Advancing verification of process mining models with quantitative model checking in stochastic environment

Fawad Ali Mangi[1*], *Guoxin* Su[1], and *Minjie* Zhang[1]

[1] School of Computing & Information Technology, University of Wollongong 2522, Australia

Abstract. The study of business process analysis and optimization has attracted significant scholarly interest in the recent past, due to its integral role in boosting organizational performance. A specific area of focus within this broader research field is *Process Mining (PM)*. Its purpose is to extract knowledge and insights from event logs maintained by information systems, thereby discovering process models and identify process-related issues. The goal of the current study is to examine how *Quantitative Model Checking (QMC)* approaches might be applied in the context of PM. Model checking is a well-known verification approach that provides thorough analysis and validation of a system's properties in comparison to a predetermined model. The adoption of QMC is aimed at improving the accuracy, reliability, and comprehensiveness of PM models in stochastic environment. We propose a novel methodology in this research direction, which integrates QMC with PM by formally modelling discovered and replayed process models and applying QMC methods to verify PM models. The potential of QMC to overcome significant drawbacks of the existing methodologies is the main driver for its use in PM. By including probabilistic model verification, it is possible to take into account the uncertainties and stochastic behaviour that are frequently present in systems that are used in real world; while statistical model checking methods utilized where probabilistic methods fails/not suitable, such as, to handle complex models and/or models with large state-spaces. *{**Keywords:** formal verification, probabilistic model checking, process mining, process models, quantitative model checking, statistical model checking, stochastic systems}*

1 Introduction

Modern information systems diligently maintain a record of operational events in the form of an event log. Historically, these system logs served primarily as a reference point to trace the sequence of events and identify any potential errors. However, with the rapid progression of technology in today's era, the focus has broadened significantly. The challenge now lies not just in detecting errors but also in deriving valuable insights from these event logs. In the modern era, information systems have become an integral part of various sectors, including business, healthcare, and education, among others. These systems generate vast amounts of data, providing a rich source of information for understanding and improving processes, particularly those running in any business environment. It is necessary to monitor, analyse, and improve the process to keep the business environment in smooth running. However, the complexity and volume of data pose significant challenges in extracting meaningful insights.

Process mining [1] has emerged as a promising approach to address these challenges. As an intersection between data mining and business process management, process mining is about discovery, monitoring, analysis and improving real-world business processes by knowledge extraction from event logs provided by information systems [2]. Real-world examples of event logs include loan applications in financial organization [3], an application of a patient's medical billing report in hospital [4], and detection of customer behaviour based on logged clicks [5]. Process mining techniques can be categorized into three types: discovery, conformance checking, and enhancement. This work discusses process discovery and conformance checking, which are relevant t these topics.

Process discovery entails utilizing an event log as an initial data source and generating a model that effectively captures the observed behaviour within the log. The objective of process discovery techniques extends beyond merely constructing models that represent the control-flow of activities. It also encompasses discovering additional dimensions, such as revealing the social network connections among the resources involved in executing these activities [6]. The utilization of process discovery techniques proves highly effective in gaining valuable insights of real-world processes.

Conformance checking involves the examination of both a process model and an event log associated with that specific process. The purpose of conformance analysis is to compare the observed behaviour recorded in the log against the behaviour permitted by the model. When the observed behaviour deviates from what is allowed by the model or vice versa, it indicates non-conformance between the log and the model. One

* Corresponding author: fam366@uowmail.edu.au

commonly utilized conformance analysis technique is outlined in [7].

Model checking (MC), on the other hand, is a method used in computer science to verify and/or analyse the properties of finite state systems. It is a powerful technique for verifying concurrent and distributed systems. Model checking involves the creation of a model of the system, specifying desirable properties in a logical language, and then using algorithms to determine whether the model satisfies these properties [8]. According to [9], the model checking approach model processes as transition systems and express properties as formulas in temporal logic. These properties signify the requirements that a particular process needs to fulfil to be correct. The application of formal verification and model checking techniques in process mining is a relatively new and promising research area [10-16]. Fig. 1 illustrates the classical model checking for process mining models.

