Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 241 (2024) 187–194

Procedia Computer Science

www.elsevier.com/locate/procedia

The 21st International Conference on Mobile Systems and Pervasive Computing (MobiSPC)
August 5-7, 2024, Marshall University, Huntington, WV, USA

Forecasting Future Behavior: Agents in Board Game Strategy

Nathan Damette[a,b,*], Maxime Szymanski[a], Yazan Mualla[a], Igor Tchappi[b], Amro Najjar[c], Mehdi Adda[d]

[a] *UTBM, CIAD UMR 7533, F-90010 Belfort cedex, France*
[b] *FINATRAX, SnT, University of Luxembourg, 6 rue Richard Coudenhove-KalergiL-1359, Kirchberg, Luxembourg*
[c] *Luxembourg Institute of Science and Technology (LIST), L-4362 Esch-sur-Alzette, Luxembourg*
[d] *Université du Québec à Rimouski, Rimouski, QC G5L 3A1, Canada*

Abstract

This paper presents findings on machine learning agent behavior prediction in a board game application developed by a group of students. The goal of this research is to create a model facilitating collaboration between a user and an AI to play together in the board game using a Human-in-the-Loop architecture. By injecting explainability, the aim is to enhance communication and understanding between the user and the AI agent. Featuring a competitive Artificial Intelligence (AI) based on the Proximal Policy Optimization model, this research explores methods to make AI decisions transparent for enhanced player understanding. Two predictive models, a Decision Tree (DT) and a Deep Learning (DL) classifier, were developed and compared. The results show that the DT model is effective for short-term predictions but limited in broader applications, while the DL classifier shows potential for long-term prediction without requiring direct access to the game's AI. This study contributes to understanding human-AI interaction in gaming and offers insights into AI decision-making processes.

Keywords: Explainable AI; Agent behavior; Explainable Reinforcement learning; Human-AI interaction

1. Introduction

SplendIA[1] is a student project created by a small group of students to develop a web application for playing the board game Splendor. This application features an AI opponent implemented using the Proximal Policy Optimization (PPO) model, allowing players to compete against the AI. The initial goal of this project was to develop solutions that render the AI's decisions explainable. This would enable users to access explanations of the AI's decisions, assisting them in better understanding the game's strategy. Additionally, a further objective was to develop a "Human-in-the-

* Corresponding author. Tel.: +33-6-95-90-20-10
E-mail address: damette.nathan@gmail.com

1877-0509
10.1016/j.procs.2024.08.026

Loop" model to explore the possibilities of human-AI collaboration. This model refers to humans whose role in the environment is active [6]. This paper discusses the research and experimentation with various methods to predict the AI's future actions, aiming to gain insights into its decision-making process.

Splendor is a strategy board game known for its simple yet deep gameplay. The objective of the game is to accumulate prestige points, primarily by purchasing development cards. The game includes the following components: (i) Gem tokens representing different resources. (ii) Development cards, categorized into three levels. (iii) Noble tiles, which provide bonus prestige points. Players take turns, during which they can perform one of three actions: (i) Collect gems, which are used to purchase development cards. (ii) Reserve a card for later purchase and take a gold token. (iii) Purchase a development card using the collected gems. Development cards provide both prestige points and gem bonuses, which can be used as discounts on future card purchases.

A game of Splendor can easily be characterized as a series of cycles, ranging from one to a few turns. In each cycle, the player begins by accumulating resources for their next purchase. The cycle culminates in the final turn, marked by a significant purchase, which we will refer to as a "remarkable move". A key element of success in Splendor is efficient resource management. Players must balance between collecting gems, reserving cards, and purchasing developments. Early in the game, focusing on acquiring cards that provide gem bonuses can significantly reduce the cost of later purchases. Additionally, players should be aware of the noble tiles, as these provide a significant number of prestige points and can be a deciding factor in the game's outcome.

This paper is structured as follows: Section 2 discusses the related works. Section 3 explains the general objectives of the project, while Section 4 presents the different experiments and their results. Finally, Section 5 concludes the paper and outlines future work.

