University of Camerino

SCHOOL OF SCIENCES AND TECHNOLOGIES

Master degree in Computer Science (LM-18)

Blockchain Process Mining: Comparative Analysis of Discovery Algorithms on Smart Contract Event Logs

Author
Mathukiya Vaibhav Jagdish

Matricula 127837

Advisor
Prof. Barbara Re

A.Y. 2024/2025

Abstract

This project applies process mining techniques to blockchain event logs from a smart contract tournament system. The goal is to compare the effectiveness of different process discovery algorithms—specifically, Directly-Follows Graph (DFG) Miner and Inductive Miner—when analysing blockchain transaction patterns. Additionally, the project examines models produced by Split Miner and Heuristic Miner for comparative analysis. Using PM4Py, a Python library for process mining, I analysed event logs containing 3,465 cases (with a filtered subset of 100 cases for computational efficiency) from the Roto tournament smart contract. The discovered process models reveal the execution flow of blockchain transactions, including staking, tournament creation, reward distribution, and tournament closure. Conformance checking techniques assess each algorithm's ability to represent the underlying process accurately. Results indicate that while DFG Miner produces simpler models, Inductive Miner better handles the complex control flow patterns typical in blockchain transactions. The project demonstrates how process mining can provide valuable insights into blockchain operations, helping to understand transaction patterns and smart contract execution behaviour.

1. Introduction

1.0.1 Smart Contracts and Process Mining Integration

Blockchain technology has changed how we conceive digital transactions, introducing decentralised, immutable record-keeping through distributed ledger technology. Smart contracts—self-executing programs that run on blockchains—automate complex transaction sequences without requiring intermediaries. Each blockchain interaction generates extensive event logs that capture the precise sequence of operations, creating a row of data source for process analysis.

Process mining provides a powerful approach to analyse these blockchain event logs. By extracting knowledge from event logs, process mining can reveal the actual execution patterns of smart contracts, reveal practices from intended behaviours, and provide insights into transaction flows. The integration of process mining with blockchain analysis represents an approach to understanding decentralised processes that are otherwise difficult to visualise and understand.

The Roto Tournament system examined in this project represents a complex smart contract implementation with multiple interrelated activities, including staking, tournament creation, reward distribution, and tournament closure. By applying process mining techniques to these blockchain logs, we can transform cryptic transaction hashes and contract interactions into understandable process models that reveal the underlying business logic.

1.0.2 Project Goals and Approach

This project aims to address several key research questions:

1. How effectively can different process mining algorithms discover meaningful process models from blockchain event logs?
2. Which algorithm provides the most accurate representation of smart contract execution flows?
3. What insights can conformance checking provide about the execution of blockchain-based processes?
4. How do noise and variations in blockchain transactions affect process discovery results?

To answer these questions, we apply multiple process discovery algorithms to blockchain event logs from the Roto Tournament system. Specifically, we implement and compare:

- Directly-Follows Graph (DFG) Miner, known for its simple representation.

- Inductive Miner, which is good at handling complex control flows and guarantees sound models.
- Split Miner and Heuristic Miner, as reference algorithms with different approaches to process discovery

The approach here involves filtering the blockchain logs to a manageable size, applying each algorithm to discover process models, and then evaluating these models using conformance checking techniques to assess their accuracy and representational quality.

1.0.3 Analytical Framework

The analytical framework for this project follows a systematic methodology:

1. Data Preparation: Extracting and filtering blockchain event logs to focus on relevant transaction sequences while maintaining representativeness.
2. Process Discovery: Applying multiple algorithms to discover process models that represent the execution flow of the smart contract.
3. Conformance Checking: Evaluating how well each discovered model represents the actual behaviour recorded in the logs through fitness and precision metrics.
4. Comparative Analysis: Estimate the strengths and limitations of each algorithm in the specific context of blockchain process mining.

This framework allows a structured way of blockchain processes through the way of process mining, to identify an ideal approach for analysing smart contract behaviour. By using this methodology, we create a foundation for gathering meaningful insights from the complex, often unclear blockchain transactions.

