NFL Sports Betting Neural Networks

Louis Cocks
School of Data Science
University of Virginia
Charlottesville, VA, USA
epx8hh@virginia.edu

Rheyan Ampoyo
School of Data Science
University of Virginia
Charlottesville, VA, USA
xcs5hg@virginia.edu

***Abstract*—The growing popularity of sports betting has spurred interest in data-driven models capable of accurately estimating the probability of game outcomes. This project proposes the use of neural networks to predict the probabilities of NFL betting of player proposition bets over/under a set line. Using large-scale historical data, including team performance metrics, sequence data, player statistics, and game specific statistics, the model aims to uncover complex nonlinear relationships that traditional statistical models often overlook. So, how can neural network architecture be optimized through techniques such as feature selection, hyperparameter tuning, adaptive learning, and dropout regularization to balance predictive accuracy and generalization? The architecture of Fusion Transformers handle these steps well with added flexibility for multi-leg parlay positional encoding and interpretable results. With these results evaluated against benchmark models such as XGBoost and Long-Short-Term-Memory (LSTM) using metrics such as Expected Calibration Error (ECE), Parlay Calibration Error (PaCE), and AUC-ROC score. Sequence models (Dual-Head LSTM / Dual-Head TFT) can improve calibration (ECE, PaCE, AUC) for NFL player props compared to strong XGBoost baselines in a testing environment, but real-world profitability remains fragile and highly sensitive to modeling assumptions. This project aims to showcase how deep learning enhances decision-making in sports analytics, specifically by generating probabilistic predictions that more effectively capture uncertainty in player statistics outcomes providing both theoretical insights and practical applications for analysts, bettors and researchers. https://github.com/Lcocks/SportsBetting_NN**

I. Motivation

Sports betting, particularly in football, has evolved into a data-rich industry where analytics plays a critical role in evaluating outcomes and odds. Despite this, many existing models used by analysts and bookmakers still rely on simplified statistical approaches that often fail to capture the complex, nonlinear dynamics of team performance, player interactions, and situational factors influencing game results. As the volume and granularity of sports data continues to increase, spanning player metrics, play-by-play statistics, injuries, and even environmental conditions there is a growing opportunity to apply machine learning (ML) techniques that can uncover deeper patterns within the data. Neural networks, known for their ability to model high-dimensional and nonlinear relationships, present a promising solution for estimating the likelihood of football betting outcomes such as match results, point spreads, and totals. This research aims to design and evaluate a neural network framework that predicting probabilistic outcomes based on historical and contextual game data. This project will investigate how Temporal Fusion Transformers (TFT) and adaptable feature sets affect predictive model performance, accounting for gaps in the use of computationally efficient modeling, and multi-leg parlay scenarios for model testing. Ultimately, this study seeks to contribute to the field of sports analytics by demonstrating how deep learning can improve the predictability and reliability of betting forecasts, offering practical and valuable insights.

II. Dataset

Data is pulled from two main public resources listed below and will be sourced through R and python package calls: nflreadr & nntrn:

- nflreadr
  - Package contains play by play and player specific data.
    - Columns examples consist of Game_id (chr), season_type(chr), Week(int) and game_date(chr)
    - CSV formatted for games.csv, pbp.csv, and more.
- nntrn
  - robust Github repository with api parameters for game by game data; team data; league data; officials data and more.

III. Literature Review

A. Systematic Review of Machine Learning in Sports Betting: Techniques, Challenges, and Future Directions

(Link)

In Galekwa et al. [1], researchers examined the growing role of ML in predicting sports outcomes and optimizing betting strategies. Following PRISMA guidelines, the authors screened hundreds of studies which excluded non-English, theoretical, or non-empirical work to identify research that applies ML methods to real betting data. Across sports, decision trees and neural networks consistently outperformed traditional statistical models along with some niche neural networks, particularly when enhanced with contextual features such as: team form, head-to-head history, player statistics, and betting odds. Researchers reviewed in this meta-analysis [2] found that optimizing models for calibration rather than accuracy yielded up to 69.86% higher betting returns, emphasizing probability reliability over simple prediction. In American football (Section 4.5), researchers leveraged play-by-play data to capture situational dynamics, while in [3] the deep neural network achieves greater accuracy than 95% in soccer prediction, the potential of deep learning. Overall, the review highlights key challenges such as data limitations in context, non-stationarity, and over-reliance on accuracy metrics and concludes that future success in sports betting ML will depend on calibrated, context-rich models that balance predictive precision with real-world profitability.

