Software Impacts 17 (2023) 100556

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

PM4Py: A process mining library for Python

Alessandro Berti, Sebastiaan van Zelst, Daniel Schuster *

Fraunhofer Institute for Applied Information Technology FIT, Germany

ARTICLE INFO

Keywords:
Process mining
Open-source software
Data analysis
Event data

ABSTRACT

PM4Py is a Python library providing a comprehensive array of tools for process mining. This paper presents an in-depth overview of the PM4Py library, including its integration with other Python libraries and its latest features, such as object-centric process mining. Furthermore, we discuss the significant impact of PM4Py within academia, industry, and the open-source community, evidenced by its wide adoption and substantial evolution. In short, the PM4Py library is an essential tool for researchers and practitioners, paving the way for advancements in process mining.

Code metadata

Current code version	**2.7.4**
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2023-301
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/6602453/tree/v1
Legal Code License	GPL-3.0 license
Code versioning system used	GIT
Software code languages, tools, and services used	Python 3.10
Compilation requirements, operating environments & dependencies	https://github.com/pm4py/pm4py-core/blob/2.7.4/requirements.txt
If available Link to developer documentation/manual	https://pm4py.fit.fraunhofer.de/
Support email for questions	pm4py@fit.fraunhofer.de

1. Introduction

Process mining [1], a link between data science and process science, enables organizations to gain insights from event data generated during the execution of operational processes to understand and improve their processes. Three central disciplines can be distinguished in process mining: process discovery [2,3], conformance checking [4,5], and process enhancement [6]. PM4Py is a Python library that contains many algorithms from these disciplines, enabling data-driven decision-making and helping organizations uncover optimization potential in their processes.

In response to the growing need for advanced process mining tools, PM4Py was developed to address the diverse challenges in this area. This library fills a noticeable gap in the market and provides a comprehensive and flexible solution for managing complex process mining scenarios. PM4Py addresses the diversity of real-world processes and covers a wide range of techniques. As an open-source project, PM4Py encourages community participation and broad adoption. Thus, it facilitates the application of process mining and fosters an evolving, interactive community in the field.

PM4Py had its start with the publication of the initial paper [7], and since then, it has grown substantially, gaining a larger user community with over 1,086,889 downloads as of July 31, 2023.[1] This growth is a testament to the library's effectiveness and has encouraged ongoing improvements. This paper will explore the latest features of PM4Py and its potential applications in the rapidly expanding field of process mining.

The remainder of this paper is organized as follows. Section 2 offers a comprehensive overview of PM4Py, detailing its essential features and illustrating its integrations. Section 3 delves into the impact of PM4Py. Finally, Section 4 concludes the paper, providing a concise summary and emphasizing the ongoing potential and future directions of PM4Py in the ever-evolving landscape of process mining.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, 53757 Sankt Augustin, Germany.
E-mail address: daniel.schuster@fit.fraunhofer.de (D. Schuster).

[1] According to https://pepy.tech/project/pm4py

https://doi.org/10.1016/j.simpa.2023.100556
Received 6 July 2023; Received in revised form 25 July 2023; Accepted 25 July 2023

Code Listing 1: An example process mining analysis conducted using PM4Py

```
import pm4py

# a log in the XES format is imported
log = pm4py.read_xes('tests/input_data/receipt.xes')

# the log is filtered on the top 5 variants
filtered_log = pm4py.filter_variants_top_k(log, 5)

# a directly-follows graph (DFG) is discovered from the log
dfg, start_activities, end_activities = pm4py.discover_dfg(filtered_log)

# a process tree is discovered using the inductive miner
process_tree = pm4py.discover_process_tree_inductive(filtered_log)
# the process tree is converted to an accepting Petri net
petri_net, initial_marking, final_marking = pm4py.convert_to_petri_net(process_tree)
    process_tree = pm4py.discover_process_tree_inductive(filtered_log)
# the accepting Petri net is converted to a BPMN diagram
bpmn_diagram = pm4py.convert_to_bpmn(petri_net, initial_marking, final_marking)

# the discovered process models are shown on the screen as .svg images
pm4py.view_dfg(dfg, start_activities, end_activities, format='svg')
pm4py.view_process_tree(process_tree, format='svg')
pm4py.view_petri_net(petri_net, initial_marking, final_marking, format='svg')
pm4py.view_bpmn(bpmn_diagram, format='svg')

# we compare the original log versus the discovered model
fitness = pm4py.fitness_token_based_replay(log, petri_net, initial_marking, final_marking)
precision = pm4py.precision_token_based_replay(log, petri_net, initial_marking, final_marking)

print(fitness) # 0.984
print(precision) # 0.758

# we check if the discovered model is a sound workflow net
is_sound_wfnet = pm4py.check_soundness(petri_net, initial_marking, final_marking)[0]
print(is_sound_wfnet) # True
```

2. Software overview

This section provides a comprehensive overview of the PM4Py library, detailing its main features, support for various standard event log and process model formats, visualization capabilities, and an illustrative example of using the library for process mining analysis.

2.1. Main features

This section briefly reviews PM4Py's central features. Section 2.1.1 overviews algorithms implemented in PM4Py. Section 2.1.2 presents the process-mining-specific data formats that are supported by PM4Py. Finally, Section 2.1.3 briefly presents built-in visualizations offered by PM4Py. The provided code in Code Listing 1llustrates a complete process mining workflow using PM4Py.