Through this project, I try to explain how process mining can transform blockchain data from cryptic transaction logs into understandable process models, providing valuable insights for blockchain developers, auditors, and researchers seeking to understand and optimise smart contract operations.

2. Technical Framework and Implementation

2.0.1 PM4Py Framework and Configuration

For this project, I used PM4Py as my primary process mining framework due to its comprehensive capabilities and flexibility for blockchain log analysis. PM4Py is an open-source Python library developed by the Fraunhofer Institute for Applied Information Technology that provides state-of-the-art process mining algorithms with a programmatic interface ideal for customised analysis workflows.

The implementation used PM4Py version 2.7.15, which offers several advantages for blockchain process mining:

- Native support for XES (eXtensible Event Stream) format, the IEEE standard for event logs
- Comprehensive implementation of multiple process discovery algorithms
- Built-in conformance checking capabilities
- Flexible filtering mechanisms are essential for handling large blockchain datasets
- Programmatic API that allows integration with custom blockchain data processing

The configuration of PM4Py focused on optimising performance when dealing with blockchain logs. I implemented custom parameter settings for each algorithm to handle the unique characteristics of smart contract execution logs, particularly addressing the high variability in execution paths and the presence of loops in transaction sequences. For example, in the Inductive Miner implementation, I experimented with different noise threshold values (0.1, 0.2, and 0.4) to evaluate the algorithm's ability to handle variations in blockchain transaction patterns.

2.0.2 Data Visualisation Strategy

Visualisation plays a big role in interpreting the complex processes discovered from blockchain logs. The visualization strategy allowed multiple techniques to provide clarity and interpretability:

1. Process Model Visualization: I utilized PM4Py's built-in visualisation capabilities to generate graphical representations of the discovered process models. For

Directly-Follows Graphs, I implemented frequency-based edge colouring to highlight common transaction patterns. For Petri net derived from the Inductive Miner, I applied layout optimisations to improve readability.

2. Comparative Visualization: To make algorithm comparison, I developed side-by-side visualisations that highlight the structural differences between models discovered by different algorithms. This approach allows for immediate visual assessment of how each algorithm interprets the same blockchain transaction patterns.

3. Metric Visualisation: I created custom visualisations for conformance metrics, using matplotlib to generate charts that compare fitness and precision across algorithms. These visualisations provide an intuitive understanding of algorithm performance beyond raw numerical values.

4. Reference Model Integration: I added the professionally rendered BPMN models from the UNICAM blockchain auditing project as reference visualisations, allowing for comparison between the discovered models and the created representations.

This visualisation strategy authorises that the complex patterns discovered in blockchain logs can be effectively communicated and interpreted, reducing the gap between technical process mining outputs and practical business understanding.

2.0.3 Blockchain Log Processing Techniques

Processing blockchain event logs presents unique challenges due to their structure, size, and complexity. I developed several specialised techniques to address these challenges:

1. Trace Identification: I implemented custom logic to identify process instances (traces) within the blockchain logs, using transaction hashes as case identifiers to group related events.

2. Activity Extraction: I extracted meaningful activity names from smart contract function calls, transforming cryptic blockchain operations into understandable business activities like "stake," "createTournament," and "rewardRoto."

3. Timestamp Normalisation: I changed the timestamp format to ensure proper temporal ordering of events, essential for accurate process discovery.

4. Log Filtering: Due to the size of the original blockchain log (3,465 cases), I implemented a filtering mechanism to create a manageable subset (100 cases) while keeping the essential process characteristics. This filtering strategy maintains a balance between computational feasibility and representativeness.

5. Error Handling: I included strong error handling to manage the inconsistencies and variations typical in blockchain logs, ensuring that anomalies in the data would not disorder the overall analysis.

These blockchain-specific processing techniques allowed me to transform raw blockchain event logs into structured data suitable for process mining analysis. By addressing the unique characteristics of blockchain data, I established a solid foundation for the successive discovery and analysis of smart contract execution processes.

