Quality assessment of discovered process models in Process Mining: the case of Process Trees

Cristina-Claudia Osman

Abstract — **Daily activities of companies generate and consume massive amounts of data. Different diagrammatic visualizations can be extracted from this data by using different Process Mining algorithms. ProM Framework provides several discovery Process Mining algorithms, mainly focused on the control-flow perspective. This paper analyses the algorithms whose output is either a Process Tree (PT), or an Efficient Process Tree (EPT). The results of several Process Mining algorithms are analyzed and qualitatively evaluated. Precision, Scaled Precision, and Fitness metrics are used for evaluating the resulted diagrammatic visualizations. Moreover, two variations of F-score are also introduced for determining the global quality of the models. The analysis considers, on one hand, two algorithms whose output is a PT and, on the other hand, five versions of an algorithm whose output is an EPT. The findings of this investigation show slightly better results on EPT compared to PT. However, the choice of the most suitable algorithm depends on the analysis type (process discovery, process improvement, audit, risk identification, etc.).**

Keywords— **Process Mining, Process Trees, Efficient Process Trees, Quality of process models, Projected Fitness, (Scaled) Precision**

I. INTRODUCTION

Massive amounts of data are stored within the Information Systems used by companies in their daily activities. This data can be converted into knowledge by using different discovering Process Mining algorithms. Process Mining is the domain that incorporates methods and techniques that a) discover diagrammatic visualizations from event logs, b) compare the discovered diagrammatic visualizations with the event log, and c) improve the discovered process models (by using prediction, enrichment of semantics, etc.) [1]. There are several use cases of Process Mining. For example, a company wants to reorganize the *Order-to-Pay* process. It can start from the existing processes and after applying suitable Process Mining and Business Process Management techniques, the existing process can be improved. But the discovery of diagrammatic visualizations can be performed even when the process is not known. For example, when two companies merge, the existing processes can be discovered by using the existing event logs. After processes are discovered, they can be compared with the existing procedures. Afterwards, improvements of the discovered processes can be suggested.

First Process Mining discovery algorithm, Alpha Miner, generates a Petri Net [2]. Afterwards, several Process Mining discovery algorithms have been developed [3]-[24]. The quality of a discovered diagrammatic visualization is measured using metrics like *Fitness*, *Precision*, *Generalization* and *Simplicity* [25]. *Fitness* measures how much the process model reproduces the event log. *Precision*, on the other hand, seeks for the behaviour captured by the process model which is not described by the event log. *Generalization* refers to capacity of the process model to support new behaviour, while *Simplicity* assesses the complexity of process models and how human readable they are. Besides simplicity, which is measured using only the process model, all quality metrics are computed using both, event log and process model.

Usually, only *Fitness* is considered for evaluating the quality of discovered process models. Although, an analysis of four Process Mining discovery algorithms whose output is a Petri Net is detailed in [26]. The algorithms considered for the study are: Alpha Miner [2], Alpha# Miner [3], Inductive Miner (IM) [4], respectively ILP Miner [5], [11]. Their quality is measured by equally weighting *Fitness* and *Precision*. The Petri Net measuring the highest overall measure is the one discovered by Inductive Miner (IM). The Petri Net discovered by IM in sound, but soundness is not guaranteed by all Process Mining algorithms. Process Trees (PTs) are process models that guarantee soundness. Therefore, the processes do not contain deadlocks. This paper provides a comparison of the algorithms providing PTs and EPTs, respectively. The analysis uses an event log describing an electronic invoicing process [27].

In this study, *F-score* from Information Retrieval [31] is used for measuring the quality of discovered process models. De Weerdt et al. [28] used for the first time, *F-score* in the context of evaluating discovered process model. The authors of the study used *F-score* in the context of Petri Nets. The current research employs a similar approach by considering *Projected Fitness*, *Precision* and *Scaled Precision* [29].

C.-C. Osman is a Teaching Assistant at Business Information Systems Department, Faculty of Economics and Business Administration, Babeș-Bolyai University, Cluj-Napoca, Romania. She also is a member of Business Informatics Research Center, Babeș-Bolyai University, Cluj-Napoca, Romania (corresponding author tel.: +40264-41 86 55; fax: +40264 41 25 70; e-mail address: cristina.osman@econ.ubbcluj.ro).

