Extended Abstract

In this project, we set out to apply deep reinforcement learning methods to the task of placing effective money-line bets on National Basketball Association (NBA) regular season games. Americans bet tens of billions of dollars on sports contests every year, and the industry is only expanding as more states legalize betting from your phone. Sports outcome prediction, and the evaluation of sports betting opportunities, is thus an area of significant import.

But while machine learning methods have long been employed in different areas of sports prediction, sports prediction problems are often posed as a supervised learning endeavor where the goal is predicting the outcome of a game or match, and thus reinforcement learning is rarely employed, and the public catalog of reinforcement learning research on NBA data is virtually non-existent.

Reinforcement learning has been applied, with great success, in the areas of finance and cryptocurrency trading. Why then cannot a researcher consider each sports contest to be a potential investment opportunity, like a stock, which has a true value that a reinforcement learning model can try to gauge so as to determine whether the bet is worth buying?

In this research we seek to establish such a framework for reinforcement learning, specifically for NBA regular season money line betting. In particular, we employ Deep Q-Learning (DQN) to test the viability of this novel research area, seeking to build a model that can identify high-value bets amongs NBA regular season money lines.

We train three versions of our DQN model, finding moderate success with two and significant success with the third. Features are constructed to capture long- and short-term measures of team quality, rest and travel advantages and disadvantages, player availability, betting line reflection of true odds, and more. Using our hand-picked features, and after casting out a handful that prove ineffectual, we train our model on NBA game outcome and betting line data for each regular season from 2007-2008 through 2018-19, holding in reserve the 2019-20 and 2020-21 regular seasons as test sets. Our best model finds significant success, achieving return on investment of over 30% in a single pass-through of the two held-out seasons (simulating a deployment of the model over those two seasons).

The success this research finds in applying deep reinforcement learning to NBA regular season money line betting holds significant implications. With such limited research in this area, we are honored to lay the groundwork for future researchers in this domain via this demonstration of the applicability of DQN to NBA betting. Future researchers should expand upon this work, expanding model capabilities to allow variable sized betting, explore more complex architectures and feature embeddings, investigate the applicability of other reinforcement learning models in this area, and tackle other forms of NBA betting such as spread or prop bets.

Reinforcement Learning for Pick(ing) and (Bank)roll: Applying Deep Q-Learning to NBA Regular Season Money Line Betting

Charles Shaviro
Department of Statistics
Stanford University
cshaviro@stanford.edu

Abstract

In this project we use deep reinforcement learning, specifically deep Q-learning, for NBA money line betting, seeking to demonstrate the applicability of these reinforcement learning paradigms to NBA bet evaluation. Designing a feature set that seeks to capture team quality, fatigue, recent travel, and betting line deviation from 'true' odds of each team winning, and merging it with 12 seasons of game outcome and betting line data, we train a DQN for this task, and find significant success with our best model, generating return on investment in excess of 30% on a single pass-through of two held-out test set regular seasons. Our findings pave the way for future researchers to continue to investigate the applicability of reinforcement learning to NBA bet evaluation and bankroll management, and future work would do well to experiment further with variable bet sizing, more granular player availability data, and other reinforcement learning frameworks.

1 Introduction

In this project we set out to train a reinforcement learning agent to make effective money-line bets on National Basketball Association (NBA) regular season games. Sports betting in the U.S. was a $14 billion industry in 2024, more than 20% larger than in 2023 [Greenberg (2025)]. Furthermore, the industry is expected to continue to grow at a blistering rate as more and more states legalize online sports betting and apps such as Draftkings and FanDuel—which allow bettors to conveniently place wagers from their smartphones with minimal required activation energy or barriers to entry—become more and more ubiquitous. The industry's skyrocketing rise means that applying modern learning methods such as those found in deep reinforcement learning to problems of game prediction, bet evaluation, and bankroll management are more relevant than ever.

