The 36th Canadian Conference on Artificial Intelligence
DOI: 0

Canadian AI 2023
5—9 June 2023, Montréal

Mimicking Electronic Gaming Machine Player Behavior Using Reinforcement Learning

Gaurav Jariwala[†,*], Vlado Keselj[†]
[†] Faculty of Computer Science, Dalhousie University

Abstract

This study looks at how Reinforcement Learning (RL) approaches can be used to understand player behavior in Electronic Gaming Machines (EGMs) found in venues like casinos. The gaming business is keen to learn about the many types of player behavior and create virtual players mimicking these behaviors. To achieve this, we trained RL models to mimic player behavior by grouping different playing styles with K-means clustering and determining termination states for one of the playing behaviors. The Proximal Policy Optimization (PPO) and Actor Critic using Kronecker-Factored Trust Region (ACKTR) models were subsequently implemented, with the agents being rewarded based on their proximity to the termination states. Our findings suggest that the ACKTR model performed better than the PPO model, with the generated playing behavior demonstrating a high level of statistical similarity to real-world player behavior within the selected cluster.

Keywords: Reinforcement Learning, Behavior Analysis, EGMs, Clustering

1. Introduction

Electronic Gaming Machines (EGMs), which are gambling machines installed in a range of locations such as casinos, bars, and hotels, have become more and more popular, attracting the attention of not just the gaming industry but also of the government and researchers. The availability of gambling activities has increased due to internet gaming [1]. According to the 2018 Canadian Community Health Survey (CCHS), nearly two-thirds (64.5%) of Canadians aged 15 or older (18.9 million) reported gambling at least once in the previous year [2]. These games can be highly addictive as there is a chance of winning a high amount of money in a short duration. In some situations, people continue to spend time and money on gambling even though it affects them mentally and financially, which is identified as a disorder named the problem gambling [3, 4]. The government is working to tackle this problem by promoting responsible gambling resources, while researchers have conducted studies [5–8] to identify the gamblers at-risk of problem gambling. Most of the work in this field has been done on limiting problem gambling, however, generating virtual players that mimic the real-world players can help industry and researchers test experiments and conduct a detailed behavioral analysis to understand the behavioral patterns in-depth without identifying the person. Future behavior in the particular situation of a real player could be anticipated using this virtual player.

The major challenge in working with EGM data is that they are anonymous, which means that these machines do not keep any identifying information about the players in their logs and that they do not distinguish between various players' sessions. In this research, we are using sessions that were sessionized based on balance and the pauses taken while playing [9]. However, these sessions do not contain any information about the experience of the player.

We developed agents using Reinforcement Learning (RL) to produce such sessions in order to mimic the behavior of real-world EGM players. To mimic one particular player's behavior, we have to separate all types of behaviors. We used an unsupervised learning

[*]gaurav.jariwala@dal.ca

algorithm, namely the K-means clustering for grouping the sessions with similar behavior of players. This similarity is measured using the Euclidean distance. Moreover, for replicating a particular behavior we are defining termination states using some selected features and values from one cluster. These termination states are used for reward calculation of the RL models, which get higher rewards as the RL agent approaches towards these termination states. The closeness of the current state of the agent to the termination states is measured by the Euclidean distance. We trained two RL algorithms, Proximal Policy Optimization (PPO) and Actor Critic using Kronecker-Factored Trust Region (ACKTR), to generate sessions with a particular behavior.

1.1. **Contributions**

The four most important contributions of this paper are as follows:

(1) Developed a methodology for mimicking the behavior of real-world players using reinforcement learning.
(2) Demonstrated the effectiveness of the proposed approach in reducing game development time and costs by using virtual players for testing and validation.
(3) Contributed to the advancement of the field of artificial intelligence and machine learning in the context of mimicking game players.
(4) Provided a valuable tool for game developers to improve the player experience and create more engaging games.

The remainder of the paper is organized in the following way. Section 2 reviews backgrounds on EGM and related work. In section 3, we discuss the methodology of this study. Section 4 details the experimental setup. In section 5, we discuss the results of Reinforcement Learning models and compare them with real-world players' sessions, followed by providing a conclusion and future directions of the concept in section 6.

2. **Background and Related Work**

Electronic Gaming Machines (EGMs) [10] are a common type of gambling machine found in casinos, clubs, and other public areas where people congregate for recreation. Although these devices, which use sophisticated technology, are actually computers, many of them still have reels that purport to spin and are evocative of earlier gambling machines. A random number generator is the base of every EGM. The computer retrieves the numbers created at that moment and transforms them into a display on the screen when a button or touch screen is pressed. The numbers represent a location on a reel map (the quantity and arrangement of symbols on each virtual reel) and a pay table (the payouts for any combination of symbols appearing on a line). For instance, the pay table will be used to map the random process's generation of three cherries to a payout of, say, two credits. These machines don't keep track of most of the play data and are stateless. Loyalty cards [11] is a major update that certain venues have implemented that are used to track customer information in the casino. As a result, well-formed data that takes into account playing sessions, games played, and money spent is produced. With this, the sessionizing task is entirely relinquished, as well as a history of user play data is also provided. Loyalty cards are not required and are not even used by the majority of venues [12], therefore typical data processing is still in use.