The reminder of this paper is structured as follows: Second Section presents a short introduction to ProM Framework and provides details about the event log used for measuring the quality of discovered Process Trees (PT) / Efficient Process Trees (EPT). Afterwards, Process Trees and their characteristics are depicted, together with the discovery algorithms whose output is a PT with focus on *Evolutionary Tree Miner Discovery* (*ETMd*) [16] and *Inductive Miner* (*IM*) [18]. Fourth section introduces the main contribution of this paper. Two evaluation metrics based on F-score [28] are presented, and a comparison between the algorithms providing PTs and EPTs is detailed. Final section exposes a summary of the findings.

II. ProM Framework and summary of the event log

A. *ProM Framework*

ProM is an open source framework that incorporates over 600 Process Mining plugins. Most Process Mining algorithms implemented in ProM focus on discovering diagrammatic visualizations. The output considers standardized notations like Petri Nets [2]-[5], 11], BPMN diagrams [7], or specific notations like Fuzzy Nets [8], Heuristics Nets [6], Process Trees [15]-[18], [21]-[24], Product Data Models [13], [14], or Social Networks [9]-[10]. Different types of plugins can be used, depending on the desired output.

The discovery algorithms focus on control-flow [2]-[8], [11], [15]-[24], data [12]-[14] or resources perspectives [9], [10]. The focus of this study is on evaluating the quality of discovered PTs and EPTs, respectively.

B. *Event Log Summary*

The event log used for this study depicts an electronic invoicing process and the tool used is ProM Framework 6.8 [30], revision 38904. A similar approach based on the same event log is used in [26], where Process Mining algorithms returning Petri Nets are examined.

Table I. Activities

Activity	**Number of occurrences**
Approve Invoice	22687
Approve Liquidated Invoices	18532
End	20135
Invoice Scanning	20135
Liquidation	21084
Marking Paid Invoices	15905
Payment Approval	15905
Register	20135
Scanning of Extra Documentation	20135

The event log describing the electronic invoicing process consists of 20135 cases and 309036 events. The process takes places over a month. This study considers only finished activities; therefore, the number of events reduces to 174653 events mapped to 9 activities, including one artificial activity (*End*). More details about the activities and their occurrences within the analysed process are depicted in Table I.

The activities are executed by 6 types of resources (from *group 1* to *group 6*). The resources performing the artificial activity *End* belong to groups 2 and 6 (they are assigned based on the previous executed activity: *Approve Invoice*, and *Marking Paid Invoices* respectively). Each case starts with *Register* activity, performed by the *System*.

Table II. Resources

Resource	**Number of occurrences in the event log**
group 1	40270
group 2	26917
group 3	21084
group 4	18532
group 5	15905
group 6	31810
System	20135

III. Discovery of Process Trees (PTs) and Efficient Process Trees (EPTs)

A. *Process Trees (PTs)*

A Process Tree (PT) is a process model represented as a directed connected graph without cycles [20]. Moreover, PTs are sounded models. The operators used by PTs are depicted in Table III. First six operators are defined in [17] and [20], while the last two are introduced in [18].

All the nodes of a Process Tree have a unique name and each leaf represents an activity. The other nodes are represented by the operators reminded earlier.

Table III. Process trees symbols and their meaning

Symbol	**Significance**
$\times$	Exclusive choice (xor)
$\rightarrow$	Sequence (seq)
$\leftarrow$	Reversed sequence
$\wedge$	Parallelism
$\vee$	Non-inclusive (exclusive) choice
$\circlearrowleft$	Loop
τ	Silent activity
$\leftrightarrow$	Interleaved

B. *PTs and EPTs in ProM Framework*

There are three Process Mining algorithms that generate PTs from event logs: a) *Evolutionary Tree Miner Discovery* (*ETMd*), b) *Inductive Miner* (*IM*) and c) *Trace Miner* (*TM*).