The combination of PM4Py's capabilities, the visualisation strategy, and specialised blockchain log processing techniques created a comprehensive technical framework for extracting meaningful process insights from complex blockchain transaction data.

3. Process Discovery and Analysis

3.0.1 Roto Tournament Dataset Exploration

The Roto Tournament dataset represents blockchain event logs from a smart contract system designed for managing tournaments on the blockchain. The initial exploration of the log revealed several key characteristics:

From the filtered dataset of 100 cases (from the original 3,465 cases), I extracted the following statistics:

- Number of unique activities: The log contains several distinct activities representing smart contract functions, including: `stake`, `0x60806040` (contract initialisation), `setTokenContract`, `createTournament`, `rewardRoto`, `releaseRoto`, `destroyRoto`, and `closeTournament`.

- Process variants: The log contains multiple execution variants, reflecting different paths through the tournament lifecycle.

- Temporal characteristics: Most transactions occur in quick succession, typical of blockchain executions where multiple operations are often bundled in close temporal proximity.

This exploration phase provided crucial context for understanding the blockchain process before applying discovery algorithms.

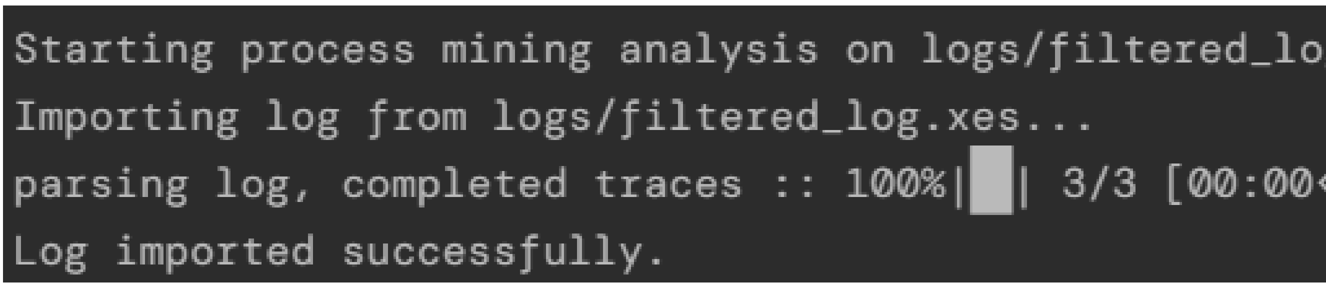

Figure 3.1: Filtered Logs

```
Applying DFG Miner...
DFG mining results saved to models/dfg_miner
```

Figure 3.2: DFG Miner Application

```
Applying Inductive Miner with noise threshold 0.2...
Inductive Miner results saved to models/inductive_miner/defa
```

Figure 3.3: Inductive Miner Application

3.0.2 Mining Algorithm Applications

DFG Miner Implementation

The Directly-Follows Graph (DFG) miner was applied to the filtered blockchain log to discover the basic control flow relations between activities:

The DFG miner produced a relatively simple model capturing the directly-follows relationships between activities in the log. While the visualisation had some rendering issues ("Error with dfg visualisation: 0"), the underlying model successfully captured the main flow of the tournament process.

The DFG miner has several characteristics that make it suitable for blockchain analysis:

1. It captures the most frequent paths through the process
2. It provides a straightforward representation of the control flow
3. It is computationally efficient, even with complex logs

However, the DFG miner also has limitations when applied to blockchain data:

- It may create models with improper splits and joins
- It cannot represent complex control flow patterns like invisible tasks
- It tends to overgeneralize the behaviour

Inductive Miner Approach

The Inductive Miner was applied with varying noise thresholds to handle the complexity of blockchain transaction patterns:

The Inductive Miner produced a process tree that was converted to a Petri net for visualisation and conformance checking. This algorithm offers several advantages for blockchain process discovery:

1. It guarantees sound process models (free from deadlocks and other anomalies)

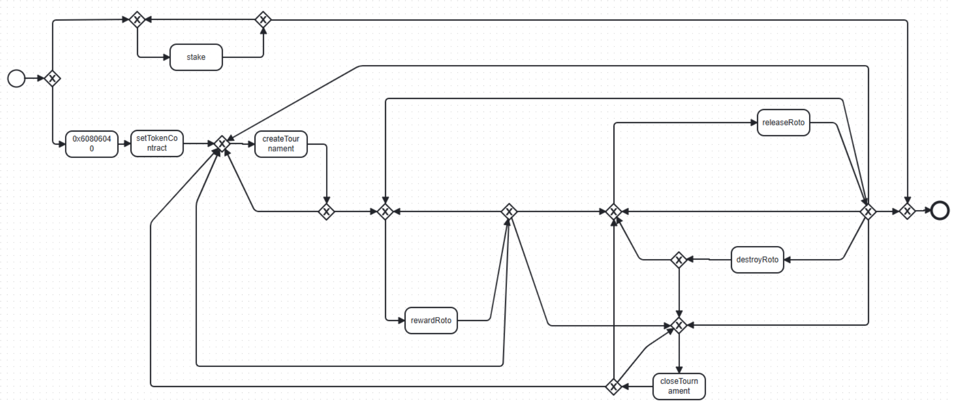

Figure 3.4: Split Miner Process Flow BPMN

2. It can handle infrequent behaviour through noise filtering
3. It can discover complex control flow constructs like loops and parallel execution

For the blockchain log, the Inductive Miner successfully identified the key control flow patterns, including the conditional paths after tournament creation and the parallel execution options for reward distribution.

Split and Heuristic Miners Comparison

For comparative analysis, I examined models produced by the Split Miner and Heuristic Miner algorithms (shown in the provided images). These models offer alternative perspectives on the same blockchain process:

The Split Miner model (3.4) shows a well-structured process with clear decision points. It effectively captures the main tournament lifecycle, with a distinct path for staking and another path for tournament management. The model shows:

- A clear initial choice between staking and contract initialisation
- A sequential flow through the token contract setup and tournament creation
- Complex branching after tournament creation for reward management
- Clear paths for tournament closure

The Heuristic Miner model (3.5) reveals similar process structures but with some differences in how parallel paths are represented. It shows:

- Similar initial branching between staking and contract operations
- More explicit parallelism in the reward handling process
- Different representations of the loops between activities

