Decoding Winning Patterns in Professional Basketball via Transformers

Joshua Zhang, Harvard University, Cambridge, MA, JoshZhang@college.harvard.edu ;
Amy Dong, Harvard University, Cambridge, MA, AmyDong@college.harvard.edu

New England Symposium on Statistics in Sports

Abstract

The NBA off-season is one of the most pivotal moments for franchises, as front offices decide whether to rebuild, maintain the status quo, or make bold moves in pursuit of a championship. These decisions hinge on expectations of future team performance, yet prior research has struggled to deliver reliable forecasting models. This gap underscores the need for a robust approach to predicting team win totals to inform strategic planning.

We utilized publicly available player-level data spanning the 2016-2025 NBA seasons to develop two transformer-based neural network models to predict the overall win record per team for the regular and playoff seasons. Leveraging the architecture's ability to capture long-range dependencies across player attributes and interactions, we represent each team as a sequence of up to 15 players, mirroring the size of an NBA roster. Each player is represented by a comprehensive set of features from the previous regular season, including statistical totals, per-game averages, per-36-minute metrics, draft position, age, physical measurements, player origin, shooting breakdowns, and other advanced analytics (e.g., BPM, VORP).

Using a 80-20 (8-2 NBA Seasons) train-test split, our regular season model achieves a mean absolute error (MAE) of 7.14 and 6.49 on the training and test set, respectively. Our playoff season model achieves a MAE of 2.69 and 2.55 on the training and test set, respectively. These results are competitive given the NBA's volatility, where small differences in win totals determine playoff qualification or seeding. Our model offers a practical tool to forecast roster impact, aiding decisions around trades, free agents, or the draft.

Data Collection and Formatting

For our data, we scraped league-wide, player-level datasets from Basketball Reference that encompass total, shooting, adjusted shooting, per game, per 36 minutes, and advanced metrics on a season-by-season basis, respectively. It is notable that there were absolutely no missing values for any of these datasets.

From there, we formatted each season's data with the following steps:

1. Concatenated all metrics at the season–player level
2. Sorted by team and descending total minutes played
3. Filtered to each player's final team designation, since post-deadline performance bests reflects roster strength for the majority of the season
4. Utilized the NBA API to append on height, position, weight, draft number (or 'UFA'), origin variables
5. Filtered to the top 15 players by minutes played to best capture team strength and mirror NBA roster size
6. Padded roster sizes less than 15 with 'Unknown' for categorical variables and 0 for numerical statistics for consistent formatting

Handling Dataset Edge Cases

Due to very rare cases of missingness within certain variables for certain players, we were required to infer and fill certain metrics. Rationally, we used the following strategies to mitigate additional noise to propel our model forward:

- **Origin -** Determined using college team, then country of origin, and finally 'Unknown' if unavailable
- **Height / Weight -** Used the mean height and weight with respect to a player's position for any missing height or weight variables to mitigate noise – almost rarely needed

Model Rationale

When modeling a feature as complex as win outcomes, we require an approach capable of capturing non-linear interactions between variables with high accuracy, while also accounting for sequential ordering and contextual relationships across years. Ultimately, we chose to utilize a transformer model because of the following reasons:

1. **Sequential Modeling -** Treats the roster as a sequence of players, enabling the model to capture positional interactions across all 15 slots
2. **Capturing Dependencies -** Identifies long-distance relationships between combinations of predictors
3. **Lack of Multicollinearity Concerns -** Correlated variables do not hinder performance

Model Layout

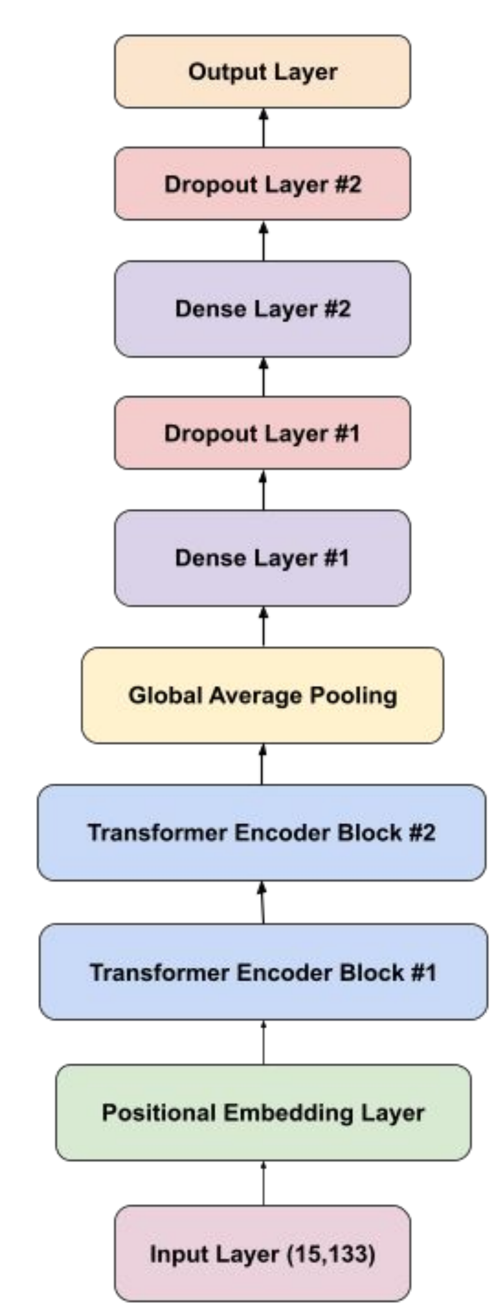

Figure 1
Model architecture for the transformer-based win prediction framework. Each input (15 players × 133 features) passes through two Transformer encoder blocks, followed by global average pooling, dense layers with dropout regularization, and a final output layer for win total prediction

Layer Rationale

1. **Input Layer -** Dimensions reflective of a 15-man main roster rotation and the 133 predictors utilized
2. **Positional Embedding Layer -** Adds player positional context; Recognizes player 1 plays the most minutes and player 15 plays the least
3. **Custom Dual Transformer Encoders**
   a. **Multi-Head Self-Attention Block -** Captures long term relationships between predictors amongst players within each respective team
   b. **FFNN -** Small 2-layer feed-forward neural network to capture non-linear relationships of the attention block
   c. **Residual Layers -** Applied to both Transformer sub-layers to mitigate gradient-related issues
4. **Global Average Pooling + Dense Layers + Dropout Layers -** Summarizes sequence into a single decision point. Adds non-linearity and stabilizes training

Results

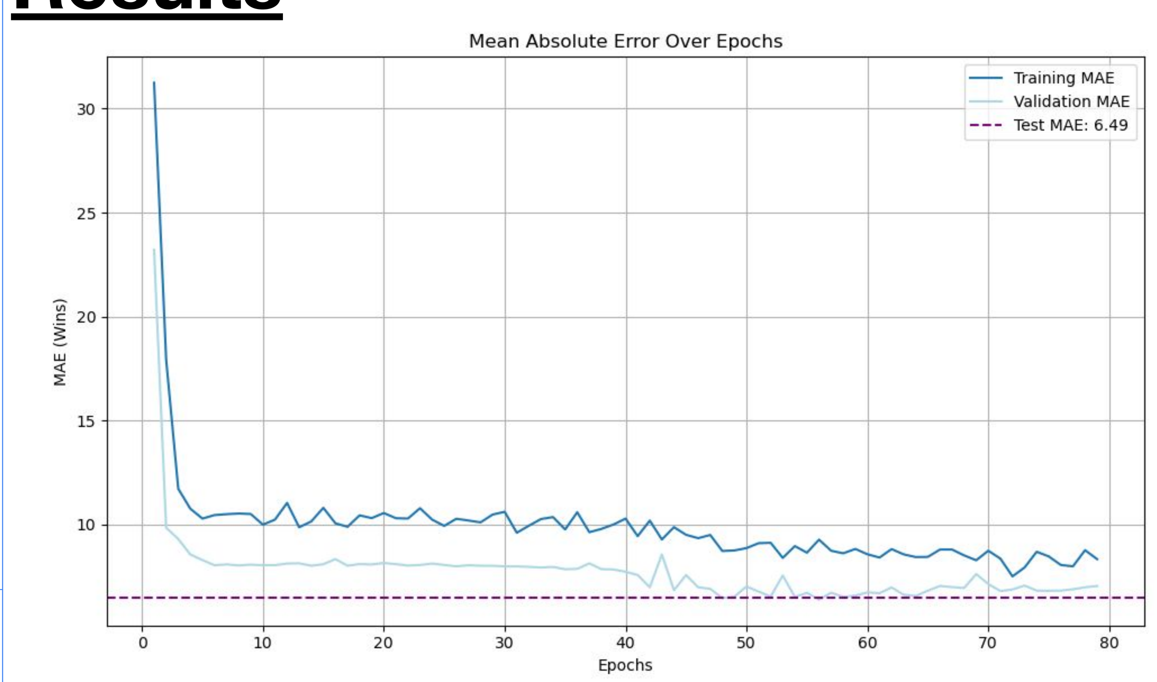

Figure 2
Performance of the regular season model, using Mean Absolute Error (MAE) over 80 epochs. MAE of 7.14 and 6.49 on the training and test set (denoted in purple line), respectively

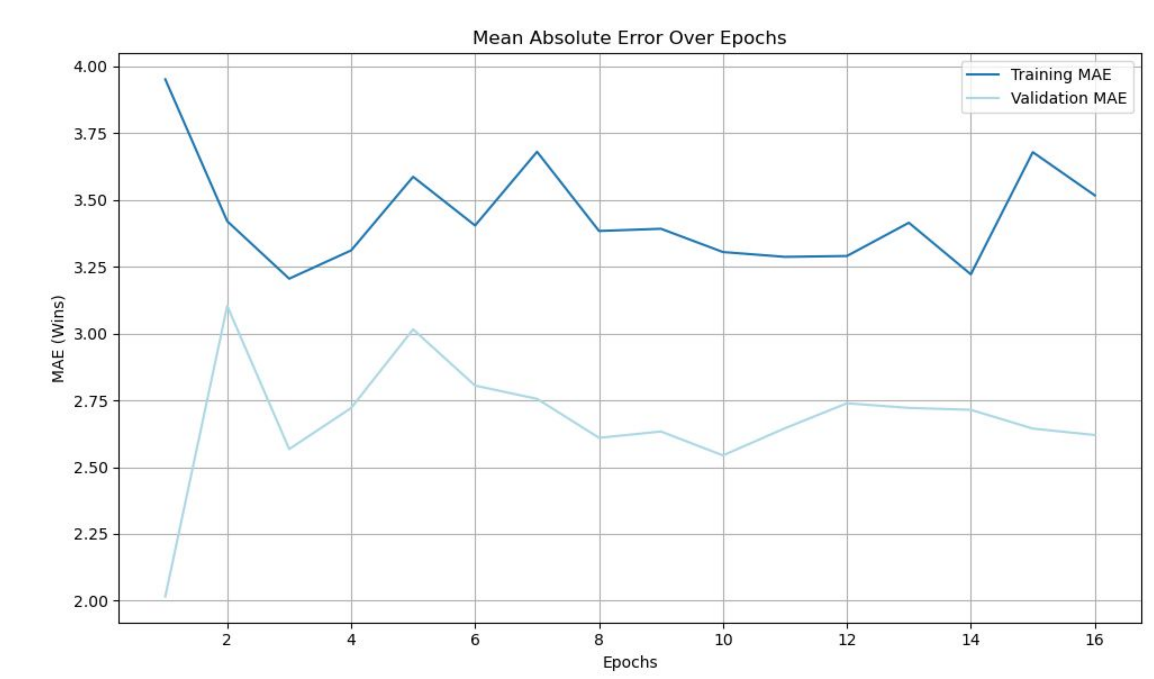

Figure 3
Performance of the playoff season model, using Mean Absolute Error (MAE) over 16 epochs. MAE of 2.69 and 2.55 on the training and test set, respectively

Conclusion / Discussion

Treating a team as a sequence of players and learning cross-player interactions with transformers yields competitive team-level forecasts despite the NBA's inherent volatility. Our models achieve MAE = 6.49 wins for the regular season and 2.12 wins for the playoffs, suggesting utility for front-office scenario planning around trades, free agency, and draft strategy. Beyond point predictions, the architecture provides a flexible foundation for analyses to estimate the win impact of roster changes, as well as a promising start for capturing players as sequence-based feature data in models.