1) Identify Gaps: A key limitation of the existing studies is the absence of binary classification metrics and modeling approaches directly applicable to multi-leg betting scenarios. Since this project aims to develop an application capable of accurately predicting the outcome of individual legs within a parlay, there is a need for models that emphasize binary or quantile prediction results, such as win-or-loss probabilities rather than continuous performance outcomes like expected yards or completion percentage. Addressing this gap will allow for a more direct alignment between predictive modeling and real-world betting decision-making.

B. Exploiting Sports-Betting Market Using Machine Learning

(Link)

Hubacek and Zelezny [4] explores how convolutional neural networks can be used to predict sports outcomes while intentionally reducing correlation with bookmaker odds. Rather than replicating market-implied probabilities, the authors design a model that identifies informational inefficiencies by de-correlating its predictions from those embedded in betting lines. Their approach operates on a two-team data pipeline, using only the features directly relevant to the competing teams in a given matchup such as form, ranking, and performance metrics, thus emphasizing game-specific rather than league-wide information. To improve profitability, the study introduces methods that 'elaborate various measures to remodel the learned model from the bookmaker's model (approximately estimated from the odds assigned), while maintaining adequate prediction accuracy.' The main models are two types of neural network tests including a standard (deep) feed-forward architecture with 4 dense layers, and a second with 1 convolutional layer followed by 3 dense layers. Confidence-thresholding is added as an enhancement in conjunction with models utilizing logistic sigmoid output which was the utilized activation function. The findings demonstrate that this de-correlation strategy can enhance profit margins by uncovering value bets overlooked by the market, showing that model independence from bookmaker behavior can be as critical as predictive accuracy.

1) Identify Gaps: There was a trade-off between the increased level of de-correlation which dropped accuracy but increased profits and vice versa. Given this, some overlooked aspects of the bookmaker model include public betting bias exploitation by which bookmakers can adjust markets and incentives for geographical location, inherent biases, and knowledge of 'whales' (extremely large bets). A portfolio theory gap includes the missing incorporation of Kelly Criterion for bet sizing which is notably a successful strategy addressed in the meta-analysis (from [5]). A final gap is a lack of realistic market simulation where closing odds with unlimited liquidity overestimates real-world profitability; artificially inflating the direct impact of this analysis on future methodologies for real-world implementation.

C. Sports Betting: an application of neural networks and modern portfolio theory to the English Premier League

(Link)

By integrating machine learning with quantitative finance principles to construct optimized betting portfolios, Jiménez et al. [6] employ Kullback–Leibler (KL) divergence to measure disparity between model-predicted probabilities and bookmaker-implied odds, allowing the system to identify wagers where the market miss-prices true outcome likelihoods. To allocate wagers efficiently, they apply concepts from modern portfolio theory constructing an efficient frontier of betting combinations that minimize the variance for a given expected return. Building on this foundation, the study borrows from modern portfolio theory, defining an efficient frontier of betting portfolios that minimize variance for a given expected return analogous to selecting the "best" number of legs in a parlay. The model further utilizes the Kelly Criterion to determine optimal stake sizing, maximizing long-term bankroll growth while controlling for risk exposure. This combination of probabilistic learning, information theory, and portfolio optimization represents a shift from traditional single-bet prediction toward system-level betting strategies that explicitly balance risk and reward.