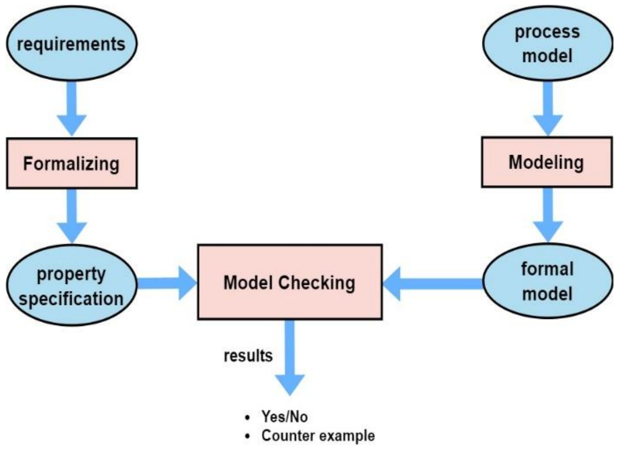

Fig. 1. Traditional model checking for process mining models

The limitation of the model checking technique is its binary (YES or NO) output, which may not be sufficient for today's business process models whose stochastic behaviour is certain. This research aims to explore the application of quantitative model checking techniques in process mining. The methodologies presented in this article are aimed at extending the current process mining techniques by introducing quantitative (*probabilistic and statistical PMC/SMC*) model checking. The probabilistic model checking approach enables the validation of qualitative and quantitative properties of the model, and, to handle the complex systems and/or systems with large state-space where it is difficult to perform probabilistic model checking, the statistical model checking has been applied. A formal framework for assessing system behaviours in an uncertain environment is built by using probabilistic model checking techniques. Utilizing probabilistic models, like Markov chains, enables the estimation of process duration's, the evaluation of the likelihood of particular process routes, and the identification of important process states. In order to handle complex models and/or models with large state-space, statistical model checking techniques give the ability to conduct quantitative analysis based on the statistical attributes and distributions of the data observed, assisting in the detection of anomalies, performance bottlenecks, and opportunities for process improvement. The case study implementation, theoretical analysis, framework proposal, and empirical evaluation are all components of the research technique. Fig. 2 illustrates the quantitative model checking for process mining models.

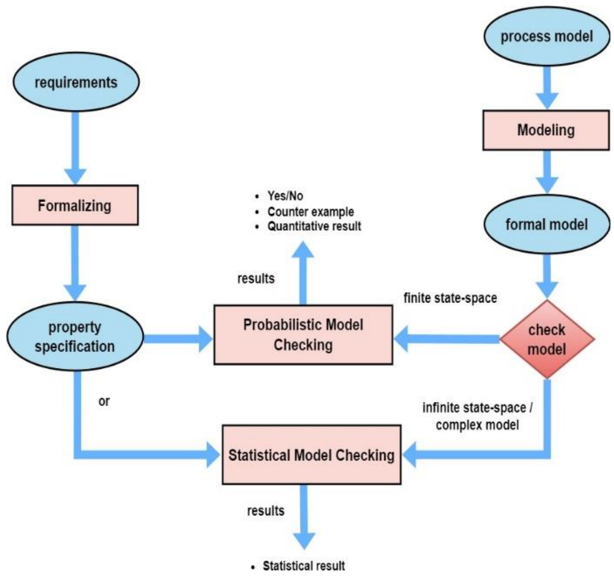

Fig. 2. Quantitative model checking for process mining models

We believe that our research will bridges the gap between model checking and process mining, and contribute to the advancement of process mining techniques and provide valuable insights for organizations to improve their processes.

The structure of this research article is as follows: Section 2 discusses relevant work. In Section 3, the proposed framework has been presented, which further divided in three stages, and each stage is discussed in detail. The implementation details of a case study and results presented in Section 4, and finally, Section 5 concludes the article, summarizing the findings and implications of this research.