Indeed, there exists one more plugin that uses PTs, but the extracted process models are Petri Nets or Directly-Follows Graphs -*Local Process Models* (*LPMs*) [19]. LPMs are process models that describe the most frequent behaviour. Thus, not all possible traces appear into the diagrammatic visualization. Quality of LPMs is measured using five metrics: support, confidence, language fit, coverage, and determinism. But LPMs are not the subject of this study as the targeted quality metrics are: *Projected*

Fitness and *(Scaled) Precision* [25]. The other two metrics used for measuring process models' quality are generalization and simplicity.

Moreover, *Indulpet Miner* provides EPTs by combining different Process Mining discovery algorithms.

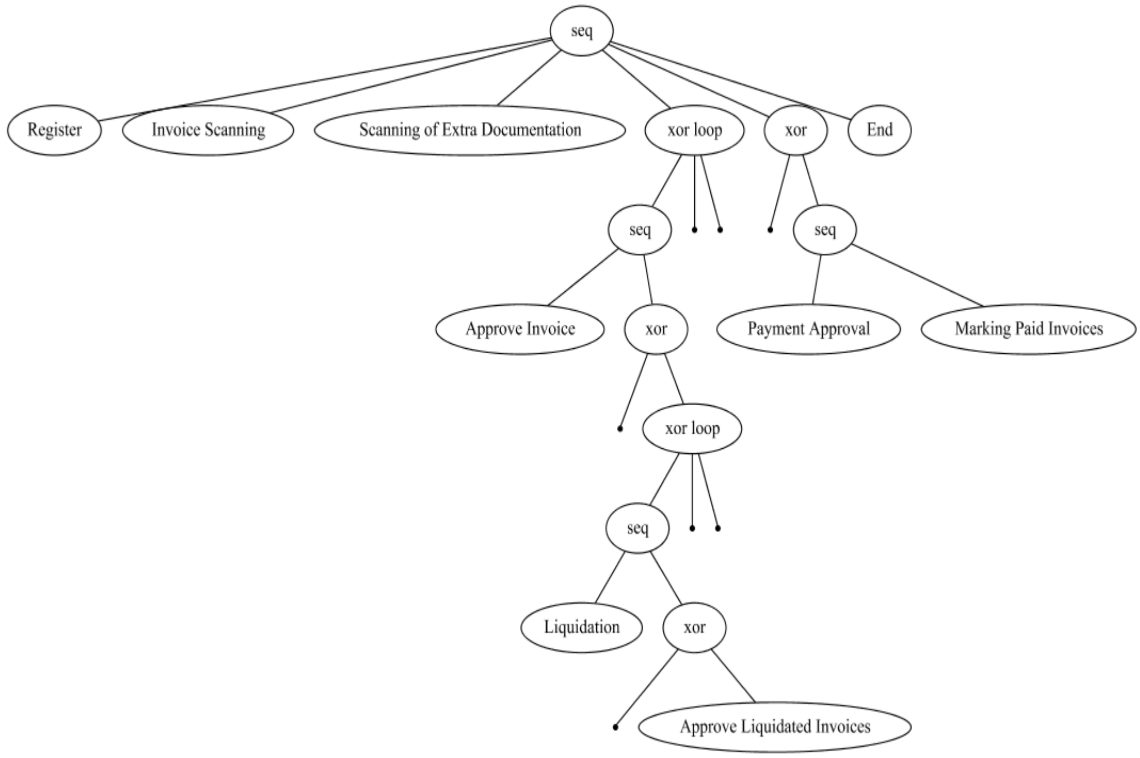

Fig. 1 Discovered process tree using IMA

1) Evolutionary Tree Miner Discovery (ETMd)

ETM (Evolutionary Tree Miner) can be used in tasks such as PTs discovery (ETMd) [16],[17]; configurable PTs discovery (ETMc) [17],[21]; and process repair (ETMr) [17].

ETMd is a genetic algorithm that primarily creates an initial population of candidate solutions [16],[17]. Afterwards, each candidate solution is evaluated using process model quality metrics (*Fitness, Simplicity, Generalization* and *Precision*). Best candidates are stored into a collection called elite. The process runs until conditions are satisfied (such as, the number of generations is reached or perfect candidate is found, etc.).