2. Related work

Understanding the decision-making processes of artificial intelligence (AI) systems is crucial for various reasons. First and foremost, it enhances transparency and accountability, enabling users to comprehend why AI systems make specific decisions or predictions. This aspect of AI research falls within the domain of Explainable AI (XAI) [17, 1, 3, 15], which aims to develop methods and techniques for making AI systems more interpretable and understandable to humans [9]. XAI has gained significant traction in recent years due to the growing adoption of AI technologies across various domains, including healthcare [18], finance, and autonomous vehicles [8, 2]. Moreover, the interaction between humans and AI models is an essential field of research. It is increasingly recognized that fostering effective communication and understanding between humans and AI systems is vital for their successful integration into various applications [19, 20, 13]. Explainability plays a pivotal role in facilitating this interaction, allowing for clearer communication and collaboration between human users and AI agents [9]. Researches recognize the importance of XAI in building trust and confidence in AI systems, particularly in high-stakes applications where the consequences of AI errors can be significant [10]. Several approaches have been proposed in the literature to achieve explainability in AI systems. One common approach involves post-hoc interpretation methods [5], where explanations are generated after the AI model has made predictions.

Understanding how AI models make decisions can be facilitated by techniques such as analyzing the importance of features [14], creating saliency maps [11], and using local surrogate models [12]. These methods provide insights into the mechanisms underlying AI decision-making. In recent studies, there has been a specific focus on improving the transparency of decision-making processes in reinforcement learning (RL) agents [4]. Researchers are developing visualization methods tailored to RL, such as Shapley value-based techniques [16]. These methods help to identify how different combinations of actions and states contribute to an RL agent's decision-making strategy, thereby putting light on the factors influencing its decisions within the realm of machine learning.

In summary, the field of AI research continues to evolve, with a growing emphasis on transparency, interpretability, and predictability in AI systems. Understanding the decision-making processes of AI agents and developing explainable AI techniques are critical steps towards building trustworthy and reliable AI systems for real-world applications, fostering better communication and collaboration between humans and AI models [7].

[1] https://github.com/kalharko/splendia

3. General objectives

The overarching objectives of this paper are to transform SplendIA into an innovative tool that aids players in understanding the game better and refining their strategies across various skill levels. To achieve this programmatic goal, an adaptation of the project's architecture was conceptualized, aiming to facilitate collaboration between human players and AI entities rather than pitting them against each other. Through collaboration, both users and AI agents stand to benefit from shared insights into the game and thought processes. This human-AI collaboration model was designed to adhere to the "Human-in-the-loop" paradigm.

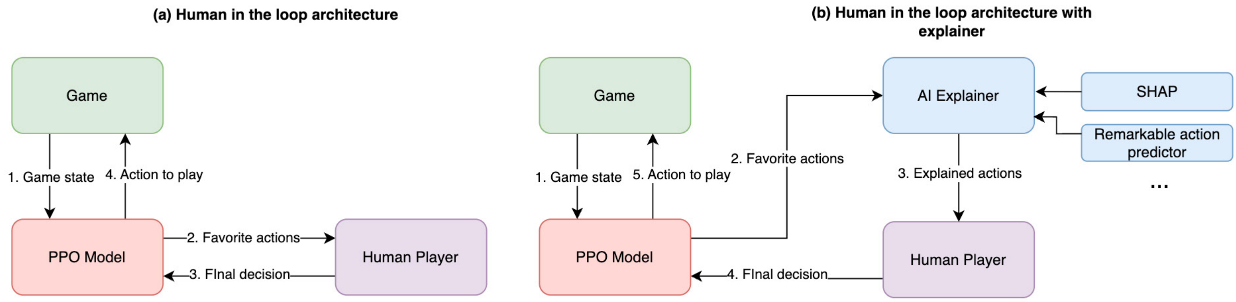

Fig. 1: Human in the loop architecture

The graph in part (a) of Figure 1 depicts the initial architecture envisioned to realize such a model. In this setup, the state of the game is relayed to the AI at each turn, which then computes its favored actions. Subsequently, a selection of the top actions is presented to the human player for final decision-making. This architecture fosters rapid collaboration between human and AI participants, leveraging their collective input to arrive at decisions efficiently. By minimizing exchanges and communication overhead, it streamlines decision-making processes and mitigates the risk of decision paralysis. Feedback in this architecture is provided to both actors via rewards based on the evolving state of the game board. These rewards inform AI training and are also communicated to the human player. Moreover, the human player's decisions can influence the reward function, thus contributing to the RL model's training process. The decision of the RL model can be represented as a list A of actions a_i a paired with their estimated values of interest v_i like such:

$$A = \{(a_1, v_1), (a_2, v_2), ..., (a_n, v_n)\}$$

3.0.1. Incorporating Explained Decisions

To facilitate collaboration between human and AI participants, it is crucial to ensure that the AI's proposed actions and their implications are comprehensible to the human player. Furthermore, enhancing the human player's understanding of why the RL model suggests certain actions improves the collective decision-making process, ultimately enhancing the model's performance. With this objective in mind, it is beneficial to augment the previously described architecture with tools that provide explanations of proposed actions to the human player. The updated protocol is illustrated in part (b) of Figure 1. Thus, for each action a_i, there are associated explanations denoted as e_i. The context of these explanations is about the action itself. Mathematically, this can be represented as:

$$A = \{(a_1, v_1, e_1), (a_2, v_2, e_2), \ldots, (a_n, v_n, e_n)\}$$

Here, e_i represents the explanation associated with action a_i. Furthermore, there are global explanations E associated with the entire list A. These global explanations offer overarching insights into the collection of actions and their values. Mathematically, this can be represented as:

$$E = \{e'_1, e'_2, \ldots, e'_m\}$$

Where e'_i represents a global explanation associated with the entire list A. These explanations play a crucial role in enhancing the human player's understanding of the AI's proposed actions and contribute to informed decision-making in the collaborative game play setting. For instance, one such algorithm is SHapley Additive exPlanations (SHAP), which computes the importance of each input in the model's decision-making process. This algorithm offers insights independent of the actions presented in the list. The result of the SHAP value will be contained in e'_i.

As discussed in Section 1, another pivotal aspect of the game Splendor is the concept of "remarkable moves". These moves form the cornerstone of players' strategies. Identifying the remarkable actions that the AI intends to execute is crucial for understanding its strategy. Consequently, an algorithm can be developed to predict the future remarkable moves that the AI plans to make for each proposed action. These explanations would then be associated with each action as e_i. The paper primarily focuses on experimental endeavors aimed at developing this last algorithm.

4. Experiments

4.1. Dataset Overview

In this research, the primary objective was to predict AI behavior. The dataset needed to represent the data the AI typically processes, ensuring the predictions closely match the actual decisions of the RL model. To achieve effective prediction, the AI's training was halted, converting it from a stochastic to a deterministic model, where the agent consistently selects the most favorable decisions without exploring other options. Although this transformation is not essential, it enhances prediction accuracy. Continuous training would require regular updates to the Classifier predictor, whereas the Decision Tree (DT) model would not.

The PPO algorithm uses the state of the board as a normalized vector, containing most information a player could glean from the game. This method was used to generate input for various predictive models. The next step was data labeling, aiming to predict specific actions of the agent. By owning the specific model, ample data was generated. The deterministic agent played against a random algorithm, and board states were recorded at each turn, marking any "remarkable" actions. The dataset was constructed by iterating backwards, assigning the next remarkable action to each board state.

Labeling involved two approaches: (i) Predicting the position of the next card the agent would purchase (positions 0 to 14, including 12 cards in the shop and 3 reserved cards). (ii) Predicting the ID of the card instead. With 90 distinct cards in Splendor, this approach broadened the classification range to 90 classes, potentially fostering more precise models by bypassing positional characteristics. For instance, an AI might reserve a card to buy later, altering its position.

Thus, the dataset was constructed with state vectors paired with the corresponding action – the next card the Agent purchased.

4.2. AI Behavior Prediction Approaches

To achieve the goal of predicting the AI's future actions, and considering the options available from the initial project, we came up with two distinct solutions, each employing a different method. The first solution relies on the creation of DTs, where the focus is on identifying and estimating the most valuable states and actions. This involves setting up a system that can evaluate different game scenarios and pinpoint the strategic choices the AI is likely to make. The second solution we envisioned involves developing and training a Deep Learning (DL) classifier. This classifier is tasked with learning the thought patterns of the Agent. It's about making sense of the data to figure out how the AI decides on its moves.

The DT approach offers a more structured way of looking at the problem, breaking down the AI's decisions into a series of logical steps. This approach aligns well with the agent's strategy of taking one action at a time, effectively imitating its decision-making behavior. In contrast, the DL classifier seeks to capture the more complex and subtle patterns in the AI's decision-making process.

