LUND UNIVERSITY

SCHOOL OF ECONOMICS AND MANAGEMENT

Master's in Data Analytics and Business Economics

Precedent-Based Adaptive Modelling Framework for Basketball Win-Prediction

Dennis Markovic

DABN01
Master's Thesis (15 ECTS)
Supervisor: Simon Reese
May 2025

Abstract

The empirical prediction of outcomes in competitive sports, particularly basketball, remains a complex challenge due to inherent game variability and the difficulty in discerning replicable strategic adaptations from random fluctuations, especially in the case of unexpected results ("upsets"). This thesis addresses this by introducing and evaluating novel features designed to enhance the predictive accuracy of NBA game outcomes. These features capture aggregate seasonal team strength markers and dynamic in-game performance indicators, building upon traditional box-score statistics.

Results demonstrate that pre-game models incorporating novel momentum features achieved 71.1% accuracy, surpassing Las Vegas betting lines (68.5%). For in-game predictions, the GRU model, leveraging the sequential nature of quarterly data, demonstrated superior performance, achieving 83.69% accuracy and an AUC-ROC of 0.9157. This outperformed a strong ensemble model (82.59% accuracy, 0.9119 AUC) and highlighted the predictive power of cumulative in-game metrics. The findings confirm that the proposed novel descriptive and explanatory features significantly improve predictive accuracy, with sequential modeling offering particular advantages for capturing evolving game dynamics. This work contributes to understanding replicability in sports outcomes and provides a robust framework for both pre-game and adaptive in-game prediction.

Keywords: Tree Based Methods, Binary Classification, RNN, LightGBM

I. Introduction

Since even before the days of the Oakland Athletics and Billy Beane (famously retold in the story of Moneyball), there has been a fascination with factors that empirically contribute to winning in competitive sports. Basketball lends itself to this consideration almost more than any other of the sports in the global zeitgeist, due to its high scoring nature, the tendency for significant swings in contests, the high volume of games played every year in professional leagues, and the abundance of informative counting statistics which have been tracked for decades.

Unexpected results, or so called "Upsets" are common in professional sports, and are typically the result of deeper contextual factors being changed between when an expectation is set and when it is realised. For example, the New York Knicks (51-31) entered the 2025 playoffs having lost all 11 matchups against teams with records better than theirs; but took surprising 2-0 and 3-1 leads in the series against the reigning champion Boston Celtics (61-21). Domain expert pundits and the major betting services placed the Knicks as overwhelming underdogs for each of the three wins in the series so far.

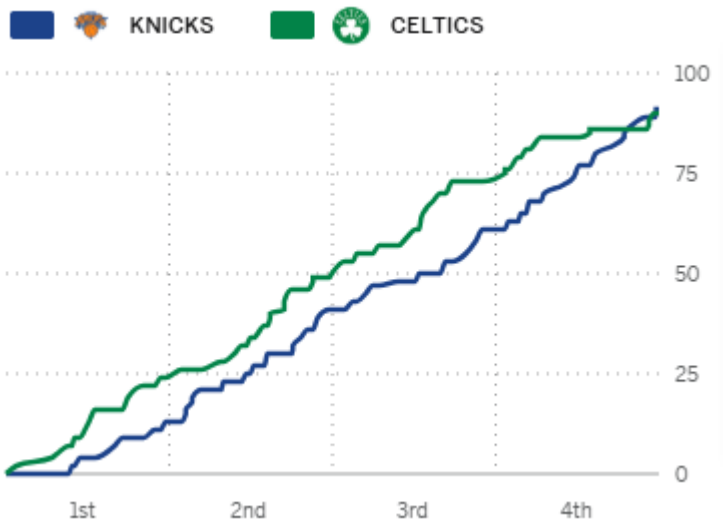

Figure 1. Knicks Celtics Game 1 Score Progression

As it pertains to this uncertainty, experts are often left having to contend with whether an unexpected result is a result of natural variate inputs that can be expected to regress to the mean, or the result of strategic adaptation that requires an update to expectations. In other words, if an unexpected event occurred, is it replicable? How should we adapt our expectations for the next prediction?

This thesis' main contribution is to provide some clarity to this question by introducing novel features that constitute aggregate success markers for teams throughout a season, and for in-game predictions to adapt our expectation as information is provided iteratively throughout the game.

This contribution builds on the predictive power of "Box-Score" features, which are all count variables gathered by the National Basketball Association (NBA) during all games for the sake of score keeping.

Table I.I: NBA Box-Score Data features			
Feature	**Description**	**Feature**	**Description**
MP	Minutes played in the game	FG	Field goals made in the game
FGA	Shots taken in open play	3P	3-point field goals made in the game
3PA	Shots attempted from behind the 3-PT line	FT	Free throws made in the game
FTA	Free throws attempted in the game	ORB	Times the ball was caught from a teammate's miss
DRB	Times the ball was caught after an opponent's miss	AST	Passes made to teammates who made a shot
STL	The ball was stolen from the opponent	BLK	Blocks in the game
TOV	Times the ball was lost to the opponent team	PF	Personal fouls committed
PTS	Points scored in the game		

The main success metric defined in Basketball discourse is "Net Rating", defined as the difference between the scores of two teams in an allotted time period. A positive Net rating means a team "won those minutes". Net rating is derived from Offensive Rating (a team's points scored per 100 possessions) and Defensive Rating (a team's points allowed per 100 possessions), where a possession is defined as a sequence of events a team has control of the ball, starting with a rebound and ending with either a field goal, free throw, or turnover. Many of the previously described Box Score features have correlations with either NRTG or its constituent parts.

Table I.II Box-Score Correlations with NRTG components					
DRTG Correlations			**ORTG Correlations**		
Feature	**Coefficient**	**Significance**	**Feature**	**Coefficient**	**Significance**
Def Rebounds	-4.432	****	Free Throws	1.707	****
Shots >20ft	1.988	****	Top Scorer Makes	1.855	****
Free Throws	1.469	****	Shots <5ft	0.664	****
Turnovers	-1.394	****	Shots >20ft	0.638	****
Lead Changes	0.485	****	Lead Changes	0.303	****
Off Rebounds	-0.942	****	Off Rebounds	-1.229	****
			Turnovers	-1.402	****
			Def Rebounds	*-0.087*	

There are a broad set of problem definitions of relevance in academic discourse around Basketball, fueled by the deterministic nature of competitive sports and the popularity of betting markets. Many models predict a team's expected number of wins in a season, or the likely championship winner, or the Most valuable player in a given year. All of these are cumulative and successive problems that build an expectation over the course of the season.

Due to the granularity of the data available (400-600 rows for each of 1280 games) I instead chose to target individual game results because the larger number of prediction circumstances allows us to make a stronger predictive model, and gain more interpretability from the model.

For this problem definition, two baseline accuracies are considered to validate the final results. The first is a lower bound, comparing the result to the accuracy of a model which predicts the home team to win each game. The accuracy of which for the prescribed dataset is 54%.

The second are Las Vegas Betting lines, operated by major betting institutions. The accuracy for this season's predictions stand at 68.5%[1] accuracy. The final "Momentum"model accuracy for this investigation beat the Las Vegas pre-game line by a margin of 2.6%.

More specifically, I consider two branches of models. Those that use pre-game context to predict the result, with no information on that particular game, and another model set that use the first three quarters of the game to predict the outcome. This secondary validation structure is adapted from Tian et. al (2020) who used the first half of NBA games to predict the final point differential based on scoring pace-related features.

[1] Ferrara, Joe. "Predicting NBA Games." *GITHUB*, 4 May 2020,

This validation method is included to quantify the importance of game state and balance for final result progressions, and also in order to introduce models that are trained on sequence data, to validate its viability for this use case.

For each game prediction, I will output a win probability for each team, as well as the team predicted to win. Based on these variables, I will subsequently validate results as follows:

AUC-ROC of Accuracy with respect to win threshold

Accuracy of Correct Game Win Predictions.

Prediction Accuracy on games that end within 12 points (informed cutoff for what is considered a close game).

And thus I propose the question

"To what extent do the proposed novel descriptive and explanatory features improve the predictive accuracy of NBA game outcomes using team level features?"

II. Literature Review

Predicting the outcomes of NBA games and playoff series has long been a pursuit of both academics and industry professionals. Early approaches relied on simple statistical models and expert heuristics, but recent years have seen a surge in Machine Learning (ML) and advanced analytics applied to this problem.

One of the foundational approaches to game prediction is the Elo rating system, originally developed for chess but adapted to sports. Elo-based models maintain a dynamic rating for each team that updates after every game based on the game's outcome and expected result. Such ratings have been used as a baseline for NBA predictions for decades. For example, FiveThirtyEight's[2] prediction system uses a CARM-Elo rating (an Elo variant incorporating player value) to generate win probabilities for each game. Traditional power ranking systems (e.g. Jeff Sagarin's rankings or John Hollinger's Basketball Power Index at ESPN) also stem from similar ideas – rating team strength by past performance and point differentials. ESPN's Basketball Power Index (BPI)[3], for instance, is described as "a measure of team strength that is meant to be the best predictor of performance going forward," expressed in points above/below average. These models do not fit into traditional definitions of Machine Learning but are nonetheless influential due to their interpretability and reasonable accuracy (often correctly predicting 70% or more of games by picking the higher-rated team).

[2] Silver, N., & Fischer-Baum, R. (n.d.). How Our NBA Predictions Work. FiveThirtyEight.
[3] "ESPN Analytics. (n.d.). Basketball Power Index (BPI) Predictions. ESPN.

Academic interest in NBA outcome prediction goes back at least to the early 2000s. Early work often applied statistical classifiers to team averages. For example, Wei (2001) applied a Naïve Bayes classifier using team stats, and Poropudas (2011) tried a Kalman filter for game predictions. A milestone study by Cao (2012) systematically compared logistic regression, Naïve Bayes, support vector machines (SVM), and neural networks on NBA game data, finding that a straightforward logistic regression was the most reliable, achieving about 68% accuracy. Around the same time, Miljković et al. (2010) also experimented with ML algorithms on NBA games, reporting roughly 67% accuracy with a Naïve Bayes model. These early studies established baselines – roughly 2/3 of games predicted correctly – and demonstrated the feasibility of applying ML to sports outcomes. They also highlighted challenges like limited data and overfitting, as simpler models often matched or beat more complex ones in those days.

"Basketball on Paper" by Dean Oliver (2004) introduced the "Four Factors" framework (shooting efficiency, turnover rate, rebounding rate, and free throw rate) which became influential in feature selection for game outcome models. Logistic regression models using these factors showed that each of the four is correlated with win probability, net rating, etc (with proportions 40-25-20-15% respectively). For instance, a logistic model by Leicht et al. (2017) testing on international basketball found a combination of factors like defensive rebounds, turnovers, and steals was highly predictive of victory.

These insights carried into NBA modeling – many legacy models incorporated variants of these core statistics. Overall, by the 2010s, logistic regression and related rating systems (like Bradley-Terry models, which are essentially logistic models for paired team comparisons) were standard tools. They provided a baseline accuracy around 60–70% and helped identify which stats correlated most with winning. For example, Horvat et al. (2020) and Houde (2021) applied logistic regression on NBA data and saw accuracy around 60–65%.

In some cases, logistic regression with carefully engineered features can perform surprisingly well: Cao's analysis noted it outperformed more complex classifiers on his NBA dataset. Modern variants like regularized logistic regression or Bayesian logistic models (to account for uncertainty) have also been used. Logistic models are also at the core of many rating systems (as the relationship between team strength difference and win probability is often modeled as a logistic curve).

Support Vector Machines are another commonality in sports prediction, as they can capture non-linear relationships via kernels. Kaur and Jain (2017) developed a hybrid fuzzy SVM model for NBA games which achieved about 72% accuracy in the 2015–16 season. By introducing fuzzy logic to handle uncertainty in inputs, their model outperformed a standard SVM on the same data. Other researchers (e.g. Cao 2012, Li et al. 2021) have reported SVM accuracies in the 70–75% range. SVMs perform well when there is a clear margin of separation in the data (e.g. strong vs weak teams), but they require careful tuning of kernel and regularization parameters. They have somewhat fallen out of favor compared to tree-based models and neural networks in recent years, due to scalability and interpretability issues.

Decision tree-based algorithms are widely used thanks to their ability to handle feature interactions, with the most common form being Random Forests. Lin et al. (2014) and Zhang et al. (2021) used random forests but saw only ~64–65% accuracy, similar to logistic regression on

their data. However, tree ensembles can improve if more feature context is added. Cai et al. (2019) reported an 84% accuracy using an ensemble of models (including tree-based learners) on Chinese Basketball Association data (The caveat being that team strength is more unbalanced in the CBA). Boosted trees have especially gained traction: gradient boosting machines like XGBoost or LightGBM allow more complex decision boundaries.

Neural network approaches range from classic multi-layer perceptrons to advanced architectures. Osken and Onay (2022) built an artificial neural network model that first clustered players into "style" types and then used those clusters with team stats to predict game outcomes. Their approach yielded about 76% accuracy across several seasons.

Deep learning models can capture non-linear patterns and interactions that simpler models miss. Recurrent or time-series neural nets have been used to incorporate form and momentum; for example, the MambaNet (2022) model is a hybrid neural network that processes a time series of team and player stats to predict game outcomes. It reportedly achieved ~73% accuracy in predicting game winners, and also identified the significance of features like three-point percentage and free throws made in its learned weights.

Another frontier method is Graph Neural Networks (GNNs) – treating players or teams as nodes in a graph. Zhao et al. (2023) applied a GNN to model the passing network of teams, integrating it with machine learning classifiers, and obtained about 71.5% accuracy. The GNN approach is novel because it incorporates interaction data (passes and team play style) rather than just aggregate stats, offering a new perspective on team dynamics. Overall, while deep learning models can improve predictive power, they require large data and careful tuning. In practice, their gains over simpler methods have been moderate for game-level predictions, but they offer more flexibility to incorporate diverse data sources (images, tracking data, play-by-play sequences, etc.).

Many high-performing systems use ensembles or hybrid approaches, combining multiple models to leverage their strengths. A hybrid of SVM and decision tree was explored by Pai et al. (2017), who argued that combining SVM's predictive power with decision trees' rule-based insights yielded better results for NBA games. Ensembles like stacking or voting classifiers have been used to blend predictions from logistic, tree-based, and neural models. The idea is to reduce variance and bias by aggregating different algorithms. Sukumaran et al. (2022)[4] conducted a systematic review and noted that several top studies reached about 84% accuracy using ensemble methods. These often incorporate domain knowledge (e.g., separate models for offense and defense that are then merged). A specific example of a hybrid is the fuzzy logic + SVM model by Kaur & Jain mentioned earlier, which can be seen as using a fuzzy system to preprocess inputs (handling uncertainty in player performance) before an SVM does classification.