2) Inductive Miner (IM)

The processes resulted using *Inductive Miner* (*IM*) are also sound, but contrarily to *ETMd*, *IM* returns a sound finite model in a finite run time. Another difference between *ETMd* and *IM* is given by the fact that the event log is decomposed into block-structured parts prior to PT construction. There exists several variants of *IM*: *IM – infrequent* [22], *IM – incompleteness* [22], *IM – exhaustive K-successor* [23], *IMlc – life cycle* [24], *IMilc – infrequent & life cycle* [24], IM_A *– all operators* [18] and *IMfa – infrequent & all operators* [18].

The variant considered for this study is (IM_A) [18] which includes silent activities, interleaved and inclusive choice operators.

3) Trace Miner

Trace Miner plugin uses a naïve algorithm which generates a PT based on Process variants. Therefore, activities are duplicated and their graphical visualization is provided by sequence and exclusive choice operators.

Subsequently, *Projected Fitness*, *Precision* and *Scaled Precision* are equal to 1. Because of these reasons, PTs provided by this algorithm are not included in this study.

4) Indulpet Miner (IN)

Indulpet Miner (IN) is an algorithm whose output is an EPT. It combines different Process Mining discovery algorithms: *IM*, *LPMs* and *ETM* together with a new bottom-up recursive technique (*BUR*) [15].

IM is used for fitness reasons, *BUR* is used to find lowest-level structure in the log, *LPMs* compute candidate process models which serves as initial population for *ETM*. Consequently, the soundness of the models discovered using IN is guaranteed.

IV. Evaluation of Algorithms

A. Process Trees Evaluation

ETMd algorithm is run for 1000 generations having a population size of 20, and an elite of 5, while the event classifier is the event name. The evaluator included into the algorithm is *Precision – Costs per node*, with a *Fitness* target of 1. The second evaluated PT is discovered using IMA. This variant of IM uses ×,→, ∧, ↺, τ, ↔, and ∨ operators.