4.2.1. First solution : Decision Tree

The initial step in the DT solution involves constructing a tree of potential actions and sequences that the AI could undertake. The root node represents the current game state received by the Agent before making its decision. This tree is then expanded from the root node to encompass all possible series of actions. Alternating between the PPO agent and a random player, the model retrieves the list of permissible actions for the current player. For each node, the game

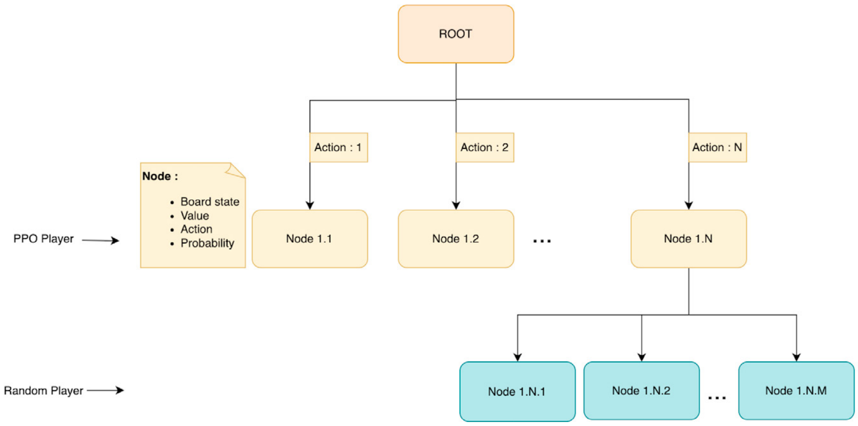

Fig. 2: Decision Tree architecture

is updated to reflect the new board state for the next player. Figure 2 shows that each node contains the board state obtained after an action is played. While actions can be seen as branches connecting nodes, they are also stored within the child node itself.

This method primarily focuses on calculating the preferred nodes and paths of the Agent to predict its future decisions. For nodes of odd generations (1, 3, 5, etc.), the algorithm computes a value directly provided by the PPO model. Supplying the current board state to the PPO enables retrieval of action probabilities. These probabilities are used to assess the desirability of each child node in accordance with the Agent's preferences. The algorithm then calculates the probabilities of each action, as shown in Figure 3. The root node starts with an initial probability of 1. For each new generation, the probabilities of the children nodes are determined as follows: This algorithm follows the logic of the deterministic agent, which consistently opts for its most preferred decision, making it as probable as the preceding game state. It also mirrors the behavior of the agent's opponent, who makes decisions through a uniform random process. Finally, to estimate the probability of each significant action, the algorithm aggregates the probabilities of the tree's leaf nodes, summing the probabilities of identical actions. The algorithm's output is a list of actions and their corresponding probabilities, which can be cross-referenced with the dataset. Optionally, the algorithm can also retrieve the card in the significant position when a remarkable action occurs, returning its ID along with the action made.

Algorithm 1 Calculate Child Node Probabilities

for each node n in the current generation of the tree **do**
    if generation of n is even **then**
        for each child c in n.children **do**
            $c.\text{prob} \leftarrow \frac{n.\text{prob}}{\text{length}(n.\text{children})}$
        end for
    else
        $b \leftarrow$ best child of n
        for each child c in n.children **do**
            $c.\text{prob} \leftarrow \begin{cases} n.\text{prob} & \text{if } c = b \\ 0 & \text{otherwise} \end{cases}$
        end for
    end if
end for

4.2.2. *Second solution : Deep Learning Classifier*

The DL classifier is designed to process the same data as the PPO agent to predict the next card purchase. Initially, the classifier was developed with an architecture identical to that of the agent. The idea was that if they shared the