2.1.1. Implemented approaches

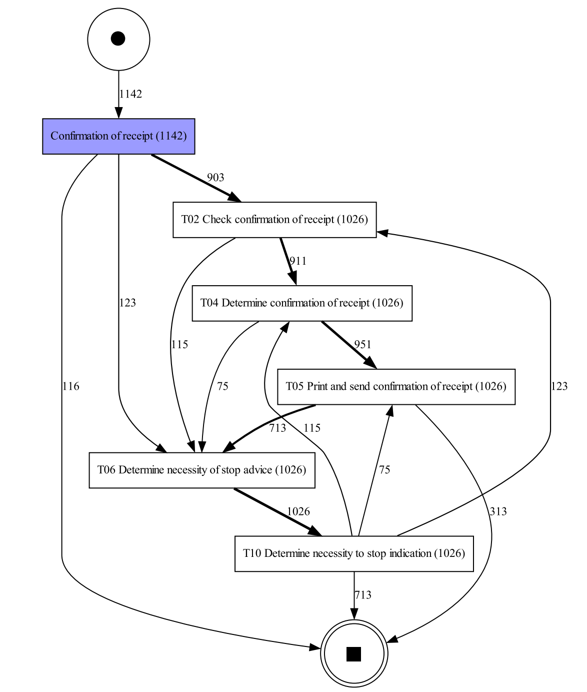
PM4Py provides an array of implemented approaches; consider Fig. 2 for a comprehensive overview. For *process discovery*, it includes the alpha miner [8], inductive mining algorithms [9–11], the heuristics miner [12], the ILP miner [13], the correlation miner [14], the prefix tree approach [15], and causal nets [1]. Moreover, it supports fundamental abstractions such as directly-follows graphs (DFGs) [1] and transition systems [1]. In the specialized area of *object-centric process mining (OCPM)* [16], PM4Py is equipped with methods like OC-DFG discovery [17], OC-Petri nets discovery [18], and feature extraction [19].

Regarding *conformance checking*, PM4Py leverages footprints [1], the token-based replay technique [20], and alignments [21] to compare observed with modeled process behavior systematically. PM4Py further refines this verification using decomposed/recomposed alignments [22] and log skeleton [23] methods. Also, conformance checking using Earth Mover Distance (EMD) [24], and LTL Checking [25] are supported by PM4Py.

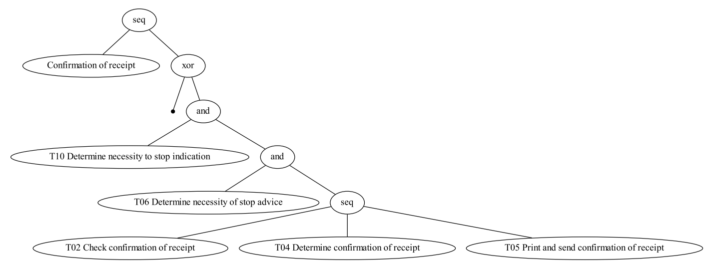
PM4Py allows evaluating the *log-model quality* through multiple metrics such as *fitness*, for instance, using token-based replay [20] or alignments [26], ETConformance *precision* [27,28], *generalization* [26], *simplicity* [29], anti-alignments [30], and multi-alignments [31]. Other process mining fields covered by PM4Py include process tree generation [32], decision mining [33], soundness checking with WOFLAN [34]), and trace clustering [35]. Beyond the process-focused perspective, PM4Py offers insights into *social and organizational aspects* of processes through social network analysis [36], role discovery [37], and resource profiling [38]. Moreover, it allows to analyze batch processing scenarios [39], in which a resource executes an activity for different cases in a short amount of time. Also, differential privacy [40] is implemented to protect sensitive information.

2.1.2. Supported data formats

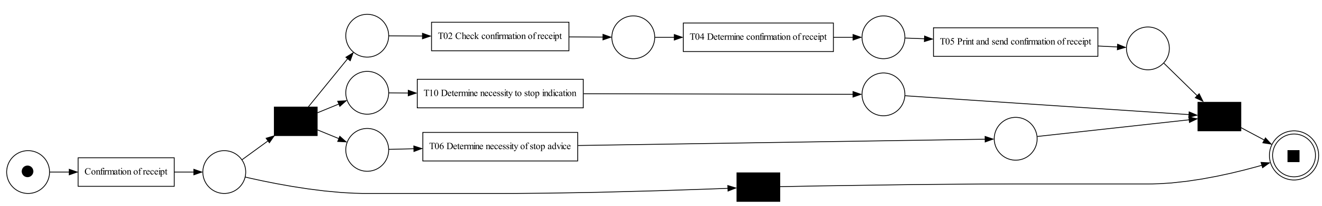
PM4Py has been designed to work seamlessly with a variety of standard event log and process model formats, making it highly adaptable to different systems. It supports the traditional event log format XES [41], and for object-centric event logs, the OCEL [42] format is utilized. Regarding process models, PM4Py supports the Petri nets format PNML [43] as well as the process trees format PTML [44]. Furthermore, the popular BPMN 2.0 [45] format is also supported. By supporting such a wide range of formats, PM4Py is capable of increasing interoperability and facilitating extensive usage.



