-based encoding and

each dataset was split into 10 subsets to account for the 10 first events on each process. To elucidate, they created a subset only with the first event information, a subset with only the first two events information and so on till they reach 10 events. The case attributes were present in all the subsets. For the RTFM log, they used only 6 events because there wasn't enough data to cover more than that.

After the labelling, property extraction and preprocessing, they evaluated each technique against each subset and collected the results. Then, they related the performance with the presence or absence of data properties. Thus, in the end they have 10 (6 for RTFM) models for each combination of event log and algorithm. In total, 184 models.

3 Experiment Setup and Execution

3.1 Objective

The main goal of this study replication is the didactical learning. Through the experience of reproducing a scientific paper, one would learn in practice the challenges of such a task and also be able to identify mistakes made by the authors that hinder the replication and the correct deeds that facilitate this task. As a secondary objective, this replication aims to validate the results of the original paper and to be able to either promote the findings' correctness or point out failures that might invalidate parts of the research.

3.2 Experiment Design

I used 4 out of the 5 datasets they used in their study leaving apart the production log (PL). The reason for that is that the production log is in a different format and they do not provide the code for labelling this dataset. To understand the labelling with only the information they provided in the paper would take much time of this experiment. Because of that and considering that I had the majority of the datasets, I decided to leave the production log out of the experiment.

The first step was to perform an exploratory analysis on the datasets to check if the properties they extracted were complying with what I see in the analysis. Secondly, I pre-processed the data just like the authors had done and generated the subsets for each event log. Lastly, I ran the algorithms against each subset and collected the three metrics for each execution. The hyper-parameters for each algorithm were specified by the authors in this appendix.

Because the algorithms take a lot of time to run, I decided to change some configuration of the experiment to speed up the process. I used 3-fold cross-validation instead of 10-fold as they used in the original experiment. I also decreased the number of trace lengths to evaluate. In the original experiment they used traces from 1 to 10 for the logs BPIC11, BPIC13 and RL, I used 3 different trace lengths: 3, 6 and 9. For the RTFM log they used traces with length from 1

to 6, and I used trace lengths of 2 and 5. These changes don't significantly affect the results of the experiment and as I applied the changes for all the datasets and all the techniques, it shouldn't affect the comparison between them.

The method to compare the results of the original paper with the results of my replication will be through a comparative analysis. I will be performing qualitative analysis by generating visualisations that compare both results and quantitative analysis by calculating Kendall's correlation between the results. For the properties comparison, I will simply compare the numbers to see if they are accurate and for the machine learning evaluation I will generate visualisations and compute correlations.

Table 3. Summary of the differences between the experiments

	Original Experiment	Replication
Datasets	BPIC11, BPIC13, RTFM, PL, RL	BPIC11, BPIC13, RTFM, RL
Trace Lengths	From 1 to 10 (for RTFM 1 to 6)	[3, 6, 9] and [2, 5] for RTFM
Folds in Cross-Validation	10	3

3.3 Execution Details

Data and Code Acquisition The first thing before starting the replication is to acquire the data. Luckily, all the datasets are publicly available. Please find below the links to datasets used in this study.

The event logs BPIC11, BPIC13, RL and RTFM are available as .xes.gz files and PL is available as a .csv file. You need to decompress the .xes.gz files before using them as input to the scripts.

The original code published by the authors is available in this dropbox directory. I made a few changes to the code to correct some errors. My version of the code is available in this github repository. The changes were necessary either because some parts of the code were malfunctioning or because it wasn't in compliance with what the authors said in the paper. I assumed that this version of the code they provided is not the final version because it has several inconsistencies and is incomplete.

Scripts Execution Inside the directory there are two different small projects: the encoder (the Log Encoding sub-directory) and the model tuning scripts, where there is one script for each one of the four algorithms.