Sports betting is a fascinating decision-making problem. Bookmakers set odds extremely carefully to ensure their profitability, balancing not just their best estimates of the true probability distribution of expected outcomes, but also concerns such as attempting to 'split the market'—a practice of setting betting lines so that similar amounts of bettors place wagers on each side of a matchup to reduce the likelihood of significant downside if things break poorly for the sportsbook [Curtis (2021)]—and making betting options attractive to potential customers. For these and other reasons, betting lines are not always perfectly efficient, and bookmakers are known to try to 'split the market', to . This results in the existence of betting lines that may systematically differ from the true probability of victory for each team, which smart decision-makers can exploit.

While machine learning methods have long been employed in different areas of sports prediction, with particular success for random forest, XGBoost, kernel methods, and others, sports prediction problems are often posed as a supervised learning endeavor where the goal is predicting the outcome of a game or match, and thus reinforcement learning is rarely employed, and has not been used it all for NBA basketball betting tasks, at least not in any public research (privately, the sportsbooks may be making significant progress in the field).

In this work, we experiment with deep Q-Learning to test the viability of this novel research area—deep reinforcement learning for NBA regular season money line betting—seeking to build a model that can effectively identify high-value bets, and to create a building block upon which future work at the intersection of NBA betting and reinforcement learning, a largely unexplored domain, can stand.

2 Related Work

A great number of studies have applied machine learning to predict sports outcomes and evaluate sports betting opportunities. In 2024, Alan Ji analyzes the effectiveness of a variety of machine learning techniques, with a focus on neural networks, for the task of predicting game outcomes, finding reasonable success [Ji (2024)]. In a CS 229 project, Bucquet et al. employ a baseline of random forest and then a fully connected neural networks as well as a long short term memory network for the task of predicting NBA game outcomes [Bucquet and Sarukkai (2018)].

More recent work uses a variety of modern learning architectures. In 2023, Zhao et al transform data from the 2012 through 2018 NBA seasons into a homogeneous undirected team graph and feed it into a graph convolutional network, reaching a baseline game-outcome prediction accuracy of 66.9%. Further, they are able to fuse random forest–based feature extraction with the GCN embeddings to attain accuracy of 71.54%, outperforming many prior methods [Zhao et al. (2023)].

While the literature on predicting sports outcomes is significant, there is comparably less research into sports betting, and specifically NBA sports betting, or bankroll management. In 2024, Devena trains and tests how reinforcement learning algorithms such as DQN, PPO, and A2C can be applied to tennis betting markets [Vena (2023)]. In 2021, Hirvi designs a simulation framework in which two 'house' or 'sportsbook' agents (one static and one tuned via reinforcement learning) provide betting odds and multiple 'gambler' agents interact with those provided odds, showing that RL-tuned odds can actually significantly exted and intensify a player's engagement [Hirvi (2021)].

In a sprawling review, Galekwa et al. examine the machine learning research into sports betting conducted over the last 15 years. Looking at 219 papers across 10 sports, and examining methods from logistic regression and random forest to convolutional neural networks and graph neural networks, to ensemble hybrids of different approaches, the authors articulate the significant effectiveness of applying machine learning techniques to sports betting. Across the sports, tree-based ensembles such as random forest, XGBoost, and LightGBM, or kernel methods such as support vector machines are some of the most common approaches, but neural architectures such as long short-term memory memory networks and graph neural networks are increasingly used, and show high performance and great promise, particularly as it comes to relational data such as networks of players. There are significant challenges with sports data, however, as data quality and availability and the dynamic construction of teams do not mirror some other more accessible and stable fields [Galekwa et al. (2024)].

Although not within the domain of sports betting, crypto and stock trading has frequently taken advantage of reinforcement learning in recent years. Jiang et al. use a model-free deep RL framework to learn portfolio allocations, managing to produce four-fold returns [Jiang et al. (2017)]. In 2016, Deng et al. construct an end-to-end 'deep direct reinforcement learning' model that is able to learn representations of real-time financial signals and respond with trading actions [Deng et al. (2017).

