The University of Melbourne

Faculty of Engineering and Information Technology

Explainable Artificial Intelligence Using Process Mining

Author: Yuanyuan Ma
yuanyuan.ma1@student.unimelb.edu.au

Supervisor: Artem Polyvyanyy
artem.polyvyanyy@unimelb.edu.au

Contents

1 Introduction **3**

2 Related work **4**

3 Methodology **7**

3.1 Foundation Setup: Establishing Environmental Rules and AI Selection 7

3.2 Data Extraction and State Simplification: Recording and Abstracting Actions and States 8

3.3 Event Log Generation: Creating Logs from Simplified States and Actions 9

3.4 Process Discovery: Applying Algorithms to Develop Process Models 9

3.5 Evaluation of Process Models: Assessing Interpretability and Accuracy 10

4 Case Study: Tic-Tac-Toe **11**

4.1 Construction of Tic-Tac-Toe game 11

4.2 Data extraction and event log generation 13

4.3 Generate Process Model Using Simplified States 19

4.4 Evaluation of Process Models 21

5 Discussion **23**

5.1 Findings 24

5.2 Limitations 25

5.3 Future works 27

6 Conclusion **28**

Abstract

As Artificial Intelligence (AI) technologies are increasingly used in high-risk environments (e.g., finance and healthcare), the interpretability of their decision-making processes becomes particularly important. Currently, many AI systems, especially deep learning-based models, are difficult to fully trust due to their 'black box' character. The aim of this study is to investigate whether process mining techniques can be used to explain the behaviours of AI agents. The methodology outlines a comprehensive pipeline that includes establishing environmental rules, extracting and simplifying data, generating event logs, applying process discovery algorithms, and evaluating the generated process models. To demonstrate the feasibility of this methodology, the article uses the classic game of Tic-Tac-Toe as a case study. The game environment was constructed in Python, and the AI agent was simulated using the Minimax algorithm. By applying process mining methods, we extracted the AI's decision-making process and generated process models to visualise the AI's strategies and decision-making logic. Preliminary results show that process mining can effectively reveal the decision-making rules of AI and enhance its decision transparency and interpretability. However, this approach relies on high-quality data and has limited explanatory capabilities for complex AI systems. Future research would explore more advanced process mining techniques to address these challenges. Overall, this research not only provides a new perspective for understanding the behaviour of complex AI systems but also lays the groundwork for interpretability studies of AI applications in other fields.

Keywords

Artificial intelligence (AI) · process mining · explainable artificial intelligence (XAI) · tic-tac-toe

1 Introduction

In today's technological environment, Artificial Intelligence (AI) has been applied in various fields. However, the opaqueness of AI decision-making processes, especially in AI systems based on deep learning, brings a significant challenge to its application in high-risk areas such as finance and healthcare(Von Eschenbach, 2021). The 'black-box' character of these algorithms often leads to a lack of trust and concerns about potential bias in the results (Durán and Jongsma, 2021). This is because the stakes and costs of error are high in these fields.

Explainable Artificial Intelligence (XAI) aims to make the outcomes of AI systems understandable to humans, addressing the transparency of traditional AI algorithms (Miller, 2019). The need for transparency and explainability in AI systems has never been more urgent. This is because humans are generally reluctant to adopt techniques that cannot be directly explained or are not trustworthy (Arrieta et al., 2020). Stakeholders, including regulators, industry professionals and end-users, expect AI systems to present users with reasonable justifications when making decisions. This would ensure fairness and reliability in crucial domains.

Process mining is a robust data analysis tool that can discover, monitor and improve real processes by extracting knowledge from event logs (Van Der Aalst, 2012). The generated process models through process mining can be used to visualise and explain the behaviour of processes (Lamghari, 2022). This tool starts with an event log where each event represents a well-defined step (Van der Aalst, 2014). Each process path stands for one or more specific cases, and the activities in each case are performed in a certain order. When applied to the investigation of AI systems, process mining can provide a unique perspective to deeply analyse the internal mechanisms of complex systems by extracting and visualising the flow of activities in transactional data.

The aim of this research was to explore whether the behaviour of AI agents could be illustrated through the application of process mining techniques to enhance the transparency of AI systems. This study utilised a

Python-based implementation of a tic-tac-toe game, where the decision-making process of the AI agent was simulated using the Minimax algorithm. The research focused on the possibility of analysing generated event logs during gameplay through process mining to reveal the AI's decision logic and visualise its decision paths.

This study, on the one hand, contributed to the development of XAI by providing concrete tools and methodologies for enhancing the interpretability of AI systems. It made AI easier to understand and explain to stakeholders who may not have deep expertise in AI technologies. On the other hand, this study demonstrated its effectiveness in real-world applications by empirically testing it in specific cases, which in turn laid the foundation for the dissemination of this technique.

Furthermore, this paper also discussed the possibility of applying process mining to other more complex AI scenarios, expanding its application from tic-tac-toe to more complex and diverse application domains, such as autonomous driving. This exploration not only advanced the field of XAI but also opened new avenues for using process mining as a standard tool for making AI systems more understandable and accountable.

2 Related work

In some of the past research, several methodologies have been developed to explain the behaviour of AI in order to make the decisions and behaviour of AI systems more transparent and understandable. These methods can be classified into two categories, which are the AI models with self-interpretation capabilities and the AI model interpretation methods (Yin et al., 2024). The former include linear models, decision trees, random forests, etc. The latter usually utilise feature attribution techniques and are model-independent. Each method presents its own advantages and limitations, making them suitable for varied application contexts.

1. Self-interpretation Models:

Decision trees: Decision trees are a model that can be easily implemented with fully transparent constraints (Vieira and Digiampietri, 2020).

It is a supervised learning model as it can automatically build predictive models from a given set of observations (Geurts et al., 2009). It shows the decision-making process through a tree structure where each node represents a decision point. The root node is the predictor variable while the leaf nodes provide the final classification (Mahbooba et al., 2021). It has a simple structure and is easy to understand. For example, in medical diagnosis, a decision tree can help doctors clearly show how to classify a disease based on symptoms.

Although decision trees excel in interpretability and are conducive to visualization, they are prone to overfitting, particularly when the tree depth is large. As a result, the model becomes highly sensitive to noise and outliers in the training data, leading to poor generalization capabilities. Their IF-THEN structure mimics human language and the way we think. However, there is an important premise that each decision condition needs to be constructed from understandable features and without too many rules (Mahbooba et al., 2021). If each decision point in the environment is difficult to represent through language, then decision trees are not a very suitable choice. Moreover, decision trees also have limitations in dealing with continuous event logs compared to process mining, making it challenging to visualise temporal and sequential dependencies in decision-making processes.