2 Related work

Formal verification in process mining has emerged as a critical research domain in recent years, bridging the gap between theoretical model analysis and real-world process execution data. This area aims to confirm that the models derived from process mining methods comply with certain desired properties, thus ensuring their validity and reliability. The verification of processes is a pivotal aspect across multiple fields, particularly where data is a foundational element for functioning and decision-making. We acknowledge that business processes without errors are of utmost importance for organizations. As such, the approach towards verification in business processes has become a cornerstone for their design, development, and enhancement. In [14, 15], the authors critically analyse the existing approaches in formal verification of business process models and provide an extensive overview of the domain. The authors underscore the significance of formal verification across various application areas, such as verifying basic process correctness, and ensuring business compliance. The work in [16] offers a perspective on automating compliance checking within business process models. This approach involves mapping business process models directly to finite state machines, specifically Kripke [17] structures. The compliance rules are expressed in a graphical language for easier understanding; these rules are subsequently translated into linear temporal logic formulae for integration. The authors in [18] introduce a verification framework for business process modelling notations, namely *BPMN*. The need for such a framework arises to overcome the lack of formal semantics in business processes. The work in [19, 20] serves as a stepping stone towards quantitative model checking, where the authors propose a novel methodology to integrate PM with PMC. The work in [21] sheds light on a research direction that merges SMC and process mining. The research aims to augment SMC by explaining the reasons behind specific estimates, which can assist in model validation or recommend improvements (enhancement).

3 Proposed framework

The proposed method is intended to uncover a diverse range of process models and subsequently validate these models via quantitative model checking techniques. This method plays an instrumental role in the evaluation of both process discovery algorithms and replay algorithms, thereby offering considerable utility to users. The methodology, illustrated in Fig. 3, is designed around a flowchart structure, ensuring clarity and ease of understanding. The methodology is segmented into three distinct yet interconnected stages: 1) *process mining model*, 2) *formal model construction and verification*, and 3) *analysis*. The initial phase revolves around the discovery of process models. In this phase, the primary objective is to discover a process model from an event log. The subsequent phase is centred on extracting a formal model *discrete-time Markov chain / continuous-time Markov chain (DTMC/CTMC)* based on the discovered process models. This phase involves translating the structural and behavioural characteristics of the process models into the quantitative framework of the formal model. This translation enables the application of quantitative model checking techniques in the subsequent phase.

The final phase entails the analysis of the constructed formal model using quantitative model checking techniques. The purpose of this phase is to formally verify the formal model by checking whether it satisfies certain specified properties. The verification results can yield invaluable insights into the performance, reliability, and safety of the processes represented by the model, thereby facilitating informed decision-making.

3.1 Process mining model

The process begins with an event log L, from a system. This event log is derived from the specific execution of a particular system. An event log is a log file that stores comprehensive information regarding all the processes executed in a system. The processes may be complete or incomplete, yet they all contribute to the richness of the event log. This log acts as an input in order to discover the model via process mining techniques. We consider a function, symbolized by f, which signifies a *process discovery algorithm*. This function establishes a mathematical relationship, mapping an event log, denoted by L, onto a process model, referred to as N. This relationship is thus written as $f(L) = N$, signifying that the function f transforms the event log L into the process model N. The transformation operation is guided by the process discovery algorithm embodied in the function f. This algorithm examines the sequence of events within the event log and consequently deduces an appropriate model that optimally symbolizes the process or processes documented in the log. After model discovery, an event log acts as an input to the discovered model for the replay process, as shown in dotted block 1 of fig. 3. The replay process can be used for conformance checking in process mining, but here we utilize this process to create a skeleton of a formal model to apply QMC techniques. A *replay algorithm* $\mathcal{R}$, has been used to perform conformance checking. A replay algorithm refers to a computational procedure that operates on an event log L, and a process model N, as its input. The objective of the algorithm is to produce a set of alignments that establish the correspondence between the event log and the process model, denoted as $\mathcal{R}(L, N) = \Gamma_{L,N}$. Here, $\Gamma_{L,N}$ represents a set of alignments that establish a correspondence between the events in the log and the states of a *Petri net* (a modelling notation for discovered process).