In this paper, we hope that by framing potential NBA bets as investments of a sort, we may be able to re-create some of the same success found at the intersection of RL and finance in our area of interest: RL and NBA money line sports betting .

3 Method

We design a custom betting environment that our RL agent is able to step through as it trains a deep Q-learning network to powers its inference.

3.1 Deep Q–Learning (DQN)

Q–learning seeks a state–action value function

$$Q^\star(s,a) = \max_\pi \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^{\,t} r_t \mid s_0 = s,\ a_0 = a,\ \pi\big] \tag{1}$$

that satisfies the Bellman optimality equation:

$$Q^\star(s,a) = \mathbb{E}_{s'}\big[\, r + \gamma\, \max_{a'} Q^\star(s',a') \mid s,a \,\big]. \tag{2}$$

In *Deep Q–Learning*, the value function is approximated by a neural network $Q_\theta(s,a)$ with parameters θ. Training uses two stabilising tricks:

1. **Experience replay** $\mathcal{D}$: store past transitions $(s,a,r,s',\texttt{done})$ and sample i.i.d. mini-batches.
2. **Target network** $Q_{\bar\theta}$: a lagged copy of Q_θ updated every K steps.

For a batch $\mathcal{B}$ the temporal-difference (TD) target is

$$y_i = r_i + \gamma\,(1 - \texttt{done}_i)\ \max_{a'} Q_{\bar\theta}(s_i', a'), \tag{3}$$

and the network minimises the Huber loss

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \mathrm{Huber}\big(y_i - Q_\theta(s_i,a_i)\big). \tag{4}$$

Parameters are updated via stochastic gradient descent

$$\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}(\theta),$$

while actions during training follow an ε-greedy policy

$$a_t = \begin{cases} \text{uniform}(A) & \text{with prob. } \varepsilon_t, \\ \arg\max_a Q_\theta(s_t,a) & \text{otherwise,} \end{cases} \qquad \varepsilon_t = \varepsilon_{\min} + (\varepsilon_0 - \varepsilon_{\min}) e^{-t/\tau}.$$

At inference time $\varepsilon = 0$, so the agent selects the action with the largest estimated Q-value.

4 Experimental Setup

4.1 Our Setting

In our setting, NBA money line betting, we have the following:

- *state* s: the current bankroll, current matchup, and a list of features describing the matchup that cover ELO ratings, fatigue, recent form, betting-odds-implied probability of victory for each team, and more.
- *action* a. We have three discrete actions { BET-HOME, SKIP, BET-AWAY}. The agent can only bet a fixed-bet size on the money line odds.
- *reward* r: the immediate reward is the profit (as determined by the line odds) or loss ($1) realized on the game depending on if the bet is correct. If the agent chooses not to bet, *reward* is 0.
- γ: discount factor, here usually set to 0.99 or 1, measures how much the agent cares about end-of-season bankroll vs. single-game swings.

- π : our policy that maps states to actions.
- $Q_\theta(s, a)$: neural approximation of the optimal action–value function; θ are the network weights we train.
- $Q_{\bar{\theta}}$: the *target network*, a periodically frozen copy of Q_θ that stabilises bootstrap targets.
- $\mathcal{D}$: replay buffer that stores the last N transitions.
- η : learning-rate used by Adam optimiser.

In our environment, a custom gym-style betting simulator that we design, a step is one NBA game. The NBA sees a feature vector describing the game, chooses from its possible actions, receives profit/loss that define its reward and an update to its bankroll, and the advances to the next game.

We defined an episode as one full regular-season (about 1000 steps). An episode terminates when the final game in a season is processed or when the bankrolls hits $0. Each final model is trained for 3,000 episodes.