Bayesian methods, while not as common, underpin some advanced approaches. A Bayesian framework allows incorporation of prior knowledge (e.g., preseason expectations or historical team strength) and provides probabilistic predictions with uncertainty estimates. Manner (2016)

[4] C. Sukumaran, D. Selvam, M. Sankar, V. Parthiban, and C. Sugumar. 2022.Application of artificial intelligence and machine learning to predict basketball match outcomes: A systematic review.*Computer Integrated Manufacturing Systems*, 28:998–1009.

introduced a heteroscedastic binary logistic model for NBA outcomes, treating team strengths and home advantage as latent parameters that differ significantly in their variance (uncertainty). By fitting this in a Bayesian way over 8 seasons of data, the model could output win probabilities that adjust for factors like back-to-back fatigue and varying team performance volatility. Another Bayesian concept is the Bradley-Terry model solved via Bayesian inference – essentially a hierarchical model where each team has a skill distribution that gets updated with each game. Such models have been applied to basketball to update team ratings dynamically and predict outcomes, similar in spirit to Elo but with a probabilistic foundation. A recent Bayesian approach by Attard et al. (2023) uses Bayesian hierarchical modeling on NBA data to jointly estimate team offensive and defensive strengths.

A summary of accuracies for all discussed literature can be found in the Appendix [II.I].

III. Data Collection

The dataset for this investigation is sourced from the basketball forum basketballreference.com, who provide extensive statistical features on different scales and resolution related to NBA games. In particular, the investigation sources the Play-by-play (PBP) data for all games of the 2024-25 season. This constitutes a time stamped sequence of every Box score event that occurred in an NBA game in the 24-25 season. This, in turn, allows for modeling the influence of specific game events and sequences on the final results.

The data was scraped using the Python Library BeautifulSoup, by looping over the itinerary page for each team respectively, meaning the dataset could be updated throughout the season as more games had been played. All games are concatenated to a master file which is then preprocessed and further processed to create the final model variables.

We aggregate the information of each game by quarter: this is done because each quarter begins and ends with a time for rest and deliberation between the constituents of each team. Teams iterate and refine their approach in the time between quarters and thus the sequence of quarters provides important feed-forward context. This data format is the most common in NBA game prediction analysis, typically in some aggregate form (game lags, season averages etc.).

There is also tracking data, which is often used internally in teams for strategic adjustment and more granular analysis for recruitment and evaluation. This is more in-depth data, including classifications of events in terms of higher level strategic classifications, like how a team created a shot for a player, where the shot was taken from, and so on. This data is scarcely available in sufficiently reliable volumes and resolutions, and is not available on the play-by-play basis used in this analysis. As a result, it is not included in any of the features in this feature set. See section [III.V] for further consideration of data limitations.

The dataset is complete and contains no null variables, I operate under the presumption that data error related to inaccurate measurement is random and without bias. All features are standardised to ensure compatibility of range. Categorical values such as home advantage, individual team identity, and game date are one-hot encoded to improve their interpretability by the utilised ML models.

IV.I Feature Selection

A primary focus of my thesis is to propose novel features that I extract from the underlying dataset and use in predictive models. In particular, I propose features that add deepened internal (matchup) and external (motivation and fatigue) context to each quarter. In this chapter I outline some of the features that I propose.

The first features I propose are the constructed variables using a framework known as “Four Factors”. These are not novel but their transformation requires some consideration. The second feature set are ones related to Efficiencies and scoring states. These features are considered for each quarter, as well as cumulatively from Q1-Q3. The third set of features are iterative elo ratings and constructed features based on them, for measuring past precedent of offensive and defensive overperformance.

IV.II Four Factors

Team based analysis considers aggregate metrics that drive winning. Since basketball teams play under a set time limit per quarter, and the game is reciprocal, meaning possession switches hands symmetrically and teams have the same number of possession in any given quarter (subject to the time constraints at the end of the quarter), the game can be seen as a two sided optimisation problem. The goal of a given team is to maximise the scoring per possession of their team and minimise the same variable for the opponent team.

Dean Oliver's seminal work, "Basketball on Paper" (Oliver, 2004), introduced the "Four Factors," a coherent framework for deconstructing the core components that drive team success and winning in basketball. These factors, which focus on critical aspects of team efficiency, are widely recognized and extensively utilized within the basketball analytics community, thus forming a foundational pillar of the feature set. The four factors have been operationalised for inclusion in the models as follows:

Table IV.I Four factors description		
Factor Name	**Description**	**Formula**
Shooting Efficiency (eFG%)	Accurately reflects shooting performance by accounting for the additional value of three-point field goals.	(Field Goals Made + 0.5 * 3-Pointers Made) / Field Goals Attempted
Turnover Rate (TOV%)	Quantifies a team's ability to maintain possession, representing the percentage of possessions ending in a turnover.	Turnovers / Possessions
Defensive Rebound Rate (DREB%)	The percentage of available defensive rebounds a team successfully secures.	Defensive Rebounds / (Defensive Rebounds + Opponent Offensive Rebounds)
Free Throw Rate (FTr)	Measures a team's ability to generate scoring opportunities from the free-throw line.	Free Throws Made / Field Goal Attempts

The calculation of rate-based metrics such as Turnover Rate and, critically, Points Per Possession (PPP) necessitates an accurate estimation of the total number of possessions a team has during a game or a segment thereof. I employ a widely accepted and empirically validated approximation for team possessions, given by the formula:

$$Possessions = 0.96 * (Field\ Goal\ Attempts + TOV + 0.44 * Free\ Throw\ Attempts - Offensive\ Rebounds)$$

These Four Factors are systematically incorporated into our predictive models. For each game, I derive these metrics for both the team in question and their opponent. For pre-game prediction models, these factors are represented as season-to-date averages. For the in-game (Q1-Q3) prediction models, these factors are calculated based on the cumulative statistics from the elapsed portion of the game. Furthermore, the differentials between a team's performance in a factor and their opponent's corresponding performance are often highly indicative of matchup advantages and are thus included as crucial features.

IV.III Game State

While pre-game metrics provide a foundational assessment of team capabilities, the actual unfolding of a basketball game introduces a wealth of dynamic information critical for refining predictions, particularly for models aiming to forecast outcomes based on partial game data. This subsection details the features engineered to capture the evolving game state, cumulative team performance up to the end of the third quarter, and quarter-specific dynamics that may signal shifts in momentum or control. These features are paramount for the in-game (Q1-Q3) prediction task.

The core philosophy behind these features is that the narrative of a game, how leads are built, how teams respond to deficits, and the efficiency demonstrated in preceding periods, provides strong signals about the likely final outcome.

Cumulative Score-Based and Efficiency Metrics (End of Q3):

These features summarize the overall state of the game and the aggregate performance of both teams leading into the final quarter.

Table IV.II Scoring Efficiency Metrics

Feature Name	Description	Calculation/Details	Purpose/Interpretation
Score Differential	The most direct indicator of game state.	Team's cumulative score minus the opponent's cumulative score at the end of Q3.	Positive value indicates the team is leading; negative value indicates a deficit.
Cumulative Points Per Possession (PPP)	Measure of offensive efficiency over the majority of the game.	Total score after three quarters / Total possessions.	Indicates how effectively each team scores per possession up to the end of Q3.
Cumulative Net Rating	The score differential adjusted for the pace of the game.	The difference between the team's and opponent's Cumulative Points Per Possession over Q1-Q3.	Provides a pace-adjusted measure of dominance or deficit over the first three quarters.
Cumulative Four Factors	In-game assessment of performance across fundamental basketball pillars.	The Four Factors (eFG%, TOV%, OREB%, DREB%, FTr),.	Assesses core aspects of basketball performance

Game Flow and Momentum Indicators (End of Q3):

Beyond simple score and efficiency, features that describe the game's trajectory and volatility offer additional predictive insight.

Table IV.III Game Flow Features			
Feature Name	**Description**	**Calculation/Details**	**Purpose/Interpretation**
Lead Changes	The total number of times the lead has switched between the two teams.	Counted during the first three quarters (Q1-Q3).	A high number indicates a closely contested and potentially unpredictable game.
Maximum Lead	The largest point advantage held by the team at any point.	Measured at any point during the first three quarters (Q1-Q3).	Indicates the peak advantage the team managed to build.
Maximum Deficit	The largest point disadvantage faced by the team at any point.	Measured at any point during the first three quarters (Q1-Q3). Represented as a positive number.	Indicates the biggest deficit the team had to overcome or is currently facing.
Momentum Swing	Captures the volatility or extent of significant shifts in the scoreline.	Absolute difference between the team's Maximum Lead (Q1-Q3) and Maximum Deficit (Q1-Q3).	A larger value suggests a game with significant runs by both teams, potentially indicating less stability in the current score.

Quarter-Specific Performance Metrics (Q1, Q2, Q3 individually):

To provide a more granular view of performance trends and to supply sequential information for models like Gated Recurrent Units (GRUs), key performance indicators are also calculated for each of the first three quarters individually.

The rationale for including these disaggregated quarterly features is to investigate whether performance in specific quarters (e.g., a strong third quarter often being a historically significant indicator) carries unique predictive weight, and to allow models to learn patterns from the sequence of quarterly performances. Preliminary analysis indicated that while cumulative Q1-Q3 data is highly predictive, the trajectory and individual quarter performances contribute additional nuanced information, with predictive signals generally strengthening as the game progresses from Q1 through Q3. The full feature significances for the final trained model are included in Appendix section [V.I]

Table IV.IV: T-value test for significant difference between wins and losses

Feature	t-stat	t p-value	Cohen's d	Mean diff	Correlation	Significance
q3_score_diff	22.87	6.89×10^{-89}	1.678	18.03	0.635	***
cum_nrtg	22.56	4.44×10^{-87}	1.655	24.27	0.629	***
q3_largest_deficit	19.74	1.35×10^{-70}	1.448	10.68	0.578	***
q3_largest_lead	18.29	2.02×10^{-62}	1.342	10.81	0.549	***

To further understand the characteristics of the feature set, an analysis of feature discriminability between winning and losing games was conducted. Features such as q3_score_diff (mean difference of 18.03 points, Cohen's d=1.678) and cum_nrtg (mean difference of 24.27, Cohen's d=1.655) exhibit highly significant differences ($p < 0.001$) and strong correlations with game outcomes. This confirms their substantial explanatory power and justifies their inclusion in the feature set. The full featureset of discriminable variables is included in the Appendix [IV.IV]

IV.IV. ELO

In order to dynamically quantify the underlying, latent strength of NBA teams and to meticulously track their evolving performance trajectories throughout the season, an adapted Elo rating system was developed. ELO ratings, originating from the world of competitive chess, offer a robust and theoretically grounded methodology for assessing and updating relative strength based on game outcomes, the strength of opponents, and the margin of victory. This paper's bespoke ELO system maintains and updates three distinct ELO ratings for each participating team:

Table IV.V: Elo Definitions	
Offensive Elo	This rating is designed to reflect a team's scoring proficiency and offensive efficiency, adjusted for the quality of the opposing defense and other contextual factors.
Defensive Elo	Conversely, this rating measures a team's ability to suppress opponent scoring and limit defensive efficiency, adjusted for the quality of the opposing offense.
Composite Elo	This serves as an overall, holistic measure of a team's strength, derived as the arithmetic mean of its current Offensive ELO Rating and Defensive ELO Rating.

All teams are initialized with an ELO rating of 1500 points across all three categories (Offensive, Defensive, and Composite) at the commencement of the 2024-25 season under ELO calculation. These ratings are then iteratively updated following each Quarter played. The core mechanism for updating ELO ratings involves calculating an expected score for both the home and away team based on their respective offensive and defensive ELO ratings, a league average baseline, and a home-court advantage adjustment. The expected scores are determined as explained in the appendix [IV.I].

The actual game scores are then compared to these expectations. The ELO update magnitude for each team is proportional to the difference between the actual outcome (considering margin of victory) and the expected outcome, scaled by a dynamic K-factor. The K-factor, which determines the responsiveness of the ELO ratings to recent results, is adjusted based on the number of games a team has played in the season, allowing for more rapid adjustments early in the season and more stable ratings as more data accumulates.

The ELO ratings for offense and defense are updated based on how a team's scoring and opponent's scoring deviate from these ELO-derived expectations. For instance, significantly outscoring the ELO-based offensive expectation leads to a substantial increase in Offensive ELO.

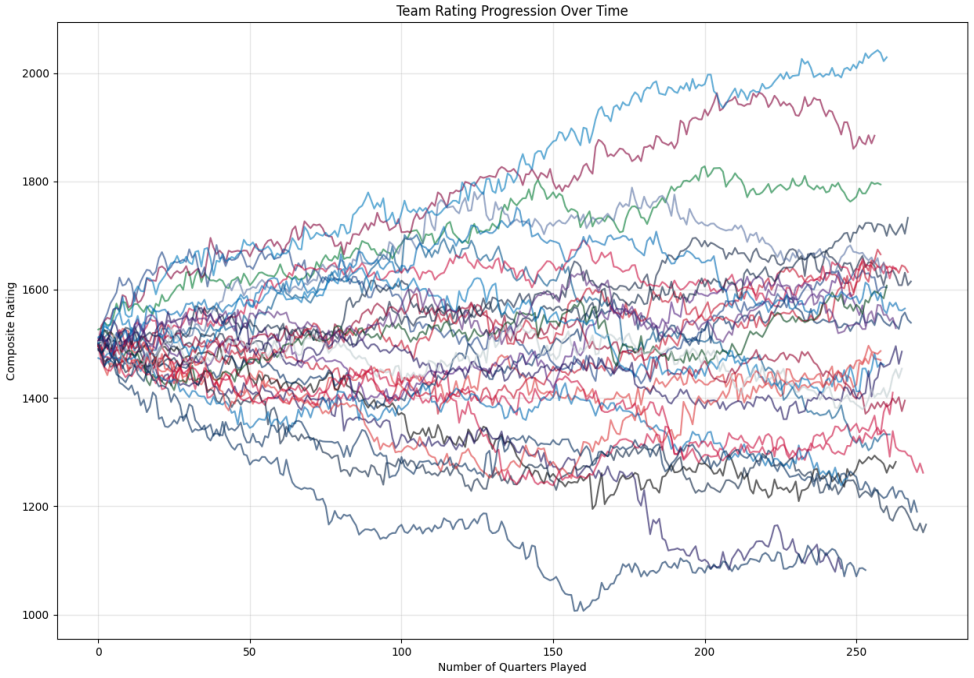

Figure IV.II. Team-separated Composite Elo after quarters played.

To further capture the crucial elements of recent team form and performance trends, which are often highly predictive of near-term outcomes, I have engineered several additional features derived from these dynamic ELO ratings:

Table IV.VI: Extended Elo feature list.

