Process Analytics Workbench (PAW)

Weixin, Wang[1,*], Eric, Amann[2], Martin, Matzner[1] and Patrick, Delfmann[2]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Fürther Straße 248, 90429 Nürnberg, Germany
[2]Universität Koblenz, Universitätsstr. 1, 56072 Koblenz, Germany

Abstract

The Process Analytics Workbench (PAW) is a comprehensive tool to streamline Predictive Process Monitoring (PPM) activities. It integrates functionalities like synthetic event log generation, model training, and prediction evaluation within a micro-service architecture, enabling agile and efficient experimentation.

Keywords

Predictive Process Monitoring, Process model, Simulation, Event log, Rapid experimentation

Metadata description	Value
Tool name	PAW
Current version	1.0
Legal code license	Apache 2.0
Languages, tools and services used	Python, Java, Pytorch, PM4Py, Camunda
Supported operating environment	Microsoft Windows, GNU/Linux
Download/Demo URL	https://uni-ko.de/paw
Documentation URL	https://uni-ko.de/paw-docs
Source code repository	https://uni-ko.de/paw-repo
Screencast video	https://uni-ko.de/paw-demo

1. Introduction

Predictive Process Monitoring (PPM) is a subcategory of process mining that is concerned with the prediction of the future state of process instances, which are currently being executed. Common predictions are, for example, the next events or the duration till the process is completed. PPM tools usually rely on Machine Learning (ML) algorithms trained on event log data [1]. Several different tools have been developed in the past. However, none of them provide complete and comprehensive support for all PPM-related activities.

Our goal is to provide a tool that supports all major aspects of PPM. We therefore propose the Process Analytics Workbench (PAW), which supports all the main PPM functionalities out of the box and can easily be extended to meet the specific application requirement. Through

ICPM 2024 Tool Demonstration Track, October 14-18, 2024, Kongens Lyngby, Denmark
*Corresponding author.
weixin.w.wang@fau.de (W. Wang); amann@uni-koblenz.de (E. Amann); martin.matzner@fau.de (M. Matzner); delfmann@uni-koblenz.de (P. Delfmann)
0009-0002-0946-5608 (W. Wang); 0009-0005-6241-6081 (E. Amann); 0000000152443928 (M. Matzner); 0000000344410311 (P. Delfmann)

components including the model training, predictions and synthetic log generation, experiments can be quickly set up and conducted with little effort.

2. Tool Description

PAW consists of a collection of micro-services, each fulfilling a single purpose. This design approach is central to the tool, providing many advantages for end users. Users can determine the modules which are relevant to their application and run the corresponding micro-services. This approach optimizes time and resource utilization while maintaining minimal complexity in the PPM environment. Figure 1 shows an overview of the currently available modules.

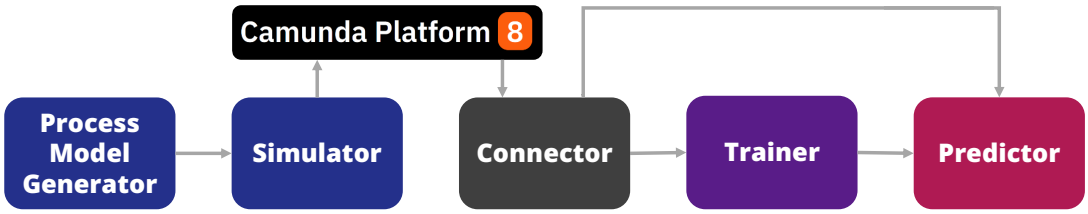

Figure 1: Micro-service modules contained in PAW.

The currently available micro-service modules can be broadly categorized into three major groups, namely the *process model generation and simulation*, the *data connector*, and the *training and prediction* module.

2.1. Process Model Generation and Simulation

The process model generation and simulation module allows users to quickly and easily create batches of different process models and simulate the respective event logs. It provides vast configurability to support a large number of experimental purposes.

We developed the process model generator based on an algorithm proposed by F. Bezerra et al.[2]. This algorithm addresses the generation of models using a *divide and conquer* strategy, recursively applying the algorithm to each divided segment of the model until it reaches the three atomic components of all models, e.g., tasks, parallel gateways, and exclusive gateways. We selected BPMN[1] as the format for the generated process models because of its widespread acceptance and readability.