Our agent starts each episode with a bankroll of $500, and wagers $1 in the bets it chooses. This means that the negative reward it can receive on a step is always —1, but the positive reward it receives can scale from a very high number (e.g. 10 if it bets on a huge underdog at +1000 odds) or a number very close to 0 (e.g. 0.2 if it bets on a huge overdog at -500 odds).

4.2 Data

We utilize data from a variety of sources for this project. Data is downloaded from Kaggle and FiveThirtyEight's github, scraped from Basketball Reference, and pulled from the NBA API.

- From Kaggle: a dataset of historical money-line betting and game outcomes, for every NBA regular-season game from the 2007-08 through the 2021-22 seasons, almost 20,000 games.
- From Basketball Reference: data on advanced player stats for the 2007-08 through 2021-22 regular seasons.
- From NBA.com via the nba underscore api: play-by-play data covering the last 33 seasons of NBA regular season basketball.
- From Kaggle: latitude/longitude location data for all 30 NBA teams' stadium locations.
- From FiveThirtyEight: RAPTOR ratings for all NBA players

Throughout, we use regular seasons 2007-08 through 2018-19 as our training, and keep seasons 2019-20 and 2020-21 in reserve as our held out test set for final evaluation of our model. The NBA playoffs (as well as the pre-season) are considered to be very different from the regular season, and are not used in this research.

4.3 Feature Engineering

One of the biggest restrictions as it comes to applying any kind of reinforcement or machine learning to sports betting is the availability and quality of data. A 48-minute basketball contest, with usually eight to 10 players from each team seeing the floor in different combinations of five on five, is a very complex game. When we further consider differences in preparation, mentality, rest, and other psychological factors, it's easy to see why 'upsets' happen so often: there is a great deal of randomness and a host of unknown factors involved. No one number metric (or even two or three number vectors) can perfectly capture the quality of a team on a given night.

We experiment with a variety of features in this work, though a great deal of future work still remains to be done.

We regarded it as imperative that our agent be able to consider the following factors for each team, and set to engineering features that describe them:

- a baseline of team quality
- a more sensitive shorter-timeline measure of team quality

- a proxy for fatigue
- the betting-line-implied team strength / likelihood of victory
- game-level player availability

4.4 Core Features

A host of metrics exist for evaluating team strength, player strength, and specific qualitative aspects of different teams. Sensitive to the possibility of noise being baked into different metrics, we settled upon the following core features for our model:

1. Home team ELO rating
2. Away team ELO rating
3. Home team win % over the last 5 games
4. Away team win % over the last 5 games
5. Home team point-differential over the last 5 games
6. Away team point-differential over the last 5 games
7. Home team days since last game
8. Away team days since last game
9. Home team on a back-to-back (played a game previous night)?
10. Away team on a back-to-back
11. Home team miles traveled last 72 hours
12. Away team miles traveled last 72 hours
13. Betting-line-implied probability of home team victory
14. Betting-line-implied probability of away team victory
15. Elo - betting market residual

Most of the metrics above are straight forward. For Elo ratings, we provide a refresher:

Elo ratings measure relative team strength on an additive (logit–scaled) point scale. Before a game the *expected score* of the home team is

$$\hat{S}_{\text{home}} \;=\; \frac{1}{1 + 10^{\left(E_{\text{away}} - E_{\text{home}}\right)/400}}, \tag{5}$$

where E_{home} and E_{away} are the current Elo ratings of the two teams (400 pts $\approx 10\times$ win–odds). After the game each team's rating is updated by

$$E_{\text{new}} \;=\; E_{\text{old}} + K\big(S - \hat{S}\big), \tag{6}$$

with $S \in \{1, 0\}$ the realised score ($1 =$ win, $0 =$ loss) and K the *K-factor* that controls update speed (K is set to 20 in our design). After computing these Elo ratings, E_{home} and E_{away}, they act as a slow-moving priors on team ability.

Point differential and win % provide a faster-moving measure of team quality.