Linear models: Linear models use the weighted sum of input features to predict targets. They are computationally efficient and can interpret linear relationships between features. Therefore, when the number of features is small, the weights of the linear model can be used to explain the prediction (Islam et al., 2021). However, this model requires the assumption of a linear relationship between the inputs and outputs, and if there is no linear relationship between the inputs and outputs, the fitting performance of this model would be diminished.

2. Feature Attribution Methods:

Another category of methods used to explain AI behaviour is feature attribution, which aims to highlight the effect of individual input features on model output. Techniques such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) provide explanations by analysing the contribution of each input feature to the model outputs.

LIME: LIME is an algorithm that locally approximates any classifier or regressor with an interpretable model (Ribeiro et al., 2016). It uses simple models to explain happenings in complex models (Gawde et al., 2024). This algorithm generates a series of modified samples by perturbing the features of selected individual data samples and subsequently forming a new dataset. Then it utilizes these data to train an interpretable model weighted by the similarity between the perturbed samples and the original instance (Magesh et al., 2020).

However, a major limitation of LIME is the stability of its explanations. The randomness of the perturbed samples can lead to inconsistent explanations across multiple executions, even if they are executed on the same dataset (Zhao et al., 2021). This inconsistency may reduce the reliability of the interpretations, potentially destroying user trust and understanding of the model's behaviour. Additionally, this method only provides local explanations and does not fully reflect the behaviour of the model across the entire dataset. This limitation further restricts its widespread application in practical scenarios.

SHAP: SHAP interprets predictions by calculating feature importance and uses Shapley values from game theory to ensure consistency of interpretation (Antwarg et al., 2021). Shapley values are a method for fairly distributing profits in a game. SHAP uses the concept of Shapley values to represent the outcome as the sum of the contributions of each feature, each of which is quantified by its Shapley value (Nohara et al., 2022).

The primary distinction between LIME and SHAP is the process of assigning weights to a linear regression model (Gohel et al., 2021). Compared to LIME, SHAP ensures consistency in explanations as it is based on the mathematically rigorous definition of Shapley values. However, SHAP incurs a high computational cost when the number of features is substantial due to the necessity of calculating all possible feature combinations. Although SHAP provides detailed explanations of feature influences and visualizations, the form of these explanations can be abstract and complex, potentially bringing interpretative challenges for non-technical users.

3 Methodology

This section summarises the pipeline for using process mining to explain AI behaviours. The entire process is divided into five sequential phases: foundation setup; abstraction; generating event logs; discovery model and evaluation.

3.1 Foundation Setup: Establishing Environmental Rules and AI Selection

To investigate artificial intelligence behaviour via process mining, the initial phase involves establishing a comprehensive set of environmental rules and selecting a suitable AI agent.

These environmental rules define the context and constraints of AI operations. They compose the behavioural framework that the AI must follow, defining how it interacts with the environment, what actions AI can take, and how it should respond in a variety of different situations. For instance, in a strategy-based game, these rules not only specify which moves are legal but also define the ending conditions of the game, how scoring is conducted, and how victory or defeat is determined. These rules ensure that the AI system can work as expected, and thus help reduce uncertainty or errors in the system. Other than that, the selection of the AI agent is equally important. It must depend on specific behaviours and characteristics that the study needs. Different AI algorithms have their strengths and weaknesses in data processing, decisions making and changes adapting. For example, decision tree algorithms excel at handling structured data, while neural networks are adept at learning complex patterns from large volumes of unlabeled data. Therefore, choosing the most suitable algorithm based on the objectives of the research is important, as it can significantly enhance the efficiency and accuracy of the AI system in solving problems.

3.2 Data Extraction and State Simplification: Recording and Abstracting Actions and States

After completing the establishment of the foundational environment and the AI agent, the next step is to record every interaction between AI and the environment systematically. This procedure involves recording each action taken by AI and the changes in the environment state triggered by these actions. However, this raw data often contains a large number of details. If it is used to construct the process model directly, it may make the model too complex to understand and interpret. To address this problem and improve the interpretability of process models, abstraction of the environmental state is necessary. Through simplification, redundant information can be removed and key state transitions retained. It not only reduces the number of paths but also enhances the intuitiveness of the model, which enables users to understand each step and decision-making process more clearly.

Another essential task in this phase is the adoption of appropriate methods for abstracting the environment states. Some common techniques include state aggregation, where similar states are grouped into a single abstract state according to specific logical rules. In terms of environments with high symmetry, this approach can substantially reduce the number of states in the model and simplify decision paths. Another popular approach is to apply mathematical techniques to reduce the data dimensionality, such as principal component analysis (PCA) or autoencoders. These methods effectively compress data while retaining the most informative features. Finally, combined expertise and domain-specific rules can also help models reduce complexity. Through these methods, the model would be easy to interpret and analyze while maintaining accuracy.

3.3 Event Log Generation: Creating Logs from Simplified States and Actions

In this stage, researchers should make use of the simplified data from the previous step to generate detailed event logs. These event logs should systematically record each simplified action or state change, and each event should include a timestamp to ensure that the sequence of events is accurately maintained. The process of generating event logs must prioritise the accuracy and integrity of the data. This requires researchers to strictly record the details of each interaction and ensure that the logs cover all critical rounds. This comprehensive coverage guarantees that all key operations and state transitions are included in the process model construction, thus avoiding the possibility of missing important steps or states.

Event logs are the foundation of building process models, which directly affect the accuracy and interpretability of the model. Therefore, it is crucial to ensure that the event logs accurately reflect the actual interactions between the AI system and its environment. Only when the precision of the event logs is high enough can the generated process models effectively reveal the system's behavioural patterns and decision-making pathways. Accurate event logs not only facilitate the construction of more precise models but also provide a solid foundation for subsequent analysis.

3.4 Process Discovery: Applying Algorithms to Develop Process Models

During the process discovery phase, researchers need to execute a rigorous analysis of various process discovery algorithms to determine the most suitable discovery algorithm for the given case. Some common process discovery algorithms are Alpha Miner, Heuristic Miner and ILP Miner, etc. Therefore, researchers have to conduct a thorough evaluation of each algorithm's strengths and weaknesses, as well as a comparative analysis of the models produced by these algorithms. The selection criteria encompass several factors, including model accuracy, algorithmic complexity, and computational efficiency. Apart from this, the effectiveness of each algorithm in capturing the underlying process patterns also needs to be assessed to identify the optimal algorithm for generating process models that are both accurate and interpretable.