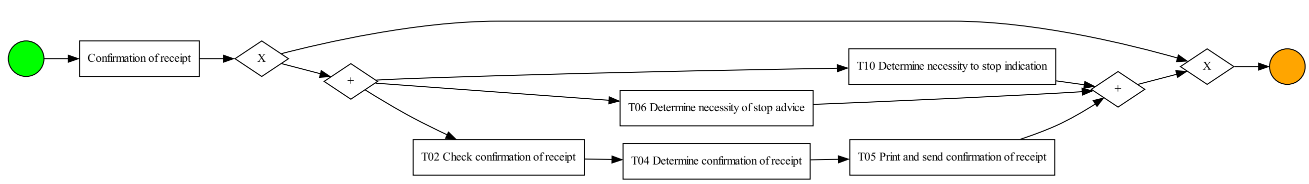
(a) Directly-Follows Graph (DFG) (cf. line 20 of Code Listing 1)



(b) Process Tree (cf. line 21 of Code Listing 1)



(c) Accepting Petri net (cf. line 22 of Code Listing 1)



(d) BPMN Diagram (cf. line 23 of Code Listing 1)

Fig. 1. Visualizations of the process models discovered in Code Listing 1.

2.1.3. *Visualizations*

PM4Py offers diverse visualizations for process mining artifacts, for instance, cf. Fig. 3. Consider Fig. 1 showing various different process model formalism visualized using PM4Py: directly-follows graphs (cf. Fig. 1(a)), process trees (cf. Fig. 1(b)), accepting Petri nets (cf. Fig. 1(c)), and BPMN models (cf. Fig. 1(d)). Besides process model visualizations, various other visualizations are implemented in PM4Py. For instance, the dotted chart visualization (cf. Fig. 3(a)) and the performance spectrum (cf. Fig. 3(b)) [46] allow to visually detect temporal performance patterns in event data. Moreover, decision trees (cf. Fig. 3(c)) elucidates decision factors. Such visualizations are important for process comprehension and optimization.

3. Impact overview

This section explores the impact of PM4Py, its role in academic innovations, and derived tools that build upon PM4Py. In terms of applications, PM4Py's potential is vast and diverse. Organizations across various sectors have used it, including healthcare for optimizing patient

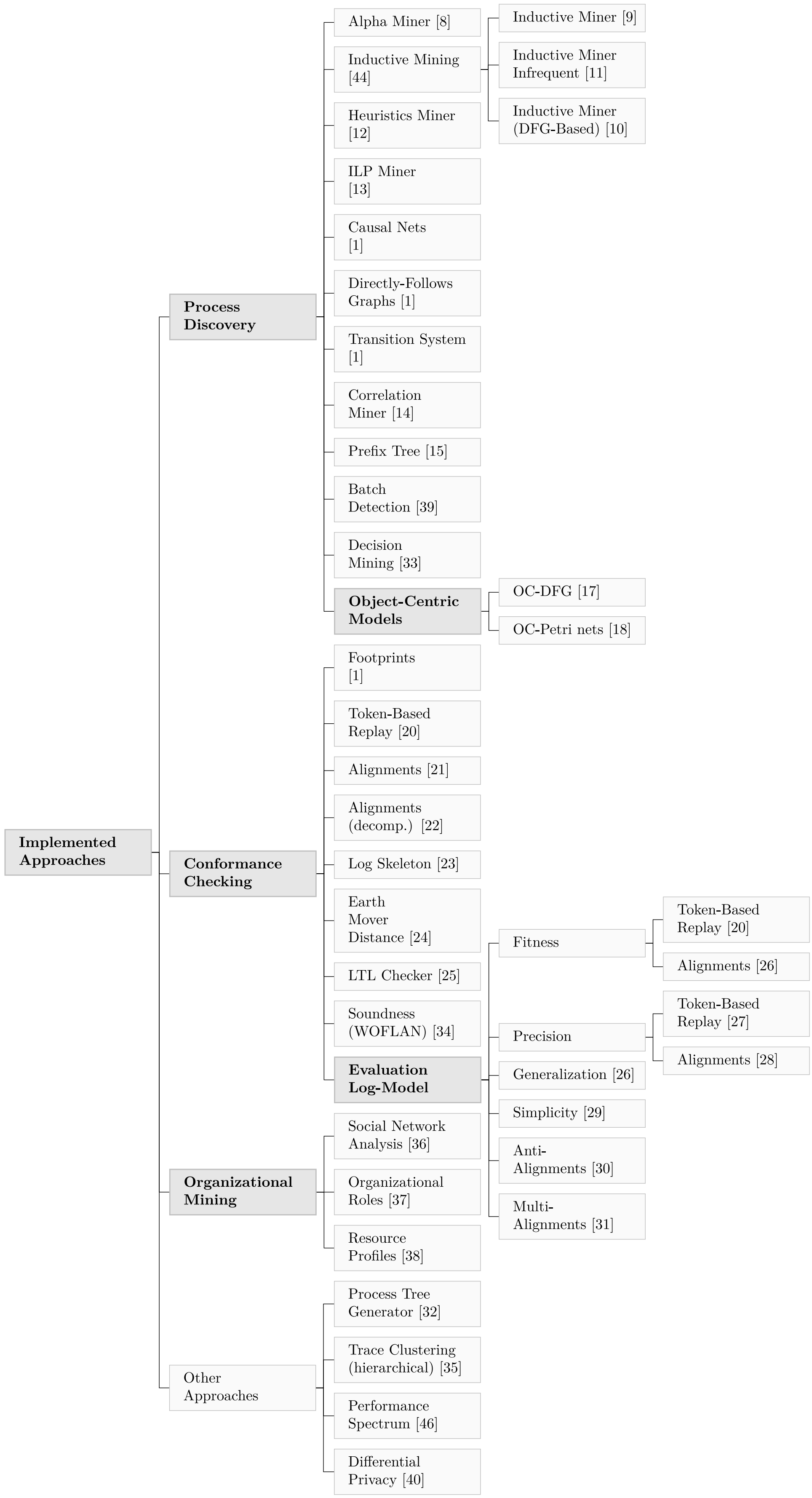
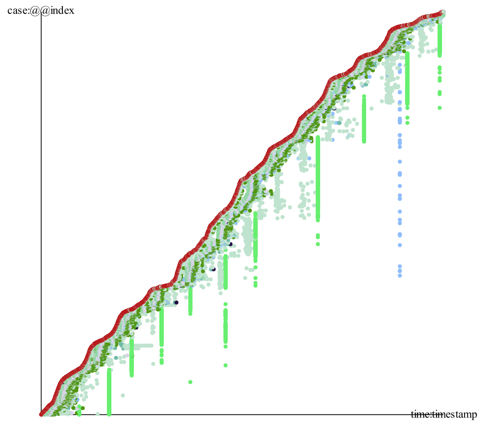
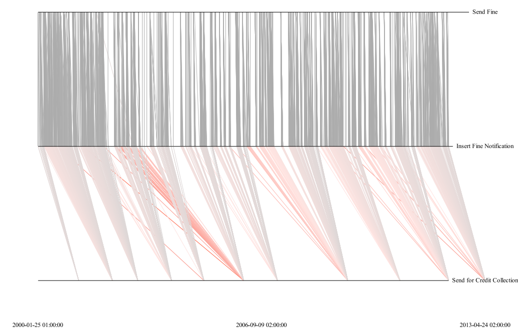


