# A WEB-SCALE LIGHTWEIGHT FRAMEWORK FOR FOOTBALL MATCH OUTCOME PREDICTION

**Adam Gábor**
Analytik - Futbalové správy
analytik@futbalovespravy.sk

August 17, 2025

**ABSTRACT**

This paper introduces a novel and lightweight framework with entire model training and selection process to the client side. Using TensorFlow.js and the Web Worker API, a standard web browser is transformed into a transient parallel-processing grid, enabling an efficient hyperparameter search without any server-side computational cost. We trained a feedforward neural network on a dataset of 105,565 matches from Europe's top seven leagues, using a minimal set of engineered features that capture team strength and momentum. A key innovation is a dual-objective selection criterion, which optimizes for overall accuracy while ensuring robust performance on the underrepresented 'draw' class. The resulting model achieves a predictive accuracy of 50.21%, representing a statistically significant improvement of 3.51 percentage points over the strongest heuristic based on ELO. This performance approaches the efficiency of commercial betting odds, validating our model as a practical and effective tool easily distributable within the client-side.

## 1 Introduction

The domain of sports analytics has seen significant growth, and association football presents a particularly challenging predictive modeling problem. The sport's low-scoring nature and inherent stochasticity result in a complex system where outcomes are difficult to forecast. A principal obstacle is the statistical class imbalance [1], where draws are significantly less frequent than wins, leading many standard classification models to achieve deceptively high accuracy simply by failing to predict this crucial minority class. Furthermore, state-of-the-art approaches often rely on computationally intensive server-side infrastructure and proprietary data, limiting their accessibility and practical application in real-time, user-facing web environments.

This paper introduces a web-scale framework designed to overcome these challenges by leveraging client-side computation. We present a system that transforms a standard web browser into a transient parallel-processing grid using TensorFlow.js and the Web Worker API. This in-browser method performs an efficient hyperparameter search to develop a lightweight yet robust neural network, specifically optimized to handle class imbalance through a dual-objective selection criterion.

## 2 Methodology

### 2.1 Dataset and Feature Engineering

This research uses the dataset compiled by Gábor (2025) [2], which aggregates historical data from numerous international leagues. We performed a curation process to construct a high-quality relevant corpus for our specific modeling objectives. The primary filtering criterion was to include only matches from a consistent set of Europe's top seven professional leagues, for which reliable Elo ratings and match-by-match form data were available. The selected leagues are English Premier League, Spanish La Liga, German Bundesliga, Italian Serie A, French Ligue 1, as well as Portuguese Primeira Liga and Dutch Eredivisie.

The data exhibits an expected class imbalance (45.0% home wins, 27.3% draws, 27.7% away wins), which is a central consideration for our training design strategy. We selected and engineered four low-latency features to serve as inputs for the model:

- **Elo_Diff:** Difference in team Elo ratings (computed using [3] )
- **Form5_Diff:** Difference in points gathered over the last 5 matches.
- **Form3_Diff:** Difference in points gathered over the last 3 matches.
- **Momentum_Diff:** A weighted form metric accentuating recent performance.

### 2.2 Model Architecture

The predictive model is a lightweight feedforward neural network, built with TensorFlow.js [4] for optimal in-browser performance. The network's sequential design starts with an Input Layer that processes four key engineered features. This feeds into a Dense Layer with 64 neurons, which employs the ReLU activation function to introduce nonlinearity. To mitigate overfitting, a Dropout Layer with a fairly high rate of 0.4 is strategically included to randomly deactivate a portion of neurons during training.

The architecture continues with a second Dense Layer of 32 neurons, also utilizing common ReLU activation. The final Output Layer consists of 3 neurons, corresponding to the three possible match outcomes: home win, draw, or away win. A Softmax activation function is applied here to generate a probability distribution across these outcomes, ensuring the model's predictions are interpretable and collectively sum to one. This streamlined design ensures the model remains computationally efficient while still capturing complex patterns in the data.

### 2.3 In-Browser Parallelized Training

To find optimal hyperparameters without server-side computation, we implement a client-side grid search using the Web Worker API. This creates a parallel processing grid within the user's browser, allowing multiple model instances to be trained simultaneously with different hyperparameter configurations. The final "champion" model is selected via a dual-objective criterion: it must exhibit the highest overall accuracy on a validation set while also surpassing a predefined performance threshold for predicting the minority 'draw' class. This ensures that the selected model is as accurate as possible, but also practically useful.