The log encoding sub-directory is a project written in C# (C-Sharp). It is responsible for pre-processing the event logs and generating the subsets corresponding to the number of events desired. The best tool to edit and execute these files is the Visual Studio IDE. The .snl file inside that folder is a configuration file to be interpreted by Visual Studio. To open the project and execute the code you might have to change the paths inside the scripts to comply with the paths

in your local machine. The main file is Program.cs. The execution of this file is not straightforward because you need to change the configurations manually to generate each subset. The configurations that need to be changed are: the path for the dataset files, the number of events to be considered and the labelling function to be called. Those changes are between the lines 11 and 41 and I left comments on the code on how to change them.

The model tuning scripts inside the folder "experiment" are the ones that train and evaluate the results for each algorithm. They are written in Python and I used PyCharm as an IDE to execute them, but you don't even need an IDE because you can run the scripts from the command line. The execution of each script is again very manual. For Random_Forest_experiments.py and SVM_experiments.py you need to pass as arguments for the script the name of the dataset, the initial number of events you want to consider (usually 1), and the final number of events you want to consider (usually 10, except for RTFM, which is maximum 6). For the deep learning approaches (DNN_experiments.py and LSTM_experiments.py), you need to execute each subset individually by passing the name of the dataset (e.g. bpic11) and the subset number you want to execute (from 1 to 10). The deep learning techniques take a lot of time to execute, so you'll need many days probably to run the whole experiment.

There's also a Jupyter notebook inside the folder (event-logs-properties.ipynb) that I created to collect the properties of the datasets and validate if they are in accordance with the original paper.

4 Results

4.1 Data Properties Check

The properties presented in the table below are the ones that have some impact in the conclusions of the original study and they also contain the properties shown in table 2.

Table 4. Data properties comparison between original study and replication

	Event Logs							
	BPIC11		BPIC13		RL		RTFM	
Feature	Orig.	Rep.	Orig.	Rep.	Orig.	Rep.	Orig.	Rep.
activity-to-instance	47/9	9/81	2/6	0/11	1/4	1/5	3/7	0/14
numeric-to-categorical	18/38	73/27	7/1	10/1	4/1	5/1	6/4	8/6
events-to-activity	33/131	33/131	4/9	3/9	12/40	10/24	4/4	4/4
variants-to-instances	1143/981	1143/981	7554/1630	7554/1511	10000/4118	10000/4118	150370/231	150370/231
class	C0: 875	C0: 875	C0: 4546	C0: 4546	C0: 4932	C0: 4932	C0: 149815	C0: 149815
imbalance	C1: 268	C1: 268	C1: 3008	C1: 3008	C1: 5068	C1: 5068	C1: 555	C1: 555

4.2 Algorithms' Performance

Qualitative Analysis Below you see the comparison of the algorithms' performances in the original study and in the replication. Each bar represents the

average value of the executions of all trace lengths. Find the table with all the results in the appendix.

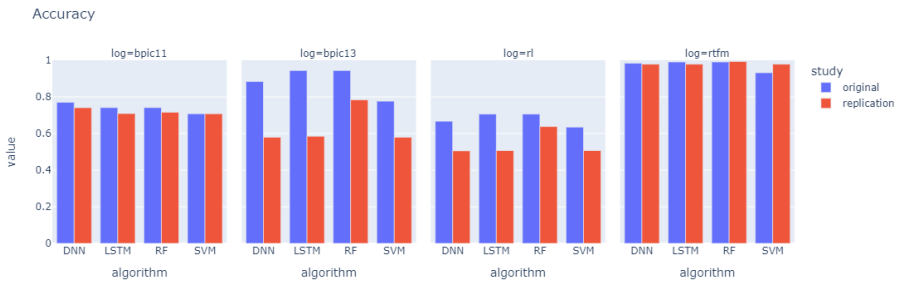

Fig. 1. Comparison of accuracy values for each event log and technique combination on the original study and on the replication.