1) Identify Gaps: While this framework provides a rigorous mathematical basis for optimizing betting portfolios, it assumes that model-predicted probabilities are perfectly calibrated and that market conditions remain static. These assumptions limit real-world applicability, particularly in fast-changing environments like NFL games where player injuries, weather, and line movements continuously alter the betting landscape. Additionally, the framework focuses on optimizing portfolio weights rather than improving the underlying predictive accuracy of individual bets or legs—an essential component for applications like the one proposed in this project. For a sports-betting model that dynamically predicts and selects parlay legs, bridging this gap requires integrating calibrated neural network predictions with portfolio optimization, allowing the system to adapt to shifting odds and evolving game contexts while still applying Kelly-style bankroll management.

IV. Methodology

This paper covers a customizable approach that takes the best raw techniques while acknowledging the gaps in the study, such as a deterministic listing of the best criterion and techniques under the implementation of a recently developed model, [7] Temporal Fusion Transformers (TFT). Alongside this is widespread real-world applicability in that we have a computationally efficient model with the implementation of multi-leg parlays (not singular bets) as a pivotal point in architecture of the model. The following breakdown describes our simplified TFT model directly. From Data Ingestion and Preprocessing, where input features (X) and target variables (y, sometimes binary) are converted into PyTorch tensors to outputs. The core of the model is the SimpleTFTBackbone, a Transformer Encoder that ingests the sequence data. This input is first mapped to a higher-dimensional space using a linear projection layer and a Positional Encoding is added to the projected data to preserve the sequential order of the time steps as the Transformer architecture is permutation-invariant without it. Computational efficiency comes from reduces dimensionality, pruning of irrelevant features, and caching of encoders for multi-leg parlays. The data then passes through multiple Transformer Encoder Layers, performing multi-head self-attention to capture complex dependencies and patterns within the player's historical data of variable length sequences. The output of the Transformer Encoder is a representation vector for each time step. The model uses the representation from the last time step of the sequence (h_last) as the summary vector for the player's history. This vector is fed into two separate Prediction Heads: a Classification Head (fc_bin) which outputs logits for a binary outcome (e.g., player achieves a stat threshold, a probability prediction) and a Regression Head (fc_reg) which outputs a single value (e.g., the predicted raw statistic). During the Training Loop, the Adam Optimizer is used to minimize the Binary Cross-Entropy Loss with Logits (BCEWithLogitsLoss), focusing only on the classification task, while the regression output is ignored in the loss calculation for this specific training function (train_tft_classifier). The model iterates over the training data in mini-batches for a configured number of epochs. Our differentiation from the original TFT model by using Dual Head for Classification (BCE loss) and Regression instead of multi-head forecasting. Focusing on the core Multi-Head Self-Attention mechanism of the transformer for capturing temporal dependencies over the whole input sequence. The Gated-Residual Networks are not implemented but rather a base encoder layer with positional encoding. LSTM follows a similar logic to simplified TFT as our model is a functional recreation of the core LSTM mechanism, which is then specifically tailored with a Dual-Head for the downstream NFL stat prediction task. The custom, explicit implementation of the LSTM cell's four gates (forget, input, output, and candidate) and their equations ($c_t = f * c_{prev} + i * g$, etc.). XGBoost itself is an ensemble of decision trees built sequentially via Gradient Boosting. Each new decision tree attempts to correct the errors (or residuals) of the previous ensemble. The algorithm is configured as an XGBClassifier with the objective of binary logistic, meaning it predicts the probability of a binary outcome. With each of these three models we have a comparison of time series handling where Dual-Head TFT uses global dependencies to capture relationships between any two time steps in the input sequence, irrespective of distance. Also using Positional Encoding to retain local order. Dual-Head LSTM for local dependencies where it captures relationships by passing a hidden state from t to $t+1$, making it good for short-term patterns but prone to forgetting long-term dependencies. Finally XGBoost where non-sequential ignores the temporal structure by flattening the time dimension ($T \cdot E$ features) such that the model must implicitly learn time's effect from the concatenated features. A comparison of these models will further support multi-leg parlays in understanding why focused calibration of expected calibration error (ECE) [2] implies success over singular accuracy.