Latifi [9] effectively sessionized user datasets using EGM logs, which contain game data and metadata but no user ID. Assumptions were made that players only use one machine and sessions start with a cash-in and end with a cash-out or playing all credits and turning off the machine. The second assumption took into account the minimum cash balance and idle time threshold. If the time gap exceeds the idle time threshold and credit is less than the minimum, the session is terminated. Multiple cash-outs within a session are allowed. Experiment results show that using idle time of a few minutes and a minimum amount of money around a couple of dollars gives actual player session duration. The study uses data sessionized using this method, making it important.

Studies have been conducted on detecting gamblers' personas and predicting at-risk problem gambling using unsupervised learning techniques like clustering [5–8]. Problem gambling is characterized by excessive gambling behavior despite negative consequences, and often co-occurs with other negative habits such as substance abuse and food disorders [3, 13, 14]. Adami et al. [5] proposed indicators based on wager volatility and number of games played to identify a group of medium-risk players that were not recognized by Braverman and Shaffer [8].

Mosquera and Keselj [6] identified game-play types based on session start and end times using k-means clustering on EGM data (win or no win to cash out). They used ANOVA and Tukey's HSD to compare clusters. To refine results, Latifi [9] advised using DBSCAN before k-means. Although the researcher uses a Multivariate Convolutional LSTM neural network to quickly classify playstyle, it does not significantly improve performance when more than 40 transactions are analyzed.

Deep neural network advancements have had a significant impact on the field of Reinforcement Learning (RL), allowing complicated decision-making issues with high-dimensional state and action spaces to be solved [15]. Deep Reinforcement Learning (DRL) has been used in a variety of fields, including robotics [16, 17] and video games [18–21]. By studying the relationship between dopaminergic neuron activity and reward prediction errors, researchers in psychology and neuroscience have found evidence that the brain uses RL algorithms [22].

Wu and Izawa [23] studied the effect of regret on motivation in problem gambling by including it in reinforcement learning. They defined regret as the gap between the maximum reward and the current reward. Their regret reinforcement learning algorithm demonstrated behavior comparable to that of addicted gamblers by selecting high-risk, high-reward options. Inverse Reinforcement Learning (IRL) [24] models human behavior by extracting reward functions from observations and optimal behavior [25–30]. However, IRL requires well-defined environment and behavior trajectories, which can be challenging to produce in EGM.

3. Methodology

In this section, we will discuss how we detected playstyles using the clustering technique. Furthermore, we will use one of the playstyle clustered data in our Reinforcement Learning (RL) model to mimic the behavior of players in that cluster. We will also discuss how we identified termination states where the players within this cluster end their session by deciding playing further is not worth it. Finally, the sessions generated by the RL are compared with the selected cluster session data. Figure 1 illustrates the flow of tasks performed for generating player behavior.

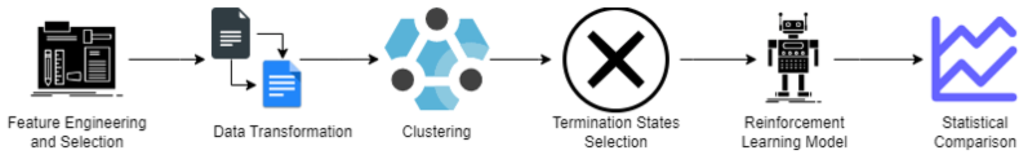
Feature Engineering and Selection
Data Transformation
Clustering
Termination States Selection
Reinforcement Learning Model
Statistical Comparison

Figure 1. Player Behavior Generation Flowchart

3.1. **Dataset Description**

The data that we used is from a less sophisticated EGM game. This game does not have any bonus rounds or secondary game rounds, which makes the implementation of RL simpler. There are around 27,027 sessions in the dataset. Each session includes the player's transactions, containing information such as cash put into or removed from the machine, bets played, and cash won for each bet. EGM data has a fairly limited set of information due to which many characteristics have been extracted from the sessions' initial features. Some of these features are mentioned and explained in Table 1.

Attribute	**Explanation**
Reward	% of wins, where win is greater than wager, # of wins (>wager) divided by number of wagers
Losses Disguised as Wins	% of wins, where win is less than wager, # of wins (<wager) divided by number of wagers
Illusion of Control	Number of times the player changed wager
Bonus Round	Frequency of bonus rounds
Total cash in amount	The total amount of money inserted by the player in the machine during a play session
Average primary wager	The average primary wager in a play session
Session length	Elapsed time is the amount of time that passes from the start of a session to the end of the session.
Total cash out	Total cash out in a session
Starting cash in	Total amount of cash in at the start of the session before he starts playing
Loss Percentage	Total number of loss divided by total number of wagers
Intensity	Intensity (wagers/minute) in a session
Cash out to cash in ratio	The ratio of cash out to cash in
Number of cash in	Number of times a player inserted money in a machine
Payout to Wager (PW) ratio	Total payout divided by total wager in a session

Table 1. Features Explanation

3.2. **Feature Selection and Data Transformation**

For detecting the playstyles of the player, we have used only a few features similar to that were used by Mosquera and Keselj [6] in the clustering algorithms. Table 2 shows the selected features for clustering and their statistics measures. It is clear from Table 2 that the data is skewed to the right showing a non-normal sample distribution.