Rest is measured with three features for this team, with miles traveled being computed using teams' stadium locations (for the purposes of this work 'last 72 hours' really means over the last 3 days; we make the simplifying assumption that each game occurs at the same time each day).

Finally, the Elo - betting market residual is the difference between the probability of home team victory implied by the teams' Elo ratings and the probability of home team victory implied by the teams' betting lines.

4.5 Supplementary Features

There are a great deal of one-number metrics that describe player quality (e.g. Player Efficiency Rating, Box Plus Minus, Win Shares / 48 Minutes), as well as team quality (e.g. Elo, Win %, Net Rating). After experimenting with a variety of them, weighting by team quality, minutes played, and adjusting with other factors still, we found limited further success, as our core features were easy for the model to learn from and well-captured a lot of the information important in determining the qualty of a betting opportunity.

We settle upon the following as supplementary features beyond the core features for our model:

- Z-score of the home team's minutes-weighted RAPTOR for the players playing in tonight's game in the context of the entire historical dataset
- Z-score of the away team's minutes-weighted RAPTOR for the players playing in tonight's game in the context of the entire historical dataset
- Percentage of the home team's top-3 players' combined RAPTOR score playing in tonight's game
- Percentage of the away team's top-3 players' combined RAPTOR score playing in tonight's game

The V1 version of our model includes the core features as well as the first two supplementary features (core features + RAPTOR), and the V2 version includes the core features along with all four supplementary featurs (core features + RAPTOR + star share).

4.6 Hyperparameter Tuning

For hyperparameter tuning we utilize Optuna, an open-source hyperparameter tuning optimization library. Optuna takes as input a search space (i.e. a numerical range) for each hyperparameter and samples different combinations, training our DQN for a short run with that combination before storing the result for us. Using a mix of random and Bayesian sampling alongside early pruning if a combination is performing significantly worse than the running median, Optuna is able to automatically and efficiently search for effective hyperparameters. We use it to tune our learning rate η, our discount factor γ, the ϵ decay rate, the final ϵ value, our memory size, and the target update interval size.

5 Results

Our core model finds significant success, turning a \$500 initial bankroll into \$673.41 (+34.7% return-on-investment) in a single passthrough on the unseen 20219-20 and 2020-21 NBA regular seasons (about 2,150 games). The core betting agent achieves a hit rate of 65.2% (i.e., winning 65.2% of bets), saw monhtly Sharpe ratio (a measure of 1.32, and experienced a maximum draw-down of just 1.8%.

The more extended models we employ are not quite as successful. V2 (core features + RAPTOR) achieves , while V3 (core features + RAPTOR + star share).

It is worth considering that any success our modeling schema find are particularly impressive due to the presence of an increased amount of variability and uncertainty, never before seen or replicated since, in the NBA regular seasons that encompass our test data, due to the Covid-19 pandemic. In the second half of the 2019-20 NBA regular season, the league took a never-before-seen mid-season halt due to the pandemic, before finishing a shortened season in the famous 2020 NBA Bubble months later where players were quarantined in a hotel in Disneyworld. The 2020-21 regular season started much later than normal, saw fewer games on a more aggressive schedule of limited rest, and involved new and significant protocols to limit disease spread.

We can see a comparison of our different models' results in Table 1.

Table 1: Performance Comparison

Method	Final Bankroll	ROI	Hit Rate (% Correct)
Random Agent Baseline	\$477.66	-4.47%	50.5%
Core Features Model	\$642.90	28.6%	65.2%
V1: Core Features + RAPTOR	\$513.50	2.70%	56.0%
V2: Core Features + RAPTOR + Star Share	\$508.07	1.61%	54.5%