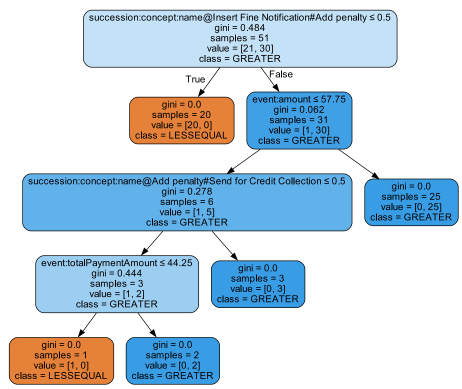
Fig. 2. Overview of various approaches and algorithms implemented in PM4Py.



(a) Dotted Chart visualizing the events from a publicly available event log [47]



(b) Performance spectrum showing temporal patterns from a publicly available event log [47] between the activities 'Send Fine', 'Insert Fine Notification', and 'Send for Credit Collection'



(c) Decision Tree (for the duration of a case) for a publicly available event log [47], correlating the features to the duration of a case (two classes, greater or smaller than the median throughput time, are identified).

Fig. 3. Example visualizations offered by PM4Py using the open-access event log [47].

pathways[2] [48], finance to improve transaction processes[3] [49], and IT for threat detection[4] [50].

3.1. Research advances

PM4Py's contribution to the academic sector is significant as an indispensable tool in process mining research and education. It encourages breakthroughs and advancements in this expanding field and serves as a foundation for prototyping and developing new algorithms in process mining. Furthermore, the library embraces recent research developments like object-centric process mining, shifting from a traditional case-centric view to capturing complex inter-object relationships. PM4Py also utilizes machine learning for predictive process mining [51], enhancing anomaly detection [52], process conformance checking [53], and forward-looking capabilities. Moreover, PM4Py has been instrumental in realizing several high-impact research studies, as demonstrated by its utilization of well-cited publications. The study [54], which outlines a data-driven approach for creating digital twins in smart factories using machine learning and process mining techniques, has been cited 55 times. Moreover, in [55], a novel solution for detecting changes in process behavior over time has been proposed, building upon PM4Py functionalities, and has garnered 32 citations.

3.2. Tools

The PM4Py library serves as the foundation for numerous process mining software tools; below, we briefly present some selected tools. The Process Mining Toolkit (PMTk) [56] includes several process mining capabilities in one end-user-oriented tool, including process discovery, visual analysis of processes, and conformance checking. OPerA [57] focuses on object-centric performance metrics, providing insight into object lifecycles. Cortado [58] innovates process discovery by utilizing domain knowledge and enabling incremental process model discovery. DTween [59] and Impacta [60] enable the creation of a digital twin of an organization. In summary, these tools, built on the

[2] PM4Py has been used for data preprocessing and process discovery purposes. In particular, the discovered models were DFGs and Petri nets discovered using the inductive miner.

[3] The discovery of directly-follows graphs helped in the audits; the discovery of social networks was used to assess the degree of automation in the financial process.

[4] Several process discovery techniques implemented in PM4Py have been used (alpha miner, heuristics miner, inductive miner) to represent the underlying process. Moreover, alignment-based conformance checking has been used to check the conformance.

PM4Py library, demonstrate the versatility of process mining while confirming the usefulness of PM4Py.