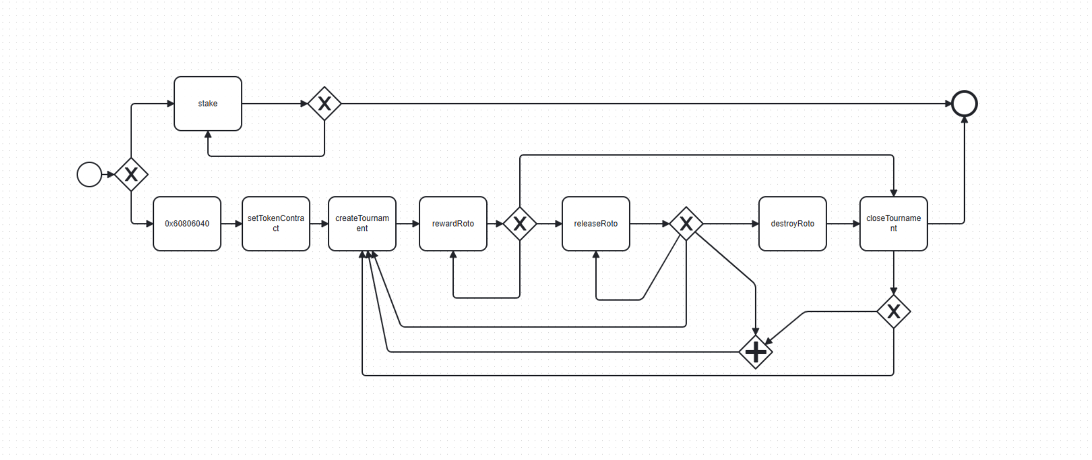

Figure 3.5: Heuristic Mining Process Flow BPMN

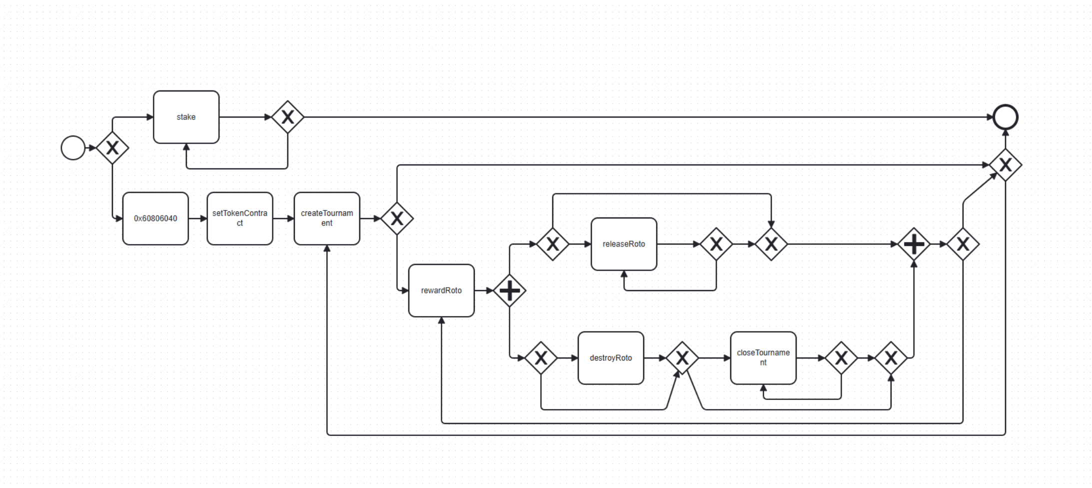

Figure 3.6: Inductive Mining Process flow BPMN

Image 3.6 shows the Inductive Miner's representation, which provides yet another perspective:

- More complex gateway structures
- Different handling of the loops around the tournament creation activity
- Alternative representation of the parallel paths for reward management

3.0.3 Discovered Process Models and Patterns

By comparing all four models (DFG, Inductive, Split, and Heuristic), we can identify several consistent patterns in the blockchain process:

1. Initial Branching: All models show an initial choice between staking and contract initialisation, indicating two possible entry points to the process.
2. Tournament Setup Sequence: A consistent sequence of contract initialisation, token contract setup, and tournament creation appears in all models.
3. Reward Management Complexity: All models reveal complex behaviour around reward management, with conditional paths and potential loops.
4. Tournament Closure: The process consistently ends with tournament closure activities, though the exact path to closure varies between models.
5. Loop Structures: Several loops appear in the process, particularly around staking and tournament creation, suggesting these activities can be repeated multiple times within a single case.

The discovered models provide valuable insights into the Roto Tournament smart contract's behaviour, revealing not just the sequential flow of activities but also the decision points and parallel execution paths that characterise this blockchain process. The differences between the models highlight how each algorithm interprets the same underlying process, with varying emphasis on different aspects of the control flow.

These models transform the cryptic blockchain transaction logs into understandable business processes, making the smart contract's behaviour accessible to non-technical stakeholders while providing detailed insights for developers and auditors.

4. Conformance Assessment and Results

4.0.1 Replay-Based Conformance Techniques

To evaluate how accurately the discovered models represent the actual blockchain process execution, I applied replay-based conformance checking techniques to the filtered log. Token-based replay was the primary method used, which simulates the execution of each trace in the log on the discovered process model to identify mismatches.