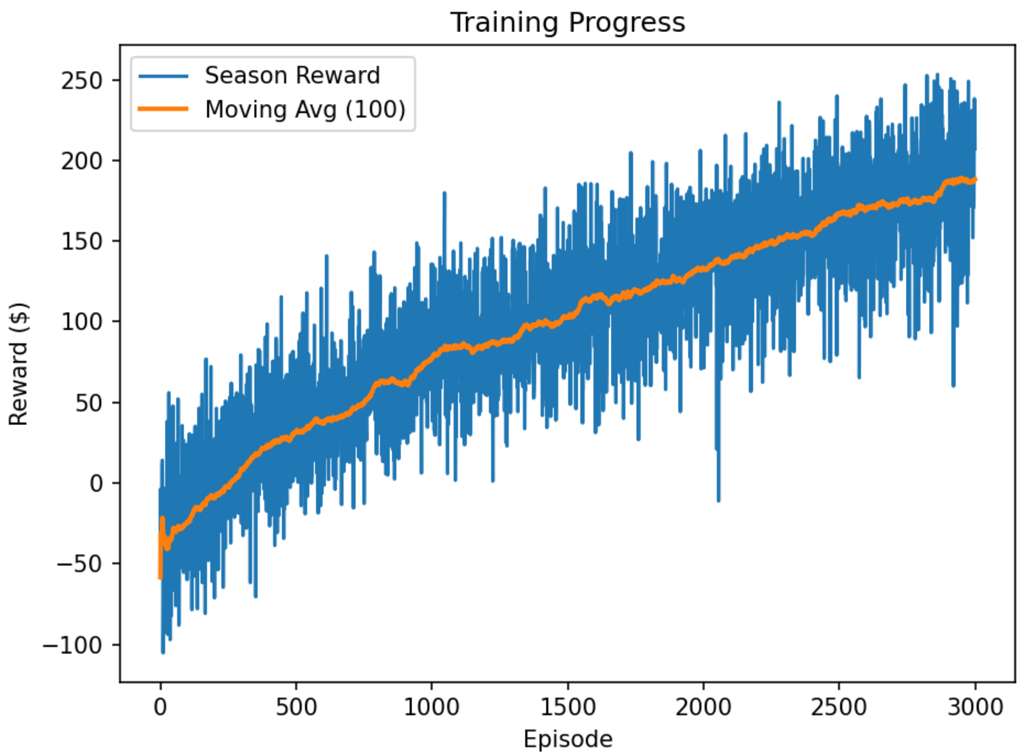

Figure 1: Training curve for ourcore features model. The blue line shows raw season reward per episode; the orange line is the 100-episode moving average. Performance continues to grow.

Finally, we present a training curves for our core features model, demonstrating the degree to which the agent is able to learn over more and more training episodes and improve its average reward (last 100 episodes running average). Each agent manages to steadily learn and improve over additional episodes.

6 Discussion

Although we can only assume that the major sportsbooks have compiled a significant amount of proprietary research on deep reinforcement learning for NBA regular season money line betting, it is still exciting to achieve this novel result not seen in the publicly available literature. Clearly, deep-Q learning can be well-applied to NBA betting, as our top model is able to achieve a significant ROI of +34.7% despite a quite conservative betting setup in which each bet size is only 0.2pct of initial bankroll (\$1 bet sizes from \$500 bankroll), and held-out testing seasons of data contain just over 2,000 games, or betting opportunities, and are made distinct from the training data due to the effects of the pandemic. To achieve a high ROI despite these confounding issues and conservative setup, RL must be well-suited to this task indeed.

Our best agent is able to find significant success by identifying discrepancies between the implied odds of different game outcomes and the ’true’ probability, once weighty factors such as cumulative fatigue are factored in. The Elo market residual feature in particular is well-relied on by the agent, and receives significant weight in the model.

It seems likely too from the training curves for our models that more improvement is possible if we train for longer, in a setting unconstrained by limited GPU resources and time.

Further work on this project is still underway and will continue beyond the scope of this course, as we seek to further train the models, enable the agent to place variable sized bets on the home or away team, and investigate the effectiveness of REINFORCE modeling or other RL frameworks for this bet evaluation task.