V. Evaluation

To evaluate the effectiveness of the models several metrics will be compared against benchmarks including ECE, and ROC-AUC. Along with raw sigmoid output, Platt scaling, and isotonic regression; we created a Parlay Calibration Error (PaCE) for joint probabilities where N games with true outcomes $\mathbf{y}_{true} = (y_1, \ldots, y_N) \in \{0,1\}^N$ and predicted probabilities $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_N) \in [0,1]^N$, we define the parlay calibration error for L-leg parlays as:

$$\epsilon_{PaCE}(L, M) = \frac{1}{M} \sum_{m=1}^{M} \left| \hat{p}^{(m)}_{parlay} - y^{(m)}_{parlay} \right| \tag{1}$$

where for each random parlay sample $m \in \{1, \ldots, M\}$:

$$\mathcal{I}^{(m)} \sim Uniform(\{I \subset \{1, \ldots, N\} : |I| = L\})$$
$$\hat{p}^{(m)}_{parlay} = \prod_{i \in \mathcal{I}^{(m)}} \hat{p}_i$$
$$y^{(m)}_{parlay} = \begin{cases} 1 & \text{if } y_i = 1 \text{ for all } i \in \mathcal{I}^{(m)} \\ 0 & \text{otherwise} \end{cases}$$

Consequently, a threshold does not exist for predictive acceptance of a leg, rather there is a quantile range for all modeled legs. A successful model will show Expected Calibrated Error (ECE)/ Parlay Calibration Error < 0.05, and AUC > 0.80. Varying each models hyperparameters and every models parameters of data feed will be used to display, compare, and contrast the differences between performance (AUC/PaCE) and confidence (ECE).

VI. Experiments

We verified all the data sources necessary to build player-level prop models. We successfully obtained nflreadr play-by-play data from 2018 to 2023, nflreadr team statistics, nntrn API metadata, and ESPN API player-level statistics across the rushing, passing, receiving, defensive, and special teams categories. These sources collectively provide all the features needed to compute player prop outcomes and evaluate combinations. However, we were unable to access player-specific odds via the nntrn APIs. Since the focus of this project is on player-prop parlays, and player odds could not be reliably sourced, we decided against the decorrelation steps and are removing them from our scope. Additionally, we validated access to play-by-play data, but found that play-by-play features were unnecessary for our modeling goals, since parlay outcomes depend on overall prop totals from a game rather than play-level sequences. Therefore, only aggregated stat-per-player data at the game-level will be used moving forward. The predictive target for this milestone is a binary variable taken from a possible parlay from a betting provider (YDSs, REC,s TDs). For a given statistic S and threshold τ, the binary outcome is defined as $y = 1\{S \geq \tau\}$. This formula captures the format of most bets, "Mahomes over 305.5 passing YDS" or " Deon Cain over 0.5 receiving TD." A model is trained for each yardage type and defines a separate binary prediction task, allowing it to produce probabilities for individual legs that later serve as inputs to multi-leg parlays. Structuring the problem as binary classification allows us to ensure consistency across different players, stat categories, and thresholds. We constructed a strong XGBoost model as our baseline classifier and engineered rolling averages over 3, 5, and 8 games; season-to-date cumulative averages; rolling hit rates for the target prop; measures of variance such as rolling standard deviations; performance trends comparing recent form to long-term form; and experience-based features like career games played. Categorical variables, including team, opponent, position, and home/away status, were label-encoded. Evaluation metrics included ROC-AUC, log-loss, accuracy, and Expected Calibration Error (ECE). Many prop categories achieved ROC-AUC values between 0.60 and 0.68, demonstrating that the problem is challenging but learnable. Several models achieved an ECE below 0.10, indicating strong probabilistic calibration. For multi-leg parlays, the baseline LSTM model's accuracy was evaluated by sampling 1000 two-leg parlays from the test set and multiplying their probabilities. Using these predictions, the LSTM model achieved a PaCE of .0464, indicating that the predicted probability of a two-leg parlay typically differs from the actual outcome by just 4.64 percentage points, which is well below our target threshold of 0.05. However, we should note that the model's single-leg parlays resulted in an ECE of .0567, which is only slightly above our desired threshold. Because we did not have access to bookmaker odds data, the Kelly Criterion return would be calculated with the goal of at least breaking even. The implied Kelly fraction for a predicted probability $\hat{p}$ is $f^* = 2\hat{p} - 1$. This is interpreted as the higher f^* the greater the chance of our parlay hitting. A flat-betting baseline was also calculated for comparison. Kelly ROI is only meaningful when predicted probabilities are well calibrated, making ECE and PaCE central to evaluation. Although the lack of real odds reduces the real-world interpretability of Kelly ROI, the synthetic approach still would allow us to study the behavior of calibrated parlay probabilities. Overall, we have successfully established a full pipeline for preparing player-level time-series features, training individual prop models using both XGBoost and LSTM architectures, computing probabilistic calibration through ECE and PaCE, and sampling random two-leg parlays from the test set across different statistics. We look to tune our models to improve performance, handle regex following the exact format of sports book parlays, and also add the Streamlit component for an easy-to-use interface.