The process model simulation is realized by integrating the agent-based workflow simulator proposed by Amann and Delfmann [3]. Combined with the process model generator, this allows for the straightforward and configurable generation of any number of synthetic event logs.

[1]https://www.bpmn.org/

Through this, predictive models' performance can be evaluated using specific types of BPMN diagrams and event logs.

To enhance reproducibility, random seeds used for stochastic functions in the process model generation and event log simulation modules can be set with presets, enabling researchers to replicate experiments without additional effort.

2.2. Data Connector

The data connector provides all process data to the trainer and predictor. It takes raw process data from a source and transforms it into a standardized format that the trainer and predictor can use. The current data connector is developed to extract event log data from the Camunda 8 workflow management system. This makes the integration into an environment running on Camunda 8 effortless. The main idea behind the connector is to provide a single piece of software for the provision of process data.For other use cases, a modified version of the connector could be developed to connect to a different workflow management system.

2.3. Model Training and Prediction

The training and prediction modules constitute the core components of PAW, facilitating the training of predictive models and the prediction of attributes for ongoing process instances. Data preprocessing described in [4] and result evaluation, respectively positioned before the model training phase and after the prediction phase, enable the modules to fulfill the complete machine learning pipeline. These stages ensure that the input data is optimally prepared for training and that the predictive outcomes are rigorously assessed, enhancing the overall effectiveness of the pipeline. At present, Long Short-Term Memory (LSTM) networks [5] implemented with PyTorch [6] and Extreme Gradient Boosting (XGBoost) [7] are implemented as the primary predictive models; However, the program interfaces provided by our architecture facilitate the seamless integration of additional artifacts from other researches within the community.

3. Case Study

To demonstrate PAW's potential application scenarios, we conducted a preliminary experiment evaluating the correlation between the control flow complexity (CFC) of a process model and the performance of a predictive model trained using event logs simulated based on this process model.

3.1. Experiment Design

Cardoso [8] classified and defined four distinct dimensions of complexity measures for business processes:

1. Control flow complexity: structure characteristics such as splits, joins, and loops.
2. Activity complexity: number of total activities.
3. Data flow complexity: interfaces and information flow within the process.
4. Resource complexity: resources required to execute the activities and the process.

First, we randomly generated 25 process models, each containing a fixed number of 10 tasks. No data dependencies or information flows were assigned to these process models. The predictive task was set as the next activity prediction, ensuring that activity, data flow, and resource complexities were excluded, leaving only CFC to influence the ground truth for prediction. The 25 process models were categorized into five groups based on their control flow complexity, with values of 1, 3, 5, 7, and 9. Event logs with 200 cases each were simulated from the process models with same simulation parameters.

In the subsequent phase, XGBoost models with identical hyperparameters were trained and evaluated with the simulated event logs, employing a split ratio of 64:16:20 for training, validation, and testing.

3.2. Result Analysis

We conducted experiments on five groups, each comprising five process models, and consolidated the results into mean and standard deviation values, reflecting the distribution of the results in each group.

Model CFC	Precision Mean ± Std	Recall Mean ± Std	F1 Score Mean ± Std
1	0.9901 ± 0.0222	0.9899 ± 0.0227	0.9898 ± 0.0228
3	0.9380 ± 0.0377	0.9321 ± 0.0392	0.9298 ± 0.0399
5	0.8960 ± 0.0947	0.9065 ± 0.0719	0.8953 ± 0.0885
7	0.9250 ± 0.0621	0.9096 ± 0.0649	0.9101 ± 0.0634
9	0.8351 ± 0.1319	0.8576 ± 0.0933	0.8368 ± 0.1149

Table 1
Prediction performance of models on event logs with different process model CFC

Based on the provided results, a general observation can be made that as the process model CFC increases, the predictive model's performance tends to decline in terms of precision, recall, and F1 score. However, an exception to this trend is observed between the CFC values of 5 and 7. This deviation suggests that the relationship between control-flow complexity and predictive performance may not be strictly linear, and other factors might influence the results.