3.3. Statistics

PM4Py's growth and popularity are evidenced by its significant GitHub and PyPI metrics. On GitHub,[5] PM4Py has received over 554 stars and has been forked over 235 times, reflecting its wide distribution. With 43 contributors, 106 pull requests, and 310 closed issues, it is a testament to a vibrant community using and contributing to PM4Py. Finally, PM4Py's distribution is substantial, with over 1,086,889 downloads.[6] PM4Py has a considerable academic influence beyond software metrics, with the first paper [7] being cited in 231 scholarly articles, affirming its standing in the process mining field.

4. Conclusion

PM4Py is a Python library implementing various process mining approaches and supporting widespread data formats. The library has demonstrated its effectiveness as a versatile open-source toolkit for process mining. Its rich feature set and extensive integration with other Python libraries provide a powerful toolbox for academic researchers and industry practitioners. The significant growth in its user community, complemented by substantial user contributions, is a testament to the tool's utility and adaptability—various software tools and implementations of novel process mining approaches built upon PM4Py. PM4Py is committed to continued innovation and community engagement to consolidate further its role as a leading platform for process mining. Through this continual evolution, PM4Py aims to keep empowering its users to harness the potential of process mining.

Acknowledgments

The authors would like to thank everyone who contributed to the development and success of PM4Py. Especially the authors would like to express gratitude to the Fraunhofer FIT colleagues that were integral to the project's development (alphabetically sorted by surname): Wil M.P. van der Aalst, Alan Alrechah, Harry Beyel, Yukun Cao, Manjari Chaudhri, Radu-Andrei Coanda, Edgar Holzmann, Eunae Jang, Chiao-Yun Li, Giorgi Lomidze, Ralf Riesen, Lukas Schade, Rafael Schimassek, Julian Schnitzler, David Wenderdel, and Jingjing Xu.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W.M.P. van der Aalst, Process Mining - Data Science in Action, Second Edition, Springer, 2016, http://dx.doi.org/10.1007/978-3-662-49851-4.

[2] W.M.P. van der Aalst, Foundations of process discovery, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, Vol. 448, Springer, 2022, pp. 37–75, http://dx.doi.org/10.1007/978-3-031-08848-3_2.

[3] A. Augusto, J. Carmona, E. Verbeek, Advanced process discovery techniques, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, Vol. 448, Springer, 2022, pp. 76–107, http://dx.doi.org/10.1007/978-3-031-08848-3_3.

[4] J. Carmona, B.F. van Dongen, A. Solti, M. Weidlich, Conformance Checking - Relating Processes and Models, Springer, 2018, http://dx.doi.org/10.1007/978-3-319-99414-7.

[5] J. Carmona, B.F. van Dongen, M. Weidlich, Conformance checking: Foundations, milestones and challenges, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, Vol. 448, Springer, 2022, pp. 155–190, http://dx.doi.org/10.1007/978-3-031-08848-3_5.

[5] https://github.com/pm4py/pm4py-core

[6] According to PyPI (https://pepy.tech/project/pm4py) assessed on 31.07.2023

[6] M. de Leoni, Foundations of process enhancement, in: Process Mining Handbook, in: Lecture Notes in Business Information Processing, Vol. 448, Springer, 2022, pp. 243–273, http://dx.doi.org/10.1007/978-3-031-08848-3_8.

[7] A. Berti, S.J. van Zelst, W.M.P. van der Aalst, Process mining for python (PM4py): Bridging the gap between process- and data science, in: Proceedings of the 1st International Conference on Process Mining (ICPM 2019), Demo Track, CEUR, 2019, pp. 13–16, URL https://ceur-ws.org/Vol-2374/paper4.pdf.

[8] W.M.P. van der Aalst, T. Weijters, L. Maruster, Workflow mining: Discovering process models from event logs, IEEE Trans. Knowl. Data Eng. 16 (9) (2004) 1128–1142, http://dx.doi.org/10.1109/TKDE.2004.47.

[9] S.J.J. Leemans, D. Fahland, W.M.P. van der Aalst, Discovering block-structured process models from event logs - a constructive approach, in: Application and Theory of Petri Nets and Concurrency - 34th International Conference, PETRI NETS 2013, Milan, Italy, June 24-28, 2013. Proceedings, in: Lecture Notes in Computer Science, Vol. 7927, Springer, 2013, pp. 311–329, http://dx.doi.org/10.1007/978-3-642-38697-8_17.

[10] S.J.J. Leemans, D. Fahland, W.M.P. van der Aalst, Scalable process discovery and conformance checking, Softw. Syst. Model. 17 (2) (2018) 599–631, http://dx.doi.org/10.1007/s10270-016-0545-x.

[11] S.J.J. Leemans, D. Fahland, W.M.P. van der Aalst, Discovering block-structured process models from event logs containing infrequent behaviour, in: Business Process Management Workshops - BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers, in: Lecture Notes in Business Information Processing, Vol. 171, Springer, 2013, pp. 66–78, http://dx.doi.org/10.1007/978-3-319-06257-0_6.

[12] A.J.M.M. Weijters, J.T.S. Ribeiro, Flexible heuristics miner (FHM), in: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, Part of the IEEE Symposium Series on Computational Intelligence 2011, April 11-15, 2011, Paris, France, IEEE, 2011, pp. 310–317, http://dx.doi.org/10.1109/CIDM.2011.5949453.