After selecting the appropriate process discovery algorithm, researchers can use the generated event logs to construct a preliminary process model. The process model shows the decision path of the AI agent to users in a transparent and easy-to-understand visualisation. Hence, choosing a suitable model is also crucial. Some of the frequently used models include Flowcharts, Petri Nets, Business Process Model and Notation (BPMN), etc. Each model has different representations and expressive capabilities. As a result, it is necessary for researchers to choose the proper model according to different usage scenarios and learning costs.

These visible representations assist users in becoming aware of capacity biases or mistakes inside the AI's decision-making process. Through a detailed examination of the paths and patterns in the process model, researchers and practitioners can exactly discover the reasons why AI may go wrong, thereby providing opportunities to refine and improve the algorithms.

3.5 Evaluation of Process Models: Assessing Interpretability and Accuracy

The final stage of the methodology is to assess the interpretability and accuracy of the process model. The purpose of the evaluation is to help researchers identify whether the model is effective in explaining the behaviour of the AI and if it can be understood by people without specialized knowledge. Several comparative analysis tools are commonly used in this phase to assist researchers analyze the discrepancies between the model and the actual event logs in order to calculate the fittedness of the model.

The performance of a process model is usually evaluated in four dimensions, which are fitness, precision, generalisation, and simplicity. Fitness evaluates how well the process model captures the observed behaviour (Rozinat et al., 2008). Precision measures the extent to which the process model allows for behaviours that are not seen in event logs (Buijs et al., 2014). Generalisation can reflect how well the model adapts to handle common behavioural changes. Simplicity ensures that the model is concise and user-friendly.

Additionally, researchers could consider using simulation testing to evaluate model performance. By manually simulating the environment to test whether the process model helps the researcher to correctly follow the steps of the model to achieve the desired results.

These assessments allow for further improvement of the model to ensure that it not only meets current analytical requirements, but also evolves with future needs and applications.

4 Case Study: Tic-Tac-Toe

To demonstrate the practical application of the above methodology, this research implemented a case study using the classic game Tic-Tac-Toe. The case study is a specific example of how process mining techniques can be utilized to enhance the understanding of AI behaviour. The following sections detail the implementation, abstraction, and findings of this case study.

4.1 Construction of Tic-Tac-Toe game

Tic-tac-toe is a simple two-player game where the goal is to be the first player to get three consecutive squares on a 3x3 grid. Players take turns placing their chess ('X' or 'O') in an empty square. This game has three types of results which can be a "win", "loss" or a "draw". The first player to align their three squares vertically, horizontally, or diagonally wins the game (Sriram et al., 2009). The game is considered a draw if all nine squares are filled and no player has three consecutive squares. It is a subject that is both simple enough to allow for in-depth analysis and complicated enough to show significant AI decision-making. The tic-tac-toe game environment was implemented in Python, utilizing a dictionary to represent the board state, where each key was a position (1 through 9) and each value indicated the occupant (’X’, ’O’, or ’ ’). For example, an empty board can be represented as {’1’: ’ ’, ’2’: ’ ’, ’3’: ’ ’, ’4’: ’ ’, ’5’: ’ ’, ’6’: ’ ’, ’7’: ’ ’, ’8’: ’ ’, ’9’: ’ ’}. Figure 1 shows a basic layout of the game board. Considering a more

1	2	3
4	5	6
7	8	9

Figure 1: Empty Board

complex game state, such as {’1’: ’X’, ’2’: ’O’, ’3’: ’X’, ’4’: ’O’, ’5’: ’O’, ’6’: ’X’, ’7’: ’X’, ’8’: ’X’, ’9’: ’O’}, we can observe the distribution of players ’X’ and ’O’ on the board. Figure 2 displays the board layout for this game state. This state representation enables each specific state of Tic-Tac-Toe

X	O	X
O	O	X
X	X	O

Figure 2: Specific game state in tic-tac-toe

to be accurately recorded and is concise and easy to understand, even for people with no coding expertise.

The AI agent for this case study was implemented using the Minimax algorithm. It aims to identify the best strategy for a player by determining the optimal move for that individual at each turn (Savelli and de Beauclair Seixas, 2008). The game tree represents all potential moves within the game and takes the form of an inverted tree (Manimegalai and Sheeba, 2021). This algorithm is particularly suitable for games like Tic-Tac-Toe due to its ability to consider all possible future game scenarios through simulation. At each step, the Minimax algorithm evaluates the potential outcomes of all available moves to determine the best move based on minimizing the possible loss for a worst-case scenario (Borovska and Lazarova, 2007). The Minimax algorithm operates with a straightforward computational mechanism. It does not require learning from extensive datasets or adjusting numerous parameters, nor does it have a complex internal structure and operation like a neural network. Although the principle behind it is simple, it still can be regarded as a black-box model in research to help researchers analyse the behaviour of AI.

4.2 Data extraction and event log generation

In this phase, the study needed to extract the event logs from each round of the game as inputs for process mining. Traditional process mining mainly focuses on business flow, and each node in the process model represents a concrete step in the real operation of business activities. However, in terms of decision-making for the Tic-Tac-Toe AI agent, it does not have specific steps like business activities. It needed to take different actions based on different game states. Therefore, what type of data could be recorded as an activity in the event log is a direction that needs to be explored.

In extracting the event logs, this research has tried different methods. Table 1 illustrates one approach. It used a combination of concrete state and action, where the state showed the current layout and the action was the decision position given by the AI. However, the study found that since there are only nine choices for action, it is easy to aggregate paths with the same action under different layouts together, making it hard for users to distinguish in process models.

Table 2 displays another method of recording. It used a list to record information about each step: the first element of the list represents the current player, the second element indicates the location of the piece, and the third element indicates which step this is. However, this recording method is only applicable to the current game environment and lacks scalability.

Through comparison between different recording methods, the study found that using game states as activities for the logs would be more con-

Table 1: Data extraction method for state-action combination

Case	Event
1	{'1': 'X', '2': ' ', '3': ' ', '4': ' ', '5': ' ', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	5
1	{'1': 'X', '2': ' ', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': 'X', '8': ' ', '9': ' '}
1	4
1	{'1': 'X', '2': ' ', '3': ' ', '4': 'O', '5': 'O', '6': ' ', '7': 'X', '8': ' ', '9': 'X'}
1	6
1	AI wins!