To rigorously evaluate the performance and validate the efficacy of our proposed framework, we conducted a series of comparative experiments. The primary goal of these experiments was to quantify the predictive advantage of our trained neural network (the "Champion Model") over an array of simpler, more conventional prediction strategies. For this purpose, the entire curated dataset (N=105,565) was partitioned according to a standard 80/20 split, with the larger partition (80%) serving as a training set for learning the model's internal parameters, while the remaining 20% was held out as a validation set.

All reported accuracy metrics are calculated exclusively on this unseen validation set to ensure an unbiased assessment of each model's generalization capabilities.

### 2.4 Benchmark Models

To create a rigorous baseline for our neural network model, we implemented six benchmark models. These models represent various simple or heuristic-based strategies, allowing us to quantify the "added value" of our approach. The first two are naive baselines: Random Guessing assigns an outcome randomly, Always Guess Home Win consistently predicts a home team victory, which is the most common outcome.

The remaining four benchmarks are more sophisticated and logic-based. Higher Form Wins predicts the winner based on the better points-per-game record over the last five matches. Higher ELO Wins predicts the team with the higher ELO rating to win. This is followed by a modified version, ELO with Draw Wiggle Room, which predicts a draw if the ELO difference is less than 50 points, as suggested by Hvattum and Arntzen in [5]. The final benchmark, ELO + Form Weighted, is a composite model that predicts the winner using a combined score derived from both ELO ratings and recent team form.

### 2.5 Performance Comparison

The predictive performance of the model we trained, selected via our in-browser parallelized training process, was evaluated against the benchmark models. The results, measured by overall predictive accuracy on the validation set, are presented in Table 1.

Table 1: Performance Comparison of Predictive Models on the Validation Set

| Model | Overall Accuracy (%) |
|---|---|
| **Our Champion Model** | **50.21%** |
| Higher ELO Wins | 46.40% |
| ELO + Form Weighted | 46.02% |
| Always Guess Home Win | 45.02% |
| ELO with Draw Wiggle Room | 43.65% |
| Higher Form (5 games) Wins | 41.13% |
| Random Guessing | 33.24% |

The results clearly demonstrate that our model substantially outperforms all heuristic and naive benchmarks. It achieves an accuracy of 50.21%, a significant improvement over the strongest baseline. This performance gap underscores the model's ability to learn non-linear relationships within the data that simple rule-based systems cannot capture, and being very close to the generally known 50 - 52% accuracy of betting odds. Crucially, this enhanced accuracy is achieved while maintaining the ability to predict draws, a primary objective of our dual-criterion selection process.

## 3 Conclusion and Outlook

The framework presented was developed as part of an initiative to create a browser-based "supercomputer" for the *Analytik* web portal. Its lightweight architecture enables the execution of thousands of simulations in parallel with only minimal input data, making it ideally suited for large-scale predictive tasks directly within the user's browser. This efficiency is further enhanced by the continual updating of Elo ratings and form metrics, allowing the system to simulate entire seasons in a matter of seconds without requiring any server-side computational resources.

#### Limitations

Although the framework demonstrates strong predictive performance, it currently relies on only four engineered features. This design choice ensures low computational cost, but may omit potentially valuable contextual factors such as player-specific information or tactical data.

## References

[1] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[2] Adam Gábor. Club Football Match Data. `https://github.com/xgabora/Club-Football-Match-Data-2000-2025/`, 2025. Online; accessed [01.08.2025].

[3] ClubElo.com. ClubElo Rating System Methodology. `http://clubelo.com/System`, 2024. Online; accessed [01.08.2025].

[4] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, Stan Bileschi, Michael Terry, Charles Nicholson, Sandeep N. Gupta, Sarah Sirajuddin, D. Sculley, Rajat Monga, Greg Corrado, Fernanda B. Viégas, and Martin Wattenberg. TensorFlow.js: Machine Learning for the Web and Beyond. *arXiv preprint arXiv:1901.05350*, 2019.

[5] L. M. Hvattum and H. Arntzen. Using ELO ratings for match result prediction in association football. *International Journal of Forecasting*, 26:460–470, 2010. `https://doi.org/10.1016/J.IJFORECAST.2009.10.002`.