VII. Ablation Study 1: Feature Sets (Tabular vs Sequence Aware)

Ablation Study 1 evaluated whether sequence-aware models meaningfully outperform traditional tabular approaches to predict prop outcomes of players. In theory, LSTM and TFT architectures should capture short-term momentum, streakiness, and volatility that single-number historical summaries cannot capture. To test this, we compared the three models: XGBoost using tabular features, and LSTM/TFT models using the last five games as true sequences, across identical receiving-yard props. The results show that all three models are very close given variable hyperparameters such as epochs, batch size, learning rate, estimators, and max depth. With (AUC = 0.8171, 0.8148, and 0.8142 see Figure 1). Very noticeably however there is variation in the ECE and custom metric PaCE, showing the most reliable probabilities with the lowest calibration error (ECE = 0.0072) for TFT. This advantage did not follow through however to the parlay level, where XGBoost achieved the best PaCE score (0.0149 vs. 0.0149 for LSTM and 0.0176 for TFT), indicating that TFT probabilities degrade less when multiplied across legs. Contrary to our initial hypothesis, the sequence models did not extract additional predictive value from the raw game-by-game sequences, suggesting that either the five-game window is too short or the tabular summaries already capture most of the relevant signal. The Dual-Head TFT provided the most confident results with the lowest ECE even though the AUc was not higher than the other models, indicating there could be greater room for calibration. Overall, this ablation highlights that strong tabular baselines remain highly competitive and, in this context, more calibrated and parlay-reliable than the sequence-based alternatives.

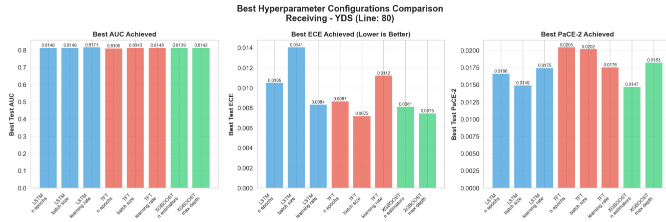

Fig. 1. AUC, ECE, and PaCE Outputs from Hyperparameter Variations in Models

VIII. Ablation Study 2: Sequence Lengths

Ablation Study 2 examined how the length of the sequence window, specifically using the last 3, 5, or 8 games, affects the performance of both the LSTM and TFT models. The motivation was to test whether longer windows meaningfully improve predictive power by capturing more historical context, or instead dilute relevant information by mixing outdated performance with recent form. The results show a consistent pattern: performance generally peaks around a 5-game window for both architectures. For LSTM, AUC rises from 0.7790 (3 games) to 0.7890 at five games, then slightly drops to 0.7860 at 8 games; TFT follows a similar trend, with an AUC peak at five games (0.7853). Calibration behavior varies more dramatically with TFT, in particular, and becomes poorly calibrated at five games (ECE = 0.0757). At the same time, LSTM remains more stable—but both models experience increased parlay error (PaCE) when windows are too short (noisy) or too long (stale).