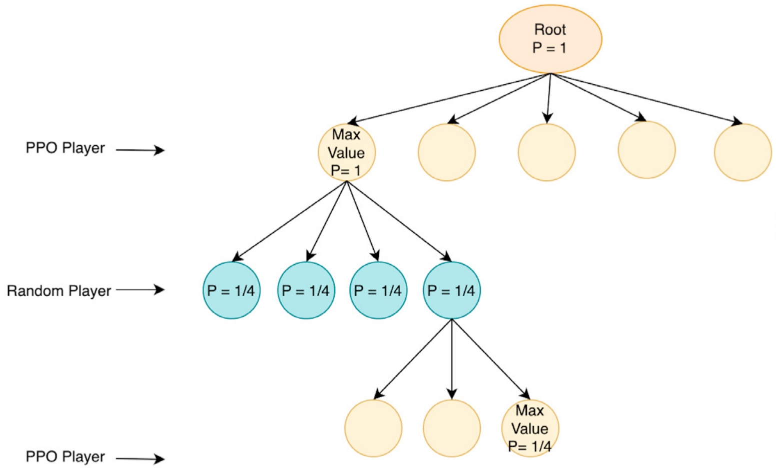

Fig. 3: Action probability estimation

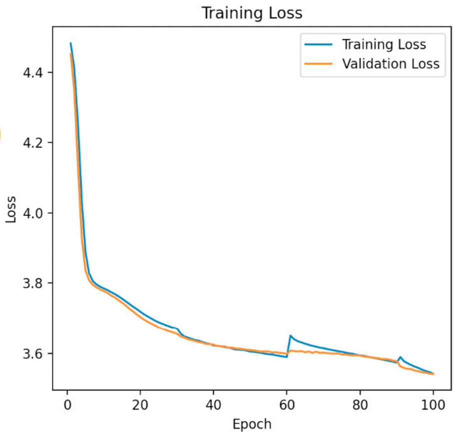

Fig. 4: Classifier loss over first 100 epochs

same neural layers and inputs, both networks would exhibit similar behaviors. However, since the classifier's objective is not to mimic the agent but to predict a specific card ID, it required a distinct architecture for effective operation. The final architecture chosen is as follows: (i) The input layer has a dimension of 88, corresponding to the size of the input data. (ii) A fully connected layer of 32x32 neurons. (iii) The output layer with a dimension of 90, each representing a different card ID. ReLU (Rectified Linear Unit) functions are used for activation between the layers.

The training of the Classifier begins with generating a dataset consisting of 500 games for training and 100 for validation. The training is conducted using batches of 500 games each. This batch size was found to be effective in preventing the model from over-fitting. To further mitigate over-fitting, the dataset of 500 games is regenerated every 30 epochs. However, the validation dataset remains unchanged, as 100 games, averaging 60 actions per game, provide sufficient diversity for effective validation. Experimentally, the optimal learning rate for this process is determined as 0.0002.

4.3. Results

4.3.1. Decision Tree results

To estimate the accuracy of this first model, we compared the actual actions taken by the agent with the estimations made by the DT, varying certain parameters. In Figure 5, we assess the efficiency of the algorithm based on the maximum depth of the tree and the K highest probabilities. Given the nature of Splendor, where players often focus on multiple cards at once to increase their chances of acquiring one, it's insightful to observe whether the action taken by the PPO agent is not just the top result of the DT prediction, but also if it appears within a relatively close range. Thus, we evaluated the accuracy based on the algorithm's K the highest predictions. It is important to note that this success rate is calculated as the ratio of correct predictions to total predictions. However, when the player is unable to purchase cards, the total number of predictions does not increase.

The initial experiments aimed to compare the actions executed by the agent with those estimated by the algorithm. This comparison primarily focused on the positions of the cards purchased. As depicted in Figure 5, the accuracy is exceptionally close to 100 percent at a tree depth of 1. This result is expected, as the model with a depth of 1 essentially mirrors the deterministic agent itself. However, the outcomes are also particularly satisfying at depths of 3 and 5. These predictions can be considered valid for forecasting the next card the agent will purchase within the subsequent 1-2 turns. The subsequent phase involved comparing the values of the card IDs, a prediction task that is inherently more uncertain due to the wider range of possible cards. For a tree depth of 1, the results are nearly identical to the previous method, since the card positions remain fixed. However, the performance of the Prediction Tree drops significantly in this scenario, highlighting its limitations. This model does not achieve sufficient precision in predicting the next card ID to be considered reliable for practical use.