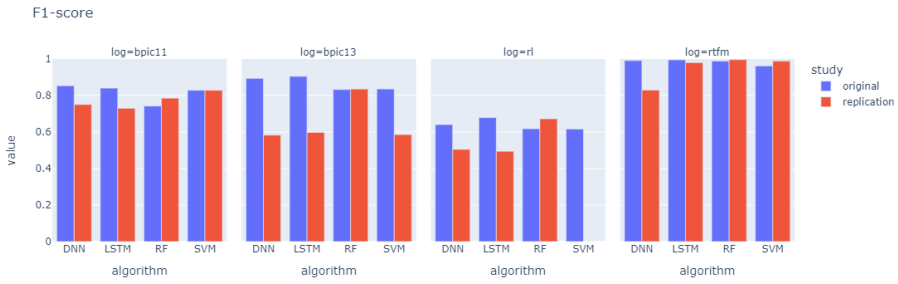

Fig. 2. Comparison of F1-score values for each event log and technique combination on the original study and on the replication.

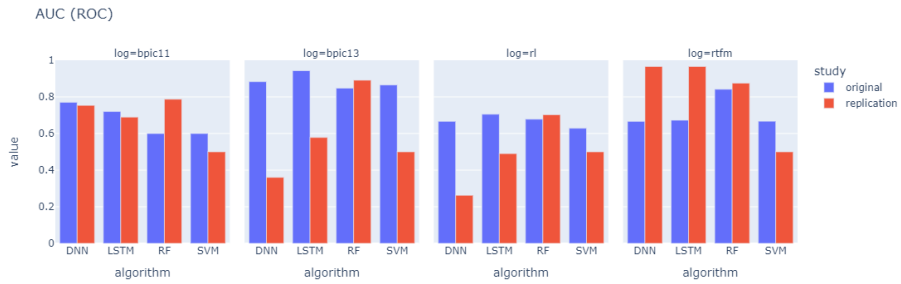

Fig. 3. Comparison of AUC (ROC) values for each event log and technique combination on the original study and on the replication.

Quantitative Analysis (Kendall's Correlation) The table below shows the Kendall's correlation value for each combination of metric and algorithm. The correlation is computed using the original values and the replication values. Each computation uses 22 observations considering the evaluation done with different trace lengths.

Table 5. Kendall's correlation between original experiment and replication results for each algorithm and metric.

	Metrics		
Algorithm	Accuracy	F1-score	AUC (ROC)
RF	0.78	0.75	0.53
SVM	0.45	0.64	—
DNN	0.48	0.48	0.11
LSTM	0.45	0.7	-0.07

5 Discussion

5.1 Results Interpretation

Data Properties Check For the first part of the results (subsection 4.1), we can see that only for the property "Class Imbalance" the values are identical across all datasets. For the other properties there is at least 1 divergent value between waht they mention in the original paper and what I saw in my data analysis. I will briefly go through each feature explaining the differences.

For the activity-to-instance ratio, I used the library PM4PY (version 2.5.1) to compute the case attributes and event attributes. So, I had to trust that the computation done by library is correct. Apart from that I didn't include

the case identifier as a case attribute because it is not included in the final prediction. The results for this feature are quite different from the ones shown in the original paper, even though I replicated the same pre-processing steps that they mentioned in the paper

For the computation of the numeric-to-categorical ratio, I manually checked each attribute by eye to see if they were categorical or numerical. The results again are expected to be different since this feature is related to the previous one in the sense that the number of attributes will be the same in both features.

For events-to-activity ratio and variants-to-instances ratio I also used PM4PY to compute the numbers and for some datasets the values are the same, but divergent for some others.

The divergence of these numbers is a crucial point in the replication, since the whole study is based on the relationship of these properties and the algorithms' performance.

The datasets are exactly the same since the class imbalance feature shows the exact same values both in the original study and in the replication. I can only assume that either I made some mistake in the code or they didn't specify very well how they obtained these features. I actually computed the features based on the brief explanation (natural language) that they give in the paper, but mostly based on my previous knowledge on process mining.