Table 2: Data extraction method for Step-by-Step Information

Case	Event
1	['X', '9', 1]
1	['O', '5', 2]
1	['X', '8', 3]
1	['O', '7', 4]
1	['X', '3', 5]
1	['O', '6', 6]
1	['X', '4', 7]
1	['O', '1', 8]
1	['X', '2', 9]
1	It's a tie!

ducive to process discovery. Therefore, in this phase, the study simulated realistic scenarios where humans interacted with the AI system and then recorded each round of the game. In order to make the process model as comprehensive as possible to contain all varied scenarios of human gameplay, this study employed a randomised algorithm to simulate human decisions, engaging with the AI agent in 10,000 rounds of tic-tac-toe. The algorithm randomly selects a position to place a piece from all available positions. The introduction of this randomness allowed for the inclusion of all the different human decision-making scenarios in this 10,000-round game, thereby enhancing the comprehensiveness of the generated event logs.

This stochastic simulation helped this study capture the various possible game states that may occur during gameplay and stored these states in event logs. Figure 3 illustrates the format of the event log. It contains two rounds of the game, each of which records every state and final result in the game. This event log was then taken to generate a basic process model using Directly Follows visual Miner. The model is shown in Figure 4, which contains all the paths in the 10,000 rounds of the game. Analysing the paths presented in the model allowed users to observe the possible actions of the AI based on a given game state, and thus replay the game scenario. However, even the most basic game such as Tic-Tac-Toe has thousands of different states, which makes the model structure very complex. It is not only difficult to read, but also violates the principle of simplicity in process models.

Case	Event
1	{'1': ' ', '2': ' ', '3': ' ', '4': ' ', '5': 'X', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'O', '2': ' ', '3': ' ', '4': ' ', '5': 'X', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'O', '2': ' ', '3': 'X', '4': ' ', '5': 'X', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'O', '2': ' ', '3': 'X', '4': ' ', '5': 'X', '6': ' ', '7': 'O', '8': ' ', '9': ' '}
1	{'1': 'O', '2': ' ', '3': 'X', '4': 'X', '5': 'X', '6': ' ', '7': 'O', '8': ' ', '9': ' '}
1	{'1': 'O', '2': ' ', '3': 'X', '4': 'X', '5': 'X', '6': 'O', '7': 'O', '8': ' ', '9': ' '}
1	{'1': 'O', '2': ' ', '3': 'X', '4': 'X', '5': 'X', '6': 'O', '7': 'O', '8': ' ', '9': 'X'}
1	{'1': 'O', '2': 'O', '3': 'X', '4': 'X', '5': 'X', '6': 'O', '7': 'O', '8': ' ', '9': 'X'}
1	{'1': 'O', '2': 'O', '3': 'X', '4': 'X', '5': 'X', '6': 'O', '7': 'O', '8': 'X', '9': 'X'}
1	It's a tie!
2	{'1': ' ', '2': ' ', '3': ' ', '4': ' ', '5': ' ', '6': ' ', '7': ' ', '8': 'X', '9': ' '}
2	{'1': ' ', '2': ' ', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': 'X', '9': ' '}
2	{'1': ' ', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': 'X', '9': ' '}
2	{'1': 'O', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': 'X', '9': ' '}
2	{'1': 'O', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': 'X', '9': 'X'}
2	{'1': 'O', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': 'O', '8': 'X', '9': 'X'}
2	{'1': 'O', '2': 'X', '3': ' ', '4': 'X', '5': 'O', '6': ' ', '7': 'O', '8': 'X', '9': 'X'}
2	{'1': 'O', '2': 'X', '3': 'O', '4': 'X', '5': 'O', '6': ' ', '7': 'O', '8': 'X', '9': 'X'}
2	AI wins!

Figure 3: concrete event logs

Therefore, simplifying the model was essential. A detailed analysis of the game's process revealed that many states are functionally similar. Figure 5 shows an example from the process model generated above, which is the first step of the game. It consists of nine different states that correspond to placing an 'X' in nine different positions on the Tic-Tac-Toe board. In fact, some of these states are identical, for instance, placing an 'X' in any of the

Figure 4: concrete process model

four corners of the board can be considered as one category of states. This categorization is illustrated in Figure 6, which shows an example of these states. Similarly, placing an ‘X’ on any of the four edges of the board can also be considered as another category, as depicted in Figure 7, highlighting the example of these placements. If concrete dictionaries are used to represent these states, then in the process model they would be divided into nine different paths. Instead, the nine paths are actually equivalent to three cases, where ’X’ is located in the corner, on the edge and in the middle of the board.

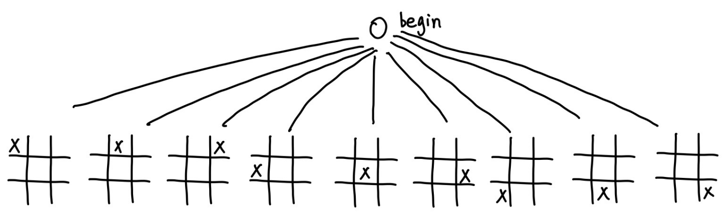

Figure 5: first step of tic-tac-toe

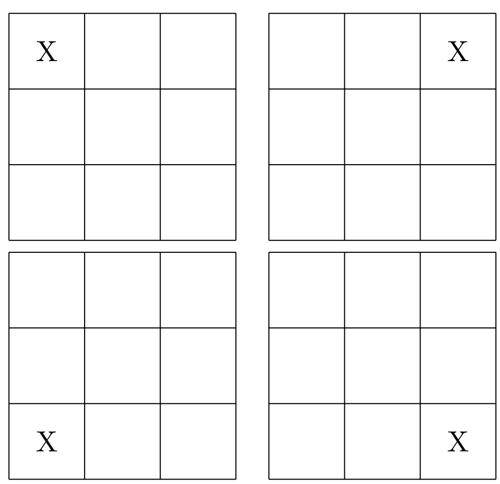

Figure 6: one ’X’ on the corner

After realising the redundancy of these equivalent states, the research adopted a simplification method of state aggregation. Through investigation, it was found that the layout of tic-tac-toe has a high degree of sym-