3.2 Formal model construction and verification

In this step, the primary focus lies in constructing a model and ensuring its validity for the system under investigation. A model is built to accurately represent the behaviour of a real-world system. The construction of this model typically involves formalizing the processes obtained in the first step (sec. 3.1), translating these into a formal language that can be used for further analysis. The dotted block 2 in Fig. 3 illustrates the modelling and verification step. There could be 2 types of formal models constructed in this step, discrete-time and/or continuous-time Markov chains.

Discrete-time Markov chains $\mathcal{D} = (\mathcal{S}, \bar{s}, \mathcal{L}, \mathcal{P})$, where $\mathcal{S}$ is a finite set of states, $\bar{s} \in \mathcal{S}$ is an initial state, $\mathcal{L}: \mathcal{S} \longrightarrow 2^{AP}$ is a labelling function, and $\mathcal{P}: \mathcal{S} \times \mathcal{S} \longrightarrow [0,1]$ is a transition probability matrix where $\Sigma_{s' \in S} \mathcal{P}(s, s') = 1$ for all $s \in \mathcal{S}$, each element $\mathcal{P}(s, s')$ of the transition probability matrix gives the probability of making a transition from state s to s'.

Continuous-time Markov chains $\mathcal{C} = (\mathcal{S}, \bar{s}, \mathcal{L}, \Re)$, where $\mathcal{S}$ is a finite set of states, $\bar{s} \in \mathcal{S}$ is an initial state, $\mathcal{L}: \mathcal{S} \longrightarrow 2^{AP}$ is a labelling function, and $\Re: \mathcal{S} \times \mathcal{S} \longrightarrow \mathbb{R}_{>=0}$ is a transition rate matrix. The logic to represent properties for formal model DTMC and CTMC is probabilistic computation tree logic *(PCTL),* and continuous stochastic logic *(CSL)* respectively [22].

A constructed model will be checked to perform SMC or PMC, if a model has infinite state-space/complex model, then SMC technique will be performed, otherwise PMC.