Algorithm's Performance For the algorithms, we can see that in most of the cases the performance on the replication is a little bit worse than in the original study. I suspect that this happens because I used less folds in the cross-validation. Anyhow, the comparison between techniques should stay the same if it was the case that the replication's results were perfectly aligned with the original study's results. I checked this alignment using Kendall's correlation and I saw that the alignment is not very strong and the correlation is even negative for one of the algorithms.

The F1-score metric showed the strongest alignment (minimum 0.48 and maximum 0.78) when comparing to the other metrics. AUC showed the weakest correlation.

The original study concludes that overall the deep learning techniques perform better than the classical approaches (on accuracy and F1-score) independently of the data properties. This is also seen in the replication even though the difference between classical approaches and deep learning approaches is smaller in the replication's results. In the original study deep learning is 8.4 pp higher in accuracy and 4.8 pp higher in F1-score. In the replication 5 pp higher in accuracy and 2 pp higher in F1-score.

5.2 Threats to Validity

This replication has one threat to internal validity given that some processes were performed manually by me during the data exploration analysis and every manual process is prone to human error and subjectivity. I tried to tackle this

threat by reviewing the task many times looking for some possible mistake that I could have made or something that I could have overlooked.

The replication has one threat to constructs validity since I used a different number of folds for cross validation, which could lead to different results in the final evaluation of the models. The reason I used a lower number of folds is that this experiment took many days to run in my machine and I had a limited deadline to finish it, so I couldn't afford waiting much longer for each model to be trained.

There's also another threat to internal validity when I assumed how the data properties were computed in section 4.1. The reason for assuming this computation is that it is not specified in the original study. To tackle this threat I reviewed the current literature on process mining to make sure I had enough knowledge to extract the properties from the datasets.

5.3 Conclusions and Lessons Learned

Through this replication I was able to identify the challenges one has to face when replicating a research paper. I was able to learn that when writing a paper you should specify every detail and every assumption you had during your research in order to facilitate the replication.

The results of this replication are overall mildly aligned with the results of the original considering the algorithms' performances. But, considering the data properties computation, the results are not very much aligned. In the latter I had to assume how the properties extraction was done based on my knowledge in process mining. Despite of that, the conclusion that deep learning techniques has a better performance on accuracy and F1-score is also observed in the replication with the observation that this performance difference is smaller in the replication.

Based on this experiment, I could identify which parts of the original paper were perfectly detailed allowing less assumptions in the replication and which parts were poorly detailed, which gave me a lot of room for assuming how things were done.

References

1. Kratsch, W., Manderscheid, J., Röglinger, M. et al. Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction. Bus Inf Syst Eng 63, 261–276 (2021). https://doi.org/10.1007/s12599-020-00645-0

6 Appendix

This appendix shows the results obtained in the original study and in the replication.