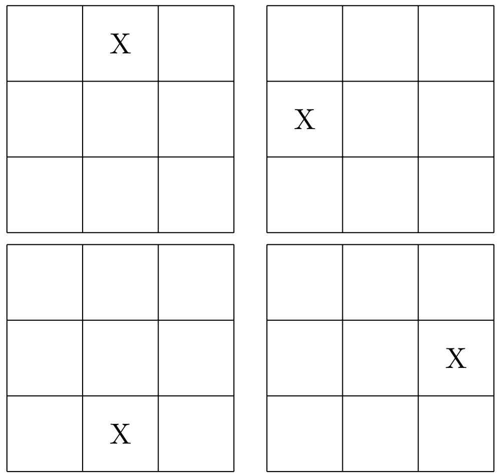

Figure 7: one ’X’ on edges

metry. States obtained by rotating an original state by 90, 180, or 270 degrees, or by flipping it vertically, horizontally, or along the diagonal, are functionally similar and can be considered as one category. Therefore, in order to reduce the complexity of the model, in this study, these transformations were summarised into a list and stored in a dictionary. The key of the dictionary is the original state and the value is the list of these seven transformed states. In the subsequent generation of event logs, if a transformed state was encountered, the original state was used to represent it.

For instance, in the following example in Figure 8, the original state is transformed into seven different states through rotations and reflections. All these variants can be represented uniformly using the dictionary of the original state {‘1’: ‘O’, ‘2’: ‘X ‘, “3”: “ ”, “4”: “ ”, ‘5’: ‘O’, ‘6’: ‘ “, ”7 ‘: “X”, “8”: “ ”, 9’: ‘ ‘} to represent, which effectively reducing the complexity of the model.

Using this method, a simplified event log was regenerated, as illustrated in Figure 9. Although some states appear discontinuous, making them less readable, they can be transformed into the desired states through various transitions. This approach significantly reduced the number of environmental states.

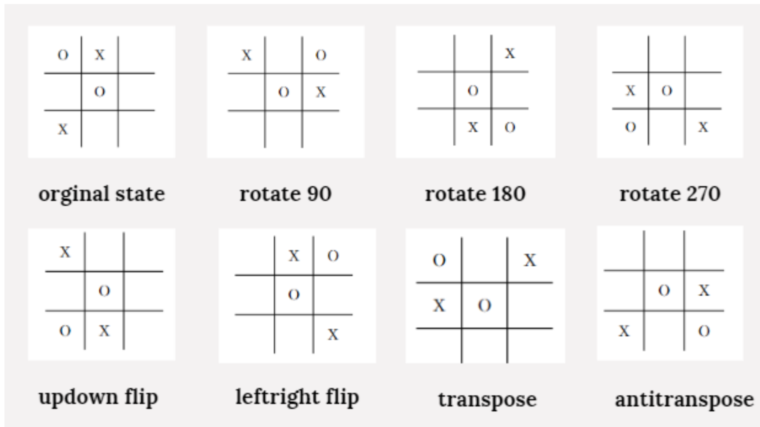

Figure 8: Transformations of Tic-Tac-Toe

This simplification strategy not only made the process model more readable but also enhanced its utility in elucidating the AI decision-making process. By reducing the number of states, the model became easier to navigate, thereby allowing users to more clearly understand the AI's strategy development and execution patterns. Furthermore, this approach of representing all states achievable through rotation or flipping with a single state provides a feasible solution to simplify the decision-making process with highly repetitive and symmetric states.

4.3 Generate Process Model Using Simplified States

Following the extraction and simplification of the Tic-Tac-Toe game data, the next crucial step in this study was to generate process models that could represent the AI's decision-making processes accurately. Using the simplified states obtained during gameplay, the Directly Follows visual Miner in ProM was employed in this phase. This method emphasises transitions between states to capture the essence of the game's progression, without cluttering the visual representation. This approach facilitates an intuitive and insightful visualization of the AI's strategies and decision-making pathways (Leemans et al., 2019).

The Directly follows models (DFMs) are more advantageous in this case

Case	Event
1	{'1': ' ', '2': 'X', '3': ' ', '4': ' ', '5': ' ', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': ' ', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'X', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'X', '2': 'X', '3': 'O', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'X', '2': 'X', '3': 'O', '4': 'X', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
1	{'1': 'X', '2': 'X', '3': 'O', '4': 'X', '5': 'O', '6': ' ', '7': 'O', '8': ' ', '9': ' '}
1	AI wins!
2	{'1': ' ', '2': ' ', '3': 'X', '4': ' ', '5': ' ', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
2	{'1': ' ', '2': ' ', '3': 'X', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
2	{'1': ' ', '2': ' ', '3': 'X', '4': ' ', '5': 'O', '6': ' ', '7': 'X', '8': ' ', '9': ' '}
2	{'1': 'O', '2': ' ', '3': 'X', '4': ' ', '5': 'O', '6': ' ', '7': 'X', '8': ' ', '9': ' '}
2	{'1': 'O', '2': ' ', '3': 'X', '4': 'X', '5': 'O', '6': ' ', '7': 'X', '8': ' ', '9': ' '}
2	{'1': 'O', '2': ' ', '3': 'X', '4': 'X', '5': 'O', '6': ' ', '7': 'X', '8': ' ', '9': 'O'}
2	AI wins!
3	{'1': ' ', '2': ' ', '3': 'X', '4': ' ', '5': ' ', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
3	{'1': ' ', '2': ' ', '3': 'X', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
3	{'1': 'X', '2': 'X', '3': ' ', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
3	{'1': 'X', '2': 'X', '3': 'O', '4': ' ', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
3	{'1': 'X', '2': 'X', '3': 'O', '4': 'X', '5': 'O', '6': ' ', '7': ' ', '8': ' ', '9': ' '}
3	{'1': 'X', '2': 'X', '3': 'O', '4': 'X', '5': 'O', '6': ' ', '7': 'O', '8': ' ', '9': ' '}
3	AI wins!

Figure 9: Simplified Event Log

than other process mining techniques such as the Alpha Algorithm, Heuristic Miner, and Inductive Miner. Each of these process mining techniques has its advantages, but there are also limitations. For example, the Alpha Algorithm can struggle with complex log structures. While generating models, this algorithm is very time-consuming, especially when the amount of event logs is very large, which can make the generated models not accurate enough and therefore impractical in real-world applications. The Heuristic Miner, although flexible, sometimes generates models that are difficult to interpret due to the inclusion of infrequent or exceptional behaviours. Similarly, the Inductive Miner, while effective in ensuring soundness and avoiding deadlocks, for a tic-tac-toe event log, it tends to over-generalize by aggregating all identical states from the second step, even if they come from different paths in the first step (Leemans et al., 2014).