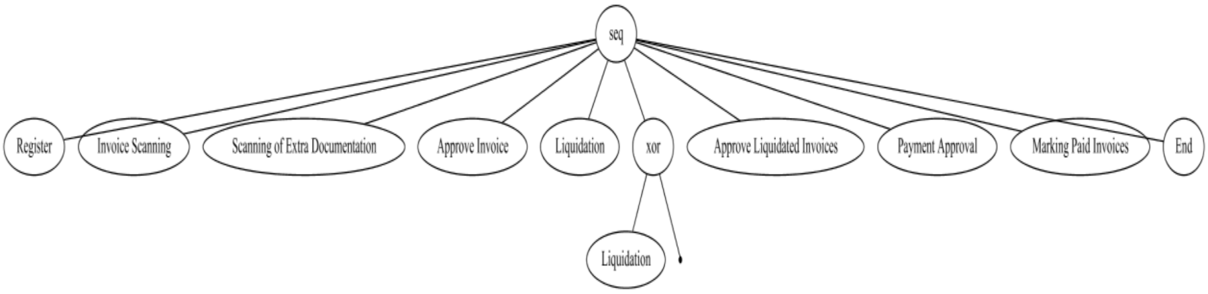

Fig. 2 Discovered process tree using *ETMd – costs*

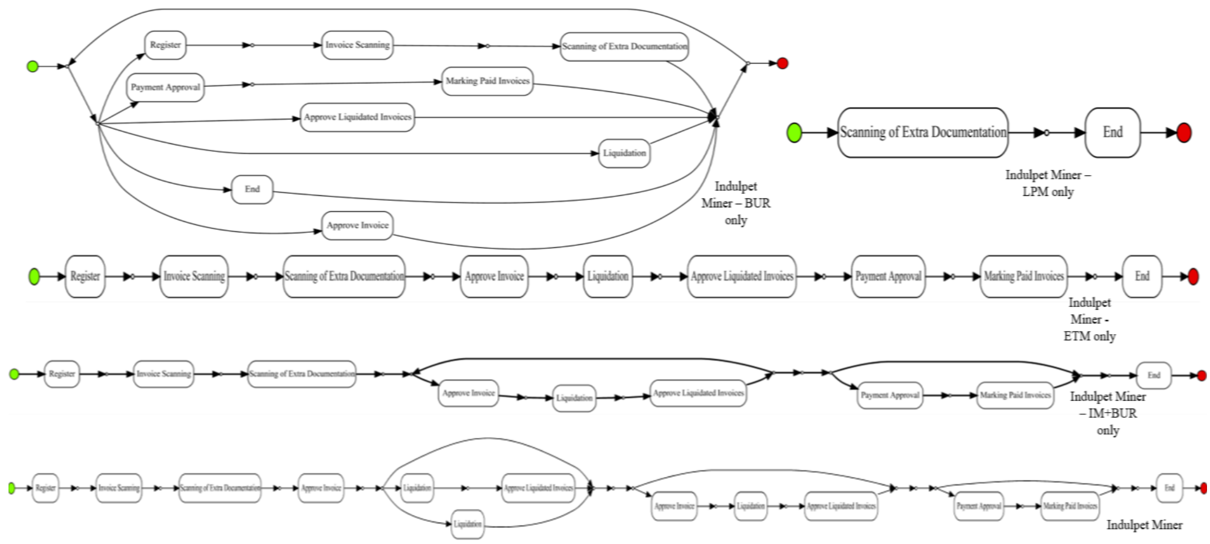

Fig. 3 Discovered efficient trees using different variants of IN

IMA identifies 9 PT operators, while *ETMd* only 2 (see Fig. 1 and Fig. 2). Both algorithms discover all the activities from the event log. However, *ETMd* duplicates *Liquidation* activity (see Fig. 2).

De Weerdt et al. [28] proposed F-score from Information Retrieval [31] for measuring the quality of discovered process models. Their approach uses artificially generated negative events. For this study we include a variation of *F-score* (*F-score`*) by considering *Fitness* and *Scaled Precision* (see equation 1). *Scaled Precision* is defined in [29] and it measures the linear precision improvement compared to a flower model.

Precision of the model and *Precision* of the flower model are used for the calculation of *Scaled Precision* metric. *F-score`* is computed using a size of projection of 2 for both PTs. *Fitness* is computed based on cost functions which measure the severity of movements in alignments [32].

$$F\text{-}score` = 2 \cdot \frac{Scaled\ Precision \cdot Projected\ Fitness}{Scaled\ Precision + Projected\ Fitness} \quad (1)$$

For this study, we use a second variation of F-score (*F-score``*) by using *Projected Fitness* and *Precision* (see equation 2). *Projected Fitness* is the *Fitness* metric of the projected PT [18]. It is computed by mapping the projected traces of the log on the projected PT. The projection of a PT is represented by a set of activities where every leaf which does not belong to the set of activities is replaced by τ. A deterministic finite automata (DFN) is generated for the model and for the log, followed by a conjunction between the behaviour permitted by both log and model generated previously is built. The aim is to catch common behaviour. *Precision* is calculated based on conjunction automaton and model automaton, for each subset of size *k*.

$$F\text{-}score`` = 2 \cdot \frac{Precision \cdot Projected\ Fitness}{Precision + Projected\ Fitness} \quad (2)$$

The *PT* discovered using *ETMd* records the same values for both *F-scores* (0.883), while the *PT* discovered using IMA registers a 6% higher *F-score``*. Although, *F-score`* values in both cases are similar (0.889 versus 0.883). The PT discovered using IMA assures a perfect fitness, while the PT discovered using *ETMd* focuses on *Precision*. Therefore, the choice of the most suitable algorithm depends on the purposes of the analysis.

Table IV. Evaluation metrics of Process Trees

Algorithm / Evaluation metric	Inductive Miner - all operators (IMA)	Evolutionary Tree Miner Discovery (ETMd)
Projected Fitness	1	0.791
Precision	0.899	1

Scaled-Precision	0.801	0.791
F-score`	0.889	0.883
F-score``	0.947	0.883

B. *Efficient Process Trees Evaluation*

For EPT evaluation, we run all the versions of *IN*. For the default version of IN, we use a noise threshold of 20% *concept:name* from XES Standard [33] as event classifier. Figure 3 depicts the process trees discovered using all the version of IN (BUR only, LPM only, ETM only, IM+BUR only, default version of IN).

