Testing Market Efficiency: Evidence from the Bundesliga (German Men’s Handball)

Adit Saxena

# Introduction

The efficient market hypothesis (EMH) has been a prevailing theory in the field of financial economics, asserting that financial markets incorporate and reflect all available information instantaneously. Its examination and verification have predominantly been within the confines of the stock markets. However, the literature increasingly extends the scope of EMH to other arenas, including sports betting markets, where similar mechanisms are present. This paper seeks to broaden this perspective by investigating market efficiency in a relatively unexplored domain: the Bundesliga, the premier professional league of German men’s handball.

The Bundesliga, with its vast and enthusiastic fanbase, presents a unique opportunity to test the predictions of the EMH in the context of sports betting. While prior research has tended to focus on larger and more globally recognized sports such as football or basketball, handball's smaller, more specialized market may offer fresh insights into how efficiently information is processed and incorporated into betting odds. By examining the betting markets for Bundesliga games, this paper will evaluate whether odds set by bookmakers reflect all publicly available information and thus whether these markets can be deemed 'efficient' in the EMH sense. The findings are expected to provide not only valuable insights for sports economists but also practical implications for sports betting.

# Data Analysis

To delve deeper into market efficiency, we employed the ELO rating model. The ELO model is a method for calculating the relative skill levels of teams in an environment where they play each other. It has seen extensive usage in various sports and is favored due to its dynamic nature, adapting to changes in team performances over time. In this analysis, the ELO model is employed to calculate the relative strengths of the teams and the probability of each possible match outcome. Subsequently, these probabilities are contrasted against the bookmakers' odds to detect any inefficiencies. One of the primary advantages of using the ELO model is its adaptability. It takes into account the strength of the opposition, so a win against a strong opponent results in a more significant increase in a team's rating than a win against a weaker opponent. This makes it particularly suitable for handball, where fluctuations in team performance are frequent.

However, the ELO model does have some limitations. It does not consider situational factors such as home court advantage, player injuries, or suspensions - variables that could considerably impact a team's performance. To combat this, I added a variable for the home court advantage which tries to mitigate this effect.

## Behind the Data

The data consists of Bundesliga matches from the following seasons: [2012, 2013, 2014, 2015, 2016, 2017, 2018, 2021, 2022]. The Bundesliga did play matches in 2019 and 2020 but I wasn’t able to find a reliable source for the match data. I used the api-sports module for handball to get the date, week, home team name, away team name, season and the score. I used these to create the ELO model. I ran into a few problems with this data, originally I wanted to use the weeks as a way to measure how teams were doing week to week but discovered that the teams often played matches later than the week they were scheduled because matches were shifted. This forced me to use the date to organize the data frame.

## Walking through the ELO model

# Initial Variables that affect ELO

initial\_rating = 1500

K = 16 # factor at which scores are updated

home\_advantage = 32 #should be in number of elo points

These are the initial values I used to determine the model, the explanation for why these numbers were chosen is explained in a later section.

def sigmoid\_strength\_of\_victory(goal\_dif, equation=1, elo\_diff=0):

"""

Calculate the strength of victory for a winning team based on the goal difference and optional parameters.

Parameters:

goal\_dif (numeric): The goal difference for the match for the winning team.

equation (int, optional): The equation used to calculate the strength of victory. Defaults to 1.

elo\_diff (numeric, optional): The Elo rating difference between the two teams. Defaults to 0.

Returns:

float: The strength of victory, representing the reward for the winning team.

The function calculates the strength of victory based on different equations depending on the chosen equation

parameter.

If equation = 1:

The function uses a sigmoid function to determine the strength of victory. A team that wins by a larger goal

difference should be rewarded more, but extremely large wins are not overvalued.

If equation = 2:

The function uses the 538 Model to calculate the strength of victory. This model takes into account

both the goal difference and the Elo rating difference between the teams. This has not been implemented yet.

Example:

goal\_dif = 5

equation = 1

sigmoid\_strength\_of\_victory(goal\_dif, equation) # Returns 1.2294464977081369

goal\_dif = 4

equation = 2

elo\_diff = 100

sigmoid\_strength\_of\_victory(goal\_dif, equation, elo\_diff) # Returns 0.8648752922373572

"""

if equation == 1:

sov = (6.2 / (2 + math.exp(-(1 / 5) \* (goal\_dif - 8))))

elif equation == 2:

sov = np.log(goal\_dif + 1) \* (2.2 / (elo\_diff + 2.2))

return sov

This function was taken from handballranking.com[[1]](#footnote-1). At this point of time, I’m not completely sold on this function. It rates a win of 8 goals as twice as strong as a win of 1 goal. This methodology might make sense for international rankings but I think in my case I might be able to have a smaller value to account for the difference in goals between the 2 situations. To illustrate the point here are the frequencies of winning goals in the Bundesliga vs. the dataset from the website.