Features	**Mean**	**Std. Dev.**	**Min**	**1st Qrt.**	**Median**	**3rd Qrt**	**Max**
Win %	0.21	0.04	0.04	0.18	0.21	0.23	0.48
Loss Disguised as Win	0.10	0.03	0	0.08	0.10	0.11	0.44
Illusion of Control %	0.04	0.06	0	0.01	0.03	0.05	0.64
Number of Wagers	259.72	377.32	4	67	145	301	9945
Intensity	0.25	0.09	0.001	0.22	0.25	0.29	1.68

Table 2. Features Statistics Measures

To address the skewness of the data, we used the Box-Cox transformation. The Q-Q plot analysis was used to verify the normal distribution of the data following the transformation. The z-score normalization technique was then used to normalize the transformed data.

3.3. **Clustering**

We used the K-means clustering method with random initialization to identify different types of gambling behavior or playstyles. Clustering algorithms, in general, divide data into k groups or clusters by analyzing cases in a data set; cases that appear similar to others are grouped together [4]. A dissimilarity function is used to define these clusters. There are numerous methods for clustering data, with k-means clustering being one of the most widely used.

The number of clusters is the only major hyper-parameter that needs to be tuned for this algorithm, which can be determined through Elbow method. This algorithm is linear in complexity and scales well to big data. The dataset was clustered by changing the value of k from 5 to 20 to identify a stable and suitable solution for k. The dissimilarity of data objects was calculated in this study based on the distance between pairs of data objects using the Euclidean distance on the normalized dataset. To mimic one of the playstyles, we selected a cluster from among all clusters that represent a group of players with similar playstyles.

3.4. **Identifying Termination States**

We want to find the termination states, where the player decides that continuing the session is not worth it, and train an RL model based on these states. The endpoints are the 25th, 50th, and 75th percentiles because they represent the majority of the distribution of the clusters and show the general playing style of the players within this cluster.

This distance is calculated by averaging 100 players' session features and then calculating the Euclidean distance for each transaction performed by the player with the termination value; i.e., the distance is calculated as the player progresses through the game. We chose win percentage, loss percentage, loss disguised as a win, PW ratio, and the illusion of control (see description in Table 1) termination values in the termination states for training the RL model because they have a smooth decreasing curve, indicating a common playing style among players in this cluster.

3.5. **Reinforcement Learning Algorithms**

In this section, we will briefly describe the Reinforcement Learning algorithms used in this study to mimic real-world player behavior.

3.5.1. **Proximal Policy Optimization**

Model-free policy search techniques, such as policy gradient approaches, are helpful for updating the policy [31], but the problem with policy gradient is finding the right step size for updation, as they are sensitive.

To eliminate this problem, researchers came up with an approach called Trust Region Policy Optimization (TRPO) [32], which applied a trust region restriction to the objective function in order to reduce the KL divergence between the existing and new policies to make sure that the new policies are not too far from the old policies. Theoretically, this can be supported by demonstrating that improving the policy within the trust region results in a guaranteed improvement in monotonic performance. TRPO is computationally inefficient for large-scale tasks, and when applied to sophisticated network architectures, it is challenging to scale up for those situations [33]. By using a clipping technique to avoid totally imposing the hard restriction, Proximal Policy Optimization (PPO) [34] greatly decreases complexity and is able to employ a first-order optimizer, such as the Gradient Descent method, to optimize the objective function which is defined as:

$$L^{CLIP}(\theta) = \widehat{\mathbb{E}}_t[\min(r_t(\theta)\widehat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_t)], \tag{3.1}$$

where θ is policy parameter, $\widehat{\mathbb{E}}_t$ is the expectation over time t, r_t is the probability ratio of old and new policies, $\widehat{A}_t$ is the estimated advantage at time t, and ϵ is a hyperparameter. By attempting to remove the reward for moving the policy away from the previous one when the probability ratio between them is outside of a clipping range, this objective function eliminates the KL constraint of TRPO while maintaining the execution of a Trust Region update.

PPO performs better overall for a broad range of tasks and is relatively easy to implement and tune while preserving the stability and reliability of a TRPO.