Table V. Evaluation metrics of Efficient Process Trees

Algorithm / Evaluation metric	Projected Fitness	Precision	F-score``
Indulpet Miner - default version	0.993	0.933	0.962
Indulpet Miner – bottom-up only	1.000	0.494	0.661
Indulpet Miner – ETM only	0.741	1.000	0.851
Indulpet Miner – IM+ bottom-up only	0.898	0.861	0.879
Indulpet Miner – LPM only	0.109	0.417	0.173

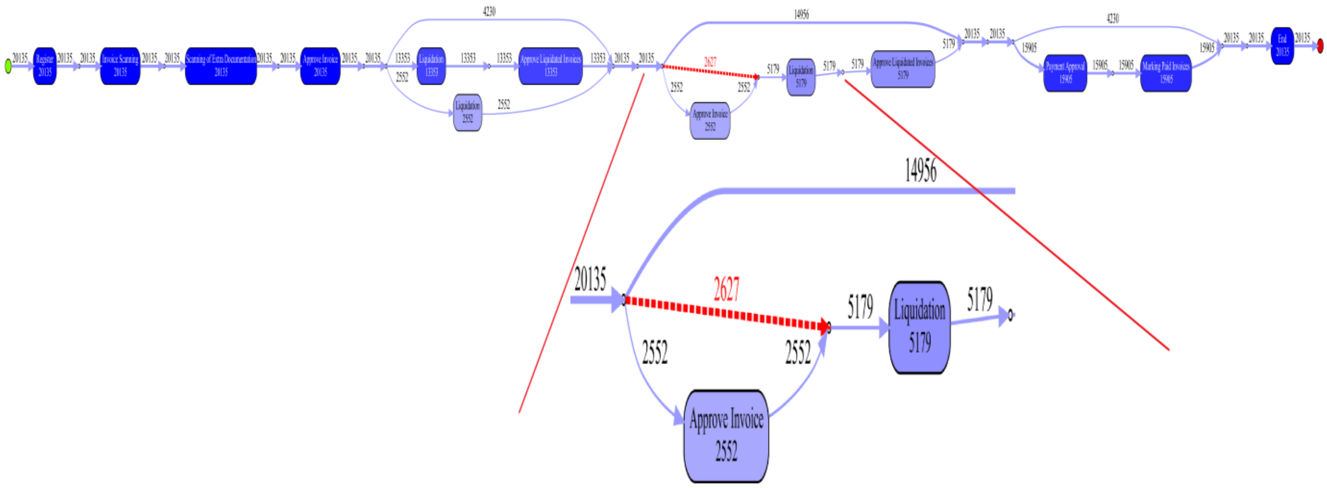

Fig. 4 Efficient tree alignment for IN (default version)

Projected Fitness and *Precision* of the discovered EPTs are computed using a size projection of 2. The percentages of *Projected Fitness* and *Precision* can be consulted in Table V. The model having the greatest *Fitness* is the one discovered using BUR, while the best *Precision* is computed for the model discovered using ETM only. Although, a *Perfect Fitness* results when BUR is used, the *Precision* of the model is only 0.494. The lowest metrics are computed when IN employs only LPM because LPMs focus on discovering frequent patterns.

The highest *F-score``* is registered by IN when it incorporates the advantages brought by IM, BUR, LPM and ETM (0.962). Again, the choice of the most suitable algorithm depends on the purposes of the analysis.

Best efficient tree alignment [34] is depicted in Fig. 4. The alignment shows a model move deviation before *Liquidation* activity, specifically *Approve Invoice* activity is optional in 2627 of cases. Therefore, *Projected Fitness* is not 1.

V. Conclusion

Process Mining discovery intends to extract diagrammatic visualizations from event logs. This paper analyses ProM algorithms whose output is either a Process Tree (PT) or an Efficient Process Tree (EPT).

For global quality assessment of process models, we introduced two measurements based on *F-score* [28]. Resulted PTs and EPTs are evaluated using metrics such as *Precision*, *Scaled Precision*, and *Fitness*. On one hand, we considered two algorithms providing PTs (ETMd and IM, respectively), and, on the other hand, we analysed the EPTs generated by Indulpet (default version, BUR only, ETM only, IM+BUR, LPM only, respectively). Concerning discovered PTs, both *F-scores* show minor differences

(0.006 on *F-score*`, and 0.064 on *F-score*``, respectively). On the other hand, discovered EPT records best quality when Indulpet Miner (default version) is used (0.962). Although, perfect *Projected Fitness* is registered when Indulpet Miner– BUR only is applied. Moreover, maxim *Precision* is recorded on the EPTs generated by Indulpet Miner – ETM.