	algorithm	log	trace_length	accuracy orig	f1 orig	auc orig	accuracy rep	f1 rep	auc rep
0	RF	bpic11	3	0.75970	0.7233	0.6004	0.789486	0.870421	0.803594
1	RF	bpic11	6	0.73250	0.7125	0.6177	0.688915	0.778367	0.723623
2	RF	bpic11	9	0.73210	0.7909	0.5820	0.669892	0.708301	0.834932
3	RF	bpic13	3	0.88640	0.7391	0.7336	0.732480	0.774593	0.897092
4	RF	bpic13	6	0.96410	0.8884	0.9335	0.803254	0.858123	0.898009
5	RF	bpic13	9	0.98170	0.8712	0.8760	0.815468	0.874041	0.878626
6	RF	rl	3	0.61430	0.5673	0.5912	0.602799	0.665771	0.660463
7	RF	rl	6	0.76440	0.6553	0.7273	0.658299	0.679683	0.723481
8	RF	rl	9	0.73960	0.6317	0.7165	0.652599	0.672306	0.724456
9	RF	rtfm	2	0.99340	0.9967	0.7312	0.993303	0.996640	0.784910
10	RF	rtfm	5	0.98810	0.9814	0.9522	0.992766	0.996261	0.965236
11	SVM	bpic11	3	0.74010	0.8506	0.6004	0.740059	0.850613	0.500000
12	SVM	bpic11	6	0.70550	0.8273	0.6177	0.705496	0.827320	0.500000
13	SVM	bpic11	9	0.67830	0.8083	0.5820	0.678272	0.808297	0.500000
14	SVM	bpic13	3	0.85440	0.8596	0.8699	0.495851	0.225958	0.500000
15	SVM	bpic13	6	0.78880	0.8443	0.8870	0.594126	0.745394	0.500000
16	SVM	bpic13	9	0.68660	0.8040	0.8386	0.648575	0.786831	0.500000
17	SVM	rl	3	0.61450	0.5771	0.6149	0.506800	0.000000	0.500000
18	SVM	rl	6	0.64690	0.6388	0.6382	0.506800	0.000000	0.500000
19	SVM	rl	9	0.64350	0.6343	0.6335	0.506800	0.000000	0.500000
20	SVM	rtfm	2	0.91040	0.9490	0.5232	0.993303	0.996640	0.500000
21	SVM	rtfm	5	0.95330	0.9755	0.8114	0.963825	0.981578	0.500000
22	DNN	bpic11	3	0.83680	0.8978	0.8368	0.769157	0.774621	0.788678
23	DNN	bpic11	6	0.74780	0.8422	0.7478	0.740653	0.754167	0.763997
24	DNN	bpic11	9	0.72630	0.8214	0.7263	0.713092	0.723380	0.708948
25	DNN	bpic13	3	0.84340	0.8414	0.8443	0.495851	0.497121	0.264829

26	DNN	bpic13	6	0.88730	0.9024	0.8897	0.594126	0.589330	0.362228
27	DNN	bpic13	9	0.92060	0.9385	0.9154	0.648575	0.662760	0.454576
28	DNN	rl	3	0.62708	0.5828	0.6260	0.506800	0.506316	0.264054
29	DNN	rl	6	0.68764	0.6589	0.6867	0.502201	0.502612	0.260350
30	DNN	rl	9	0.68704	0.6810	0.6871	0.506800	0.506316	0.264054
31	DNN	rtfm	2	0.99340	0.9967	0.5000	0.993303	0.993371	0.988681
32	DNN	rtfm	5	0.97390	0.9864	0.8327	0.963825	0.666667	0.943400
33	LSTM	bpic11	3	0.75970	0.8558	0.6951	0.740059	0.746212	0.696230
34	LSTM	bpic11	6	0.73250	0.8347	0.7488	0.706960	0.722222	0.696832
35	LSTM	bpic11	9	0.73210	0.8313	0.7166	0.680006	0.722222	0.675998
36	LSTM	bpic13	3	0.88640	0.7591	0.8876	0.512687	0.536732	0.528389
37	LSTM	bpic13	6	0.96410	0.9698	0.9637	0.593944	0.604167	0.588542
38	LSTM	bpic13	9	0.98170	0.9860	0.9785	0.648567	0.653274	0.618350
39	LSTM	rl	3	0.61430	0.5375	0.6122	0.506836	0.494792	0.490458
40	LSTM	rl	6	0.76440	0.7605	0.7643	0.506836	0.494792	0.490458
41	LSTM	rl	9	0.73960	0.7387	0.7397	0.506836	0.494792	0.490458
42	LSTM	rtfm	2	0.99340	0.9967	0.5072	0.993303	0.993371	0.988681
43	LSTM	rtfm	5	0.98810	0.9939	0.8386	0.963825	0.967448	0.943400