	model	window_len	auc	ece	pace2	n
0	LSTM	3	0.779028	0.020154	0.175534	5307
1	TFT	3	0.774747	0.012516	0.159462	5307
2	LSTM	5	0.789034	0.019783	0.156100	4448
3	TFT	5	0.785321	0.075735	0.214891	4448
4	LSTM	8	0.786017	0.015815	0.180853	3352
5	TFT	8	0.783902	0.051245	0.168026	3352

Fig. 2. Separate Test Metrics Comparing Sequence Model performance across differing Sequence Lengths

Fig. 3. Variable `N_PAST_GAMES` (number of games look-back) Performance on Each Model With Test Metrics

Overall, these findings suggest that the five most recent games strike the best balance between capturing proper player form and avoiding outdated or low-signal history. Longer windows do not reliably improve performance and may even worsen calibration, supporting the hypothesis that excessive historical context can obscure the signal needed for accurate prop- and parlay-probability estimation.

IX. Ablation Study 3: Calibration Methods

Ablation Study 3 evaluated the impact of probability calibration on XGBoost's predicted hit rates by comparing three methods: raw sigmoid output, Platt scaling, and isotonic regression. The raw model is reasonably well-aligned with the perfect calibration line, but still exhibits noticeable overconfidence in the highest-probability bins, as reflected in its ECE. Applying Platt scaling, a logistic regression fit on the validation scores, smooths the probability mapping and reduces miscalibration in the low- to mid-probability regions, producing a more consistent reliability curve. However, isotonic regression delivers the most substantial correction overall: by fitting a non-parametric, monotonic function, it nearly eliminates structural bias across all bins and yields the closest match to the ideal diagonal. This improvement is evident in the high-probability range, where isotonic regression corrects the overconfident tail seen in the raw model. Together, these results show that even a well-performing classifier benefits significantly from calibration, and

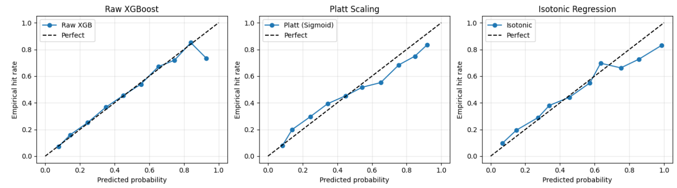

Fig. 4. Calibration comparison showing raw, Platt-scaled, and isotonic probabilities against perfect reliability.

isotonic regression in particular provides substantial gains in both reliability and parlay-focused error metrics such as PaCE.

X. Conclusion

Across all three ablation studies, the empirical results reveal that while different modeling approaches Dual-Head LSTM sequence encoders, Dual-Head TFT-style transformers, and tabular models of XGBoost. Overall each exhibits strengths in certain evaluation dimensions; no single model consistently or decisively outperforms the others. This outcome reflects a deeper characteristic of player-prop forecasting: the data-generating process is highly stochastic and context-dependent. Individual player outcomes are influenced by rapidly shifting factors such as game flow, injuries, defensive matchups, and coaching adjustments, all of which introduce substantial variance and regime instability into the task. Even well-designed deep learning architectures struggle to capture these irregularities when historical patterns are weak predictors of future behavior. Consequently, model performance gains were modest and often overlapping, suggesting that the predictive ceiling for this domain requires high-degrees of engineering and architectural modeling. These findings underscore the importance of not only choosing an appropriate architecture but also leveraging robust preprocessing, feature engineering, and calibration strategies to extract the signal present in the data. The results highlight that modeling volatility-dominated phenomena demands tempered expectations, with improvements measured in incremental refinement rather than large leaps in accuracy. Ultimately, Sequence models (Dual-Head LSTM / Dual-Head TFT) can improve calibration (ECE, PaCE, AUC) for NFL player props compared to strong XGBoost baselines; however, fine-tuning and over-confidence in a testing environment can lead to failed model assumptions.