Overall, the findings of this study show slightly better results on EPTs compared then on PTs (0.962 vs. 0.947). Nevertheless, the choice of the most suitable algorithm depends on the aim of the investigation (process discovery, process improvement, audit, risk identification, etc.).

REFERENCES

[1] W. M. P van der Aalst, Process Mining: Data science in action, Springer, Berlin, Heidelberg, 2016.

[2] W. M. P. van der Aalst, T. Weijters, and L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128-1142, 2004.

[3] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huan, and J. Sun, “Mining process models with prime invisible tasks,” *Data & Knowledge Engineering*, vol. 69, no. 10, pp. 999- 1021, 2010.

[4] S. J. J. Leemans, D. Fahland, and W. M .P. van der Aalst, “Process and deviation exploration with Inductive visual Miner,” in *CEUR Workshop Proceedings of the BPM Demo Sessions 2014 Co-located with the 12th International Conference on Business Process Management,* vol. 1295, Eindhoven, 2014, pp. 46-50.

[5] J. M. E. Van der Werf, B. F. van Dongen, C .A. Hurkens, and A. Serebrenik, “Process discovery using integer linear programming,” in *Applications and Theory of Petri Nets. PETRI NETS 2008. Lecture Notes in Computer Science,* vol. 5062, Springer, Berlin, Heidelberg, Xi’an, 2008, pp. 368-387.

[6] A. J. M. M. Weijters, J. T. S. Ribeiro, “Flexible Heuristics Miner (FHM),” *BETA Working Paper Series*, WP 334, Eindhoven University of Technology, Eindhoven, 2010.

[7] R. Conforti, M. Dumas, L. García-Bañuelos, and M. La Rosa, “BPMN Miner: Automated discovery of BPMN process models with hierarchical structure” *Information Systems*, 56, pp. 284-303, 2016.

[8] C. W. Günther, and W. M. P. van der Aalst, “Fuzzy mining–adaptive process simplification based on multi-perspective metrics,” in *Business Process Management. BPM 2007. Lecture Notes in Computer Science*, vol. 4714, 2007, Springer, Berlin, Heidelberg, Brisbane, pp. 328-343.

[9] W. M. P van der Aalst, H. A. Reijers, and M. Song, “Discovering social networks from event logs,” *Computer Supported Cooperative Work (CSCW)* vol. 14, no. 6, pp. 549-593, 2005.

[10] M. Song, and W .M. P. van der Aalst, “Towards comprehensive support for organizational mining,” *Decision Support Systems*, vol. 46, no. 1, pp. 300-317, 2008.

[11] J. M. E. M. van der Werf, B.F. van Dongen, C. A. J. Hurkens, and A. Serebrenik, “Process Discovery using Integer Linear Programming,” *Fundamenta Informaticae*, vol. 94, no. 3-4, pp. 387-412, 2010.

[12] F. Mannhardt, M. de Leoni, H. A. Reijers, W. M. P. van der Aalst, “Data-driven Process Discovery - Revealing Conditional Infrequent Behavior From Event Logs,” in *Advanced Information Systems Engineering. CAiSE 2017. Lecture Notes in Computer Science*, vol. 10253, Springer, Cham, 2017, Essen, pp. 545–560.

[13] R. Petrușel, I. Vanderfeesten, C. C. Dolean, and D. Mican, “Making decision process knowledge explicit using the decision data model,” in *Business Information Systems. BIS 2011. Lecture Notes in Business Information Processing*, vol. 87, 2011, Springer, Berlin, Heidelberg, pp. 172-184.

[14] C.-C. Dolean, “Mining Product Data Models: A Case Study,” *Informatica Economica*, vol. 18, no. 1, pp. 69-82, 2014.

[15] S. J. J. Leemans, N. Tax, and A. H. M. ter Hofstede, “Indulpet miner: Combining discovery algorithms” in *OTM 2018 Conferences. OTM 2018. Lecture Notes in Computer Science*, vol. 11229, 2018, Springer, Cham, Valletta, pp. 97-115.