4. Evaluation

During the literature review, we could not identify publicly available BPMN generators with comprehensive documentation besides PLG2[9]. Multiple publications adopted it as the generation tool to generate artificial event logs. The limitations of PLG2 render it unsuitable for our implementation, as the experimental workflow necessitates precise control over generation parameters, including the exact numbers of tasks and gateways in the generated process model. Similar to our case study, PLG2 was employed to generate synthetic event logs in a study on the relations between event log characteristics and optimal machine learning algorithms [10]. Our case study demonstrated how PAW is designed and specialized to streamline such experimental workflow compared to the existing tools such as PLG2.

We also identified other popular and publicly available BPMN model simulators, namely BIMP[2] and Apromore[3], and compared them to our tool. Table 2 provides an overview of the main differences. The major distinction between our BPMN simulator and others lies in the basic approach of the simulation engine. While the established simulators rely on the propagation of simulated tokens through the process model, our approach simulates the whole environment in which the process runs. This provides many more configuration options and allows for a high complexity in the simulation. For example, while the established simulators select a path after an XOR gateway by defining fixed probabilities for each path, our approach handles the selection based on the values of variables in the simulated environment. This allows for very high flexibility and simulates dynamic scenarios, including concept drift. However, because of the added complexity, our simulation approach takes several minutes, while the established simulators usually finish in several seconds.

	BIMP	Apromore	PAW
Simulation Approach	Token-Based	Token-Based	Agent-Based
Simulation Complexity	Low	Low	High
Simulation Time	Fast	Fast	Slow

Table 2
Differences between common BPMN simulators and our approach

No comparable applications within the Business Process Monitoring (BPM) domain have been identified that simplify and automate the machine learning pipeline for PPM. As demonstrated in the case study, our tool enables rapid experiment execution, produces statistically robust results, and effectively addresses randomness issues in experimental design.

5. Conclusion

PAW provides a comprehensive solution for PPM activities. It supports many different application scenarios, from industry utilization to research. The micro-service architecture provides plenty of opportunities for customization and future expansions. Our case study showed how PAW can facilitate and support experiments in the PPM domain.

However, this work only represents the initial development iteration of the tool and does not constitute a final and complete state of the software. Currently, its complex installation process is not friendly for users without solid computer and programming knowledge. Also, the current user interface does not allow for the control of all aspects of the tool. We plan to address these shortcomings with the upcoming releases. We also plan to expand the existing platform further to include additional features in the form of new micro-service modules, such as hyperparameter optimization and data clustering.

[2] https://bimp.cs.ut.ee/
[3] https://apromore.com/

Acknowledgments

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 456415646, 432399058.

References

[1] C. Di Francescomarino, C. Ghidini, Predictive process monitoring, in: Process Mining Handbook, Springer International Publishing Cham, 2022, pp. 320–346.

[2] F. Bezerra, J. Wainer, Algorithms for anomaly detection of traces in logs of process aware information systems, Information Systems 38 (2013) 33–44.

[3] E. Amann, P. Delfmann, An agent-based workflow simulator, Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations Resources Forum co-located with 22nd International Conference on Business Process Management (BPM 2024) (2024).

[4] I. Teinemaa, M. Dumas, M. L. Rosa, F. M. Maggi, Outcome-Oriented Predictive Process Monitoring: Review and Benchmark, ACM Transactions on Knowledge Discovery from Data 13 (2019) 1–57.

[5] N. Tax, I. Verenich, M. La Rosa, M. Dumas, Predictive Business Process Monitoring with LSTM Neural Networks, in: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering, Springer International Publishing, Cham, 2017, pp. 477–492.

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 721, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 8026–8037.

[7] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794.

[8] J. Cardoso, Control-flow complexity measurement of processes and weyuker's properties, in: 6th international enformatika conference, volume 8, 2005, pp. 213–218.

[9] A. Burattin, PLG2: multiperspective process randomization with online and offline simulations, in: L. Azevedo, C. Cabanillas (Eds.), Proceedings of the BPM Demo Track 2016 Co-located with the 14th International Conference on Business Process Management (BPM 2016), Rio de Janeiro, Brazil, September 21, 2016, volume 1789 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016, pp. 1–6.

[10] C. Drodt, S. Weinzierl, M. Matzner, P. Delfmann, Predictive Recommining: Learning Relations Between Event Log Characteristics and Machine Learning Approaches for Supporting Predictive Process Monitoring, in: Intelligent Information Systems, Springer International Publishing, 2023, pp. 69–76.