[13] S.J. van Zelst, B.F. van Dongen, W.M.P. van der Aalst, ILP-based process discovery Using Hybrid Regions, in: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, ATAED 2015, Satellite Event of the Conferences: 36th International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2015 and 15th International Conference on Application of Concurrency To System Design ACSD 2015, Brussels, Belgium, June 22-23, 2015, in: CEUR Workshop Proceedings, Vol. 1371, CEUR-WS.org, 2015, pp. 47–61, URL https://ceur-ws.org/Vol-1371/paper04.pdf.

[14] S. Pourmirza, R.M. Dijkman, P. Grefen, Correlation miner: Mining business process models and event correlations without case identifiers, Int. J. Cooperative Inf. Syst. 26 (2) (2017) 1742002:1–1742002:32, http://dx.doi.org/10.1142/S0218843017420023.

[15] M.R. Przybylek, Skeletal algorithms in process mining, in: Computational Intelligence: Revised and Selected Papers of the International Joint Conference, IJCCI 2011, Paris, France, October 24-26, 2011, Springer, 2013, pp. 119–134, URL https://link.springer.com/chapter/10.1007/978-3-642-35638-4_9.

[16] W.M.P. van der Aalst, Object-centric process mining: Dealing with divergence and convergence in event data, in: Software Engineering and Formal Methods, in: Lecture Notes in Computer Science, Vol. 11724, Springer International Publishing, Cham, 2019, pp. 3–25, http://dx.doi.org/10.1007/978-3-030-30446-1_1.

[17] A. Berti, W.M.P. van der Aalst, OC-PM: analyzing object-centric event logs and process models, Int. J. Softw. Tools Technol. Transf. 25 (1) (2023) 1–17, http://dx.doi.org/10.1007/s10009-022-00668-w.

[18] W.M.P. van der Aalst, A. Berti, Discovering object-centric Petri nets, Fundam. Informaticae 175 (1–4) (2020) 1–40, http://dx.doi.org/10.3233/FI-2020-1946.

[19] A. Berti, J. Herforth, M.S. Qafari, W.M.P. van der Aalst, Graph-based feature extraction on object-centric event logs, Int. J. Data Sci. Analyt. (2023) http://dx.doi.org/10.1007/s41060-023-00428-2.

[20] A. Berti, W.M.P. van der Aalst, A novel token-based replay technique to speed up conformance checking and process enhancement, Trans. Petri Nets Other Model. Concurr. 15 (2021) 1–26, http://dx.doi.org/10.1007/978-3-662-63079-2_1.

[21] A. Adriansyah, B.F. van Dongen, W.M.P. van der Aalst, Conformance checking using cost-based fitness analysis, in: Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2011, Helsinki, Finland, August 29 - September 2, 2011, IEEE Computer Society, 2011, pp. 55–64, http://dx.doi.org/10.1109/EDOC.2011.12.

[22] W.L.J. Lee, H.M.W. Verbeek, J. Munoz-Gama, W.M.P. van der Aalst, M. Sepúlveda, Recomposing conformance: Closing the circle on decomposed alignment-based conformance checking in process mining, Inform. Sci. 466 (2018) 55–91, http://dx.doi.org/10.1016/j.ins.2018.07.026.

[23] H.M.W. Verbeek, The log skeleton visualizer in prom 6.9, Int. J. Softw. Tools Technol. Transf. 24 (4) (2022) 549–561, http://dx.doi.org/10.1007/s10009-021-00618-y.

[24] S.J.J. Leemans, W.M.P. van der Aalst, T. Brockhoff, A. Polyvyanyy, Stochastic process mining: Earth movers' stochastic conformance, Inf. Syst. 102 (2021) 101724, http://dx.doi.org/10.1016/j.is.2021.101724.

[25] W.M.P. van der Aalst, H.T. de Beer, B.F. van Dongen, Process mining and verification of properties: An approach based on temporal logic, in: On the Move To Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, OTM

Confederated International Conferences CoopIS, DOA, and ODBASE 2005, Agia Napa, Cyprus, October 31 - November 4, 2005, Proceedings, Part I, in: Lecture Notes in Computer Science, Vol. 3760, Springer, 2005, pp. 130–147, http://dx.doi.org/10.1007/11575771_11.

[26] J.C.A.M. Buijs, B.F. van Dongen, W.M.P. van der Aalst, Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity, Int. J. Cooperative Inf. Syst. 23 (1) (2014) http://dx.doi.org/10.1142/S0218843014400012.

[27] J. Munoz-Gama, J. Carmona, A fresh look at precision in process conformance, in: Business Process Management - 8th International Conference, BPM 2010, Hoboken, NJ, USA, September 13-16, 2010. Proceedings, in: Lecture Notes in Computer Science, Vol. 6336, Springer, 2010, pp. 211–226, http://dx.doi.org/10.1007/978-3-642-15618-2_16.

[28] A. Adriansyah, J. Munoz-Gama, J. Carmona, B.F. van Dongen, W.M.P. van der Aalst, Measuring precision of modeled behavior, Inf. Syst. E Bus. Manag. 13 (1) (2015) 37–67, http://dx.doi.org/10.1007/s10257-014-0234-7.

[29] B. Vázquez-Barreiros, M. Mucientes, M. Lama, ProDiGen: Mining complete, precise and minimal structure process models with a genetic algorithm, Inform. Sci. 294 (2015) 315–333, http://dx.doi.org/10.1016/j.ins.2014.09.057.

[30] T. Chatain, M. Boltenhagen, J. Carmona, Anti-alignments - measuring the precision of process models and event logs, Inf. Syst. 98 (2021) 101708, http://dx.doi.org/10.1016/j.is.2020.101708.