[16] M. L. van Eck, J. C. A. M. Buijs, and B. F. van Dongen, “Genetic process mining: Alignment-based process model mutation,” *in International Conference on Business Process Management, Business Process Management Workshops. BPM 2014. Lecture Notes in Business Information Processing,* vol. 202, 2014, Springer, Cham, Eindhoven, pp. 291-303.

[17] J.C.A.M. Buijs, “Flexible evolutionary algorithms for mining structured process models,” Ph.D. Thesis, Eindhoven University of Technology, Netherland, 2014.

[18] S. J. J. Leemans, “Robust process mining with guarantees,” Ph.D. thesis, Eindhoven University of Technology, Netherlands, 2017.

[19] N. Tax, N. Sidorova, R. Haakma, and W. M. P. van der Aalst, “Mining local process models,” *Journal of Innovation in Digital Ecosystems,* vol. 3, no. 2, pp. 183-196, 2016.

[20] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “A genetic algorithm for discovering process trees,” in *IEEE Congress on Evolutionary Computation*, Brisbane, pp. 1-8, 2012.

[21] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “Mining Configurable Process Models from Collections of Event Logs,” in Business Process Management. Lecture Notes in Computer Science, vol. 8094, Springer, Berlin, Heidelberg, 2013, Beijing, pp. 33-48.

[22] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Discovering block-structured process models from event logs containing infrequent behaviour,” in *Business Process Management Workshops. BPM 2013. Lecture Notes in Business Information Processing*, vol. 171, Springer, Cham, 2013, Beijing, pp. 66-78.

[23] M. Weidlich, and J. M. van der Werf, “On profiles and footprints– relational semantics for petri nets,” in *Application and Theory of Petri Nets. PETRI NETS 2012. Lecture Notes in Computer Science,* vol. 7347, Springer, Berlin, Heidelberg, Xi’an, 2008, pp. 148-167.

[24] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Using life cycle information in process discovery,” in *Business Process Management Workshops. BPM 2016. Lecture Notes in Business Information Processing*, vol. 256. Springer, Cham, Innsbruck, 2016, pp. 204-217.

[25] 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,” in *International Journal of Cooperative Information Systems*, vol. 23, no. 1, pp. 1-39, 2014.

[26] C.-C. Osman, “Quality assessment of process models in Process Mining: the case of Petri Nets,” in *Proceedings of 18th International Conference on INFORMATICS in ECONOMY. Education, Research and Business Technologies*, Bucharest, 2019, pp.199-205.

[27] A. Djedović, “Electronic Invoicing Event Logs,” 4TU, Centre for Research Data, https://doi.org/10.4121/uuid:5a9039b8-794a-4ccd-a5ef-4671f0a258a4.

[28] J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens, “A robust F-measure for evaluating discovered process models,” in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Paris, 2011, pp. 148-155.

[29] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Scalable process discovery and conformance checking,” *Software & Systems Modeling*, vol. 17, no. 2, pp. 599-631, 2018.

[30] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, W. M. P. van der Aalst, “Prom 6: The process mining toolkit„” in *BPM 2010 Demonstration Track*, New Jersey, 2010, pp. 34-39.
[31] C. J. van Rijsbergen, Information Retrieval, Butterworth, 1979.
[32] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen, “Replaying history on process models for conformance checking and performance analysis,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182-192, 2012.
[33] C. W. Günther, and E. Verbeek, “XES Standard Definition,” https://xes-standard.org/_media/xes/xesstandarddefinition-2.0.pdf
[34] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Exploring processes and deviations,” in *Business Process Management Workshops. BPM 2014. Lecture Notes in Business Information Processing*, vol. 202, Springer, Cham, Eindhoven, 2014, pp. 304-316.

Cristina-Claudia Osman is a Teaching Assistant at Business Information Systems Department, Faculty of Economics and Business Administration, Babeș-Bolyai University, Cluj-Napoca, Romania. She holds a bachelor degree in Business Informatics, a master degree in E-Business and a PhD degree in Cybernetics and Economical Statistics from 2014. Her current research interest includes Process Mining, Business Process Management, Business Process Modelling, Learning Analytics and Text Mining.