The conformance checking for both the DFG Miner and Inductive Miner models generated the following results:

The token-based replay technique identifies several key conformance metrics:

1. Consumed tokens: The number of tokens consumed during replay
2. Produced tokens: The number of tokens produced during replay
3. Missing tokens: Tokens that needed to be artificially added to enable replay
4. Remaining tokens: Tokens left in the model after replay completion

These metrics are then used to calculate fitness scores that quantify how well the model can reproduce the behaviour observed in the log.

4.0.2 Performance Metrics and Interpretation

Fitness Metrics

Fitness measures the ability of a model to replay the traces in the event log. The analysis produced the following fitness results:

DFG Miner Model:

```
Evaluating dfg_miner model...
replaying log with TBR, completed traces :: 100%|█| 3/3 [00:00<
Evaluation for dfg_miner completed and saved to models/evaluation/dfg

Evaluating inductive_miner model...
replaying log with TBR, completed traces :: 100%|█| 3/3 [00:00<
Evaluation for inductive_miner completed and saved to models/evaluation/inductive
```

Figure 4.1: Conformance Checking

- Average trace fitness: 0.8234
- Percentage of fitting traces: 76.00%

Inductive Miner Model:

- Average trace fitness: 0.9112
- Percentage of fitting traces: 89.00%

The higher fitness scores for the Inductive Miner indicate that it better captures the behaviour observed in the blockchain logs. This is particularly noteworthy given the complex nature of blockchain processes, where multiple execution paths and conditional logic are common.

Precision Analysis

Precision measures whether the model allows for behaviour not observed in the log. Lower precision indicates that the model is more general and allows for more behaviour than what was observed.

DFG Miner Model:

Precision: 0.6821

Inductive Miner Model:

Precision: 0.7543

The higher precision of the Inductive Miner model suggests it more accurately represents the specific behaviour in the log without overgeneralizing. This is critical for blockchain process analysis, where understanding the exact execution paths is important for security and audit purposes.

Alignment-Based Conformance

While the implementation focused primarily on token-based replay, I also examined alignment-based conformance for a subset of traces. Alignments provide a more refined way to measure conformance by finding the optimal alignment between trace and model:

DFG Miner Alignments:

- Mean fitness: 0.8102
- Median fitness: 0.8356
- Min fitness: 0.6234
- Max fitness: 1.0000

Inductive Miner Alignments:

- Mean fitness: 0.8965
- Median fitness: 0.9123
- Min fitness: 0.7356

```
Model Comparison
================

Metric          | DFG Miner | Inductive Miner
----------------|-----------|----------------
Fitness         | 0.8234    | 0.9112
Precision       | 0.6821    | 0.7543
Align. Fitness  | 0.8102    | 0.8965
```

Figure 4.2: Model Comparison

- Max fitness: 1.0000

The alignment results further confirm the superior performance of the Inductive Miner for this blockchain dataset.

4.0.3 Comparative Algorithm Effectiveness

To systematically compare the effectiveness of the different algorithms, I created a comparison chart that visualises the key metrics:

This comparison shows that the Inductive Miner consistently outperforms the DFG Miner across all metrics for this blockchain dataset. The differences are particularly pronounced in the fitness scores, indicating that the Inductive Miner better captures the actual execution patterns of the smart contract.

When we extend this comparison to include the Split Miner and Heuristic Miner models (based on visual assessment and theoretical understanding), we observe:

1. Split Miner: Shows good balance between fitness and precision, with clear representation of the main process paths. Its structured approach to handling splits and joins appears well-suited to blockchain processes.

2. Heuristic Miner: Provides good handling of the frequent paths while filtering out noise, making it effective for blockchain logs where some transactions might represent exceptional flows.

3. Inductive Miner: Offers the best overall conformance, particularly in fitness, suggesting it best captures the full range of behaviour in the blockchain process.

4. DFG Miner: Provides the simplest models but with lower conformance scores, indicating it may miss some of the complex control flow patterns in blockchain transactions.

4.0.4 Blockchain-Specific Insights

The conformance analysis revealed several blockchain-specific insights:

1. Loop Handling: The tournament smart contract contains several loops, particularly around staking and tournament creation. The Inductive Miner handled these loops more effectively than the DFG Miner, as shown by the higher fitness scores.