In contrast, the DFMs is able to generate more streamlined and understandable models by focusing on direct inheritance between states. This makes it suitable in particular for the visualisation of decision paths in games. Using this technique, we created a simplified process model, as shown in Figure 10. It not only displays all the decision paths for these 10,000 rounds of the game but also highlights the frequency of occurrence of each path. Moreover, the number of states in this model has been reduced significantly compared with the concrete model.

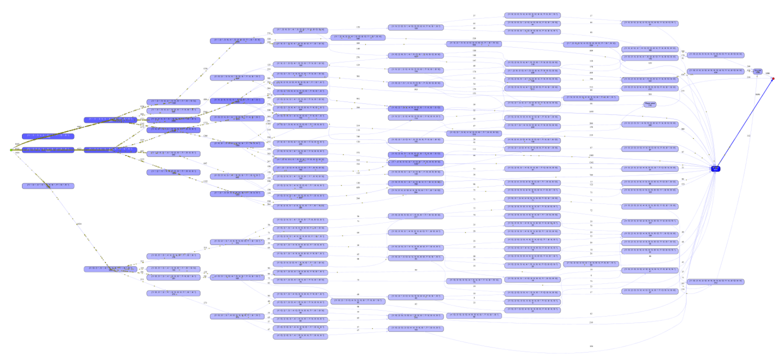

Figure 10: simplified Process Model

4.4 Evaluation of Process Models

The final phase of the case study was to assess the performance of the process model, which included accuracy, interpretability and practical utility. The purpose of the assessment is to verify that the model realistically and understandably reflects the decision-making process of the AI. In this case, the evaluation relies heavily on simulation games, where specific paths in the process model are manually examined to confirm that they are consistent with the expected behaviours.

In the following, the study chose a path to show the situation during a round of game simulation to evaluate the model performance. The premise of the game is that the human as the first player uses 'X' against AI. The expected result of this evaluation is that no matter how the human chooses the position to play, the AI can do its best to prevent the human from winning. As shown in Figure 11, after entering the initial node, the human placed 'X' in position 2. Observing the layout of the board, the AI agent placed 'O' at position 5 attempting to occupy the centre of the board. Following this move, human faced four potential choices. As illustrated in Figure 12, if the human chose to place the 'X' at square 8, the AI would take advantage of this by placing the 'O' at square 1, thereby establishing a potential victory path along the diagonal. Figure 13 indicated the human realised that the AI might win and placed an 'X' at position 9 to prevent the AI from connecting three consecutive pieces on the main diagonal. Subsequently, the AI counterattacked by occupying the square 7. This move was crucial and was the highlight of this round, as it put the AI in a very beneficial position. As can be seen in Figure 14, it prevented the human from connecting a line in the third row and also allowed itself to form two potential victory lines in the sub-diagonal and the first column. Analysis showed that for the current layout, if the AI's performance is good enough, then the AI would definitely win. This conclusion was consistent with the process model illustrated in Figure 15, demonstrating that regardless of the choices the human makes among the remaining three positions, the outcome would inevitably be a win for the AI. This indicated that the process model has the ability to replay the majority of situations in a real tic-tac-toe game, which confirmed its accuracy and interpretability.

This process model presented above effectively constructed the various states of the tic-tac-toe game into sequential process paths and provided an intuitive visual model for users to interpret the AI's behaviour. It contained all the possible positions that a human player could choose, and then the AI made different decisions according to different situations. This allowed users to simulate the game in such a way that no matter which position a human chooses, the user would be able to find the path in the model and get the corresponding response from the AI. The model facilitated human visualization of various game outcomes from placing pieces at different positions, thereby guiding users to adjust their strategies in time during the game. This comprehensiveness enables users to analyse the strategies adopted by the AI by concluding the regularities in the model, such as seizing the centre, occupying the corners and constructing dual threats, etc. In addition, this model can also be used as a teaching tool to help new players understand

and master the basic strategies of Tic-Tac-Toe effectively.

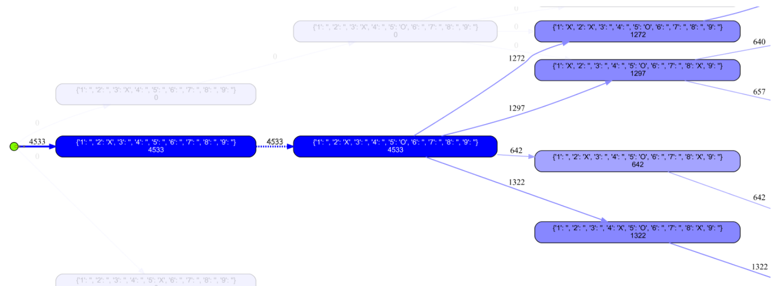

Figure 11: Sample Example 1

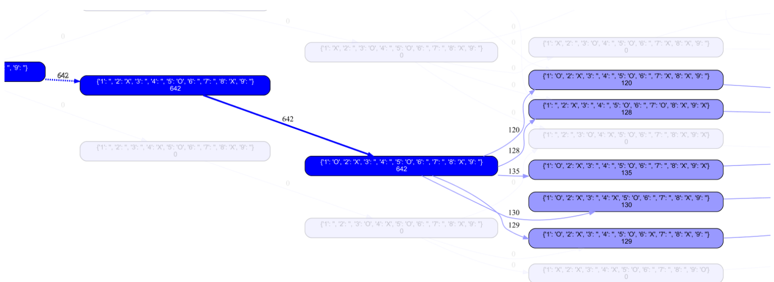

Figure 12: Sample Example 2

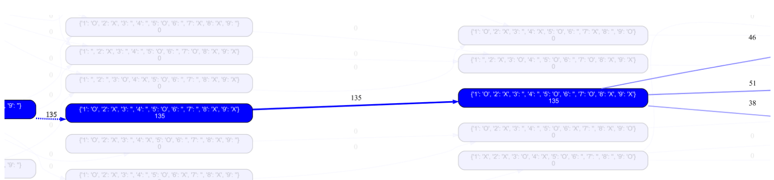

Figure 13: Sample Example 3

5 Discussion

This section discusses the main findings of this research, specifically how process mining helps users to understand AI behaviours. Additionally, it outlines some limitations of the pipeline methodologies taken in the current research, as well as possible future approaches that could be used to