3.5.2. Actor Critic using Kronecker-Factored Trust Region

Actor Critic using Kronecker-Factored Trust Region (ACKTR) [35] uses actor-critic methods in which the actor performs an action while the critic estimates the value function, distributed Kronecker factorization [36], and trust region optimization [32]. It creates a scalable approximation of the natural gradient using the Kronecker-Factored Approximated Curvature (K-FAC) [36, 37]. K-FAC uses a Kronecker-factored approximation to the Fisher matrix to perform efficient approximate natural gradient updates. It approximate small block F_l corresponding to layer l as $\widehat{F_l}$ by calculating:

$$F_l \approx \mathbb{E}[aa^T] \otimes \mathbb{E}[\nabla_s L(\nabla_s L)^T] := A \otimes S := \widehat{F_l} \tag{3.2}$$

By assuming that there is no correlation between the second-order statistics of the activations and the backpropagate derivatives, this approximation can be understood. The Fisher metric for RL objectives is defined as:

$$F = \mathbb{E}_{p(\tau)}[\nabla_\theta \log \pi(a_t|s_t)(\nabla_\theta \log \pi(a_t|s_t))^T], \tag{3.3}$$

where $p(\tau)$ is the distribution of trajectories stated as:

$$p(s_0)\prod_{t=0}^{T} \pi(a_t|s_t)p(s_{t+1}|s_t, a_t). \tag{3.4}$$

The Fisher matrix is used to update both the actor and the critic by approximating it by applying K-FAC. ACKTR then applies trust region formulation of K-FAC [38] to update the policy distribution. With both discrete and continuous action spaces, ACKTR is adaptable to learning the model's action probability distribution from an observation. It returns the probability mass for discrete action spaces whereas it returns the probability density for continuous action spaces [39].

4. Experimental Setup

For this study, we used PPO2, which is implemented for GPU by OpenAI, and for multiprocessing, it uses vectorized environments compared to PPO1 which uses MPI [40]. Both models, PPO2 and ACKTR, were trained for around 1 million iterations.

4.1. RL Environment

4.1.1. Action Space

The agent can mainly take two types of actions that are either it can make a wager or cash out. The agent also has to decide the amount of money to wager. As in the real EGM game, the agent can also only bet 2, 4, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, and 100 dollars. If the agent decides to cash out, it will cash out all of the money in the machine because the majority of the players in the original cluster only cashed out once.

4.1.2. State/Observation Space

The agent looks at the observation space in order to take action. Initially, the agent gets some random credit in the machine to start with as we have no cash in action. The agent considers the amount waged and received in the previous transaction, the current credit amount in the machine, the win percentage, the loss percentage, the loss disguised as a win, the payout to the wager, and the illusion of control. The agent uses these features to predict which actions will result in a higher reward and acts accordingly.

4.1.3. **Reward**

The agent's goal is to maximize the total reward of the game episode. If an agent makes an invalid move (for example, wagering more than credit) for each step, it will be penalized by 15. If the Euclidean distance between the termination state and the current state decreases, the agent receives a reward of 1. If the agent cashes out from the machine, he receives 15 as a reward. If the Euclidean distance between the current state and the termination state is less than 2, it receives a reward of 5.

5. **Results and Discussion**

For clustering different playstyles, the optimal value of k was found to be 9 which was then verified using silhouette analysis (Figure 2). Further, we chose cluster number 4 to mimic the playstyle of players in that group, which contains 4483 sessions. This cluster represents intense gamblers, with a mean of 0.26 bets per second, who are unconcerned about losing a lot of money, as evidenced by an 82% loss.

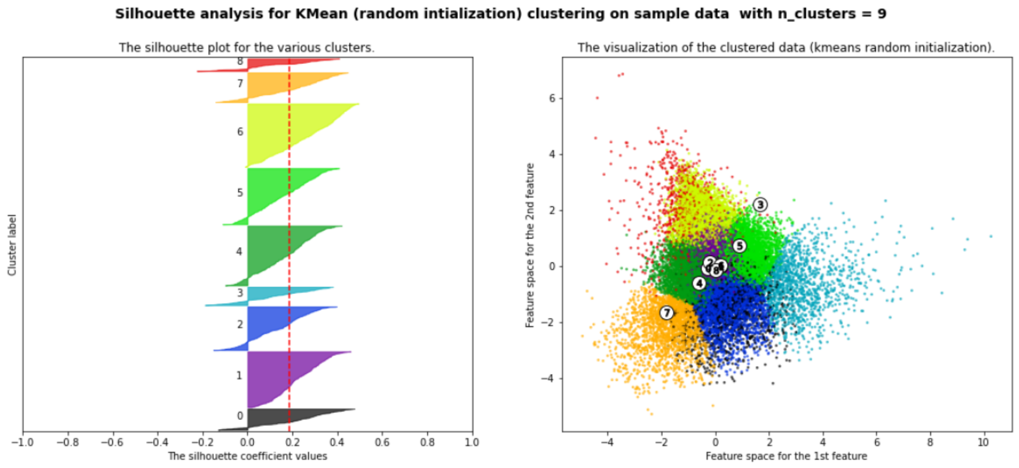

Figure 2. Silhouette Analysis

We generated around 1000 sessions of agents playing the game by both models trained using the termination states of the chosen cluster. These session data were then used to compute the win percentage, loss percentage, loss disguised as a win, and PW ratio (see description in Table 1). To evaluate the model's performance, we compared statistical measures like the minimum, 25th percentile, median, 75th percentile, and maximum values of the features and the real player cluster to see if the model's agent playstyle matched with the real player playstyle of the selected cluster.