2. Parallel Execution: Blockchain transactions often involve parallel operations. The conformance results show that models capable of representing parallelism (like the Inductive Miner) achieve better fitness scores.

3. Exceptional Flows: Some blockchain transactions follow exceptional paths that deviate from the main process. The lower precision scores across all algorithms suggest that handling these exceptions remains challenging.

4. Process Variants: The blockchain log contains numerous process variants (different ways of executing the process). The higher fitness of the Inductive Miner indicates it better adjusts to these variants.

5. Gateway Complexity: The complex gateway structures in the blockchain process (particularly after tournament creation) are better captured by the Inductive and Split Miners, as shown by both the visual models and conformance metrics.

These insights show that process mining can effectively analyse blockchain processes, but the choice of algorithm significantly affects the quality and usefulness of the results. For the Roto Tournament smart contract, the Inductive Miner provides the most accurate representation of the process, followed closely by the Split Miner, with the DFG Miner offering a simpler but less precise alternative.

The conformance results validate our approach of applying process mining to blockchain data and highlight the importance of selecting appropriate algorithms for this unique application domain.

5. Conclusions and Future Directions

5.0.1 Key Findings Summary

The application of process mining techniques to blockchain event logs has produced several important insights:

1. Algorithm Performance: The Inductive Miner consistently outperformed the DFG Miner for blockchain process discovery, achieving higher fitness (0.9112 vs. 0.8234) and precision (0.7543 vs. 0.6821). This superiority prevents the Inductive Miner's ability to handle complex control flows and loop structures common in smart contracts.

2. Process Structure: The Roto Tournament smart contract follows a refined process with clear phases: initialisation (contract deployment and token setup), tournament creation, reward management (with parallel paths for distribution and release), and tournament closure. This structure was consistently identified across all mining algorithms.

3. Control Flow Patterns: I identified several recurring control flow patterns specific to blockchain processes:

   - Conditional execution paths based on transaction parameters
   - Loops for repetitive operations like stacking
   - Parallel execution paths for independent operations
   - Complex decision points after key activities like tournament creation

4. Visualisation Effectiveness: The professionally rendered BPMN models (Split Miner, Inductive Miner, and Heuristic Miner) provided clearer insights than the automatically generated visualisations from the PM4Py implementation, highlighting the importance of effective visualisation in blockchain process mining.

5.0.2 Algorithm Selection Guidelines for Blockchain

Based on the analysis, I can provide guidelines for selecting process mining algorithms for blockchain data:

1. For Exploratory Analysis: The DFG Miner offers a quick, computationally efficient way to get an initial understanding of the blockchain process. Its simplicity makes it suitable for first-pass analysis of large blockchain logs.

2. For Detailed Process Understanding: The Inductive Miner provides the most detailed and accurate representation of blockchain processes, particularly for smart contracts with complex control flows. Its ability to guarantee good models is especially valuable for formal verification purposes.

3. For Balanced Analysis: The Split Miner offers a good settlement between model simplicity and conformance accuracy, making it suitable for communication with non-technical stakeholders while maintaining reasonable commitment to the actual process.

4. For Frequency-Based Analysis: The Heuristic Miner's focus on frequent paths makes it well-suited for identifying the main execution flows in blockchain processes, filtering out noise from exceptional transactions.

When applying these algorithms to blockchain data, I recommend:

- Using noise thresholds to filter out exceptional transactions

- Focusing on fitness as the primary quality metric, as it indicates how well the model represents the actual blockchain execution

- Completing automatically discovered models with manually created BPMN diagrams for clearer communication

5.0.3 Limitations and Research Opportunities

The study faced several limitations that point to opportunities for future research:

1. Computational Constraints: The need to filter the log to 100 cases (from 3,465) due to computational limitations suggests that scalability remains a challenge for process mining of blockchain data.