[31] M. Boltenhagen, T. Chatain, J. Carmona, Optimized SAT encoding of conformance checking artefacts, Computing 103 (1) (2021) 29–50, http://dx.doi.org/10.1007/s00607-020-00831-8.

[32] T. Jouck, B. Depaire, PTandLogGenerator: A generator for artificial event data, in: 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, in: CEUR Workshop Proceedings, Vol. 1789, CEUR-WS.org, 2016, pp. 23–27, URL https://ceur-ws.org/Vol-1789/bpm-demo-2016-paper5.pdf.

[33] M. de Leoni, W.M.P. van der Aalst, Data-aware process mining: discovering decisions in processes using alignments, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013, ACM, 2013, pp. 1454–1461, http://dx.doi.org/10.1145/2480362.2480633.

[34] H.M.W. Verbeek, T. Basten, W.M.P. van der Aalst, Diagnosing workflow processes using woflan, Comput. J. 44 (4) (2001) 246–279, http://dx.doi.org/10.1093/comjnl/44.4.246.

[35] S.J. van Zelst, Y. Cao, A generic framework for attribute-driven hierarchical trace clustering, in: Business Process Management Workshops - BPM 2020 International Workshops, Seville, Spain, September 13-18, 2020, Revised Selected Papers, in: Lecture Notes in Business Information Processing, Vol. 397, Springer, 2020, pp. 308–320, http://dx.doi.org/10.1007/978-3-030-66498-5_23.

[36] W.M.P. van der Aalst, H.A. Reijers, M. Song, Discovering social networks from event logs, Comput. Support. Cooperative Work. 14 (6) (2005) 549–593, http://dx.doi.org/10.1007/s10606-005-9005-9.

[37] A. Burattin, A. Sperduti, M. Veluscek, Business models enhancement through discovery of roles, in: IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, 16-19 April, 2013, IEEE, 2013, pp. 103–110, http://dx.doi.org/10.1109/CIDM.2013.6597224.

[38] A. Pika, M. Leyer, M.T. Wynn, C.J. Fidge, A.H.M. ter Hofstede, W.M.P. van der Aalst, Mining resource profiles from event logs, ACM Trans. Manag. Inf. Syst. 8 (1) (2017) 1:1–1:30, http://dx.doi.org/10.1145/3041218.

[39] N. Martin, M. Swennen, B. Depaire, M. Jans, A. Caris, K. Vanhoof, Batch processing: Definition and event log identification, in: Proceedings of the 5th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2015), Vienna, Austria, December 9-11, 2015, in: CEUR Workshop Proceedings, Vol. 1527, CEUR-WS.org, 2015, pp. 137–140, URL https://ceur-ws.org/Vol-1527/paper11.pdf.

[40] S.A. Fahrenkrog-Petersen, H. van der Aa, M. Weidlich, PRIPEL: privacy-preserving event log publishing including contextual information, in: Business Process Management - 18th International Conference, BPM 2020, Seville, Spain, September 13-18, 2020, Proceedings, in: Lecture Notes in Computer Science, Vol. 12168, Springer, 2020, pp. 111–128, http://dx.doi.org/10.1007/978-3-030-58666-9_7.

[41] G. Acampora, A. Vitiello, B. Di Stefano, W.M.P. van der Aalst, C. Gunther, E. Verbeek, IEEE 1849: The XES standard: The second IEEE standard sponsored by IEEE computational intelligence society [society briefs], IEEE Comput. Intell. Mag. 12 (2) (2017) 4–8, URL https://ieeexplore.ieee.org/document/7895272.

[42] A.F. Ghahfarokhi, G. Park, A. Berti, W.M.P. van der Aalst, OCEL: a standard for object-centric event logs, in: New Trends in Database and Information Systems - ADBIS 2021 Short Papers, Doctoral Consortium and Workshops: DOING, SIMPDA, MADEISD, MegaData, CAoNS, Tartu, Estonia, August 24-26, 2021, Proceedings, in: Communications in Computer and Information Science, Vol. 1450, Springer, 2021, pp. 169–175, http://dx.doi.org/10.1007/978-3-030-85082-1_16.

[43] L. Hillah, F. Kordon, L. Petrucci, N. Trèves, PNML framework: An extendable reference implementation of the Petri net markup language, in: Applications and Theory of Petri Nets, 31st International Conference, PETRI NETS 2010, Braga, Portugal, June 21-25, 2010. Proceedings, in: Lecture Notes in Computer Science, Vol. 6128, Springer, 2010, pp. 318–327, http://dx.doi.org/10.1007/978-3-642-13675-7_20.

[44] S.J.J. Leemans, Robust Process Mining with Guarantees - Process Discovery, Conformance Checking and Enhancement, in: Lecture Notes in Business Information Processing, Vol. 440, Springer, 2022, http://dx.doi.org/10.1007/978-3-030-96655-3.

[45] G. Aagesen, J. Krogstie, BPMN 2.0 for modeling business processes, in: Handbook on Business Process Management 1, Introduction, Methods, and Information Systems, second ed., in: International Handbooks on Information Systems, Springer, 2015, pp. 219–250, http://dx.doi.org/10.1007/978-3-642-45100-3_10.