Figure 3 shows that both ACKTR and PPO2 generated sessions that are very similar to those of real players. We can see that the 25th percentile, median, and 75th percentile values of both models are nearly identical to those of the original cluster for all the selected features. Both models did not perform well in the max value due to noise in the data. Figure 4 depicts the distribution comparison. We can see that the distribution of sessions generated by both models is similar to that of the original cluster. ACKTR depicted the wager features more accurately, such as the number of unique wagers, average wager, and the illusion of control, because the agent learned to change the wager, whereas the PPO2 agent played the entire session with only one wager value. ACKTR agent was intelligently changing the wager value depending on the losses and the wins to gain maximum reward out of the session. PPO2 was better than ACKTR in playing for a longer time, as PPO2 produced sessions with a number of wagers of around 200, while ACKTR produce sessions with a maximum of 100 numbers of wagers.

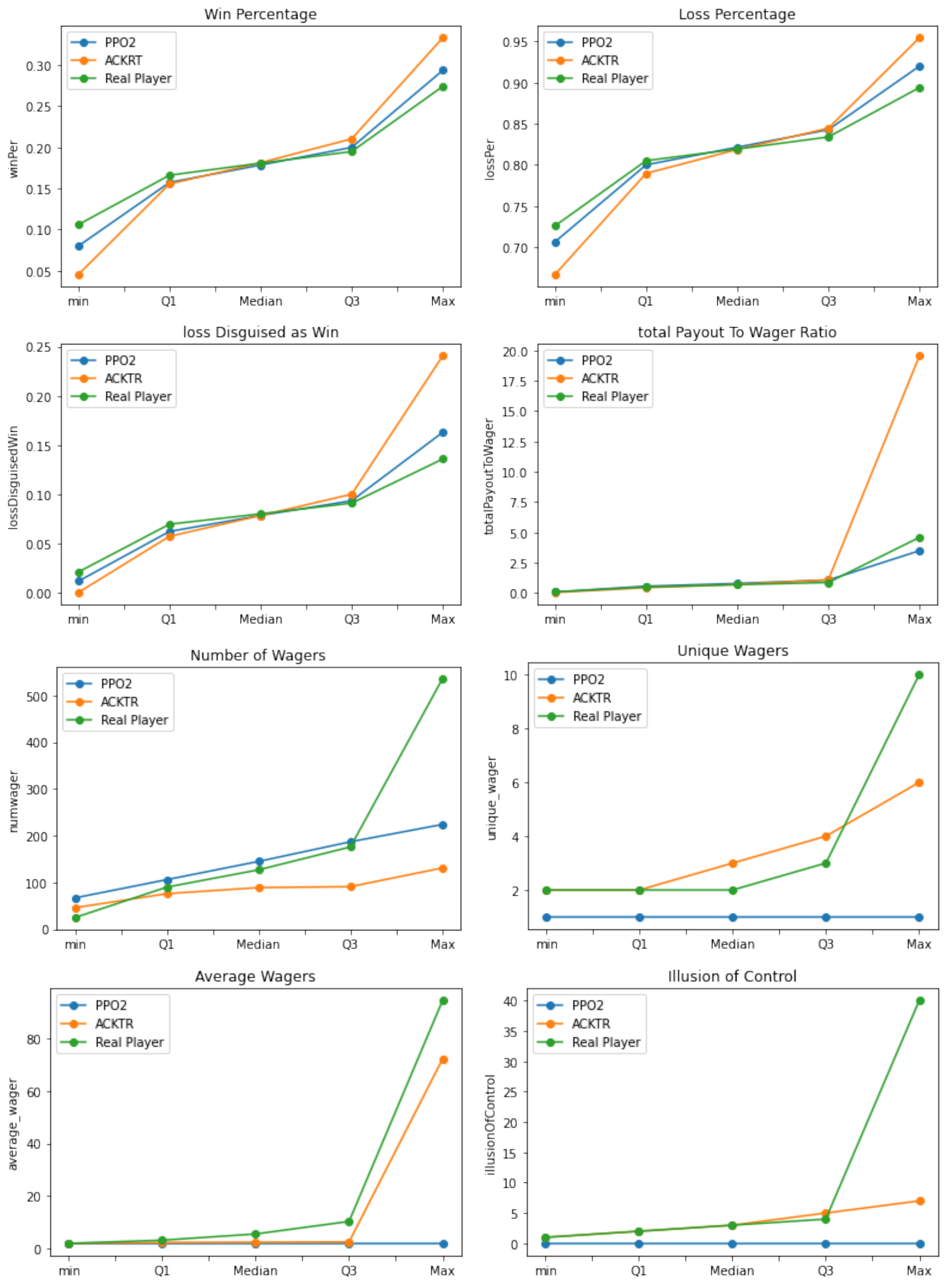

Figure 3. Statistical Measures Comparison

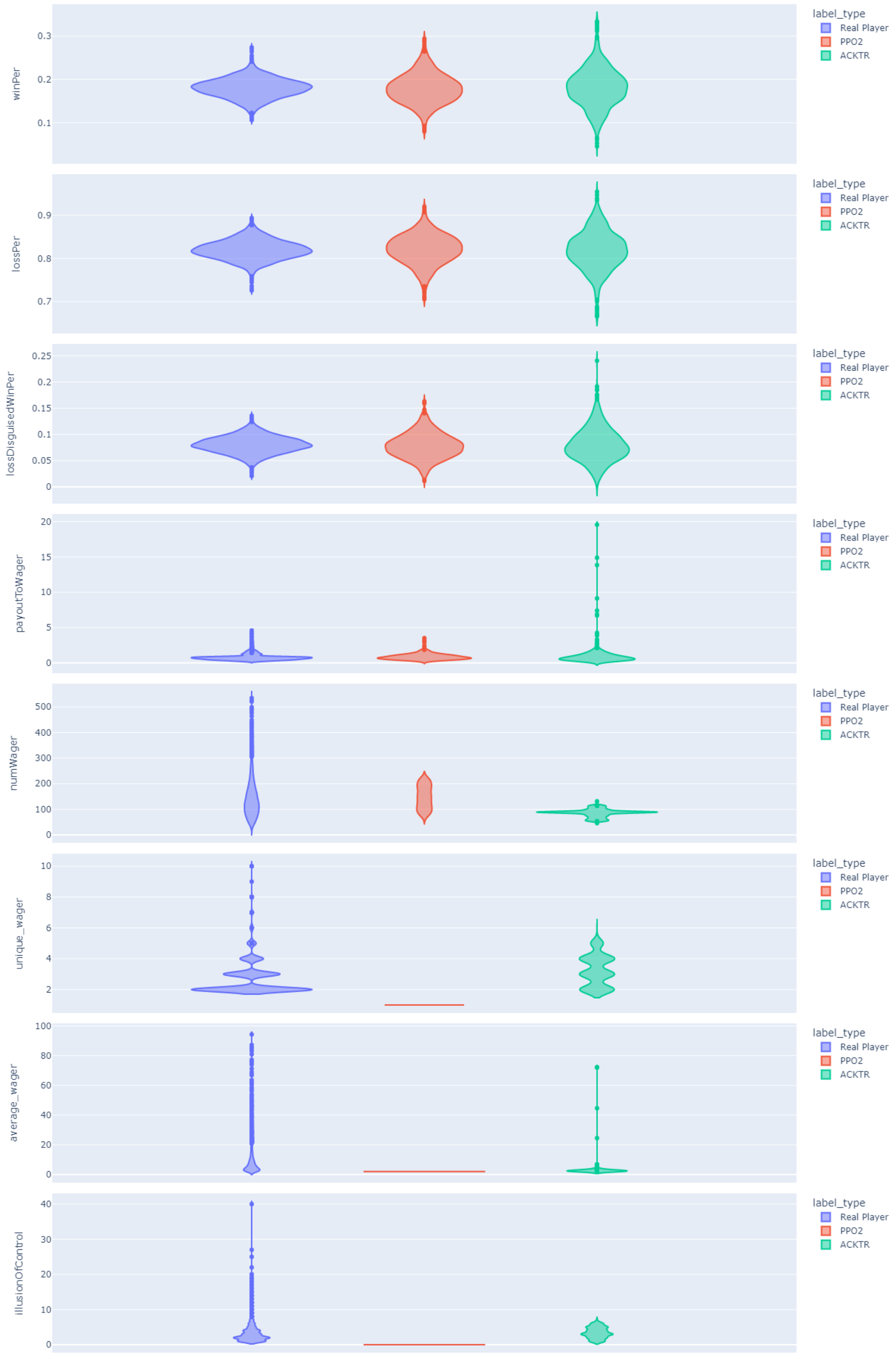

Figure 4. Features Distribution Comparison

6. Conclusion

This study mimics player behavior to obtain a good evaluation of a reinforcement learning model as a feasible substitute for real-world players. K-means gave an excellent separation of the player behaviors. We selected the cluster with intense gamblers and their behavioral attributes used to define the termination states for the RL model. The reward for the agents was calculated based on the Euclidean distance between the current state and the termination states. Both the models, PPO2 and ACKTR, succeeded in producing behavior similar to that of the selected cluster behavior. Though PPO2 was able to produce sessions with longer duration or more number of wagers, ACKTR slightly outperformed PPO2 as the ACKTR agent was intelligently able to change wager values during the session, which is an important attribute to say that the agent mimicked the player behavior.

6.1. Limitations

The RL algorithms employed to mimic player behavior in this study were only used for one playstyle and game, hence they may not be generalizable to other playstyles and games. Since the RL models have not been tested in production, they may be subject to unforeseen limitations.