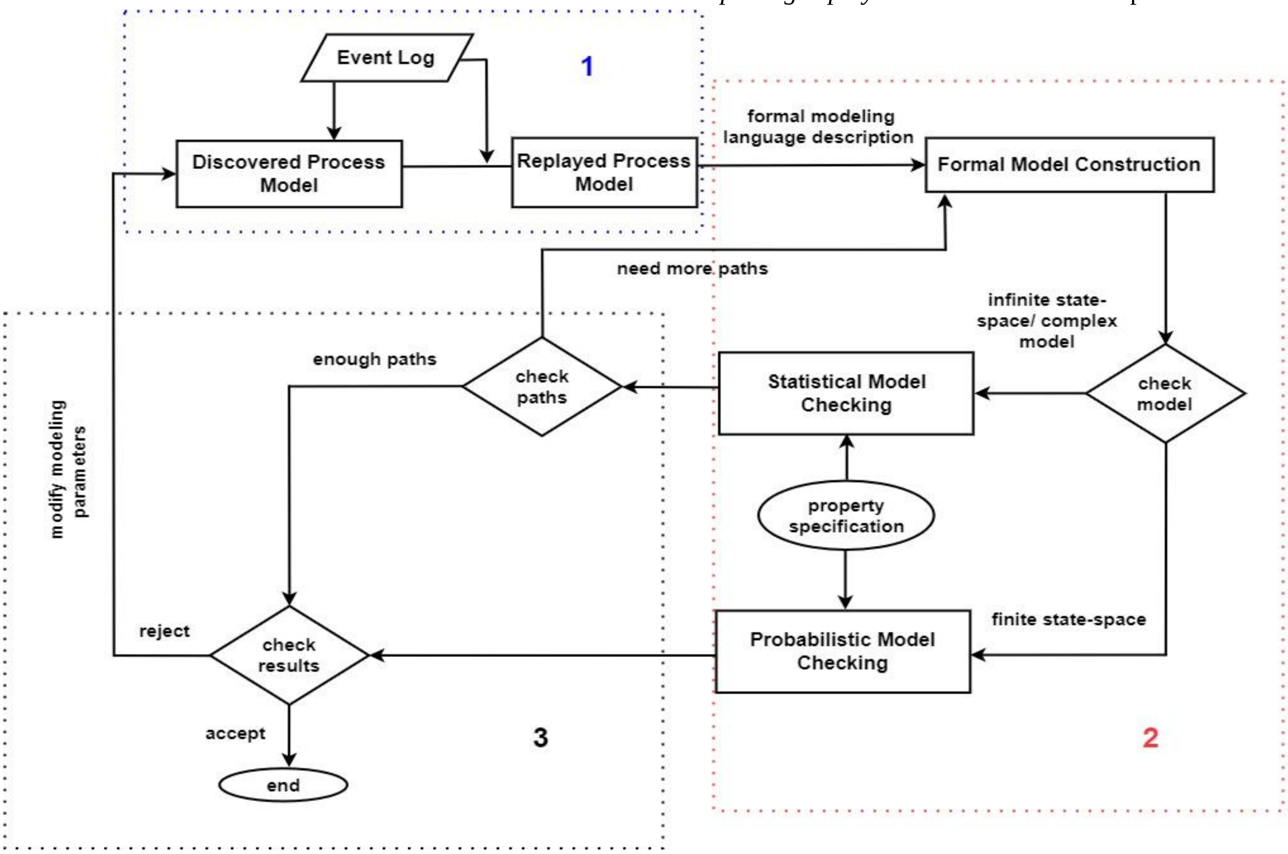

Fig. 3. Proposed framework

3.3 Analysis

The final step of the proposed framework is crucial for both (SMC and PMC) techniques. For SMC, it is related to checking number of paths to perform statistical estimation and tests [23-25], as illustrated in the dotted block 3 of fig. 3 As, SMC is based on sample traces, and therefore, checking paths is crucial in evaluating the results and making conclusive decision. For PMC, we check the verification results and make a decision whether to accept or reject it. The acceptance of QMC results will end the process, while rejection of results will lead to modify modelling parameters of process mining models and start the complete process one more time to obtain reliable/expected results.

4 Implementation and results

The implementation of the proposed framework is based on an event log [3]. We employed a range of process discovery algorithms, including *Alpha miner, Decomposed discovery, Mine for heuristic net, and Mine transition system*. For each discovered process model, we implemented various replay algorithms, *Heuristic cost-based fitness with Integer Linear Programming (ILP), A* cost-based fitness with ILP, Best first search simple string distance calculation, Dijkstra-based replayer, ILP-based replayer, LP-based replayer, Prefix based A* cost-based fitness, and Splitting replaye*r. To showcase the implementation of proposed framework, Table 1-4 presents the verification results for specific properties associated with QMC techniques. Table 1 and 2 shows the verification results of PMC techniques for DTMC/CTMC model, while Table 3 and 4 shows the verification results of SMC techniques. The property in Table 1 calculates the probability of reaching state *y* before state *x*. The property in Table 2 shows the transient behaviour for a CTMC model, it defines the probability of the process being in a specific state at a given time, based on its current state. The property in Table 3 presents the estimated results with *99% confidence level* of arriving at state *x* eventually within a time frame of *10* units. In Table 4, the property verifies a true/false condition, i.e., "The probability is greater than or equal to *0.5* that state *x* is not reached until the *'visited'* is satisfied". [Where *visited* is a combination of states defined in the model]. This property can be used to find how many samples needed to validate if the system behaves as expected with a threshold of *50%* or more. The results in Table

1-4 show that one process discovery algorithm is better in one condition, while another is better under a different condition. With the given number of properties we can deduce that *Decomposed discovery* gives average best results.