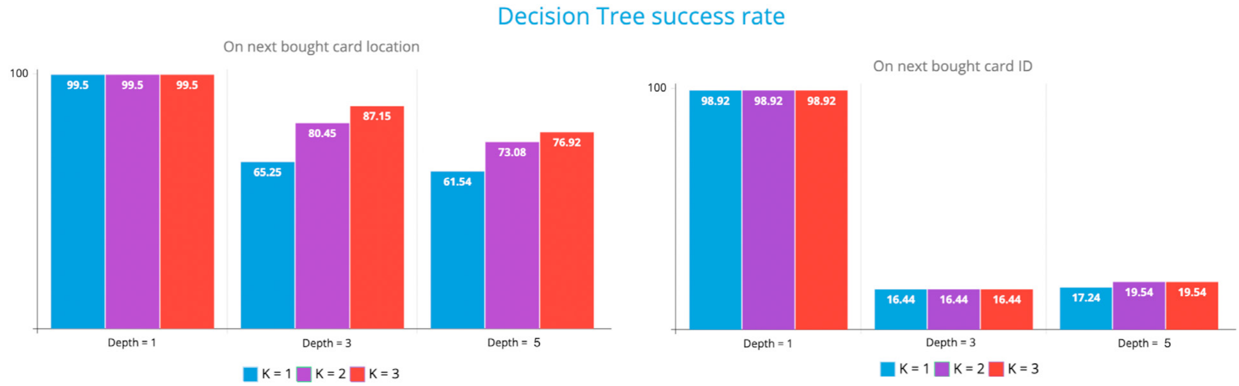

Fig. 5: Decision Tree predictions accuracy

4.3.2. *Classifier results*

As outlined in its introduction, the Classifier's primary function is to estimate the ID of the next card that the Agent will purchase. This task was proved to be significantly more challenging with the previous solution due to the large number of classes it entails, compared to the positions on the board. This complexity, coupled with the vast array of possible game states and the numerous parameters influencing each action, makes training the Classifier network particularly arduous. Through experimentation, it was observed that the best progress in training occurred when the classifier was fed a substantial volume of data simultaneously. Generating and processing such a large dataset is demanding and consequently prolongs the training duration.

The training regimen for the model was segmented into three phases, each comprising 500 epochs. Referring to Figure 6, we observe the model's loss and validation rate. Unfortunately, the model has not yet converged to its maximum accuracy and currently stands at a mere 16 percent, which complicates analysis. However, the validation rate has already reached the accuracy of the DT and exhibits a consistent upward trend, offering hope for continued improvement and to overpass the previous solution.

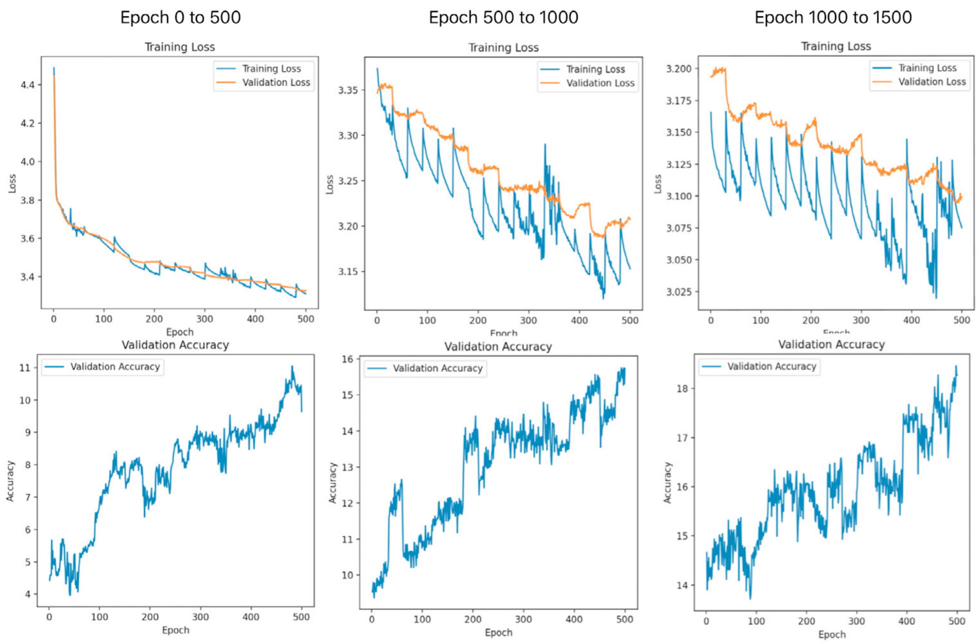

Fig. 6: Classifier Loss and accuracy evolution over 1500 epoch

4.3.3. Discussion

The comparison of the models highlights the advantages and drawbacks of each. The DT effectively predicts the PPO agent's short-term behavior but lacks consistency in varying scenarios and becomes computationally heavy for medium-term predictions. It also requires access to both the agent and the game model. Conversely, the classifier offers more consistent accuracy over the long term, does not require access to the agent or game model, but needs substantial data for training.