7 Conclusion

In this paper, we have made clear the viability of reinforcement learning application to the domain of NBA regular season money-line betting. Despite the constraints on this project, the significant positive ROI achieved by the trained models is a bullish sign for future research in the domain of deep reinforcement learning and NBA betting.

Future research in this area should seek to build upon this result and expand in a number of directions. First, further researchers might seek to explore a dynamic bet-sizing approach, ideally allowing the agent to pick size from a continuous scale, something that has been well achieved by reinforcement learning models for playing poker. Further, a more sophisticated embedding of players, trained upon play-by-play data and other advanced stats, could more ably catch discrepancies between different games such as the absence of important players, the occurrence of a recent trade, or other rapid or sharper faster changes in roster quality. Future work could also tweak our proposed reward framework to accept greater or lesser risk or focus upon high-upside low-conversion bets, explore dueling DQN, PPO, or any other of the variety of deep reinforcement learning frameworks that may bear fruit in this space. The applications of reinforcement learning to NBA prediction have only just begun, and we are excited to see what future work can accomplish.

8 Team Contributions

As the only group member, I did all the tasks.

Changes from Proposal No huge changes, scope of project was decreased as difficulty of 1-person team became clearer amid bandwidth constraints and sprawling feature engineering investigation taking significant time. Rather than evaluating several models or finishing implementing variable bet sizing, I stuck more to feature engineering and trying to hone this version of the model to be as effective as possible.

References

Alexandre Bucquet and Vishnu Sarukkai. 2018. *The Bank is Open: AI in Sports Gambling*. CS229 Project Report. Stanford University. `https://cs229.stanford.edu/proj2018/report/3.pdf`

Anthony Curtis. 2021. Do Books Try To Balance Their Action On Both Sides Of A Bet? `https://www.lasvegasadvisor.com/question/balancing-sports-book-action/`

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. 2017. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems* 28, 3 (2017), 653–664. `https://doi.org/10.1109/TNNLS.2016.2522401`

René Manassé Galekwa, Jean Marie Tshimula, Etienne Gael Tajeuna, and Kyamakya Kyandoghere. 2024. A Systematic Review of Machine Learning in Sports Betting: Techniques, Challenges, and Future Directions. *arXiv preprint* arXiv:2410.21484 (2024). https://arxiv.org/abs/2410.21484 Accessed 25 Apr 2025.

Doug Greenberg. 2025. U.S. Sports Betting Industry Posts Record $13.7B Revenue for '24. https://www.espn.com/espn/betting/story/_/id/43922129/us-sports-betting-industry-posts-record-137b-revenue-24 Accessed: 2025-04-23.

Andrei Hirvi. 2021. *Reinforcement Learning for Maximizing Gaming Experience: An Application in Sports Betting*. Master's thesis. Tallinn University of Technology. https://digikogu.taltech.ee/en/Item/480468d2-ad69-4783-adcc-0eb8b83c96d6 Master's thesis supervised by Innar Liiv; accessed 2025-04-25.

Alan Ji. 2024. *Machine Learning for NBA Betting: An Analysis and Exploitation*. Senior Thesis. Princeton University, Department of Operations Research and Financial Engineering. http://arks.princeton.edu/ark:/88435/dsp01s7526g767 Advisor: Ramon van Handel; Finance Program.

Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. *arXiv preprint arXiv:1706.10059* (2017). https://arxiv.org/abs/1706.10059

William De Vena. 2023. Reinforcement Learning for Market Making in the Sports Trading (Betting) Market. https://github.com/williamdevena/RL_for_Market_making_in_Sports_Trading. GitHub repository; accessed 2025-04-25.

Kai Zhao, Chunjie Du, and Guangxin Tan. 2023. Enhancing Basketball Game Outcome Prediction through Fused Graph Convolutional Networks and Random Forest Algorithm. *Entropy* 25, 5 (2023), 765. https://doi.org/10.3390/e25050765