Table 1. Property: P =? [¬ (s=x) U (s=y)]

P.D. Alg.	R. Alg. 1	R. Alg. 2	R. Alg. 3	R. Alg. 4	R. Alg. 5	R. Alg. 6	R. Alg. 7	R. Alg. 8
Alpha miner	0.355	0.437	0.199	0.844	0.437	0.355	0.356	0.284
Decomposed discovery	0.355	0.355	0.355	0.355	0.355	0.355	0.355	0.355
Mine for heuristic net	0.531	0.435	0.396	0.383	0.435	0.419	0.394	0.452
Mine transition system	0.313	0.214	0.320	0.357	0.241	0.241	0.332	0.291

Table 2. Property: P =? [$F^{[5,5]}$ (s=x)]

P.D. Alg.	R. Alg. 1	R. Alg. 2	R. Alg. 3	R. Alg. 4	R. Alg. 5	R. Alg. 6	R. Alg. 7	R. Alg. 8
Alpha miner	$1.25*10^{-10}$	$1.67*10^{-10}$	$1.25*10^{-10}$	$5.87*10^{-10}$	$2.35*10^{-10}$	$1.26*10^{-10}$	$1.01*10^{-11}$	$1.18*10^{-10}$
Decomposed discovery	$5.44*10^{-6}$	$5.44*10^{-6}$	$5.44*10^{-6}$	$5.44*10^{-6}$	$5.44*10^{-6}$	$5.44*10^{-6}$	$5.44*10^{-6}$	$5.44*10^{-6}$
Mine for heuristic net	$2.68*10^{-10}$	$2.59*10^{-10}$	$2.10*10^{-10}$	$9.37*10^{-10}$	$2.59*10^{-10}$	$2.93*10^{-10}$	$6.13*10^{-10}$	$2.52*10^{-10}$
Mine transition system	$1.49*10^{-10}$	$8.18*10^{-11}$	$2.18*10^{-10}$	$5.32*10^{-11}$	$5.85*10^{-10}$	$3.17*10^{-10}$	$3.38*10^{-10}$	$1.45*10^{-12}$

Table 3. Property: P =? [F <= 10 (s=x)]

P.D. Alg.	R. Alg. 1	R. Alg. 2	R. Alg. 3	R. Alg. 4	R. Alg. 5	R. Alg. 6	R. Alg. 7	R. Alg. 8
Alpha miner	0.105	0.092	0.053	0.202	0.091	0.096	0.101	0.104
Decomposed discovery	0.1	0.101	0.104	0.105	0.098	0.107	0.102	0.106
Mine for heuristic net	0.161	0.271	0.154	0.274	0.252	0.187	0.238	0.275
Mine transition system	0.266	0.101	0.105	0.107	0.132	0.103	0.24	0.103

Table 4. Property: P >=0.5 [¬ (s=x) U "visited"]; samples required =?

P.D. Alg.	R. Alg. 1	R. Alg. 2	R. Alg. 3	R. Alg. 4	R. Alg. 5	R. Alg. 6	R. Alg. 7	R. Alg. 8
Alpha miner	28	29	25	27	29	28	28	28
Decomposed discovery	28	30	27	30	29	27	29	29
Mine for heuristic net	22	29	34	28	28	29	34	24
Mine transition system	25	31	29	27	29	26	28	26

5 Conclusion

The primary aim of this research is to explore the application of QMC techniques to PM models, aiming to enhance the reliability, accuracy, and comprehensibility of the mined models. Our work finds its motivation in the noticeable absence of a robust and universally accepted approach towards quantitative modelling for PM models in stochastic systems. Consequently, our effort was to propose a methodology and subsequently implement it, so that it enables the quantitative verification of properties inherent to PM models. The approach that we have outlined has an advantage over conventional model checking. One such advantage is the capacity to analyse systems with uncertain behaviours. However, our approach does not merely stop at the verification step. It provides an iterative feedback loop to the users, enabling them to select the most fitting model, and even further refine it. Given the output of the QMC as feedback, the users can dynamically fine-tune the parameters of their model. This active interaction between the user and the QMC results allows continuous improvement and refinement of process models.

References

1. W. van der Aalst, *Process Mining: Discovery, conformance and enhancement of business processes*, Media; Springer: Berlin/Heidelberg, Germany, vol. **136**, (2011).
2. W. van der Aalst, *Process mining: Overview and opportunities*, ACM (TMIS), vol. **3**, no. 2, pp. 1 17, (2012).
3. B. van Dongen, *Bpi challenge (2017)*, 2017. [Online].Available: https://data.4tu.nl/articles/_/12696884/1.
4. F. Mannhardt, M. De Leoni, H. A. Reijers, and W. M. Van Der Aalst, *Data-driven process discovery revealing conditional infrequent behaviour from event logs*, Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, (2017), Proceedings 29, pp. 545–560.
5. M. Dees and B. van Dongen, *Bpi challenge (2016)*: Clicks logged in 2016. [Online]. Available: https://data.4tu.nl/articles/_/12674816/1.
6. W. M. Van Der Aalst and M. Pesic, *Decserflow: Towards a truly declarative service flow language*, Web Services and Formal Methods: Third International Workshop, WS-FM 2006 Vienna, Austria, September 8-9, 2006, Proceedings 3, pp. 1–23, 2006.
7. A. Adriansyah, *Aligning observed and modelled behaviour*, Ph.D. dissertation, Mathematics and Computer Science, 2014, isbn: 978-90-386- 3574-3
8. C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, (2008).
9. E.M. Clarke, F. Lerda, *Model checking: Software and beyond*. J. UCS (2007), **13**, 639–649.
10. M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, et al. *Fundamentals of business process management*; Vol. **2**, Springer, (2018).
11. W. van der Aalst, T. Weijters, L. Maruster, *Workflow mining: Discovering process models from event logs*. IEEE transactions on KDE (2004), **16**, 1128–1142.
12. W. van der Aalst, H.T. de Beer, B.F. van Dongen, *Process mining and verification of properties: An approach based on temporal logic*. 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 (2005), pp. 130–147.
13. M. Räim, C. Di Ciccio, F.M. Maggi, M. Mecella, J. Mendling, *Log-based understanding of business processes through temporal logic query checking*. On the Move to Meaningful Internet Systems: OTM (2014) Conferences: Confederated International Conferences: CoopIS, and ODBASE (2014), Amantea, Italy, October 27-31, 2014, Proceedings (2014), pp. 75–92.
14. S. Morimoto, *A survey of formal verification for business process modelling*, International Conference on Computational Science, pp. 514 522, 2008.
15. H. Groefsema and D. Bucur, *A survey of formal business process verification- from soundness to variability*, Third International Symposium on Business Modeling and Software Design, vol. **1**, pp. 198–203, (2013).
16. O. M. Kherbouche, A. Ahmad, and H. Basson, *Formal approach for compliance rules checking in business process models*, 2013 IEEE 9th International Conference on Emerging Technologies (ICET), pp. 1–6, (2013).
17. S. Kripke, *Semantical considerations on modal logic*, Acta Philosophica Fennica, vol. **16**, pp. 83–94, (1963).
18. F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi, and A. Vandin, *Bprove: A formal verification framework for business process models*, 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 217 228, (2017).
19. F. A. Mangi, G. Su, and M. Zhang, *Pm2pmc: A probabilistic model checking approach in process mining*, 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), pp. 1–6, (2023).
20. F. A. Mangi, G. Su, and M. Zhang, *Integrating process mining with probabilistic model checking via continuous time Markov chains*, FCS'23 The 19th International Conference on Foundations of Computer Science July 24-27, 2023, pp. 1-6. (Proceedings to be published).
21. R. Casaluce, A. Burattin, F. Chiaromonte, A. Vandin, *Process Mining Meets Statistical Model Checking: Towards a Novel Approach to Model Validation and Enhancement*. International Conference on Business Process Management (2022), pp. 243–256.
22. M. Kwiatkowska, G. Norman, and D. Parker, *Stochastic model checking*, Formal Methods for Performance Evaluation: 7th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2007, Bertinoro, Italy, May 28-June 2, 2007, Advanced Lectures 7, pp. 220–270, (2007).
23. Younes, H.L.S. *Verification and planning for stochastic processes with asynchronous events*, Carnegie Mellon University, (2004).
24. K. Sen, M. Viswanathan, and G. Agha, *On statistical model checking of stochastic systems*, Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005. Proceedings 17, pp. 266–280, (2005).
25. G. Agha and K. Palmskog, *A survey of statistical model checking*, ACM TOMACS, vol. **28**, no. 1, pp. 1–39, (2018).