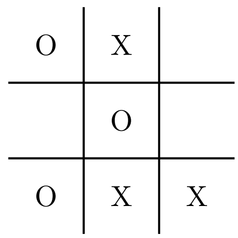

Figure 14: A Crucial Step in the Example

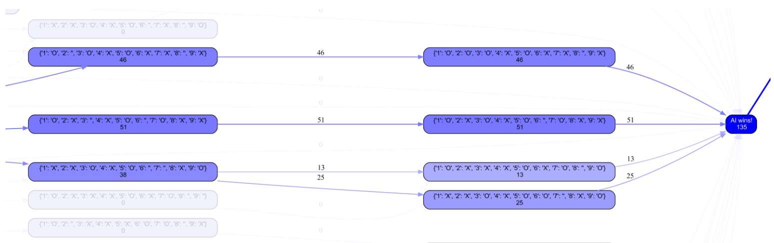

Figure 15: Sample Example 4

increase the likelihood of using process mining to explain more complex AI behaviours.

5.1 Findings

Using the case of tic-tac-toe, this study demonstrated the effectiveness of the methodology in explaining AI behaviour. The process mining technique is able to show the entire decision-making process from input to output in detail. It can capture the AI's decisions and chronologically visualise them to generate a clear flowchart. This presentation lays out the antecedents and consequences of each decision point and how these steps depend on each other to reach the final result.

This process-oriented and comprehensive presentation makes AI decision-making no longer a black-box, but transparent and easy to understand. As a result, users can not only observe the outcomes of decisions but also understand the specific steps that led to those consequences. Process models provide insight into the reasons behind the AI's decisions in various scenarios by mapping states in the environment or the AI's behaviour to nodes in the visual model, allowing users to track the AI's decisions in the order of activities. This transparency also enables users to build trust in AI decision-making. In addition, this approach also makes it faster and easier for non-technical users to understand the logic of AI models.

5.2 Limitations

Even though process mining was creatively used in this study to explain AI decision-making processes, it still has several limitations:

1. Dependency on Data Quality and Coverage: The accuracy and reliability of the process model depend heavily on the quality and completeness of the event logs. If these logs contain missing steps or insufficiently comprehensive coverage of the state, these inaccuracies could lead to an incomplete process model that does not accurately reflect the AI's decision-making process. For example, in the case of tic-tac-toe, this study took a randomised algorithm to simulate human behaviours so that each available position had a chance to be chosen, which makes the event log very comprehensive. Nevertheless, if some scenarios are missed during simulations, the process model would become incomplete. Under these circumstances, when the logs are replayed and these unrecorded states are encountered, corresponding paths in the process model cannot be found, thereby hindering the accurate identification of the AI's decision-making paths.

Moreover, we need to abstract the states and actions in the AI decision-making process in order to simplify the process model. This simplification may cause some features missing, which may affect the accuracy of the model. In the tic-tac-toe game, an initial attempt was made to represent the distribution of X and O pieces across corners, edges, and the centre using a list of six elements. The first three elements of this list denote the number of X pieces in the corners, on the edges, and in the centre, respectively. The last three elements similarly represent the count of O pieces in these positions. Although this representation is straightforward to understand, it results in an overly abstracted event log. For instance, one state represented by [1,1,0,0,0,1] indicates a board layout with two 'X' pieces and one 'O' piece, where one 'X' is in a corner, the other 'X' is on an edge, and the 'O' is in the centre. However, this layout can correspond to multiple actual game states, as shown in Figure 16. This abstraction leads to the loss of concrete features, making it difficult for users to determine the exact board layout and reach the desired results.

X	X	
	O	

X		
	O	
	X	

Figure 16: Two X and one O layout

2. Scalability challenge: Process mining techniques have been proven effective in simple environments, such as scenarios with a limited number of states. However, when attempting to extend this approach to broader or more complex AI systems, there would be significant obstacles. As the complexity of AI behaviours and operational environments increases, the event logs and process models invariably grow in complexity. This phenomenon is particularly evident in chess or real-time strategy games, where the number of potential states is much larger than that in basic games like tic-tac-toe. This makes it challenging for researchers to find suitable state representations as activity in event logs since it causes the model to become very complex. In this case, the lack of effective state simplification methods or feature extraction strategies may make the generated process models difficult to use for analysing the AI's behaviours.

3. Interpretability and Subjectivity: While process mining has helped users visualise the behaviour of AI to some extent, the interpretation of each node in the process model still relies strongly on the individual understanding and expertise of users (Tiwari et al., 2008). Something that is easy to understand for one user may be difficult to comprehend for another. This subjectivity might lead to process models having different validity among different stakeholders. This includes developers, end users who lack knowl-

edge of AI or process mining techniques, etc. In this case study, if the state aggregation strategy used in Tic-Tac-Toe had not been clearly explained to the users beforehand, it could cause many users to misunderstand that the process model is discontinuous. That is, one node could not be linked to the next node.

4. Applicability challenge: Process mining techniques were originally designed for traditional business process management and optimization, primarily aimed to deal with structured transactional data. When applied to the interpretation of AI behaviours, especially in complex AI systems such as deep learning models, these techniques may struggle to manage unstructured decision processes effectively. This is because such decision-making processes often do not follow clear predefined paths. The highly dynamic and unpredictable structure of AI decision-making processes can limit the effectiveness of process mining.

5.3 Future works

The above limitations emphasise the need for continued research in this field, and therefore future work should focus on several strategic areas to overcome current limitations and improve the effectiveness and applicability of the technology. The following are a few suggested directions for further research and development:

1. Applying Process Mining to Complex AI Systems: One possible direction for future research is to extend the process mining approach used in this study to more complex AI systems, such as self-driving cars and the medical domain. These systems have extensive datasets and intricate decision-making processes, where increased transparency and comprehensibility would be particularly beneficial. Developing approaches that can capture and explain the subtle behaviours of such AI systems is critical to enhancing safety, reliability and public trust.

2. Advanced Algorithms for Process Simplification: Alongside expanding the application domains, there is an urgent need to develop more advanced algorithms that could simplify process models automatically. When dealing

with large-scale datasets generated by complex systems such as self-driving cars, process models can become very complex if trying to include each specific step in the process model. Therefore, appropriate state merging or step simplification is particularly important for generating models.