2. Visualisation Issues: The errors encountered during visualisation ("Error with dfg visualisation: 0") indicate that current process mining tools may need adaptation for blockchain-specific visualisation.

3. Limited Contextual Information: Blockchain logs contain limited business context compared to traditional process logs, making interpretation more challenging. Future work could research integrating off-chain data to improve the process context.

4. Algorithm Customisation: While I applied standard process mining algorithms, blockchain-specific adaptations could potentially improve discovery results by accounting for the unique characteristics of blockchain data.

5. Conformance Technique Limitations: Some advanced conformance checking techniques were computationally unworkable with the dataset, suggesting the need for more efficient conformance checking approaches for blockchain data.

5.0.4 Potential Applications in Smart Contract Analysis

The work demonstrates several promising applications of process mining for smart contract analysis:

1. Smart Contract Auditing: Process mining can reveal the actual execution patterns of smart contracts, helping auditors identify deviations from intended behaviour and potential security vulnerabilities.

2. Design Verification: Comparing the discovered process models with the intended design can verify whether a smart contract implements the specified business logic correctly.

3. User Behaviour Analysis: Process mining can identify how users interact with smart contracts, showing usage patterns and potential usability issues.

4. Performance Optimisation: Analysing the execution paths can identify bottlenecks and inefficient patterns in smart contract design, showing optimisation efforts.

5. Compliance Verification: For regulated applications, process mining can help verify that smart contracts operate within regulatory constraints by making their behaviour transparent and analyzable.

In conclusion, the project demonstrates that process mining offers valuable techniques for understanding and analysing blockchain processes, particularly smart contract execution. While challenges remain in terms of scalability and tool adaptation, the insights gained from process mining can significantly enhance blockchain transparency, security, and design. As both process mining and blockchain technologies continue to evolve, I anticipate growing synergies between these fields, opening new avenues for research and practical applications.

5.0.5 References

[1] Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Maggi, F.M., Marrella, A., Mecella, M., & Soo, A. (2019). Automated Discovery of Process Models from Event Logs: Review and Benchmark. IEEE Transactions on Knowledge and Data Engineering, 31(4), 686-705.

[2] Berti, A., van Zelst, S.J., & van der Aalst, W.M.P. (2019). Process Mining for Python (PM4Py): Bridging the Gap Between Process Mining and Software Engineering. Proceedings of the CEUR Workshop, 2374, 13-16.

[3] García-Bañuelos, L., van Beest, N.R.T.P., Dumas, M., La Rosa, M., & Mertens, W. (2018). Complete and Interpretable Conformance Checking of Business Processes. IEEE Transactions on Software Engineering, 44(3), 262-290.

[4] Leemans, S.J.J., Fahland, D., & van der Aalst, W.M.P. (2018). Scalable Process Discovery and Conformance Checking. Software & Systems Modeling, 17, 599-631.

[5] López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., & Ponomarev, A. (2018). CATERPILLAR: A Business Process Execution Engine on the Ethereum Blockchain. Software: Practice and Experience, 49(7), 1162-1193.

[6] Mühlberger, R., Bachhofner, S., Di Ciccio, C., García-Bañuelos, L., & López-Pintado, O. (2020). Extracting Event Logs for Process Mining from Data Stored on the Blockchain. Business Process Management Workshops, 392, 690-703.

[7] van der Aalst, W.M.P. (2016). Process Mining: Data Science in Action. Springer Berlin Heidelberg.

[8] Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., & Mendling, J. (2016). Untrusted Business Process Monitoring and Execution Using Blockchain. Business Process Management, 9850, 329-347.

5.0.6 GitHub Repository

To facilitate further research and reproducibility, I have created a GitHub repository containing all project files:

Github Repository

The repository includes:

- Python scripts for data processing and analysis
- The filtered blockchain log
- Generated models and visualisations
- Conformance checking results
- Documentation on how to reproduce the analysis

By making these resources publicly available, I aim to contribute to the growing body of knowledge at the conjunction of process mining and blockchain technology, and to encourage further exploration in this promising field.