6.2. Future Work

For future work, a different clustering algorithm could be used to have a better distinction of the behaviors of the player. We can include more features in the termination states, this might make the agent behave more like real-player as it will have more behavioral attributes to think about. It will be interesting to see how the agent will behave by tweaking the reward function from using the Euclidean distance to some other distance. Currently, the model does not have a cash in action as it starts with some random number credit in the machine, this action could be added so that it is performed by the agent, and also the agent could be modeled so it can perform some intermediate cash in and cash out based on the net loss of the session. Finally, it will be interesting to change this EGM environment to simulate it as a video game such as Atari games.

References

[1] S. Dragicevic, G. Tsogas, and A. Kudic. "Analysis of casino online gambling data in relation to behavioural risk markers for high-risk gambling and player protection". In: *International Gambling Studies* 11.3 (2011), pp. 377–391. DOI: 10.1080/14459795.2011.629204. URL: https://doi.org/10.1080/14459795.2011.629204.

[2] M. Rotermann and H. Gilmour. *Who gambles and who experiences gambling problems in Canada*. 2022. URL: https://www150.statcan.gc.ca/n1/pub/75-006-x/2022001/article/00006-eng.htm.

[3] V. V. MacLaren, J. A. Fugelsang, K. A. Harrigan, and M. J. Dixon. "The personality of pathological gamblers: A meta-analysis". In: *Clinical Psychology Review* 31.6 (2011), pp. 1057–1067. ISSN: 0272-7358. DOI: https://doi.org/10.1016/j.cpr.2011.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0272735811000274.

[4] H. J. Shaffer and D. A. Korn. "Gambling and Related Mental Disorders: A Public Health Analysis". In: *Annual Review of Public Health* 23.1 (2002). PMID: 11910060, pp. 171–212. DOI: 10.1146/annurev.publhealth.23.100901.140532. URL: https://doi.org/10.1146/annurev.publhealth.23.100901.140532.

[5] N. Adami, S. Benini, A. Boschetti, L. Canini, F. Maione, and M. Temporin. "Markers of unsustainable gambling for early detection of at-risk online gamblers". In: *International Gambling Studies* 13.2 (2013), pp. 188–204. DOI: 10.1080/14459795.2012.754919. URL: https://doi.org/10.1080/14459795.2012.754919.

[6] M. G. Mosquera and V. Keselj. “Identifying electronic gaming machine gambling personae through unsupervised session classification”. In: *Big Data & Information Analytics* 2.2 (2017), pp. 141–175.

[7] W. Ni. “Application of Clustering, Logistic Regression and Decision Tree Induction on EGM Data for Detection and Prediction of At-Risk and Problem Gamblers”. MA thesis. Dalhousie University, 2014.

[8] J. Braverman and H. J. Shaffer. “How do gamblers start gambling: identifying behavioural markers for high-risk internet gambling”. In: *European Journal of Public Health* 22.2 (Jan. 2010), pp. 273–278. ISSN: 1101-1262. DOI: 10.1093/eurpub/ckp232. eprint: https://academic.oup.com/eurpub/article-pdf/22/2/273/1366659/ckp232.pdf. URL: https://doi.org/10.1093/eurpub/ckp232.

[9] S. Latifi. “Electronic gaming machine playstyle detection and rapid playstyle classification using multivariate convolutional LSTM neural network architecture”. MA thesis. Dalhousie University, 2021.

[10] C. Livingstone. *How electronic gambling machines work*. English. Published online on 11/7/17 by AGRC/AIFS. Australian Gambling Research Centre - AIFS, July 2017. URL: \url{https://aifs.gov.au/sites/default/files/publication-documents/1706_argc_dp8_how_electronic_gambling_machines_work_0.pdf}.

[11] T. Schellinck and T. Schrans. “Intelligent design: How to model gambler risk assessment by using loyalty tracking data”. In: *Journal of Gambling Issues* 26 (Jan. 2011), pp. 51–68.

[12] V. Keselj. *Use of data mining on electronic gaming machine session variables for responsible gaming support*. (internal report). Dalhousie University, Faculty of Computer Science, 2011.

[13] National Research Council (US) Committee on the Social and Economic Impact of Pathological Gambling. *Pathological Gambling: A Critical Review*. Washington (DC): National Academies Press (US), 1999.

[14] G. Banks, R. Fitzgerald, and L. Sylvan. *Gambling: Productivity Commission Inquiry Report*. 2010.

[15] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. “Deep Reinforcement Learning: A Brief Survey”. In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 26–38. DOI: 10.1109/msp.2017.2743240. URL: https://doi.org/10.1109%2Fmsp.2017.2743240.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel. “End-to-end training of deep visuomotor policies”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373.

[17] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection”. In: *The International journal of robotics research* 37.4-5 (2018), pp. 421–436.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. *Playing Atari with Deep Reinforcement Learning*. 2013. DOI: 10.48550/ARXIV.1312.5602. URL: https://arxiv.org/abs/1312.5602.