XI. Further Experimentation & Disclaimer

The original Temporal Fusion Transformer [7] adds layers around the core Dual-Head Self Attention transformer to manage heterogeneous time series data (static, known, and observed time-varying features), control the model's complexity with gating layers, and provide interpretability via the variable selection weights. Portfolio theory is often used in the best models Galekwa et al. [1] & Jiménez et al. [6] that incorporate odds data; however, Kelly criterion is seldom mentioned. It could be used to diminish hidden assumptions in the variable selection network when adjusting for different parlay legs in a manner of risk management. Although not implemented here only referenced, Kelly criterion is set to provide a reliable determination of profitability. Also de-correlation of any bookmaker data from the datasets to support Hubacek and Zelezny's [4] findings. Using these three novel ideas further experimentation is recommended to build on this baseline and expand into each of the separate yet related research projects and greatly enhance sports players predictability. While the models developed in this study demonstrate measurable improvements in predictive and calibration performance, it is essential to emphasize that sports betting carries substantial financial risk, and model outputs should not be interpreted as guarantees of profitability. Player-prop outcomes are influenced by many unobserved or unpredictable factors such as injuries, coaching decisions, in-game variance, and random chance, which no statistical or machine-learning model can fully anticipate. Moreover, real-world betting environments impose additional constraints, including shifting odds, market efficiency, bookmaker margins, bet limits, and adversarial pricing strategies, all of which can eliminate or outweigh any edge suggested by model forecasts. As such, even models that appear well-calibrated in offline evaluation may fail to generate positive expected value in practice. These results should be viewed as analytical tools, not financial advice, and any real-money application should be approached with caution, strict bankroll management, and an understanding that losses are not only possible but likely.

REFERENCES

[1] R. M. Galekwa, J. M. Tshimula, E. G. Tajeuna, and K. Kyandoghere, "A systematic review of machine learning in sports betting: Techniques, challenges, and future directions," 2024. [Online]. Available: https://arxiv.org/abs/2410.21484

[2] C. Walsh and A. Joshi, "Machine learning for sports betting: Should model selection be based on accuracy or calibration?" *Machine Learning with Applications*, vol. 16, p. 100539, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266682702400015X

[3] W. Deng and E. Zhong, "Analysis and prediction of soccer games: An application to the kaggle european soccer database," *Insight - Statistics*, vol. 3, p. 1, 11 2020. [Online]. Available: https://www.researchgate.net/publication/346862578_Analysis_and_Prediction_of_Soccer_Games_An_Application_to_the_Kaggle_European_Soccer_Database

[4] O. Hubacek, G. Sourek, and F. Zelezny, "Exploiting sports-betting market using machine learning," *International Journal of Forecasting*, vol. 35, no. 2, pp. 783–796, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016920701930007X

[5] U. Matej, G. Sir, O. Hubacek, and Z. Filip, "Optimal sports betting strategies in practice: an experimental review," *IMA Journal of Management Mathematics*, vol. 32, 02 2021. [Online]. Available: https://www.researchgate.net/publication/349139435_Optimal_sports_betting_strategies_in_practice_an_experimental_review

[6] V. Jiménez, R. Alberto, L. Ontiveros, J. Manuel, and E. Possani, "Sports betting: an application of neural networks and modern portfolio theory to the english premier league," arXiv.org, Papers 2307.13807, Jul 2023. [Online]. Available: https://ideas.repec.org/p/arx/papers/2307.13807.html

[7] B. Lim, S. O. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207021000637