Feature Name	**Definition**
ELO Momentum	The first derivative (rate of change) of the team's Composite ELO rating, calculated over their last 5 games
ELO Variance	The statistical variance of a team's Composite ELO rating over their last 10 games.
10-Game Rolling Win Rate	The percentage of games won by the team out of their last 10 games played.
Rating Differentials	The absolute difference in Composite ELO ratings between the two competing teams.
Matchup-Specific ELO Differentials	Cross differences in ELO (Team A offense - Team B defense).
ELO-derived Win Probability	The implied win probability for the team in the upcoming game, calculated based on ELO rating differences and incorporating home-court advantage.

ELO momentum and win rate are included in order to quantify recent form in the models. Momentum, particularly, is a measure of a given team's recent trend in their ability to outperform their expected offense and defense, taking into account the strengths of recent opponents. This is complimented by the Elo Variance, which quantifies the consistency of a team's recent performance relative to their expectation, and as such acts as a measure of uncertainty.

The differential terms are employed as interaction terms so the model can learn potentially important feature relations, since good offensive teams might have quantifiable advantages against bad defensive teams.

These ELO-based features, particularly the Composite ELO rating itself, its differentials, and the matchup-specific ELO differentials, form a critical component of the input feature set for pre-game prediction models, providing a robust and dynamically updated assessment of team strength and competitive context. The seasonal progression of each team's offensive, defensive, and composite ELO ratings offers a compelling narrative of the shifting hierarchy within the league that cannot be captured by simpler metrics like win rate%, since they signal the qualitative strength that characterise the teams' profiles.

Table IV.VII : Elo correlations with empirical win probability

Feature	Correlation	P-value	Significance
Elo_derived win_probability	0.418	2.27E-42	***
composite_rating	0.299	1.51E-21	***
offensive_rating	0.229	5.17E-13	***
opp_composite_rating	0.299	1.51E-21	***
opp_offensive_rating	0.229	5.17E-13	***
defensive_rating	0.163	3.34E-07	***
opp_defensive_rating	0.163	3.34E-07	***
off_def_matchup	0.042	1.88E-01	ns
def_off_matchup	-0.042	1.88E-01	ns

Here the significance of the pre-game Elo ratings are measured for their correlation with the empirical win probability at the end of the game. The Win_probability feature is the softmax output for a model trained on the rest of the highlighted features. This, alongside the composite, offensive, and opp_defensive features show the strongest correlation (0.418. 0.299, 0.229, and 0.299) with win probability. None of these features have a correlation above 0.5 which strengthens the case for the necessity of in-game feature consideration for accurate prediction.

Additionally, the specific matchup features have non-significant correlations with the outcome variable. The presumption that underlines their inclusion is that they inform the win probability feature with added context, as their ablation reduces the correlation of the feature from 0.418 to 0.378.

IV.V Feature Limitations

One important consideration is that models built on the four factors work best to evaluate teams, particularly on a macro scale, as opposed to predicting individual results. Even more so important, is the fact that a team is simply the aggregate of individuals. By using counting stats I ignore who those statistics are the product of, which players were allocated which minutes, whether there were individuals missing from either team, etc. Teams are in general more liberal with their minute allocation in the regular season compared to the playoffs, simply because these minutes mean less, this expression is colloquialised as “shortening the rotation”.

This introduces the most significant weakness of any team-based modelling approach: Basketball teams are iteratively adaptive networks of strategists, and changes can occur over a wide variance of time horizons. Injuries are the most immediate, often impacting a team’s ability to field a competitive roster, especially in games against the best opponents.

Another caveat of all models built on counting stats is that the box score does not fairly distribute defensive impact. Steals, blocks, and rebounds are the best proxies available for defensive impact, but these are noisy and subject to bias.

V.I Methodology

This section describes the models and methodology used to predict NBA game results with the feature sets described in the previous section. The intent here is to benchmark different Machine-Learning algorithms and feature sets to infer in which contexts different insights are valuable in predicting game results.

The models primarily use information as of the third quarter of the game to predict the final result. This validation method has not been previously analysed to a great extent and as such the results hopefully represent the state of the art. For those models that use pre-game prediction, the baseline is Vegas pre-game odds which stand at approximately 68% correct predictions.

For all games, the models are cross validated in five sections using time series cross-validation, meaning t=1 is the training validation by t=2 and so on. The results presented are trained on the t=1,2 (512 games) and validated on t=3,4,5 (768 games). The goal is to maximise the accuracy and AUC of the models. The results of the models trained on each of these time series folds are included in Appendix figure [V.I]

V.II LightGBM

The predictive modeling in this thesis explored a variety of machine learning algorithms, including Naive Bayes, Support Vector Machines, various regression techniques, an extensive set of tree-based methods, and transformer networks. Among these, ensemble methods based on gradient boosting demonstrated strong performance.

Gradient Boosting is a powerful machine learning technique that builds predictive models in an additive, stage-wise fashion. It constructs an ensemble of weak learners, typically decision trees, where each new tree attempts to correct the errors (residuals) of the ensemble of trees trained so far. By iteratively fitting new models to the negative gradient of the loss function with respect to the predictions of the current ensemble, gradient boosting algorithms can achieve high predictive accuracy on complex, non-linear datasets.

For this research, LightGBM (Light Gradient Boosting Machine) was selected as the primary gradient boosting framework and serves as a robust baseline model. LightGBM is a highly efficient and scalable implementation of Gradient Boosted Decision Trees (GBDTs) developed by Microsoft (Ke et al., 2017). Its selection was informed by initial cross-validation experiments which indicated its performance was competitive with other ensemble methods tested (as detailed in Section VI.III], while offering significant advantages in training speed and memory usage, crucial for iterative experimentation and hyperparameter tuning. The parameters used for the LightGBM models are specified in Appendix [V.I] and were optimized via grid search to maximize the Area Under the ROC Curve (AUC) for win probability predictions.

Traditional GBDT algorithms weigh all data instances equally when calculating information gain for splits. However, instances with larger gradients (i.e., those that are currently poorly predicted by the ensemble) contribute more significantly to the learning process. GOSS intelligently addresses this by retaining all instances with large gradients while performing random sampling on instances with small gradients. This approach allows the model to focus more on the harder-to-learn examples without completely discarding the information from 'easier' well-predicted instances, thereby maintaining accuracy while significantly reducing the number of data instances used for training each tree.

In the context of NBA game prediction, GOSS is advantageous because it allows the model to more effectively learn from challenging or surprising game scenarios (e.g., unexpected upsets, or games where standard features might be misleading) which carry larger prediction errors. By focusing on these informative instances, the model can refine its decision boundaries for these difficult cases while still efficiently processing more straightforward matchups, ultimately improving overall predictive power.

Finding the optimal split points for continuous features can be computationally intensive in traditional GBDTs, as they require iterating through all unique values. LightGBM employs a histogram-based algorithm, which discretizes continuous feature values into a fixed number of bins. Instead of evaluating every unique feature value as a potential split point, LightGBM considers only the discrete bin boundaries. This significantly reduces the computational cost of

finding splits, especially for datasets with many continuous features or a large number of instances.

This is particularly beneficial for our feature set, which includes continuous features like 'point differential at Q3,' 'ELO rating difference (rating_diff),' and 'cumulative net rating (cum_nrtg).' These features can have a wide range of unique values. The histogram-based approach allows LightGBM to efficiently identify optimal 'cut-off points' within these features without the prohibitive cost of examining each individual differential or rating. This not only speeds up training but can also act as a form of regularization, preventing overfitting to noise in continuous feature values.

Unlike many GBDT implementations that grow trees level-wise (expanding all nodes at the current depth before moving to the next), LightGBM typically employs a leaf-wise (or best-first) growth strategy. It splits the leaf node that is expected to yield the largest reduction in the loss function. While this can lead to deeper, more complex individual trees and potentially overfit on smaller datasets if not regularized, it often allows the model to converge to a better solution more quickly by focusing on the most promising splits.

For NBA predictions, this means it can quickly hone in on specific combinations of feature values that are highly predictive. For example, it might rapidly develop a deep path in a tree that identifies that a team with a moderate ELO disadvantage (rating_diff < -100) but a surprisingly high cum_nrtg (Q1-Q3) and playing at home (is_home=1) still has a decent chance of winning, capturing a potential "upset brewing" scenario that a level-wise tree might take longer to model effectively or might miss if depth is too constrained.

For pregame prediction the LightGBM was trained on three distinct feature sets. The first two are the ELO and Momentum feature sets individually, and the third is their aggregate. The Elo-Only model is above the lowest baseline of predicting the home team to win (54%) but below the expected Vegas accuracy (68.5%).

Table V.I. Final Average Performance Metrics Across All Cross-Validation Folds (Pre-Game)

Metric	Elo-Only	Momentum	Elo + Momentum
Test Accuracy	0.6371	0.711	0.6757
Test AUC-ROC	0.7001	0.7718	0.7351
Test Precision	0.6601	0.7488	0.688
Test Recall	0.6855	0.7125	0.7455
Test F1	0.6711	0.7273	0.715

Full Feature sets for all specified models are included in Appendix Table V.II

The Momentum feature set, when used exclusively, yielded the highest predictive accuracy (0.711) and AUC-ROC (0.7718) among the three pre-game LightGBM configurations. This model also demonstrated a good balance between precision (0.7488) and recall (0.7125), resulting in the highest F1-score (0.7273). This suggests that short-term team performance dynamics, as captured by the momentum features, are highly informative for pre-game win probability prediction using the LightGBM framework.

V.III Ensemble

While the LightGBM model (Section VI.II) provided a strong baseline, this research further explored an ensemble approach to potentially enhance predictive accuracy, improve model robustness, and leverage the diverse strengths of multiple learning algorithms. The primary goal of this ensemble is to achieve a more generalized and resilient prediction by combining the outputs of several high-performing models, thereby reducing the risk of overfitting to the idiosyncrasies of any single model and capturing a broader range of predictive patterns within the data.

For the construction of this ensemble, the quarter indexed, cumulative, and ELO + Momentum feature sets were combined: The feature-set is henceforth referred to as the "Full Model" feature set. This comprehensive dataset, as detailed in Section [IV. Feature Engineering], incorporates pre-game ELO and momentum metrics, team-level statistics, and crucial in-game dynamic features such as q3_score_diff and cum_nrtg. This feature set was selected based on preliminary individual model evaluations which indicated its superior predictive performance (as alluded to in Section VI.II)

The ensemble comprises five distinct base models, selected for their proven efficacy in classification tasks and their varied approaches to learning from complex data:

LightGBM, as detailed in the previous section, with its leaf-wise tree growth and GOSS, contributes its efficiency and ability to model complex interactions quickly.

LightGBM, XGBoost, Catboost, Random Forests, Extra Trees.

XGBoost (Chen & Guestrin, 2016) is another highly popular and powerful open-source implementation of Gradient Boosted Decision Trees. Like LightGBM, it is designed for speed and performance. However, it traditionally utilized a level-wise tree growth strategy (though newer versions offer options) and incorporates sophisticated regularization techniques (L1 and L2) directly into its objective function.". XGBoost was included in the ensemble for its robust regularization capabilities, which are particularly beneficial for mitigating overfitting when working with high-dimensional or densely informative feature sets like our 'Full Model' set. Its different default tree structure and regularization approach compared to LightGBM can lead it to learn different aspects of the feature space, contributing to the ensemble's diversity.

CatBoost (Prokhorenkova et al., 2018) is a GBDT algorithm specifically designed to excel with datasets containing categorical features. A key innovation in CatBoost is its use of ordered boosting, a permutation-driven approach to training on different data shuffles, and an efficient method for converting categorical features to numerical representations using target statistics while avoiding target leakage. This typically involves calculating statistics like the average target value for each category, often combined with smoothing techniques.

Its inclusion is motivated by the presence of categorical variables in our dataset such as team_id, opponent_id, time_between_games, and game_number . CatBoost's specialized handling of these features is intended to extract predictive signals that might be less effectively captured by other GBDT implementations that require manual pre-processing (e.g., one-hot encoding, which can

lead to sparsity) or simpler categorical handling. This offers a distinct perspective on these features, enhancing ensemble diversity.

For aggregating the predictions from these five base models, a soft voting (also known as averaging probabilities) strategy was employed. Each of the five models was independently trained on the 'Full Model' feature set using the optimized hyperparameters determined during their individual validation phases. For a given game instance in the test set:

1. Each base model outputs a predicted probability of the home team winning.
2. These five probabilities are then averaged to produce the ensemble's final win probability for the home team.
3. For binary classification, this averaged probability is then converted to a class label using a standard threshold of 0.54.

The rationale for combining these specific models initially aimed to leverage their potential diversity. Gradient boosting models (LightGBM, XGBoost, CatBoost) employ different tree growth strategies, distinct regularization techniques, and varied approaches to feature handling. Bagging models (Random Forests, Extra Trees) offer a different learning paradigm.

The ensemble, using the soft voting aggregation method, achieved a test accuracy of 0.8259 and an AUC-ROC of 0.9119. While the ensemble's accuracy is comparable to the best individual models (XGBoost and Random Forest), its AUC-ROC is slightly higher than XGBoost's and very close to CatBoost's leading individual AUC. This suggests that the aggregation of probabilities helped in refining the ranking of predictions, leading to better discrimination between classes, even if the direct accuracy improvement was modest. The ensemble benefited from the consensus, correcting some individual model errors. The overall strong performance of the ensemble, particularly its high AUC, indicates its effectiveness in combining the strengths of its diverse components.

This heterogeneity was hypothesized to result in base models making different types of errors. However, an analysis of the prediction correlations between the base models on the test set revealed a high degree of agreement

Table V.II: Prediction Correlations for Ensemble

Model	LGB	XGB	CAT	RF	ET
LGB	1	0.972	0.991	0.977	0.979
XGB	0.972	1	0.916	0.913	0.984
CAT	0.911	0.916	1	0.949	0.911
RF	0.977	0.913	0.949	1	0.994
ET	0.979	0.984	0.911	0.994	1

As shown in Table X, the predicted probabilities from the base models are very highly correlated, with pairwise correlations exceeding 0.9, particularly among the gradient boosting variants, and also remaining very high with the bagging models. This indicates that, while architecturally different, the models learned to interpret the strong signals within the 'Full Model' feature set in a largely similar manner. This is likely attributable to the dominance of highly predictive in-game features such as q3_score_diff and cum_nrtg which guide capable models towards similar decision boundaries.

While these high correlations suggest that the benefit from traditional 'error-correction diversity' might be limited, ensembling highly accurate (and highly correlated) models can still offer advantages. Averaging outputs can reduce the variance of the final prediction, making it less sensitive to the specific stochastic elements of any single model's training process or minor variations in the data.

Table V.III Test accuracies for ensemble models and aggregate

Model	LightGBM	XGBoost	CatBoost	RF	Extra Trees	Ensemble
Test AUC-ROC	0.9066	0.9087	0.915	0.9003	0.8942	0.9119
Test Accuracy	0.8128	0.8313	0.8189	0.8251	0.8107	0.8259

Even a small improvement in metrics like AUC-ROC, as observed in Table [V.II], can be valuable. The ensemble may be smoothing out minute imperfections or biases present in individual models. The consensus prediction is arguably more robust than relying on a single best model, which might have performed optimally on this specific test set due to chance.

Therefore, despite the high inter-model correlation, the ensemble was retained as the final predictive approach due to its empirically demonstrated slightly superior performance.

V.IV Gated Recurrent Unit

Recognizing that NBA games are inherently sequential processes, where events and team performance in one period can significantly influence subsequent periods and the ultimate outcome, this research explored the application of Recurrent Neural Networks (RNNs). Unlike traditional machine learning models that operate on flattened feature vectors (as discussed for LightGBM in Section VI.II and the ensemble in VI.III), RNNs are specifically designed to process sequences of data, making them well-suited for capturing temporal dynamics.

A standard RNN processes a sequence by maintaining a hidden state that is updated at each time step, incorporating information from the current input and the previous hidden state. This allows the network to, in theory, capture dependencies across the sequence. However, vanilla RNNs face challenges when learning long-range dependencies due to the vanishing gradient problem. During backpropagation through time, gradients can shrink exponentially as they are propagated back through many layers (time steps). This makes it difficult for the network to learn the influence of

early inputs on later outputs, as the gradient signal becomes too weak to effectively update the network weights for those earlier steps.

To address these limitations, more sophisticated RNN architectures like the Gated Recurrent Unit (GRU) (Cho et al., 2014) were developed. Conceptually, GRUs enhance the vanilla RNN by incorporating gating mechanisms – specifically, an update gate and a reset gate.

- The **reset gate** determines how much of the previous hidden state should be forgotten. This allows the model to discard irrelevant information from the past.
- The **update gate** decides how much of the new candidate hidden state (computed similarly to a vanilla RNN) should be combined with the previous hidden state to form the next hidden state. This gate effectively controls the flow of information, allowing the network to selectively retain important information across longer sequences and thus better mitigate the vanishing gradient problem by maintaining a more stable gradient flow.

These gates, which are essentially small neural networks with sigmoid activations, learn to control the information flow within the recurrent unit, enabling GRUs to capture longer-term dependencies more effectively than vanilla RNNs.

GRUs were chosen over LSTMs due to their comparable performance on many tasks with a slightly simpler architecture and fewer parameters, potentially leading to faster training and less risk of overfitting on datasets of this scale.

For this study, each game is represented as a sequence of three feature vectors, corresponding to the cumulative statistics and performance metrics at the end of Q1, Q2, and Q3 respectively. Each vector (time step) encapsulates a comprehensive statistical profile of the game up to that quarter's end, including metrics such as effective Field Goal Percentage (eFG%), Net Rating (NRTG), and Turnover Percentage (TOV%) for both the team and its opponent, totaling [22] distinct features per time step. Pre-game features were used to initialize the GRU's hidden state. This sequential representation allows the GRU to learn the evolving narrative of the game.

To capture increasingly abstract temporal patterns, a **stacked GRU architecture** with 2 layers was implemented. In this configuration:

1. The first GRU layer processes the input sequence of quarterly feature vectors. Its output at each time step is a sequence of hidden states representing learned features from the raw input.
2. This entire sequence of hidden states from the first GRU layer is then passed as input to the second GRU layer.

The second GRU layer can thus learn higher-order temporal dependencies and more complex relationships between the patterns identified by the first layer across the game's progression.

This hierarchical processing allows the network to build a richer understanding of the game's dynamics. To mitigate overfitting, which can be a concern with deeper networks, dropout is applied to the outputs of each GRU layer except the last, as handled by the Neural Network.

Upon processing the full input sequence (up to Q3), the final hidden state of the last GRU layer is taken as the learned game_embedding. This embedding, which encapsulates the relevant sequential information, is then passed through a feed-forward network comprising a linear layer reducing dimensionality, followed by a ReLU activation, Batch Normalization, and another dropout layer . Finally, an output linear layer transforms this into a single logit, which is passed through a sigmoid activation function to yield the predicted win probability for the home team.

Table IV.IV Final Average Performance Metrics Across All Cross-Validation Folds (Q1-Q3)

Metric	Cumulative	Quarter Indexed	Full Model	Ensemble	GRU
Test Accuracy	0.754	0.8009	0.7977	0.8259	0.8369
Test AUC-ROC	0.8314	0.871	0.8867	0.9119	0.9157
Test Precision	0.7739	0.8217	0.8034	0.8465	0.8666
Test Recall	0.773	0.8093	0.8314	0.8309	0.8253
Test F1	0.7734	0.8152	0.7734	0.817	0.8443

Table [IV.IV] presents the final average performance metrics for various models when predicting outcomes using data available up to the end of the third quarter. The GRU model, designed to capture temporal dependencies, achieved the highest performance across all key metrics, with a test accuracy of 0.8369 and an AUC-ROC of 0.9157. This represents a slight improvement over the strong performance of the ensemble model (Accuracy: 0.8259, AUC: 0.9119). Notably, the GRU model also demonstrated superior precision (0.8666) compared to the ensemble (0.8465), while maintaining a competitive recall (GRU: 0.8253, Ensemble: 0.8309), resulting in the highest F1-score (0.8443).

VI. Discussion

The Elo-only model, while outperforming a naive home-team-wins baseline (54%) with its 63.7% accuracy, proved to be the weakest of the pre-game approaches and fell short of Las Vegas betting lines (68.5%). Elo rating describes the cumulative macro state of a team's offense and defense but struggles to capture the intricacies of uncertainty, matchup viability, and, crucially, recent variations in performance beyond opponent strength. The aggregative nature of Elo, even with the novel quarterly updates, likely smooths out these vital short-term dynamics. A potential issue is the inconsistency in the magnitude of Elo ratings across a season; a team might reach 1600 Elo early on but progress to 2200 later, and while time series cross-validation partially addresses this, it may still introduce noise.

Furthermore, the equal weighting of offensive and defensive Elo components into a composite score might mask situations where a specific imbalance (elite offense vs. mediocre defense) is a stronger predictor against certain opponent archetypes than the composite ELO difference suggests. This limitation is particularly evident in the model's reduced accuracy (58%) when predicting games between heavy favorites and underdogs, as illustrated by Figure VI.I, where large ELO differences might overshadow other subtle factors or where the K-factor isn't dynamic enough to capture true upset potential.

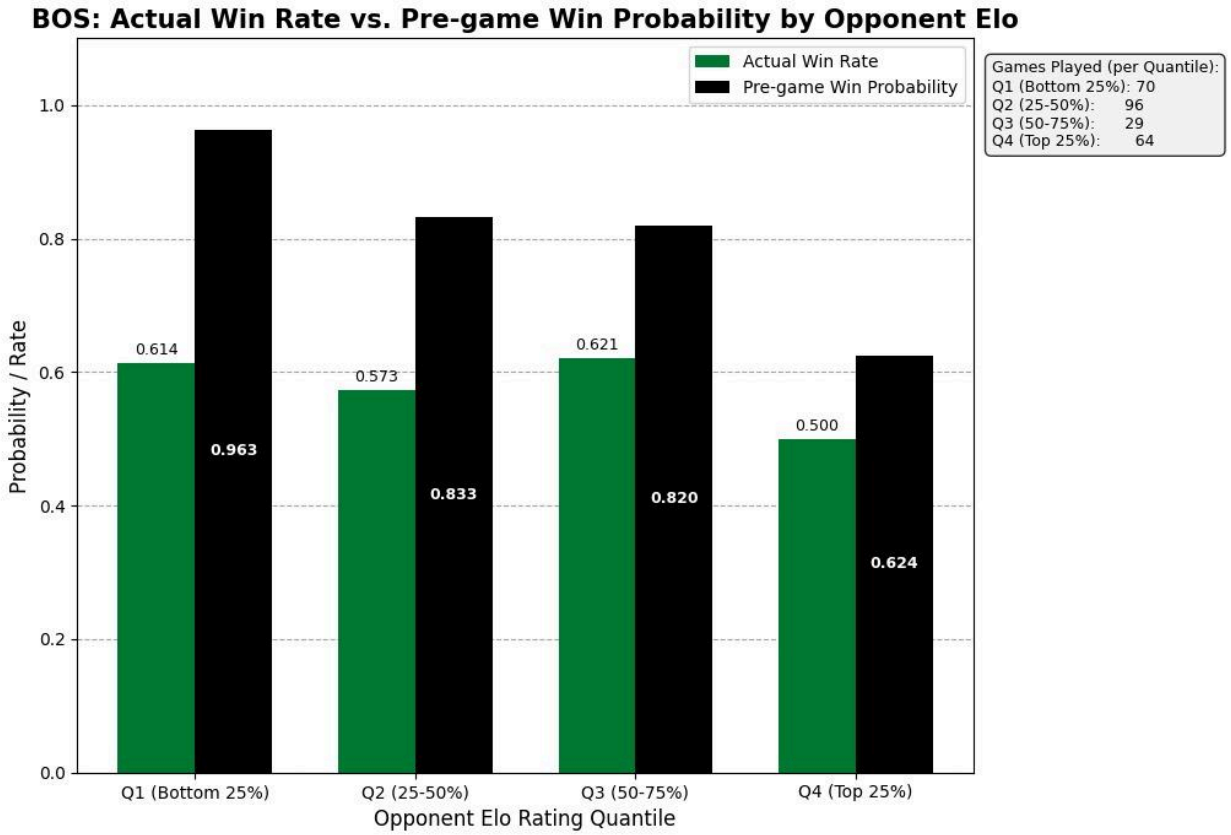

Figure VI.I Pre-game Win probability and Empirical win rate by opponent Elo Boston (t = 1)

There is larger variance between teams in average points scored *(off)* than in average points *(def)* conceded. And scoring volume is also greater among the best teams than the defensive equivalent.

These two factors mean that the composite rating is skewed towards offensively successful teams, which downweights the tangible importance of defensive solidity for winning. This is one possible reason for the previously discussed gap between empirical win percentage, particularly for teams that are offensively leaning.

This dissonance speaks to the variety of ways teams can create winning advantages. Boston has a tendency to outscore opponents in large margins leading to large ELO gains, only to lose the leads due to natural variance in shooting.

When incorporating third-quarter context, model accuracy increased dramatically, as expected. The GRU model achieved the highest performance at 83.7% (averaged across the final three folds), yet this still leaves a significant margin for unpredictability.

This "unpredictability" likely arises from specific fourth-quarter phenomena that aggregate statistics struggle to anticipate. These could include individual players exhibiting clutch shooting or defensive plays that deviate significantly from their game/season averages, radical defensive schemes or offensive set changes in the fourth quarter not foreshadowed by Q1-Q3 patterns, unquantified fatigue differentials, or psychological momentum shifts triggered by singular high-impact plays.

Table VI.II: Win rate by closing margin

Subgroup Definition	Games	Actual Win Rate	Accuracy	ROC AUC
Close Games (Margin ≤ 12)	222	0.54	0.6622	0.6985
Blowout Games (Margin > 12)	563	0.54	0.8899	0.9613

The model's notably lower accuracy of 66% in "Close Games" (margin ≤ 12 points), as shown in Table VI.II, strongly suggests these unmodeled factors are amplified when the score is tight, as small deviations in performance have an outsized impact on the outcome.

Nevertheless, the subgroup analysis outperforms a common heuristic used in betting, stating that late in games if the points differential is greater than the number of minutes left, the leading team wins 90% of the time. When applied to the end of the third quarter, previous literature predicts 80% of results correctly, a mark we beat here by an approximate 9% margin.

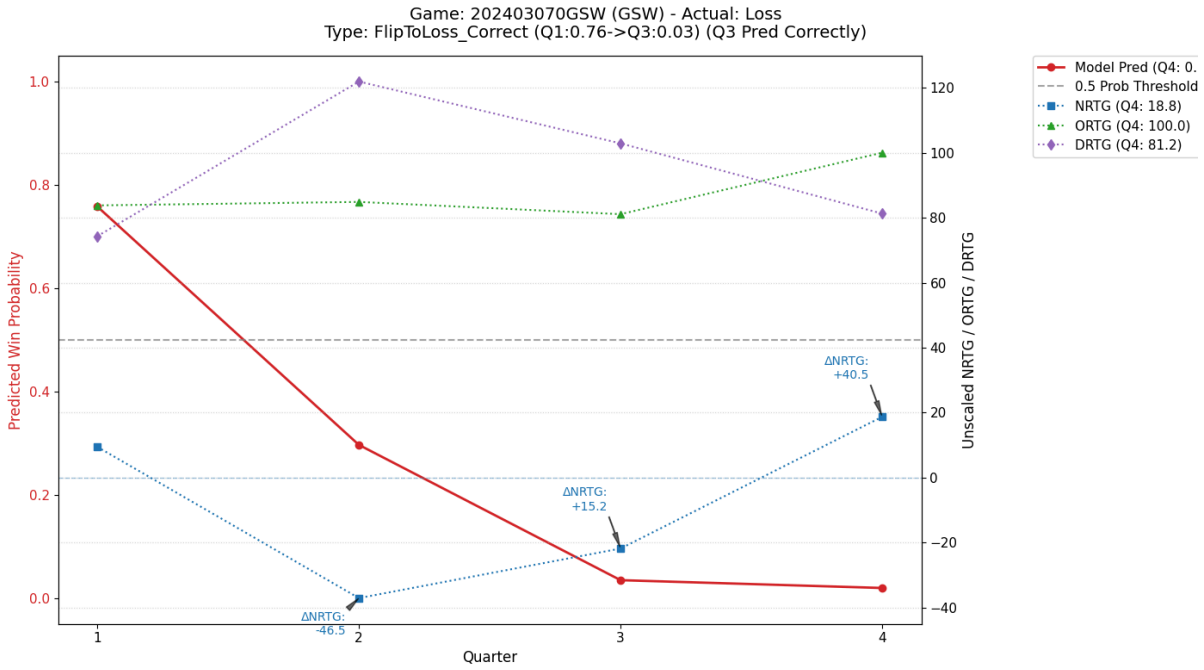

Figure VI.II Progression of win prediction throughout the game (Correct, significant change).

To illustrate the models ability to adapt expectation, an example is here included of a case where the pre-game prediction was flipped as of the end of the third quarter. The prior game prediction was a 76% win probability, but due to a significant loss signal in the second quarter (-46.5 NRTG), the prediction falls to 32%. Since the model learns cumulative NRTG, the positive signal in the third quarter (+15.2 NRTG) still results in a negative cum_nrtg, meaning the model still predicts the team to lose.

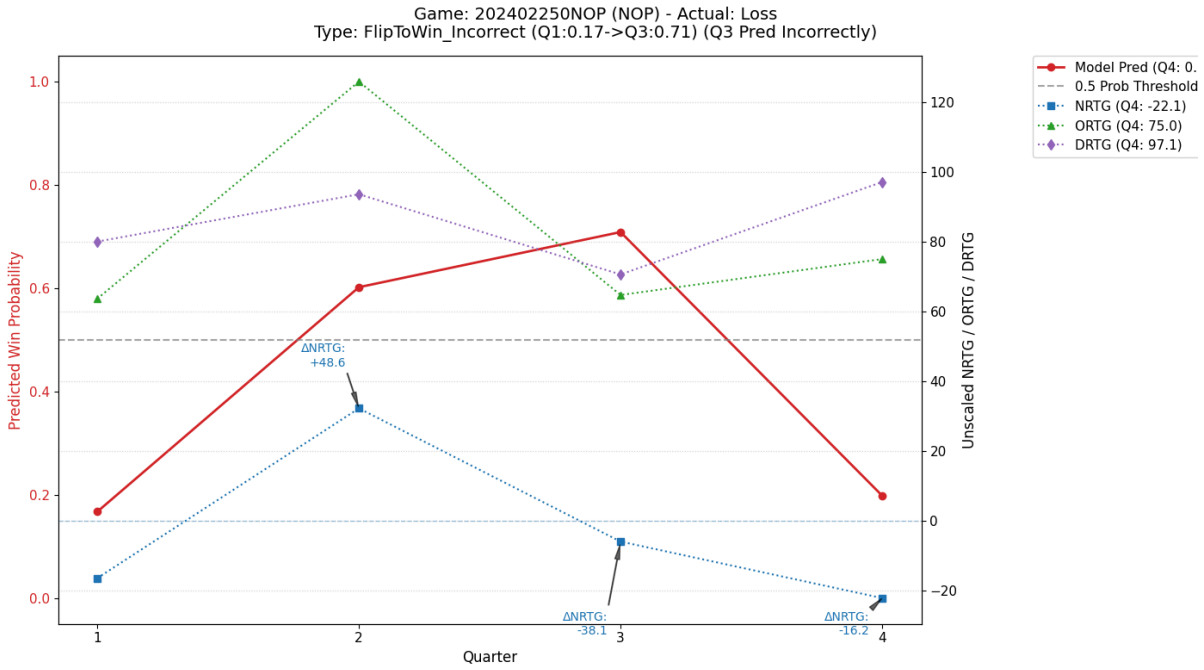

Figure VI.III Progression of win prediction throughout a game (Incorrect)

However, the model still requires an ample signal as of the third quarter to make an accurate prediction. A counter-example here is provided where the cumulative NRTG as of the third quarter is still positive but with high variability throughout the game, which results in the model following the general trend and fails to predict the break, where the team underperformed in the fourth quarter and lost the game.

This in turn suggests that the model could gain from some mean reversion expectation that adjusts for games with high variability of NRTG, potentially targeting such for teams that started with low win probability as of the pre-game prediction.

The quarter indexed models show strong convergence performance, with the final accuracy being reached after the first two time series folds, and the result after the first time fold also showcasing an AUC above 0.9. A rolling model was also trained to investigate convergence in more detail, the results of which are displayed below.

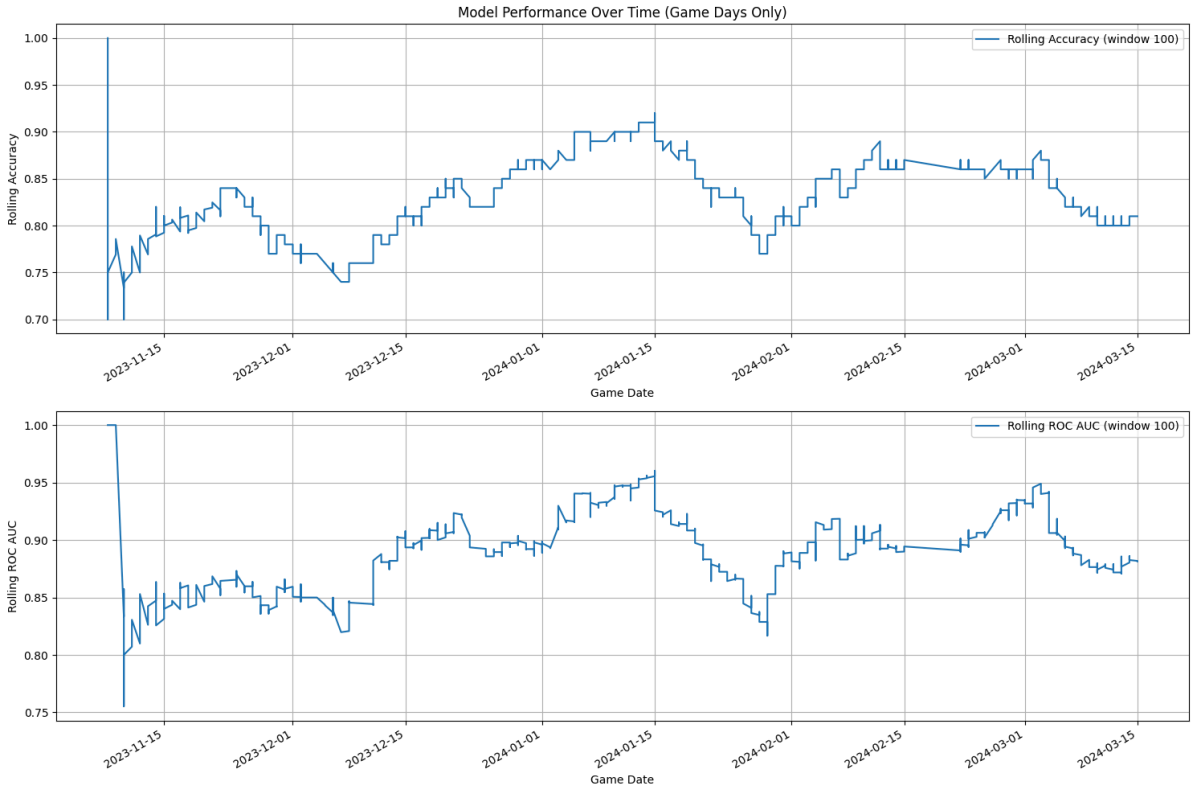

***Figure VI.IV** GRU game-by-game training Model performance over time*

The iterative improvement of the GRU model over time is showcased here, with the structural break after the peak 90% accuracy to be attributable to the *Trade Deadline* at which point a large number of teams made significant changes to their rosters, and as such the learned expectations upon specific teams must be relearned, especially with reference to the Elo ratings, which are not quickly adaptive.

Regardless, there is significant variance to the feature's predictive power throughout the season in a manner which is nonlinear. This suggests as expected that there is significant unexplained variance in the dataset, attributable to the aggregation mechanism.

Overall, a key finding is the significance of the diverse feature set, particularly the extended Elo features within the "Momentum" pre-game model, which achieved 71.1% accuracy, exceeding Las Vegas betting odds. The GRU method provides a robust framework for implementing added context throughout a game, its value justified by the accuracy increase over the ensemble method and its ability to correct initially flawed pre-game judgments. The superior performance of the GRU over the ensemble, albeit slight, hints that it extracts additional value from the sequence of quarterly metrics, perhaps identifying performance trajectories like consistent improvement versus erratic play, which a flattened feature set might obscure.

VII. Future Work

The implication for future work is guided by the efficiency of the GRU model to predict game results better than a flattened vector baseline. This suggests that models which include longer term dependencies, for example learning from the context of the full season as one sequence, are feasible and may provide better team-specific dependencies. These models may be able to make better predictions for the most difficult use cases.

To enhance the foundational Elo system, future work could develop a player-based Elo component that aggregates to team Elo. This would allow for more granular strength updates, particularly when individual players are injured or traded, potentially initializing player Elos from advanced metrics like Box Plus-Minus. Complementing this, implementing a dynamic, event-driven K-factor for Elo updates,(one that increases sensitivity after significant roster changes or during pronounced winning/losing streaks) could allow for faster recalibration of team strength.

As stated in section VI, there is ample room for new feature creation addressing the dependencies between quarter features. To better capture the unique dynamics of late-game situations and address the model's struggles with high NRTG variability, future research should focus on engineering features specifically reflecting fourth-quarter volatility and clutch tendencies. This could include metrics like a team's historical variance in fourth-quarter net rating, their frequency of comebacks or squandered leads (particularly from disadvantaged Q3 scenarios), or even specialized free-throw percentages in clutch moments. Exploring regime-switching models or developing separate predictive models specifically for 'close game' scenarios, trained on features indicative of late-game performance, could also prove fruitful.

Within sequential modeling, the GRU architecture can be further refined. Incorporating attention mechanisms could allow the model to dynamically weigh the importance of different quarters (e.g., paying more attention to a Q3 surge if Q1 and Q2 were lackluster) or specific features within each quarter when making its final prediction. Experimenting with longer input sequences, such as data from several preceding games for each team, might capture inter-game momentum or fatigue patterns beyond current ELO-derived features. To specifically address events like the trade deadline, explicitly introducing a 'roster change magnitude' feature or mechanisms to temporarily

increase the GRU's learning rate or reset parts of its hidden state for affected teams could improve adaptability.

Expanding the feature set to include more external contextual data is another promising direction. This includes explicit travel and rest features like days since last game, back-to-back game indicators, travel distance, and time-zone changes, which could directly inform the model or modulate Elo ratings. If ethically and practically feasible, proxies for player availability and injury impact, such as a binary 'star player missing' feature or the percentage of team usage missing, would add significant context. Interaction with betting markets also offers rich possibilities. Using the Las Vegas betting line itself as an input feature would allow models to learn from and potentially correct the market's aggregated wisdom. Beyond predicting outright wins, developing models to predict the probability of 'beating the spread' would have direct practical applications.

Finally, a more systematic application of explainable AI (XAI) techniques, such as SHAP or LIME, particularly to the GRU and ensemble models for misclassified upsets or blown leads, is crucial. This would help identify whether model failures stem from specific feature interactions or if crucial information for these hard-to-predict scenarios is fundamentally missing from the current feature set, thereby guiding the next iteration of feature engineering and model development.

VIII. Bibliography

Attard, P., Suda, D., & Sammut, F. (2016). Bayesian Hierarchical Modelling of Basketball Team Performance: An NBA Regular Season Case Study.

Cai, W., Yu, D., Wu, Z., Du, X., & Zhou, T. (2019). A hybrid ensemble learning framework for basketball outcomes prediction. *Physica A: Statistical Mechanics and its Applications, 528*, 121461. https://doi.org/10.1016/j.physa.2019.121461

Cao, C. (2012). Sports Data Mining Technology Used in Basketball Outcome Prediction. *International Journal of Computer Science and Network Security, 12*(2), 116–120.

Chen, W.-J., Jhou, M.-J., Lee, T.-S., & Lu, C.-J. (2021). Hybrid basketball game outcome prediction model by integrating data mining methods for the national basketball association. *Entropy, 23*(4), 477. https://doi.org/10.3390/e23040477

C. Sukumaran, D. Selvam, M. Sankar, V. Parthiban, and C. Sugumar. 2022.Application of artificial intelligence and machine learning to predict basketball match outcomes: A systematic review.Computer Integrated Manufacturing Systems, 28:998–1009.

Cheng, G., Zhang, Z., Kyebambe, M., & Nasser, K. (2016). Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle. *Entropy, 18*(12), 450. https://doi.org/10.3390/e18120450

Oliver, D. (2004). *Basketball on Paper: Rules and Tools for Performance Analysis*. Potomac Books, Inc."

Dubbs, A. (2018). Statistics-free sports prediction. *Model Assisted Statistics and Applications, 13*(2), 173-181.

ESPN Analytics. (n.d.). *Basketball Power Index (BPI) Predictions*. ESPN.

Ferrara, Joe. "Predicting NBA Games." *GITHUB*, 4 May 2020, joe-ferrara.github.io/2020/05/04/basketball.html#:~:text=How%20good%20is%20Vegas%20at,predicting%20NBA%20games

Goldsberry, K., & Weiss, E. (2013). The Dwight Howard Paradox: A New Metric for Valuing Rebound Prowess. In *Proceedings of the MIT Sloan Sports Analytics Conference*.

Hobbs, W., Wu, P., Gorman, A., Mooney, M., & Freeston, J. (2020). Bayesian hierarchical modelling of basketball tracking data - a case study of spatial entropy and spatial effectiveness. *Journal of Sports Sciences, 38*(11-12), 1-11. https://doi.org/10.1080/02640414.2020.1736252

Horvat, Tomislav & Job, Josip. (2020). The use of machine learning in sport outcome prediction: A review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 10. e1380. 10.1002/widm.1380.

Jain, S., & Kaur, H. (2017). Machine learning approaches to predict basketball game outcome. In *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)* (pp. 1–7). IEEE.

Jones, E.S. (2016). *Predicting outcomes of NBA basketball games*. Ph.D. thesis, North Dakota State University.

Kannan, A., Kolovich, B., Lawrence, B., & Rafiqi, S. (2018). Predicting national basketball association success: A machine learning approach. *SMU Data Science Review, 1*(3), 7.

Khanmohammadi, R., Saba-Sadiya, S., Esfandiarpour, S., Alhanai, T., & Ghassemi, M. (2024). MambaNet: A Hybrid Neural Network for Predicting the NBA Playoffs. *SN Computer Science, 5*, 628. https://doi.org/10.1007/s42979-024-02977-0

Locker, J., & Parpart, P. (2018). Predicting NBA Game Outcomes Using Machine Learning: A Study on Feature Importance and Model Performance. In *International Conference on Data Mining Workshops (ICDMW)* (pp. 714-721).

Leicht AS, Gómez MA, Woods CT. Explaining match outcome during the men's basketball tournament at the Olympic Games. J Sport Sci Med. 2017;16(4):468–473. https://www.jssm.org/jssm-16-468.xml%3EFulltext# pmid:29238245

Loeffelholz, B., Bednar, E., & Hwsalek, K. (2009). *Predicting NBA Games Using Neural Networks*. University of Nebraska-Lincoln Department of Computer Science and Engineering Senior Design Project.

Manner, H. (2016). Modeling and forecasting the outcomes of nba basketball games. *Journal of Quantitative Analysis in Sports, 12*(1), 31–41.

Miljković, D., Gajić, L., Kovačević, A., & Konjović, Z. (2017). A Comparative Analysis of Machine Learning Techniques for NBA Match Outcome Prediction. *Journal of Sports Analytics, 3*(4), 283-295.

Osken, Cem & Onay, Ceylan. (2022). Predicting the winning team in basketball: A novel approach. Heliyon. 8. e12189. 10.1016/j.heliyon.2022.e12189.

Ouyang, Y., Li, X., Zhou, W., Hong, W., Zheng, W., et al. (2024). Integration of machine learning XGBoost and SHAP models for NBA game outcome prediction and quantitative analysis methodology. *PLOS ONE, 19*(7), e0307478. https://doi.org/10.1371/journal.pone.0307478

Pai, Ping-Feng & ChangLiao, Lan-Hung & Lin, Kuo-Ping. (2017). Analyzing basketball games by a support vector machines with decision tree model. Neural Computing and Applications. 28. 10.1007/s00521-016-2321-9.

Pelechrinis, K., & Winston, W. L. (2019). In-Play Prediction of NBA Game Outcomes Using High-Frequency Data. *Journal of Quantitative Analysis in Sports, 15*(3), 185-200.

Puranmalka, K. (2014). *Modelling the NBA to make better predictions*. MIT Thesis. Retrieved from https://dspace.mit.edu/bitstream/handle/1721.1/85464/870969496-MIT.pdf?sequence=2&isAllowed=y

Tian, X., Gao, Y., & Shi, J. (2020). Modeling basketball games by inverse Gaussian processes. *Communications in Statistics - Simulation and Computation, 51*(11), 6246–6256. https://doi.org/10.1080/03610918.2020.1798461

Tsagris, M., Adam, C., & Pantatosakis, P. (2024). On predicting an NBA game outcome from half-time statistics. *Discovery Artificial Intelligence, 4*, 111. https://doi.org/10.1007/s44163-024-00201-9

Zhao, K., Du, C., & Tan, G. (2023). Enhancing basketball game outcome prediction through fused graph convolutional networks and random forest algorithm. *Entropy, 25*(5), 765. https://doi.org/10.3390/e25050765

IX. Appendix

Accounting for generative AI use

AI Tools Used:

- Gemini Google AI Studio
- Cursor
- ChatGPT

Cursor was used throughout the process for programming applications, primarily for implementing data validation checks, creating visualisations, and some usage for core functionality. All code was controlled throughout to maintain coherence and for personal Understanding. Gemini was used to understand theoretical concepts throughout the writing of the thesis, as well as for some text formatting and narrative construction, but all text in the final submission is written by me. The "Deep Research" function on ChatGPT's interface was used for preliminary research and to find a wealth of sources, but all inclusion of those sources was then researched independently and written into the previous literature section. ChatGPT was also used for early ideation on matching structures, for example for differentiation between different RNNs and their aptitude for the specific application to the thesis.

IX.A Introduction

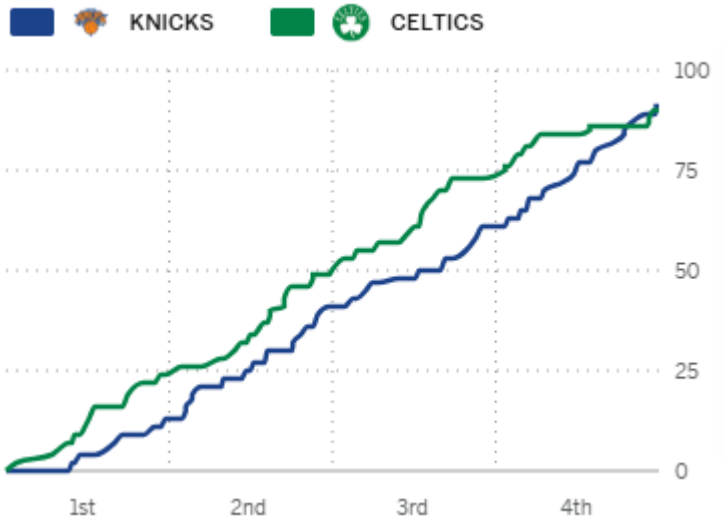

Figure I.I. Knicks Celtics Game 1 Score Progression
Source: "Knicks 91-90 Celtics (May 7, 2025) Final Score." *ESPN*, ESPN Internet Ventures, 14 May 2025, www.espn.com/nba/game/_/gameId/401769747/knicks-celtics.

Table I.I: NBA Box-Score Data features

Feature	Description	Feature	Description
MP	Minutes played in the game	FG	Field goals made in the game
FGA	Shots taken in open play	3P	3-point field goals made in the game
3PA	Shots attempted from behind the 3-PT line	FT	Free throws made in the game
FTA	Free throws attempted in the game	ORB	Times the ball was caught from a teammate's miss

DRB	Times the ball was caught after an opponent's miss	AST	Passes made to teammates who made a shot
STL	The ball was stolen from the opponent	BLK	Blocks in the game
TOV	Times the ball was lost to the opponent team	PF	Personal fouls committed
PTS	Points scored in the game		

Table I.II: Box-Score Correlations with NRTG components

DRTG Correlations			ORTG Correlations		
Feature	**Coefficient**	**Significance**	**Feature**	**Coefficient**	**Significance**
Def Rebounds	-4.432	****	Free Throws	1.707	****
Shots >20ft	1.988	****	Top Scorer Makes	1.855	****
Free Throws	1.469	****	Shots <5ft	0.664	****
Turnovers	-1.394	****	Shots >20ft	0.638	****
Lead Changes	0.485	****	Lead Changes	0.303	****
Off Rebounds	-0.942	****	Off Rebounds	-1.229	****
			Turnovers	-1.402	****
			Def Rebounds	*-0.087*	

IX.B Literature Review

Table II.I: Previous Literature Prediction Accuracy

Study	Model(s) Used	Accuracy

Elo-based Systems (General)	Elo Rating Variants (e.g., CARM-Elo)	~70%+
Cao (2012)	Logistic Regression, Naïve Bayes, SVM, Neural Networks	~68%
Miljković et al. (2010)	Naïve Bayes	~67%
Horvat et al. (2020)	Logistic Regression	~60-65%
Houde (2021)	Logistic Regression	~60-65%
Kaur and Jain (2017)	Hybrid Fuzzy SVM	~88%
Cao (2012) - SVM	Support Vector Machine (SVM)	~70-75%
Li et al. (2021)	Support Vector Machine (SVM)	~70-75%
Lin et al. (2014)	Random Forests	~64-65%
Zhang et al. (2021)	Random Forests	~64-65%
Osken and Onay (2022)	Artificial Neural Network (ANN)	~76%
MambaNet (Manis et al., 2022)	Hybrid Neural Network (Time Series)	~83%
Zhao et al. (2023)	Graph Neural Network (GNN) + ML Classifiers	~71.5%

IX.C Feature Selection

Table IV.I Four factors description

Factor Name	**Description**	**Formula**
Shooting Efficiency (eFG%)	Accurately reflects shooting performance by accounting for the additional value of three-point field goals.	(Field Goals Made + 0.5 * 3-Pointers Made) / Field Goals Made
Turnover Rate (TOV%)	Quantifies a team's ability to maintain possession, representing the percentage of possessions ending in a turnover.	Turnovers / Possessions

Defensive Rebound Rate (DREB%)	The percentage of available defensive rebounds a team successfully secures.	Defensive Rebounds / (Defensive Rebounds + Opponent Offensive Rebounds)
Free Throw Rate (FTr)	Measures a team's ability to generate scoring opportunities from the free-throw line.	Free Throws Made / Field Goal Attempts

Table IV.II Scoring Efficiency Metrics

Feature Name	Description	Calculation/Details	Purpose/Interpretation
Score Differential	The most direct indicator of game state.	Team's cumulative score minus the opponent's cumulative score at the end of Q3.	Positive value indicates the team is leading; negative value indicates a deficit.
Cumulative Points Per Possession (PPP)	Measure of offensive efficiency over the majority of the game.	Total score after three quarters / Total possessions.	Indicates how effectively each team scores per possession up to the end of Q3.
Cumulative Net Rating	The score differential adjusted for the pace of the game.	The difference between the team's and opponent's Cumulative Points Per Possession over Q1-Q3.	Provides a pace-adjusted measure of dominance or deficit over the first three quarters.
Cumulative Four Factors	In-game assessment of performance across fundamental basketball pillars.	The Four Factors (eFG%, TOV%, OREB%, DREB%, FTr),.	Assesses core aspects of basketball performance

Table III.III Game Flow Features

Feature Name	Description	Calculation/Details	Purpose/Interpretation
Lead Changes	The total number of times the lead has switched between the two teams.	Counted during the first three quarters (Q1-Q3).	A high number indicates a closely contested and potentially unpredictable game.

Maximum Lead	The largest point advantage held by the team at any point.	Measured at any point during the first three quarters (Q1-Q3).	Indicates the peak advantage the team managed to build.
Maximum Deficit	The largest point disadvantage faced by the team at any point.	Measured at any point during the first three quarters (Q1-Q3). Represented as a positive number.	Indicates the biggest deficit the team had to overcome or is currently facing.
Momentum Swing	Captures the volatility or extent of significant shifts in the scoreline.	Absolute difference between the team's Maximum Lead (Q1-Q3) and Maximum Deficit (Q1-Q3).	A larger value suggests a game with significant runs by both teams, potentially indicating less stability in the current score.

Table IV.IV: T-value test for significant difference between wins and losses

Feature	t-stat	t p-value	U p-value	Cohen's d	Mean diff	Correlation	Significance
q3_score_diff	22.87	6.89×10^{-89}	2.43×10^{-74}	1.678	18.03	0.635	***
cum_nrtg	22.56	4.44×10^{-87}	7.54×10^{-73}	1.655	24.27	0.629	***
q3_largest_deficit	19.74	1.35×10^{-70}	4.17×10^{-63}	1.448	10.68	0.578	***
q3_largest_lead	18.29	2.02×10^{-62}	7.10×10^{-61}	1.342	10.81	0.549	***
momentum_swing	17.79	1.26×10^{-59}	1.94×10^{-50}	1.305	41.23	0.538	***
max_lead	16.76	4.90×10^{-54}	2.14×10^{-50}	1.229	9.15	0.516	***
q2_score_diff	15.45	3.75×10^{-47}	6.32×10^{-42}	1.133	11.27	0.485	***
max_deficit	13.65	3.71×10^{-38}	4.23×10^{-38}	1.001	4.27	0.440	***

cum_ppp	12.52	7.04×10^{-33}	9.89×10^{-32}	0.918	0.12	0.410	***
opp_composite_rating	-7.74	3.10×10^{-14}	1.07×10^{-15}	-0.568	-139.89	-0.268	***
off_def_matchup	6.96	7.06×10^{-12}	4.49×10^{-12}	0.511	282.42	0.243	***
q3_ppp	8.50	9.60×10^{-17}	2.59×10^{-15}	0.624	0.16	0.292	***
q1_opp_turnover_rate	3.38	7.59×10^{-4}	4.17×10^{-4}	0.248	0.02	0.120	**

Table IV.V: Elo Definitions

Offensive Elo	This rating is designed to reflect a team's scoring proficiency and offensive efficiency, adjusted for the quality of the opposing defense and other contextual factors.
Defensive Elo	Conversely, this rating measures a team's ability to suppress opponent scoring and limit defensive efficiency, adjusted for the quality of the opposing offense.
Composite Elo	This serves as an overall, holistic measure of a team's strength, derived as the arithmetic mean of its current Offensive ELO Rating and Defensive ELO Rating.

Figure IV.I : Calculation of expectation on points for Elo Model.

Let team_off be the offensive ELO of the team, team_def be the defensive ELO of the team.

Let opp_off be the offensive ELO of the opponent, opp_def be the defensive ELO of the opponent.

NBA_AVG_POINTS = 112.0

BETA = 400 (scaling factor for ELO differences to probability)

HOME_ADVANTAGE_ELO = 65 (home court advantage expressed in ELO points)

Expected score for Home Team (exp_score_home):
exp_score_home = NBA_AVG_POINTS + (team_off_home - opp_def_away) / BETA + HOME_ADVANTAGE_ELO / BETA

Expected score for Away Team (exp_score_away):
exp_score_away = NBA_AVG_POINTS + (team_off_away - opp_def_home) / BETA - HOME_ADVANTAGE_ELO / BETA

- K = BASE_K * 1.5 (where BASE_K = 30) if team games played < 10
- K = BASE_K * 1.2 if 10 ≤ team games played < 20
- K = BASE_K if 20 ≤ team games played ≤ 50
- K = BASE_K * 0.8 if team games played > 50

Figure IV.II. Team-separated Composite Elo after quarters played.

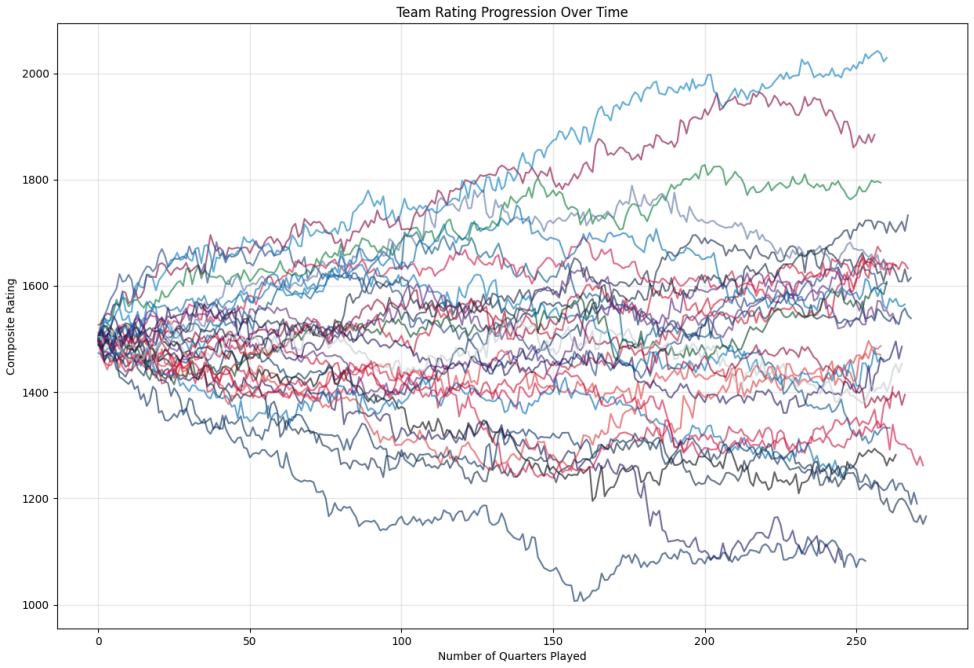

Table IV.VI: Extended Elo feature list.

Feature Name	Definition
ELO Momentum	The first derivative (rate of change) of the team's Composite ELO rating, calculated over their last 20 quarters
ELO Variance	The statistical variance of a team's Composite ELO rating over their last 10 games.
10-Game Rolling Win Rate	The percentage of games won by the team out of their last 10 games played.
Rating Differentials	The absolute difference in Composite ELO ratings between the two competing teams.

Matchup-Specific ELO Differentials	Cross differences in ELO (Team A offense - Team B defense).
ELO-derived Win Probability	The implied win probability for the team in the upcoming game, calculated based on ELO rating differences and incorporating home-court advantage.

Table IV.VII : Elo correlations with empirical win probability			
Feature	Correlation	P-value	Significance
win_probability	0.418	2.27E-42	***
composite_rating	0.299	1.51E-21	***
offensive_rating	0.229	5.17E-13	***
opp_composite_rating	0.299	1.51E-21	***
opp_offensive_rating	0.229	5.17E-13	***
defensive_rating	0.163	3.34E-07	***
opp_defensive_rating	0.163	3.34E-07	***
off_def_matchup	0.042	1.88E-01	ns
def_off_matchup	-0.042	1.88E-01	ns

Table IV.VIII: Statistically Significant Non-Redundant Features, Correlation with Game Outcome, and Associated p-values

Feature Index	Feature Name	Correlation with Win	p-value (Correlation)	p-value (t-test)	p-value (U-test)
64	cum_nrtg	0.6294	4.44e-87	4.44e-87	7.54e-73
44	q3_score_diff	0.6345	6.89e-89	6.89e-89	2.43e-74
47	q3_largest_deficit	0.5782	1.35e-70	1.35e-70	4.17e-63
46	q3_largest_lead	0.5489	2.02e-62	2.02e-62	7.10e-61
69	momentum_swing	0.5382	1.26e-59	1.26e-59	1.94e-50
67	max_lead	0.5155	4.90e-54	4.90e-54	2.14e-50
26	q2_score_diff	0.4850	3.75e-47	3.75e-47	6.32e-42
68	max_deficit	0.4399	3.71e-38	3.71e-38	4.23e-38

29	q2_largest_deficit	0.4390	5.47e-38	5.47e-38	9.42e-34
28	q2_largest_lead	0.4236	3.18e-35	3.18e-35	5.53e-34
65	cum_ppp	0.4099	7.04e-33	7.04e-33	9.89e-32
51	q3_opp_score	-0.4098	7.28e-33	7.28e-33	7.55e-31
57	win_probability (pre-game ELO)	0.3902	1.08e-29	1.08e-29	3.99e-27
61	rating_diff (ELO based)	0.3719	6.20e-27	6.20e-27	5.79e-26
36	q3_nrtg	0.3617	1.86e-25	1.86e-25	2.04e-23
8	q1_score_diff	0.3564	1.03e-24	1.03e-24	9.68e-23
33	q2_opp_score	-0.3505	6.72e-24	6.72e-24	5.51e-22
10	q1_largest_lead	0.2962	3.20e-17	3.20e-17	4.66e-17
18	q2_nrtg	0.2954	3.97e-17	3.97e-17	5.33e-16
37	q3_ppp	0.2919	9.60e-17	9.60e-17	2.59e-15
11	q1_largest_deficit	0.2827	9.31e-16	9.31e-16	1.65e-13
58	opp_offensive_rating **(*)**	-0.2214	4.28e-10	4.28e-10	8.76e-11
56	composite_rating **(*)**	0.2550	5.19e-13	5.19e-13	2.22e-13
60	opp_composite_rating **(*)**	-0.2677	3.10e-14	3.10e-14	1.07e-15
1	q1_ppp	0.2519	1.01e-12	1.01e-12	3.90e-13
15	q1_opp_score	-0.2466	3.03e-12	3.03e-12	2.98e-11

62	off_def_matchup	0.2425	7.06e-12	7.06e-12	4.49e-12
13	q1_opp_ppp	-0.2387	1.54e-11	1.54e-11	1.92e-10
49	q3_opp_ppp	-0.2274	1.40e-10	1.40e-10	1.34e-10
31	q2_opp_ppp	-0.2005	1.68e-08	1.68e-08	5.74e-08
54	offensive_rating **(*)**	0.1866	1.57e-07	1.57e-07	1.90e-07
3	q1_three_point_efficiency	0.1762	7.55e-07	7.55e-07	1.18e-06
39	q3_three_point_efficiency	0.1760	7.85e-07	7.85e-07	3.64e-07
59	opp_defensive_rating **(*)**	-0.1558	1.28e-05	1.28e-05	7.95e-07
19	q2_ppp	0.1508	2.40e-05	2.40e-05	2.47e-05
66	cum_turnover_rate	-0.1478	3.48e-05	3.48e-05	1.03e-04
40	q3_close_range_efficiency	0.1422	6.92e-05	6.92e-05	5.40e-05

VIII.C Methodology

Figure V.I LightGBM Configuration:
- boosting_type: gbdt - class_weight: None - colsample_bytree: 0.8 - importance_type: split - learning_rate: 0.05 - max_depth: 8 - min_child_samples: 20 - min_child_weight: 0.001 - min_split_gain: 0.001 - n_estimators: 500 - n_jobs: None - num_leaves: 63

- objective: binary
- random_state: 42
- reg_alpha: 0.1
- reg_lambda: 0.2
- subsample: 0.8
- subsample_for_bin: 200000
- subsample_freq: 0
- scale_pos_weight: 0.8202247191011236
- boost_from_average: True
- verbose: -1
- min_gain_to_split: 0.1
- feature_fraction: 0.8
- bagging_fraction: 0.8
- bagging_freq: 5
- categorical_feature: [67, 68]
- metric: auc
- is_unbalance: False

Table V.I. Final Average Performance Metrics Across All Cross-Validation Folds (Pre-Game)

Metric	Elo-Only	Momentum	Elo + Momentum
Test Accuracy	0.6371	0.711	0.6757
Test AUC-ROC	0.7001	0.7718	0.7351
Test Precision	0.6601	0.7488	0.688
Test Recall	0.6855	0.7125	0.7455
Test F1	0.6711	0.7273	0.715

Table V.II: Full Model Feature_sets

```
"Elo-Only": {
  "model_type": "LGBMClassifier",
  "training_features": [
    "offensive_rating",
    "defensive_rating",
    "composite_rating",
    "win_probability",
    "rating_diff",
    "off_def_matchup",
    "def_off_matchup",
    "opp_offensive_rating",
    "opp_defensive_rating",
    "Opp_composite_rating"
```

```
"Momentum": {
  "model_type": "LGBMClassifier",
  "training_features": [
    "def_momentum",
```

```
    "composite_trend",
    "off_trend",
    "def_volatility",
    "composite_volatility",
    "composite_avg",
    "off_volatility",
    "off_avg",
    "composite_momentum",
    "def_off_matchup",
    "off_def_matchup",
    "def_trend",
    "composite_rating",
    "def_avg",
    "offensive_rating",
    "off_momentum",
    "Defensive_rating"
```

```
"Combined Elo+Momentum": {
  "model_type": "LGBMClassifier",
  "training_features": [
    "opp_defensive_rating",
    "off_trend",
    "off_volatility",
    "off_def_matchup",
    "composite_rating",
    "off_momentum",
    "win_probability",
    "composite_trend",
    "rating_diff",
    "composite_momentum",
    "def_trend",
    "def_avg",
    "def_volatility",
    "composite_volatility",
    "off_avg",
    "opp_composite_rating",
    "defensive_rating",
    "opp_offensive_rating",
    "def_momentum",
    "composite_avg",
    "def_off_matchup",
    "Offensive_rating"
```

```
"Quarter-Indexed": {
  "model_type": "LGBMClassifier",
  "training_features": [
    "q1_turnover_rate",
    "q1_three_point_efficiency",
    "q1_close_range_efficiency",
    "q1_score_diff",
    "q1_momentum_swing",
    "q1_team_longest_streak",
```

```
    "q1_top_players_total_points_created",
    "q1_team_turnovers",
    "q1_team_off_rebounds",
    "q1_opp_team_nrtg",
    "q1_opp_offensive_ppp",
    "q1_opp_turnover_rate",
    "q1_opp_three_point_efficiency",
    "q1_opp_close_range_efficiency",
    "q2_turnover_rate",
    "q2_three_point_efficiency",
    "q2_close_range_efficiency",
    "q2_score_diff",
    "q2_momentum_swing",
    "q2_team_longest_streak",
    "q2_top_players_total_points_created",
    "q2_team_turnovers",
    "q2_team_off_rebounds",
    "q2_opp_team_nrtg",
    "q2_opp_offensive_ppp",
    "q2_opp_turnover_rate",
    "q2_opp_three_point_efficiency",
    "q2_opp_close_range_efficiency",
    "q3_turnover_rate",
    "q3_three_point_efficiency",
    "q3_close_range_efficiency",
    "q3_score_diff",
    "q3_momentum_swing",
    "q3_team_longest_streak",
    "q3_top_players_total_points_created",
    "q3_team_turnovers",
    "q3_team_off_rebounds",
    "q3_opp_team_nrtg",
    "q3_opp_offensive_ppp",
    "q3_opp_turnover_rate",
    "q3_opp_three_point_efficiency",
    "Q3_opp_close_range_efficiency"
```

```
"Cumulative": {
  "model_type": "LGBMClassifier",
  "training_features": [
    "cum_nrtg",
    "cum_ppp",
    "cum_turnover_rate",
    "max_lead",
    "max_deficit",
    "total_momentum_swing_q123",
    "total_lead_changes_q123",
    "Score_diff_end_q3"
```

Table V.III: Prediction Correlations for Ensemble

Model	LGB	XGB	CAT	RF	ET
LGB	1	0.972	0.991	0.977	0.979
XGB	0.972	1	0.916	0.913	0.984
CAT	0.911	0.916	1	0.949	0.911
RF	0.977	0.913	0.949	1	0.994
ET	0.979	0.984	0.911	0.994	1

Figure V.II: Model Architecture for Ensemble

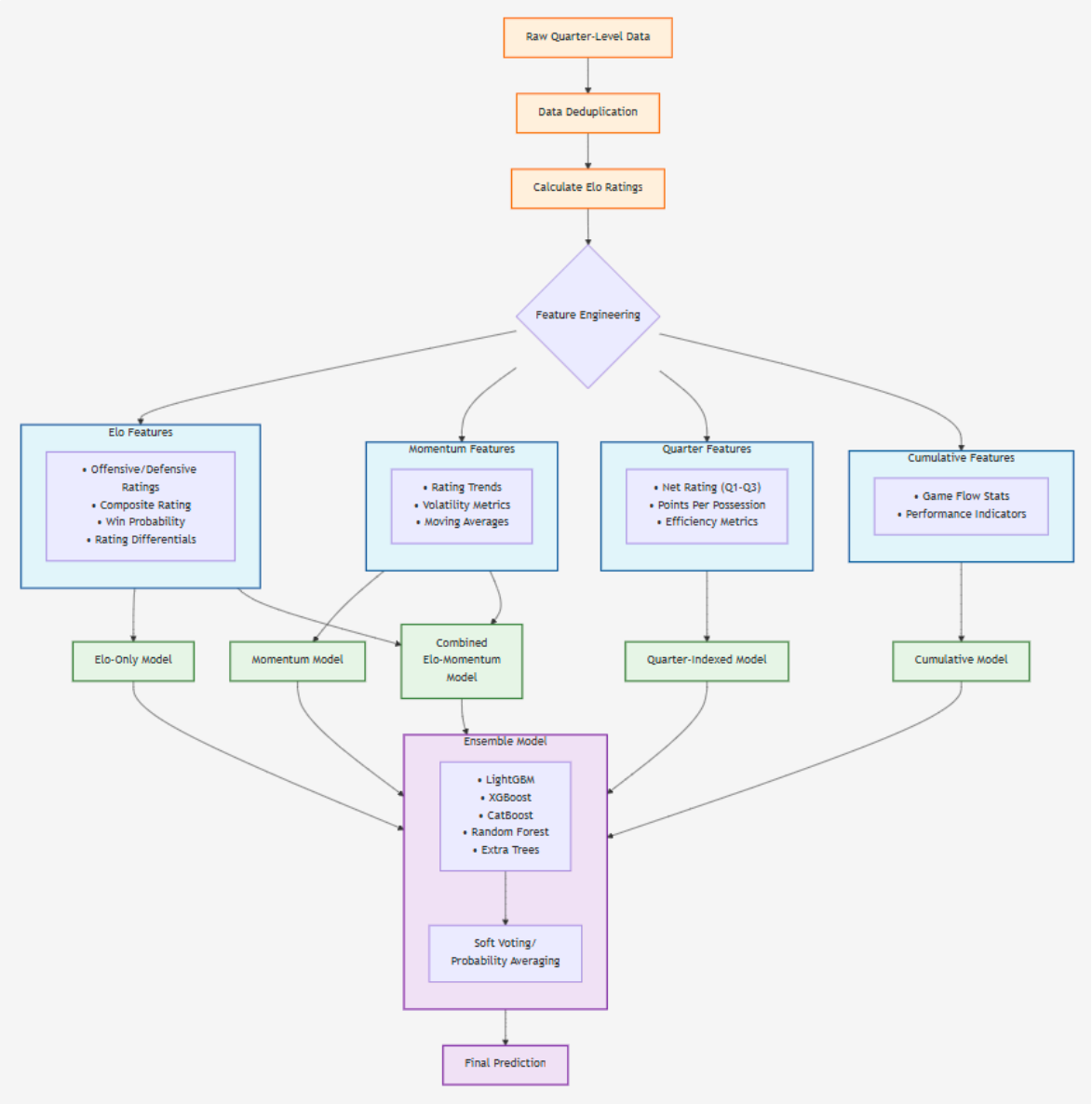

Table V.IV: Test accuracies for ensemble models and aggregate

Model	LightGBM	XGBoost	CatBoost	RF	Extra Trees	Ensemble
Test AUC-ROC	0.9066	0.9087	0.915	0.9003	0.8942	0.9119
Test Accuracy	0.8128	0.8313	0.8189	0.8251	0.8107	0.8259

Table V.V Final Average Performance Metrics Across All Cross-Validation Folds (Q1-Q3)

Metric	Cumulative	Quarter Indexed	Full Model	Ensemble	GRU
Test Accuracy	0.754	0.8009	0.7977	0.8259	0.8369
Test AUC-ROC	0.8314	0.871	0.8867	0.9119	0.9157
Test Precision	0.7739	0.8217	0.8034	0.8465	0.8666
Test Recall	0.773	0.8093	0.8314	0.8309	0.8253
Test F1	0.7734	0.8152	0.7734	0.817	0.8443

Table V.VI Top Significant Features and Their Relative Importance for Ensemble Model

Rank	Feature Name	Relative Importance (%)
1	opp_offensive_rating	100.00
2	cum_nrtg	99.02
3	cum_ppp	87.58
4	q3_opp_ppp	87.25
5	q3_assist_ratio	84.97
6	opp_composite_rating	84.97
7	composite_rating	84.31
8	q3_opp_three_point_efficiency	84.31

9	q3_opp_turnover_rate	82.68
10	q3_score_diff	82.03
11	q2_assist_ratio	77.12
12	offensive_rating	75.49

Figure V.III: Model training for ensemble.

```
--- Temporal Fold 1/5 ---
Train size: 161, Val size: 157
Fold 1 Ep 1: TrLoss:0.592, VLoss:0.680, VAUC:0.865
Fold 1 Ep 2: TrLoss:0.417, VLoss:0.670, VAUC:0.886
Fold 1 Ep 3: TrLoss:0.408, VLoss:0.658, VAUC:0.875
Fold 1 Ep 4: TrLoss:0.285, VLoss:0.642, VAUC:0.862
Fold 1 Ep 5: TrLoss:0.268, VLoss:0.617, VAUC:0.866
Fold 1 Ep 6: TrLoss:0.297, VLoss:0.587, VAUC:0.870
Fold 1 Ep 7: TrLoss:0.264, VLoss:0.552, VAUC:0.872
Fold 1 Ep 8: TrLoss:0.248, VLoss:0.516, VAUC:0.869
Fold 1 Ep 9: TrLoss:0.257, VLoss:0.486, VAUC:0.867
Fold 1: Early stopping at epoch 9.
Fold 1 Final Val ROC AUC (Q1-Q3): 0.8667

--- Temporal Fold 2/5 ---
Train size: 318, Val size: 157
Fold 2 Ep 1: TrLoss:0.490, VLoss:0.681, VAUC:0.895
Fold 2 Ep 2: TrLoss:0.373, VLoss:0.655, VAUC:0.898
Fold 2 Ep 3: TrLoss:0.321, VLoss:0.616, VAUC:0.900
Fold 2 Ep 4: TrLoss:0.283, VLoss:0.558, VAUC:0.898
Fold 2 Ep 5: TrLoss:0.279, VLoss:0.490, VAUC:0.898
Fold 2 Ep 6: TrLoss:0.279, VLoss:0.439, VAUC:0.905
Fold 2 Ep 7: TrLoss:0.291, VLoss:0.378, VAUC:0.920
Fold 2 Ep 8: TrLoss:0.226, VLoss:0.351, VAUC:0.918
Fold 2 Ep 9: TrLoss:0.237, VLoss:0.389, VAUC:0.926
Fold 2 Ep 10: TrLoss:0.179, VLoss:0.351, VAUC:0.925
Fold 2 Ep 11: TrLoss:0.184, VLoss:0.360, VAUC:0.923
Fold 2 Ep 12: TrLoss:0.166, VLoss:0.343, VAUC:0.930
Fold 2 Ep 13: TrLoss:0.177, VLoss:0.364, VAUC:0.928
Fold 2 Ep 14: TrLoss:0.155, VLoss:0.364, VAUC:0.928
Fold 2 Ep 15: TrLoss:0.151, VLoss:0.352, VAUC:0.930
Fold 2 Ep 16: TrLoss:0.146, VLoss:0.347, VAUC:0.929
Fold 2 Ep 17: TrLoss:0.166, VLoss:0.357, VAUC:0.934
Fold 2 Ep 18: TrLoss:0.121, VLoss:0.316, VAUC:0.941
Fold 2 Ep 19: TrLoss:0.096, VLoss:0.354, VAUC:0.931
Fold 2 Ep 20: TrLoss:0.139, VLoss:0.351, VAUC:0.935
Fold 2 Ep 21: TrLoss:0.118, VLoss:0.365, VAUC:0.931
Fold 2 Ep 22: TrLoss:0.079, VLoss:0.361, VAUC:0.929
```

```
Fold 2 Ep 23: TrLoss:0.076, VLoss:0.373, VAUC:0.929
Fold 2 Ep 24: TrLoss:0.105, VLoss:0.370, VAUC:0.931
Fold 2 Ep 25: TrLoss:0.103, VLoss:0.368, VAUC:0.928
Fold 2: Early stopping at epoch 25.
Fold 2 Final Val ROC AUC (Q1-Q3): 0.9279
```

IX.D Discussion

Table VI.I Prediction Accuracy by post-season team composite rating

Quartile Difference	Description	Games	Accuracy
-3	Team playing Q4 vs. Opponent Q1 (Team significantly weaker)	53	62.3%
-2	Team vs. Opponent 2 quartiles stronger	100	65.0%
-1	Team vs. Opponent 1 quartile stronger	147	62.6%
0	Teams in same quartile	164	61.0%
1	Team vs. Opponent 1 quartile weaker	158	63.9%
2	Team vs. Opponent 2 quartiles weaker	102	61.8%
3	Team playing Q1 vs. Opponent Q4 (Team significantly stronger)	53	56.6%

Table VI.II: Win rate by closing margin

Subgroup Definition	Games	Actual Win Rate	Accuracy	ROC AUC
Close Games (Margin ≤ 12)	222	0.54	0.6622	0.6985
Blowout Games (Margin > 12)	563	0.54	0.8899	0.9613

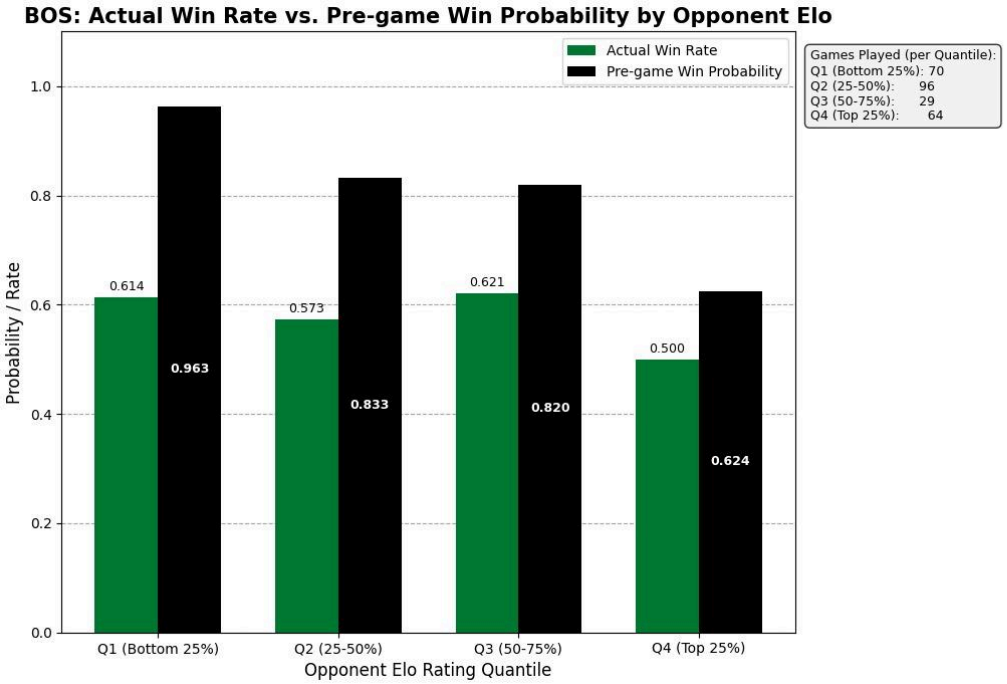

Figure VI.I Pre-game Win probability vs Empirical win rate, Boston (t = 1)

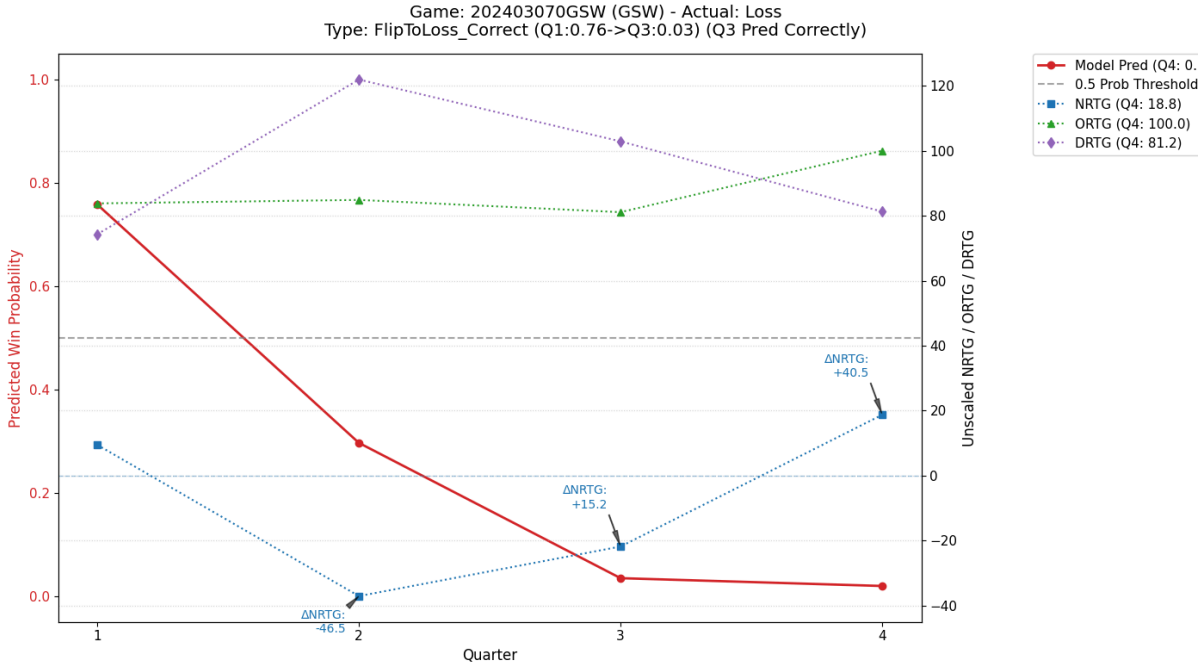

Figure VI.II Progression of win prediction throughout the game (Correct, significant change).

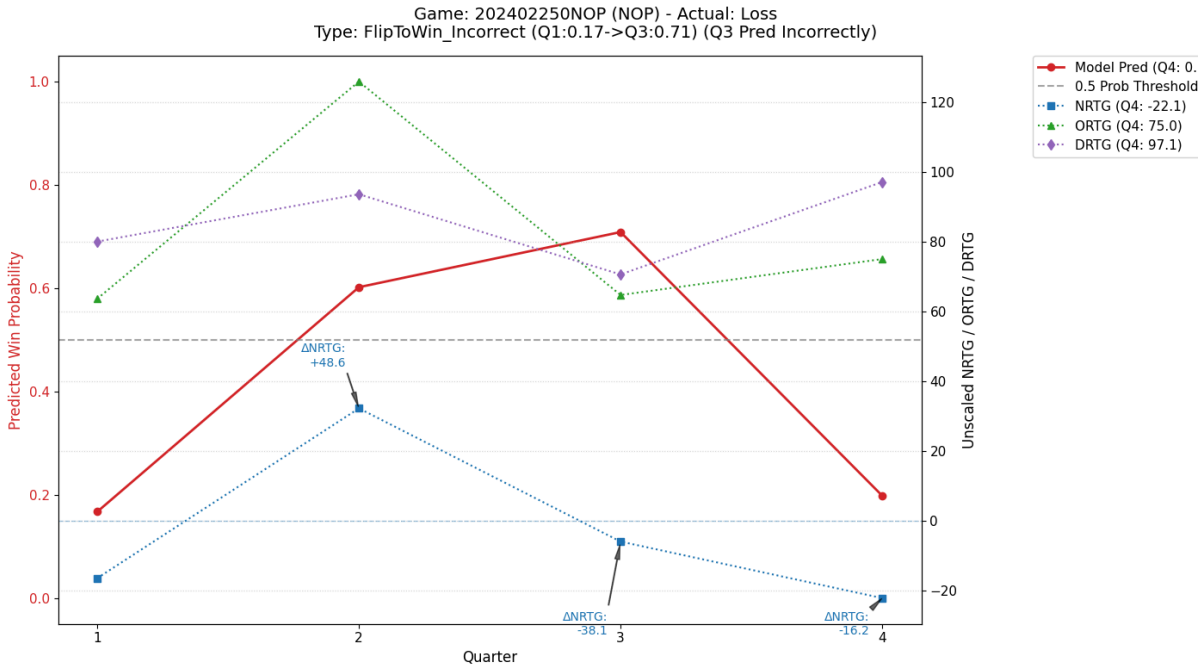

Figure VI.III Progression of win prediction throughout a game (Incorrect)

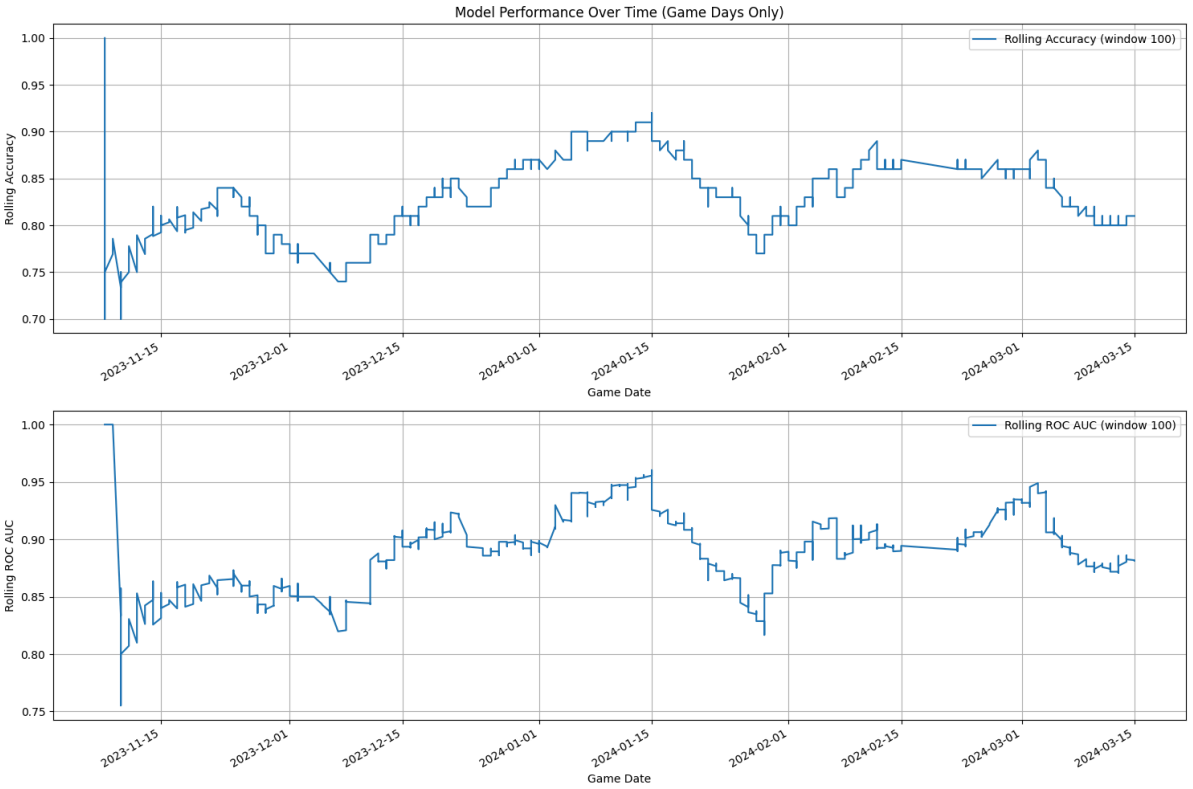

Figure VI.IV Iterative Training accuracy throughout season