3. Enhanced Data Collection and Quality Control: The collection and quality assurance of event logs also have a significant impact on process modelling. Future work could focus on how to help the system generate higher-quality event logs. This includes but is not limited to, creating algorithms that are capable of detecting and correcting errors in data records to ensure thorough coverage of all possible states and transitions in the AI decision-making process.

4. Ethical Considerations and Guidelines: With the growth of process mining applications in AI, it is crucial to consider the ethical implications as well. Process mining increases the transparency of deep learning, but it is also important to ensure that this transparency does not compromise the privacy or security of individuals. If the pursuit of excessive transparency leads to a leakage of personal privacy, it would have serious impacts on society. Therefore it is equally important for developers to use the explanations provided by the tools responsibly.

6 Conclusion

This research demonstrated the potential of process mining as a powerful tool in elucidating the decision-making process of AI systems, especially using Tic-Tac-Toe as a case study. Through a pipeline methodology of establishing environments, extracting and simplifying event logs, and applying process discovery algorithms, researchers were able to intuitively access the strategic decision paths of the AI agents. The generated process models not only enhance the transparency of AI operations but also make AI decisions more interpretable and rational.

The findings indicated that process mining can make a significant contribution to the field of XAI by providing a methodological foundation for analysing AI behaviours. This is particularly important in high-risk environments such as finance and healthcare, where AI decisions have profound implications. Through a clearer understanding of AI processes, stakeholders are able to better make informed decisions about the deployment and management of AI systems.

However, despite the results achieved, it is recognised that current approaches face challenges in terms of data quality, model simplification, generalisation, and scalability across a wider range of AI applications. Future work needs to address these limitations and develop more refined data handling and process simplification techniques, which could enhance the interpretability and applicability of AI systems in diverse environments.

By continuing to explore the application of process mining in the field of XAI, it is possible to further improve the transparency and credibility of AI systems. This improvement would contribute to safer and more responsible deployment of AI systems across a wide range of societal domains.

References

Antwarg, L., Miller, R. M., Shapira, B., and Rokach, L. (2021). Explaining anomalies detected by autoencoders using shapley additive explanations. *Expert systems with applications*, 186:115736.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

Borovska, P. and Lazarova, M. (2007). Efficiency of parallel minimax algorithm for game tree search. In *Proceedings of the 2007 international conference on Computer systems and technologies*, pages 1–6.

Buijs, J. C., van Dongen, B. F., and van der Aalst, W. M. (2014). Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *International Journal of Cooperative Information Systems*, 23(01):1440001.

Durán, J. M. and Jongsma, K. R. (2021). Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical ai. *Journal of Medical Ethics*, 47(5):329–335.

Gawde, S., Patil, S., Kumar, S., Kamat, P., Kotecha, K., and Alfarhood, S. (2024). Explainable predictive maintenance of rotating machines using lime, shap, pdp, ice. *IEEE Access*, 12:29345–29361.

Geurts, P., Irrthum, A., and Wehenkel, L. (2009). Supervised learning with decision tree-based methods in computational and systems biology. *Molecular Biosystems*, 5(12):1593–1605.

Gohel, P., Singh, P., and Mohanty, M. (2021). Explainable ai: current status and future directions. *arXiv preprint arXiv:2107.07045*.

Islam, S. R., Eberle, W., Ghafoor, S. K., and Ahmed, M. (2021). Explainable artificial intelligence approaches: A survey. *arXiv preprint arXiv:2101.09429*.

Lamghari, Z. (2022). Process mining: Basic definitions and concepts. *International Journal*.

Leemans, S. J., Fahland, D., and Van Der Aalst, W. M. (2014). 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 11*, pages 66–78. Springer.

Leemans, S. J., Poppe, E., and Wynn, M. T. (2019). Directly follows-based process mining: Exploration & a case study. In *2019 International Conference on Process Mining (ICPM)*, pages 25–32. IEEE.

Magesh, P. R., Myloth, R. D., and Tom, R. J. (2020). An explainable machine learning model for early detection of parkinson's disease using lime on datscan imagery. *Computers in Biology and Medicine*, 126:104041.

Mahbooba, B., Timilsina, M., Sahal, R., and Serrano, M. (2021). Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. *Complexity*, 2021:1–11.

Manimegalai, R. and Sheeba, A. (2021). Game optimization using minimax and alpha-beta pruning. *Journal of Information Technology and Society*, 1(01).

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Nohara, Y., Matsumoto, K., Soejima, H., and Nakashima, N. (2022). Explanation of machine learning models using shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine*, 214:106584.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rozinat, A., de Medeiros, A. K. A., Günther, C. W., Weijters, A., and van der Aalst, W. M. (2008). The need for a process mining evaluation framework in research and practice: position paper. In *Business Process Management Workshops: BPM 2007 International Workshops, BPI, BPD, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia, September 24, 2007, Revised Selected Papers 5*, pages 84–89. Springer.

Savelli, R. M. and de Beauclair Seixas, R. (2008). Tic-tac-toe and the minimax decision algorithm. *Luiz H. de F., Waldemar, C., ve Roberto L.(Eds.), Lua programming gems*, pages 239–245.

Sriram, S., Vijayarangan, R., Raghuraman, S., and Yuan, X. (2009). Implementing a no-loss state in the game of tic-tac-toe using a customized decision tree algorithm. In *2009 International Conference on Information and Automation*, pages 1211–1216. IEEE.

Tiwari, A., Turner, C. J., and Majeed, B. (2008). A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal*, 14(1):5–22.

Van Der Aalst, W. (2012). Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, 3(2):1–17.

Van der Aalst, W. M. (2014). Process mining in the large: a tutorial. *Business Intelligence: Third European Summer School, eBISS 2013, Dagstuhl Castle, Germany, July 7-12, 2013, Tutorial Lectures 3*, pages 33–76.

Vieira, C. P. R. and Digiampietri, L. A. (2020). A study about explainable articial intelligence: Using decision tree to explain svm. *Revista Brasileira de Computação Aplicada*, 12(1):113–121.

Von Eschenbach, W. J. (2021). Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622.

Yin, S., Li, H., Sun, Y., Ibrar, M., and Teng, L. (2024). Data visualization analysis based on explainable artificial intelligence: A survey. *IJLAI Transactions on Science and Engineering*, 2(3):24–31.

Zhao, X., Huang, W., Huang, X., Robu, V., and Flynn, D. (2021). Baylime: Bayesian local interpretable model-agnostic explanations. In *Uncertainty in artificial intelligence*, pages 887–896. PMLR.