[46] V. Denisov, E. Belkina, D. Fahland, W.M.P. van der Aalst, The performance spectrum miner: Visual analytics for fine-grained performance analysis of processes, in: Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 Co-Located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018, in: CEUR Workshop Proceedings, Vol. 2196, CEUR-WS.org, 2018, pp. 96–100, URL https://ceur-ws.org/Vol-2196/BPM_2018_paper_20.pdf.

[47] Massimiliano de Leoni, Felix Mannhardt, Road Traffic Fine Management Process - Event Log, Eindhoven University of Technology, 2015, http://dx.doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5.

[48] J.L. Lay, J. Neveu, B. Dalmas, V. Augusto, Automated generation of patient population for discrete-event simulation using process mining, in: Annual Modeling and Simulation Conference, ANNSIM 2022, San Diego, CA, USA, July 18-20, 2022, IEEE, 2022, pp. 42–53, http://dx.doi.org/10.23919/ANNSIM55834.2022.9859406.

[49] M. Werner, M. Wiese, A. Maas, Embedding process mining into financial statement audits, Int. J. Account. Inf. Syst. 41 (2021) 100514, http://dx.doi.org/10.1016/j.accinf.2021.100514.

[50] M. Macák, I. Vanat, M. Merjavy, T. Jevocin, B. Buhnova, Towards process mining utilization in insider threat detection from audit logs, in: Seventh International Conference on Social Networks Analysis, Management and Security, SNAMS 2020, Virtual Event, France, December 14-16, 2020, IEEE, 2020, pp. 1–6, http://dx.doi.org/10.1109/SNAMS52053.2020.9336573.

[51] W. Rizzi, L. Simonetto, C.D. Francescomarino, C. Ghidini, T. Kasekamp, F.M. Maggi, Nirdizati 2.0: New features and redesigned backend, in: Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 Co-Located with 17th International Conference on Business Process Management, BPM 2019, Vienna, Austria, September 1-6, 2019, in: CEUR Workshop Proceedings, Vol. 2420, CEUR-WS.org, 2019, pp. 154–158, URL https://ceur-ws.org/Vol-2420/paperDT8.pdf.

[52] H. van der Aa, A. Rebmann, H. Leopold, Natural language-based detection of semantic execution anomalies in event logs, Inf. Syst. 102 (2021) 101824, http://dx.doi.org/10.1016/j.is.2021.101824.

[53] J. Peeperkorn, S. vanden Broucke, J.D. Weerdt, Supervised conformance checking using recurrent neural network classifiers, in: Process Mining Workshops - ICPM 2020 International Workshops, Padua, Italy, October 5-8, 2020, Revised Selected Papers, in: Lecture Notes in Business Information Processing, Vol. 406, Springer, 2020, pp. 175–187, http://dx.doi.org/10.1007/978-3-030-72693-5_14.

[54] J. Friederich, D.P. Francis, S. Lazarova-Molnar, N. Mohamed, A framework for data-driven digital twins for smart manufacturing, Comput. Ind. 136 (2022) 103586, URL https://www.researchgate.net/profile/Sanja-Lazarova-Molnar/publication/357150951_A_framework_for_data-driven_digital_twins_of_smart_manufacturing_systems/links/61ea35a6dafcdb25fd3dd175/A-framework-for-data-driven-digital-twins-of-smart-manufacturing-systems.pdf.

[55] A. Yeshchenko, C.D. Ciccio, J. Mendling, A. Polyvyanyy, Visual drift detection for event sequence data of business processes, IEEE Trans. Vis. Comput. Graph. 28 (8) (2022) 3050–3068, http://dx.doi.org/10.1109/TVCG.2021.3050071.

[56] A. Berti, C.-Y. Li, D. Schuster, S.J. van Zelst, The process mining ToolKit (PMTK): Enabling advanced process mining in an integrated fashion, in: Proceedings of the ICPM Doctoral Consortium and Demo Track 2021, in: CEUR Workshop Proceedings, CEUR Workshop Proceedings, 2021, pp. 43–44, URL https://ceur-ws.org/Vol-3098/demo_206.pdf.

[57] G. Park, J.N. Adams, W.M.P. van der Aalst, Opera: Object-centric performance analysis, in: Conceptual Modeling - 41st International Conference, ER 2022, Hyderabad, India, October 17-20, 2022, Proceedings, in: Lecture Notes in Computer Science, Vol. 13607, Springer, 2022, pp. 281–292, http://dx.doi.org/10.1007/978-3-031-17995-2_20.

[58] D. Schuster, S.J. van Zelst, W.M. van der Aalst, Cortado: a dedicated process mining tool for interactive process discovery, SoftwareX 22 (2023) 101373, http://dx.doi.org/10.1016/j.softx.2023.101373.

[59] G. Park, W.M.P. van der Aalst, Realizing a digital twin of an organization using action-oriented process mining, in: 3rd International Conference on Process Mining, ICPM 2021, Eindhoven, the Netherlands, October 31 - Nov. 4, 2021, IEEE, 2021, pp. 104–111, http://dx.doi.org/10.1109/ICPM53251.2021.9576846.

[60] G. Park, M. Comuzzi, W.M.P. van der Aalst, Analyzing process-aware information system updates using digital twins of organizations, in: Research Challenges in Information Science - 16th International Conference, RCIS 2022, Barcelona, Spain, May 17-20, 2022, Proceedings, in: Lecture Notes in Business Information Processing, Vol. 446, Springer, 2022, pp. 159–176, http://dx.doi.org/10.1007/978-3-031-05760-1_10.