5. Conclusion

This study explored machine learning agent behavior prediction in the board game application SplendIA. The main objective was to develop methods enhancing player understanding of AI decisions and human-AI collaboration. By comparing two predictive models, a DT and a Deep Learning classifier, this paper provides insights into AI decision-making and predictability.

The DT model effectively predicts short-term behavior, closely mirroring the PPO agent. However, it struggles with broader and long-term predictions and has high computational costs for medium-term predictions, requiring access to both the agent and game model.

The Deep Learning classifier shows promise for long-term predictions without needing direct access to the game's AI or model. Although it struggles in short-term predictions compared to the DT, it offers more consistent long-term accuracy and can predict information not directly linked to the original model but requires substantial training data.

Overall, this study contributes to understanding AI decision-making. Future research could refine these models, explore additional features, and investigate real-time adaptation during gameplay to enhance collaboration with AI agents. These models can also be applied beyond gaming to improve AI decision-making and human-AI interaction in various contexts.

References

[1] Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., et al., 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion .
[2] Atakishiyev, S., Salameh, M., Yao, H., Goebel, R., 2023. Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions arXiv:2112.11561.
[3] Das, A., Rad, P., 2020. Opportunities and challenges in explainable artificial intelligence (xai): A survey. IEEE .
[4] Du, M., Liu, N., Hu, X., 2019. Techniques for interpretable machine learning. Commun. ACM 63, 68–77.
[5] Mahya, P., Fürnkranz, J., 2023. An empirical comparison of interpretable models to post-hoc explanations. AI 4, 426–436.
[6] Mualla, Y., 2020. Explaining the Behavior of Remote Robots to Humans : An Agent-based Approach. Theses. UBFC.
[7] Mualla, Y., Kampik, T., Tchappi, I.H., Najjar, A., Galland, S., Nicolle, C., 2020a. Explainable agents as static web pages: Uav simulation example, in: Int Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems, Springer, Cham. pp. 149–154.
[8] Mualla, Y., Najjar, A., Kampik, T., et al., 2019. Towards explainability for a civilian uav fleet management using an agent-based approach .
[9] Mualla, Y., Tchappi, I., Kampik, T., et al., 2022. The quest of parsimonious xai: A human-agent architecture for explanation formulation. Artificial Intelligence 302, 103573.
[10] Mualla, Y., Tchappi, I., Najjar, A., Kampik, T., Galland, S., Nicolle, C., 2020b. Human-agent explainability: an experimental case study on the filtering of explanations, in: 12th International Conference on Agents and Artificial Intelligence, Valletta, Malta, February 22-24, 2020.
[11] Mundhenk, T.N., Chen, B.Y., Friedland, G., 2020. Efficient saliency maps for explainable ai arXiv:1911.11293.
[12] Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "why should i trust you?": Explaining the predictions of any classifier arXiv:1602.04938.
[13] Rong, Y., Leemann, T., Nguyen, T., et al., 2023. Towards human-centered explainable ai: A survey of user studies for model explanations .
[14] Saarela, M., Jauhiainen, S., 2021. Comparison of feature importance measures as explanations for classification models. SN Appl Sci 3.
[15] Saeed, W., Omlin, C., 2023. Explainable ai : A systematic meta-survey of current challenges and future opportunities. Knowl-Based Syst .
[16] Sundararajan, M., Najmi, A., 2020. The many shapley values for model explanation 119, 9269–9278.
[17] Swamy, V., Frej, J., Käser, T., 2023. The future of human-centric explainable artificial intelligence (xai) is not post-hoc explanations .
[18] Tjoa, E., Guan, C., 2021. A survey on explainable artificial intelligence (xai): Toward medical xai. IEEE Transactions on Neural Networks and Learning Systems 32, 4793–4813. doi:10.1109/TNNLS.2020.3027314.
[19] Wang, D., Churchill, E., Maes, P., Fan, X., Shneiderman, B., Shi, Y., Wang, Q., 2020. From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people , 1–6.
[20] Wang, D., Maes, P., Ren, X., Shneiderman, B., Shi, Y., Wang, Q., 2021. Designing ai to work with or for people?, in: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA.