UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Computer Science Department

Master's Degree in Computer Science

Machine Learning

Tennis Match Prediction using GNN

Student: Donato Festa

Professor: Claudia D'Amato
Professor: Nicola Di Mauro

Academic Year 2024-2025

Contents

Abstract

The prediction of professional tennis match outcomes is a challenging task due to the complex interplay of player skills, dynamic form, and contextual factors. Traditional machine learning approaches often rely on hand-crafted features and treat matches as independent events, potentially overlooking the rich relational structure of the sport. This project addresses this gap by proposing a Graph Neural Network (GNN) model that learns player representations directly from the network of historical match results.

We construct a directed graph where players are nodes and matches are edges, allowing the model to capture latent attributes of skill and reputation. We conduct a rigorous comparative analysis of our GNN model against a strong XGBoost baseline, which is enriched with a comprehensive set of engineered features including Elo ratings, head-to-head statistics, and fatigue metrics. The evaluation is performed using a robust temporal cross-validation protocol repeated over 10 independent runs.

Our results demonstrate that the GNN model achieves a mean Log-Loss of approximately 0.50, a statistically significant improvement ($p < 0.001$) over the baseline's 0.63. Qualitative analysis of the learned embeddings via t-SNE further reveals that the model successfully clusters players by their competitive era and relational standing. This work shows that modeling the relational structure of player interactions is a highly effective strategy, offering superior predictive performance by uncovering patterns not easily captured by traditional methods.

1 Introduction

The prediction of outcomes in professional sports has long been a captivating challenge for both statisticians and machine learning practitioners. Among these, tennis stands out for its dynamic nature, where match results are determined by a complex interplay of individual skill, physical endurance, psychological state, and environmental factors such as the playing surface. Developing accurate predictive models not only holds commercial value in areas like betting markets but also provides deeper insights for coaches and analysts into the factors that drive player performance.

However, traditional modeling approaches often fall short. Many methods rely on aggregated statistics or ranking systems like the official ATP rankings. While useful, these metrics are often lagging indicators of a player's true current form and, more importantly, they typically treat each match as an independent event. This overlooks a fundamental truth of competitive sports: a player's strength is relational and best defined by the context of whom they have defeated and to whom they have lost. Pioneering work in sports analytics has often focused on creating sophisticated hand-crafted features, such as the Elo rating system, to better capture player form [3]. While effective, these methods still do not explicitly model the complete network of interactions.

Recent advancements in Graph Neural Networks (GNNs) offer a powerful new paradigm for this problem. GNNs are designed to learn from data structured as graphs, making them ideally suited to model the relational ecosystem of a professional sports league. While GNNs have seen successful application in team sports like soccer and basketball [2], their application to individual sports like tennis, specifically for outcome prediction based on historical match graphs, remains a less explored area.

This project aims to fill this gap by investigating the efficacy of a GNN-based approach for tennis match prediction. Our main contributions are as follows:

- We develop a robust baseline model using XGBoost, trained on a rich set of engineered features including dynamic Elo ratings, head-to-head statistics, and fatigue metrics.
- We propose a GNN model based on a GraphSAGE architecture that learns player embeddings directly from a directed graph of match outcomes, capturing latent relational strengths.
- We conduct a rigorous comparative evaluation using a temporal cross-validation protocol and demonstrate that the GNN model achieves a statistically significant improvement in predictive performance over the strong baseline.

- We provide a qualitative analysis of the learned embeddings, showing that they capture meaningful structures, such as clusters of players from the same competitive era, thereby offering insights beyond simple prediction accuracy.

2 Related Work

The task of predicting sporting outcomes has been approached from various angles, evolving from purely statistical models to complex deep learning architectures. This section reviews the primary methodologies relevant to this project, highlighting the gaps that our graph-based approach aims to address.

2.1 Feature-Based and Statistical Models

The most traditional approach to sports prediction relies on meticulous feature engineering. In tennis, this often involves using player rankings, recent performance metrics, and surface-specific statistics. A cornerstone of this approach is the **Elo rating system**, originally developed for chess [3], which provides a dynamic measure of a player's strength by updating their score after each match based on the outcome and the opponent's rating. Many successful predictive models have been built upon Elo ratings and other carefully crafted features, often employing classic machine learning algorithms like logistic regression or, more recently, powerful ensemble methods such as Gradient Boosted Trees [1].

While highly effective, these models fundamentally operate on a per-match basis. They excel at capturing a player's current form but treat each contest as an isolated event, defined only by the explicit features provided. They do not inherently model the broader network of relationships and the "reputational" strength a player holds within the entire competitive ecosystem.

2.2 Graph-Based Approaches in Sports Analytics

A more recent paradigm in sports analytics involves representing the domain as a graph, where entities (players, teams) are nodes and their interactions (matches, passes) are edges. This structure allows for the application of Graph Neural Networks (GNNs), which can learn representations that capture complex relational dependencies.

This methodology has shown significant promise, particularly in team sports. For example, GNNs have been used to analyze team formations in soccer, predict player movements in basketball, and forecast play outcomes in American football. A recent survey by Drexler [2] highlights the breadth of GNN applications in sports, emphasizing their ability to model dynamic, interconnected systems.

However, the application of GNNs for a global, historical outcome prediction in individual sports like tennis is a less-explored area. Most existing graph-based analyses in tennis focus on fine-grained, intra-match events (e.g., classifying stroke types from video) rather than modeling the entire player-vs-player network to predict match results. Our project directly addresses this specific niche, investigating

whether learning from this global relational structure provides a predictive edge over state-of-the-art, feature-based methods.

3 Dataset and Preprocessing

This section details the data source used for this project, the preprocessing steps applied, and the key insights derived from an initial Exploratory Data Analysis (EDA).

3.1 Data Source

The study is based on the comprehensive match-by-match dataset for the men's ATP tour, compiled and maintained by Jeff Sackmann[1]. This publicly available repository provides detailed statistics for all ATP tour level matches, including Challenger and Futures events, from the beginning of the Open Era. For this project, we focused on all singles matches from the year 2000 to the end of the most recent complete season available, ensuring a modern and extensive dataset of over 70,000 matches. Each record in the dataset contains granular information, including tournament details, match-specific data (e.g., surface, round, score), player attributes (e.g., rank, age), and in-match statistics (e.g., aces, double faults, break points).

3.2 Data Preprocessing

Before any feature engineering, a series of preprocessing steps were applied to ensure data quality and consistency. The raw data, distributed across multiple CSV files per year, was first concatenated into a single DataFrame. Key columns, such as the match date, were converted to the appropriate 'datetime' format. All statistical columns were cast to numeric types, with any conversion errors or missing entries being treated as 'NaN' (Not a Number) values to be handled during the feature engineering phase. Finally, the entire dataset was rigorously sorted chronologically by match date and an internal match number to ensure that all historical and time-dependent features could be calculated without data leakage.

3.3 Exploratory Data Analysis (EDA)

An initial EDA was conducted to understand the fundamental properties of the dataset. Two key insights emerged that directly informed the modeling approach.

First, the analysis confirmed that the **playing surface** is a critical contextual factor, with a majority of matches played on Hard courts (approx. 50%), followed by Clay (approx. 35%) and Grass (approx. 15%). This validated the decision to include surface type as a key feature in our baseline model.

[1] `https://github.com/JeffSackmann/tennis_atp`

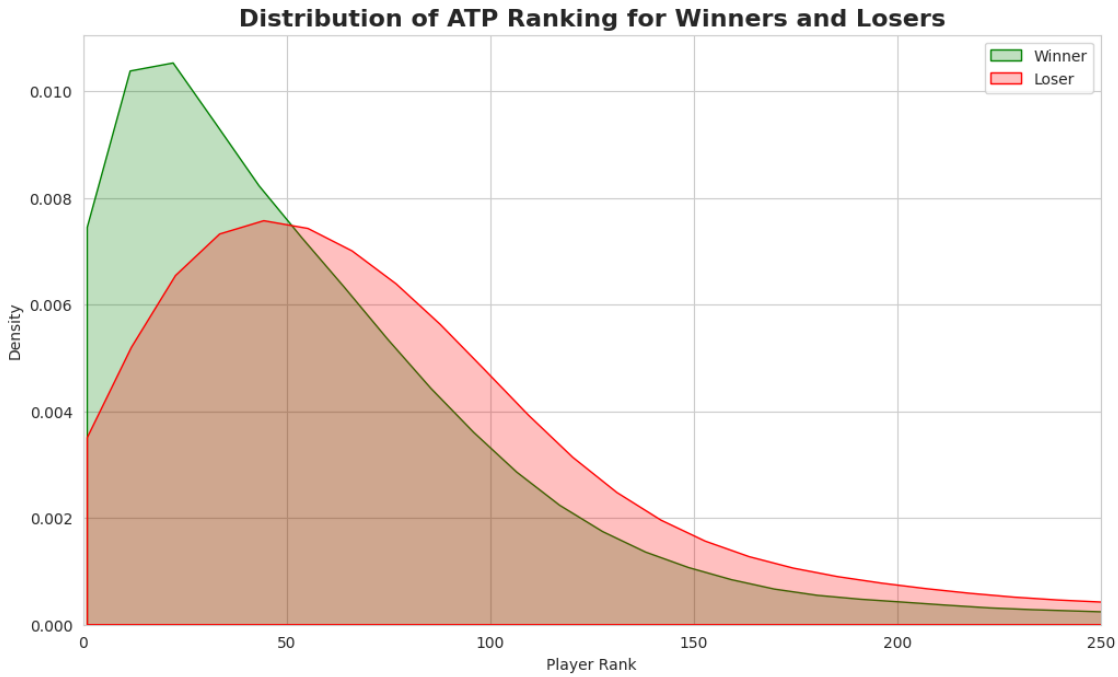

Figure 1: Probability density distribution of player rankings for match winners versus losers. A significantly lower (better) rank is highly correlated with winning.

Second, as illustrated in Figure 1, a clear and stark difference exists between the rank distributions of winners and losers. Winners are overwhelmingly concentrated in the top-tier of the rankings, while losers' ranks are more broadly distributed. This confirmed that rank-based features, and by extension more dynamic rating systems like Elo, would be highly predictive and must form the core of any strong baseline model.

4 Methodology

This project employs a comparative approach, evaluating a traditional feature-based model against a novel graph-based model. This section details the feature engineering process, the architecture of both models, and the robust evaluation protocol used to compare them.

4.1 Baseline Model: XGBoost with Engineered Features

The baseline model is designed to be a strong competitor, representing a state-of-the-art approach based on traditional feature engineering. We use XGBoost [1], a powerful gradient boosting algorithm, as our classifier. Its performance relies heavily on a rich set of features calculated for each match.

4.1.1 Feature Engineering

For each match between Player 1 and Player 2, we engineered a vector of differential and contextual features. The key features are summarized in Table 1. The most critical engineered feature is the **Elo rating** [3], a dynamic strength measure calculated iteratively over the entire dataset. For each match, we compute the expected win probability E_1 for Player 1 as:

$$E_1 = \frac{1}{1 + 10^{(R_2 - R_1)/400}} \tag{1}$$

where R_1 and R_2 are the pre-match Elo ratings of Player 1 and Player 2, respectively. The ratings are then updated based on the actual outcome. This provides a more responsive measure of a player's current form than the official rankings.

Table 1: Key features engineered for the XGBoost baseline model.

Feature Category	**Description**
Dynamic Ratings	'elo_diff': Difference in pre-match Elo ratings. 'rank_diff': Difference in official ATP rankings.
Historical Context	'h2h_diff': Difference in head-to-head win counts. 'streak_diff': Difference in current match winning streaks.
Physical State	'fatigue_diff': Difference in minutes played in the last 7 days. 'age_diff': Difference in player ages.
Match Context	'surface': One-hot encoded vector for Clay, Hard, Grass, or Carpet.

4.2 Graph-Based Model: GNN for Link Prediction

Our primary model frames the prediction task as a **link prediction** problem on a directed graph. This approach aims to learn player representations (embeddings) not from their individual statistics, but from their relational position within the entire network of matches.

4.2.1 Graph Construction

We construct a single, global graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where:

- $\mathscr{V}$ is the set of nodes, with each node representing a unique player.
- $\mathscr{E}$ is the set of directed edges. A directed edge $(u, v) \in \mathscr{E}$ exists if player u defeated player v in a match.

This structure captures the complex web of historical dominance and upsets across the entire ATP tour.

4.2.2 Model Architecture

We implement a 'GNNLinkPredictor' model using PyTorch Geometric [4]. The architecture consists of an encoder and a decoder. The **encoder** is a two-layer GraphSAGE network [5]. It generates a final embedding $\mathbf{z}_u \in \mathbb{R}^d$ for each player u by aggregating feature information from their local neighborhood in the graph. The initial player features are learnable vectors from a 'torch.nn.Embedding' layer.

The **decoder** is a simple yet effective dot product function. Given the embeddings $\mathbf{z}_u$ and $\mathbf{z}_v$ for two players, it calculates a logit score s_{uv} for the potential edge (u, v):

$$s_{uv} = \mathbf{z}_u \cdot \mathbf{z}_v = \sum_{i=1}^{d} z_{ui} z_{vi} \tag{2}$$

This score is then passed through a sigmoid function to produce the final win probability. The entire model is trained end-to-end by minimizing a binary cross-entropy loss, augmented with an L2 regularization term to prevent overfitting on the embedding weights.

4.3 Evaluation Protocol

To ensure a robust and fair comparison, both models were evaluated using an identical, rigorous protocol:

- **Temporal Cross-Validation:** We employed a ‘TimeSeriesSplit‘ with 5 folds. This method ensures that the model is always trained on past data to predict future, unseen matches, mimicking a real-world scenario and preventing data leakage.
- **Multiple Runs:** The entire 5-fold CV process was repeated 10 times with different hyperparameter configurations sampled from a predefined grid. This yields a stable distribution of performance scores for each model.
- **Primary Metric:** Our primary evaluation metric is **Log-Loss**, which measures the accuracy of the predicted probabilities, heavily penalizing over-confident incorrect predictions.

The final comparison between the models is based on a paired t-test performed on the 10 Log-Loss scores generated by this procedure.

5 Results and Analysis

This section presents the results of our experimental evaluation. We first provide a quantitative comparison of the predictive performance of the XGBoost baseline and the GNN model. We then delve into a qualitative analysis of the GNN's learned embeddings to interpret what the model has captured.

5.1 Quantitative Performance Comparison

The performance of both models was evaluated using the robust protocol described in Section 4. For each of the 10 independent runs, we computed the overall out-of-fold Log-Loss. Table 2 summarizes the descriptive statistics of these scores for both models.

Table 2: Summary of model performance over 10 evaluation runs. The GNN model demonstrates a substantially lower (better) mean Log-Loss and comparable stability.

Metric	**XGBoost Baseline**	**GNN Model**
Mean Log-Loss	0.6337	**0.5016**
Std. Dev. Log-Loss	0.0081	0.0149
Min Log-Loss (Best)	0.6223	0.4871
Max Log-Loss (Worst)	0.6480	0.5288

The results clearly indicate the superiority of the GNN model. On average, it achieves a Log-Loss of 0.5016, a significant reduction compared to the baseline's 0.6337. To verify that this improvement is not a product of random chance, we performed a paired t-test on the 10 Log-Loss scores from each model. The test yielded a p-value of less than 10^{-6}, which is far below the standard significance level of $\alpha = 0.05$. We can therefore confidently reject the null hypothesis and conclude that the GNN model's superior performance is **statistically significant**.

5.2 Qualitative Analysis of Player Embeddings

To understand what the GNN has learned beyond simple predictive accuracy, we analyzed the 64-dimensional player embeddings generated by the best-performing model run. We used the t-SNE algorithm to project these high-dimensional vectors into a 2D space for visualization. The resulting plot is shown in Figure 2.

The visualization reveals that the GNN has learned a meaningful organization of players. Distinct clusters and structures have emerged automatically from the graph data. A prominent cluster, highlighted in Figure 2, contains the dominant

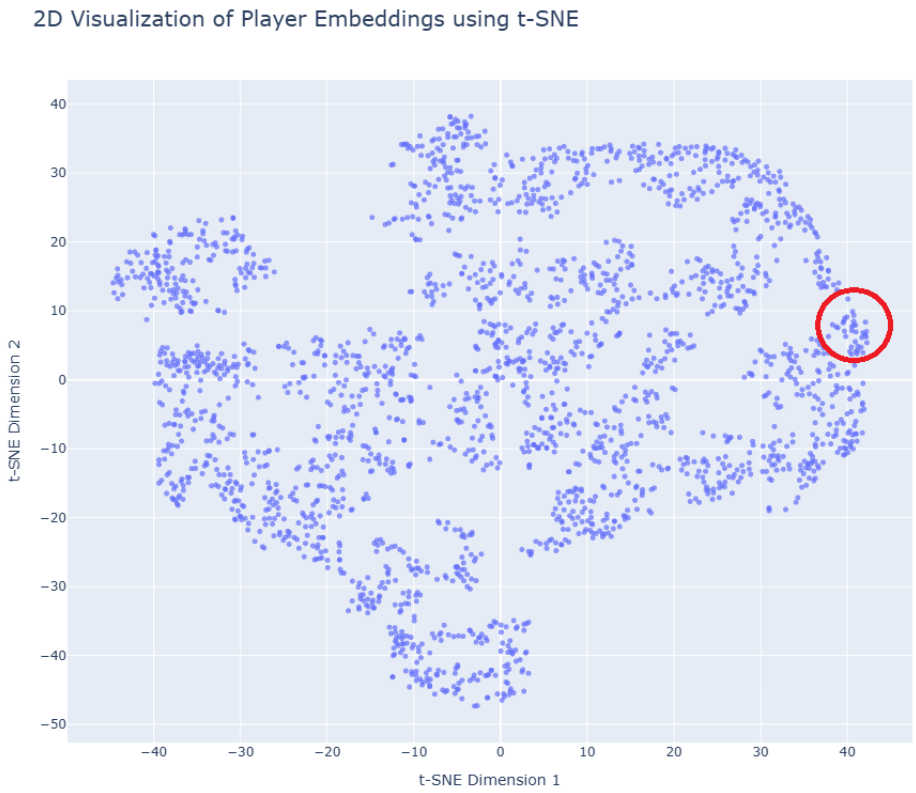

Figure 2: 2D t-SNE visualization of the learned player embeddings. The model has automatically organized players into distinct clusters based on their relational patterns in the match graph. The highlighted region shows the "Golden Era" cluster.

figures of tennis's "Golden Era", including Roger Federer, Rafael Nadal, Novak Djokovic, and Andy Murray.

Interestingly, this cluster also includes players like Andreas Seppi and Gaël Monfils. While not typically ranked alongside the "Big 4", their inclusion is revealing. These players are characterized by exceptional longevity, and as such, they are highly-connected nodes in the graph who frequently played against the same elite opponents. This suggests that the GNN has learned to represent not just raw skill, but also a player's role and context within their competitive era. This ability to capture deep, latent relational features is likely the primary reason for its superior predictive performance.

6 Conclusion and Future Work

6.1 Conclusion

This project successfully developed and compared two distinct models for predicting the outcome of professional tennis matches: a feature-rich XGBoost baseline and a novel Graph Neural Network (GNN) model. Our primary goal was to determine if a relational approach could outperform traditional, instance-based feature engineering.

Our findings confirm this hypothesis. The GNN model, which learns player representations directly from the network of match histories, demonstrated a **statistically significant** improvement in predictive power over the strong XGBoost baseline. With a mean Log-Loss of approximately 0.50, compared to the baseline's 0.63, the GNN provides more accurate and better-calibrated probabilistic predictions. Furthermore, a qualitative analysis of the learned embeddings revealed that the GNN successfully captured nuanced, real-world structures, grouping players by their shared competitive era and relational standing—an insight beyond the reach of traditional models.

A limitation of the proposed GNN, however, was also identified: a tendency towards over-confident predictions (probabilities near 100%) for well-established historical rivalries. This suggests that while the model effectively learns long-term dominance, it may not fully capture the inherent uncertainty of a single match.

6.2 Future Work

While this project achieved its objectives, our experiments highlighted several exciting avenues for future research that could address the identified limitations and further enhance the model's capabilities:

- **Advanced Model Architectures:** To mitigate over-confidence, a natural next step is to explore more sophisticated decoders, such as a Multi-Layer Perceptron (MLP), which could learn more complex interactions between player embeddings. Our preliminary experiments indicated this approach is less stable and requires a much more rigorous hyperparameter optimization process, for instance, by using a framework like Optuna.
- **Richer Graph Features:** The current GNN learns from a simple, unweighted graph. Its performance could be significantly enhanced by enriching the graph with:
  - **Node Features:** Initializing player nodes with static attributes like height and handedness.

  - **Edge Features:** Adding contextual information to each match edge, such as the court surface and tournament importance.

- **Hybrid Models:** A powerful and pragmatic direction would be to create a hybrid model that combines the strengths of both approaches. This would involve using the GNN's learned embeddings as rich, relational features within the high-performing XGBoost framework, alongside the dynamic, up-to-the-minute features like Elo and fatigue. This could yield a single, comprehensive model that leverages both long-term reputational knowledge and short-term player form.

- **Temporal Graph Networks (TGNs):** The most advanced future step would be to move from a static graph to a dynamic one. Implementing a TGN would allow player embeddings to evolve over time, directly capturing a player's changing form and momentum within the graph structure itself, offering the most holistic approach to the problem.

References

[1] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. 5, 9

[2] Tim Todd Drexler. Sports analytics with graph neural networks and graph convolutional networks. *Preprints.org*, 2024. Version 1, Posted: 1 October 2024. 3, 5

[3] Arpad E. Elo. *The Rating of Chessplayers, Past and Present.* Arco, New York, 1978. 3, 5, 9

[4] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 10

[5] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pages 1024–1034, 2017. 10