[19] H. van Hasselt, A. Guez, and D. Silver. “Deep Reinforcement Learning with Double Q-Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 30.1 (2016). DOI: 10.1609/aaai.v30i1.10295. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10295.

[20] OpenAI et al. *Dota 2 with Large Scale Deep Reinforcement Learning*. 2019. DOI: 10.48550/ARXIV.1912.06680. URL: https://arxiv.org/abs/1912.06680.

[21] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. “Grandmaster level in StarCraft II using multi-agent reinforcement learning”. In: *Nature* 575.7782 (2019), pp. 350–354.

[22] W. Schultz, P. Dayan, and P. R. Montague. “A Neural Substrate of Prediction and Reward”. In: *Science* 275 (1997), pp. 1593 –1599.

[23] Y. Wu and J. Izawa. “The Regret Motivated Reinforcement Learning”. In: *2021 International Symposium on Micro-NanoMehatronics and Human Science (MHS)*. Nagoya, Japan: IEEE Press, 2021, 1–5. DOI: 10.1109/MHS53471.2021.9767180. URL: https://doi.org/10.1109/MHS53471.2021.9767180.

[24] A. Y. Ng, S. Russell, et al. “Algorithms for inverse reinforcement learning.” In: *Icml*. Vol. 1. 2000, p. 2.

[25] C. L. Baker, R. Saxe, and J. B. Tenenbaum. "Action understanding as inverse planning". In: *Cognition* 113.3 (2009). Reinforcement learning and higher cognition, pp. 329–349. ISSN: 0010-0277. DOI: https://doi.org/10.1016/j.cognition.2009.07.005. URL: https://www.sciencedirect.com/science/article/pii/S0010027709001607.

[26] T. D. Ullman, C. L. Baker, O. Macindoe, O. Evans, N. D. Goodman, and J. B. Tenenbaum. "Help or Hinder: Bayesian Models of Social Goal Inference". In: *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. NIPS'09. Vancouver, British Columbia, Canada: Curran Associates Inc., 2009, 1874–1882. ISBN: 9781615679119.

[27] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard. "Socially compliant mobile robot navigation via inverse reinforcement learning". In: *The International Journal of Robotics Research* 35.11 (2016), pp. 1289–1307. DOI: 10.1177/0278364915619772. eprint: https://doi.org/10.1177/0278364915619772. URL: https://doi.org/10.1177/0278364915619772.

[28] M. Kuderer, S. Gulati, and W. Burgard. "Learning driving styles for autonomous vehicles from demonstration". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 2641–2646. DOI: 10.1109/ICRA.2015.7139555.

[29] G. Neu and C. Szepesvári. "Apprenticeship Learning Using Inverse Reinforcement Learning and Gradient Methods". In: *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*. UAI'07. Vancouver, BC, Canada: AUAI Press, 2007, 295–302. ISBN: 0974903930.

[30] B. Kim and J. Pineau. "Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning". In: *International Journal of Social Robotics* 8 (2016), pp. 51–66.

[31] J. Peters and S. Schaal. "Reinforcement learning of motor skills with policy gradients". In: *Neural Networks* 21.4 (2008). Robotics and Neuroscience, pp. 682–697. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2008.02.003. URL: https://www.sciencedirect.com/science/article/pii/S0893608008000701.

[32] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. "Trust region policy optimization". In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897.

[33] Y. Wang, H. He, and X. Tan. "Truly Proximal Policy Optimization". In: *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. Ed. by R. P. Adams and V. Gogate. Vol. 115. Proceedings of Machine Learning Research. PMLR, 2020, pp. 113–122. URL: https://proceedings.mlr.press/v115/wang20b.html.

[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint* (2017). DOI: 10.48550/ARXIV.1707.06347. URL: https://arxiv.org/abs/1707.06347.

[35] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba. "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation". In: *Advances in neural information processing systems* 30 (2017).

[36] J. Martens and R. Grosse. "Optimizing neural networks with kronecker-factored approximate curvature". In: *International conference on machine learning*. PMLR. 2015, pp. 2408–2417.

[37] R. Grosse and J. Martens. "A kronecker-factored approximate fisher matrix for convolution layers". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 573–582. DOI: 10.48550/ARXIV.1602.01407.

[38] J. Ba, R. B. Grosse, and J. Martens. "Distributed Second-Order Optimization using Kronecker-Factored Approximations". In: *ICLR*. 2017.

[39] Y. Chu, Z. Wei, G. Sun, H. Zang, S. Chen, and Y. Zhou. "Optimal home energy management strategy: A reinforcement learning method with actor-critic using Kronecker-factored trust region". In: *Electric Power Systems Research* 212 (2022), p. 108617. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2022.108617. URL: https://www.sciencedirect.com/science/article/pii/S0378779622006915.

[40] *PPO2*. https://stable-baselines.readthedocs.io/en/master/modules/ppo2.html. Accessed: 2022-11-06.