A blue and white graph

Description automatically generated A graph of different sizes and colors

Description automatically generated

Figure 1. Bundesliga scores vs. International Handball scores

As you can see from the histograms, the Bundesliga has far more ties and closer matches and less blowouts as compared to all competitive handball matches. One adjustment that can be made to the model is to change this strength\_of\_victory function as it doesn’t seem to have a much of an effect on the model accuracy. I tried numerous iterations of this function, but they didn’t seem to affect the accuracy of the model (still ranging between 66.5-68.5). In the future I might consider using the 538 method if I can find the right equation for handball as the current equation is currently used for the NFL.

def calculate\_expected(team1, team2):

"""

Calculate the expected outcome of a match between two teams, taking into account the home team advantage.

Parameters:

team1 (numeric): Elo rating for the first team.

team2 (numeric): Elo rating for the second team.

Returns:

float: The expected probability of the first team winning, considering the home team advantage.

The function uses the Elo rating system formula to calculate the expected outcome of a match. The Elo rating

represents the skill level of a team or player. The higher the Elo rating, the stronger the team.

The home team advantage is factored into the calculation by adjusting the Elo rating of the home team. The home

team advantage is represented by a numeric value.

Example:

team1 = 1600

team2 = 1500

home\_advantage = 100

calculate\_expected(team1, team2) # Returns 0.7597468482494313

team1 = 1500

team2 = 1600

home\_advantage = 100

calculate\_expected(team1, team2) # Returns 0.5

"""

team1\_expected = 1 / (1 + math.pow(10, (team2 - (team1 + home\_advantage)) / 400))

return team1\_expected

Next, we have this function which takes into account the home\_advantage and provides the probability for that team to win.

def update\_ratings(home\_team\_rating, away\_team\_rating, goal\_difference):

"""

Update team ratings based on match outcome and goal difference.

Parameters:

home\_team\_rating (numeric): The rating of the home team.

away\_team\_rating (numeric): The rating of the away team.

goal\_difference (numeric): The difference in goals between the home and away team.

Returns:

float: The updated rating of the home team.

The function updates the ratings of the home team based on the match outcome and goal difference. The outcome is

determined by the goal difference, where a goal difference of 0 results in a draw (outcome = 0.5), a positive goal

difference indicates a win for the home team (outcome = 1), and a negative goal difference indicates a loss for the

home team (outcome = 0).

The expected outcome is calculated using the 'calculate\_expected' function, which estimates the probability of the

home team winning based on their rating and the away team's rating.

The strength of victory is calculated using the 'sigmoid\_strength\_of\_victory' function, which determines the

reward for the home team based on the goal difference.

The updated rating is computed using the formula:

updated\_rating = home\_team\_rating + (K \* strength\_of\_victory \* (outcome - expected))

Example:

home\_team\_rating = 1500

away\_team\_rating = 1600

goal\_difference = 2

update\_ratings(home\_team\_rating, away\_team\_rating, goal\_difference) # Returns the updated rating of the home team

"""

outcome = 0.5 if goal\_difference == 0 else 1 if goal\_difference > 0 else 0

expected = calculate\_expected(home\_team\_rating, away\_team\_rating)

strength\_of\_victory = sigmoid\_strength\_of\_victory(goal\_difference)

updated\_rating = home\_team\_rating + (K \* strength\_of\_victory \* (outcome - expected))

return updated\_rating

This function updates the rating of teams taking into account the two factors mentioned earlier, a similar function is applied for the away team if they are the winning team.

Finally, we have a regression factor in between seasons. This is a way to hedge against trades, coaching changes and player acquisitions. Currently the regression is set up to regress teams 2/3 of the way back to the mean.

Now that the base of our ELO model is set up, we can start creating rating for all teams and updating them on a per match basis. There were 3 parameters I had to tune, the weight constant K, the bonus points for goal difference, and the home advantage. Beginning with the weight constant, the source uses the following.

A screenshot of a game

Description automatically generated

Figure 2. Weight Factor based on Match type

The Bundesliga falls in the Continental championships, so I wanted to use a K factor of approximately 30, but after running my model through various K factors I came up with the following diagram. A graph with blue lines

Description automatically generated

Figure 3. Tuning the Scaling Factor

As seen in the graph, it looks like the model’s accuracy isn’t really affected by the scale factor (home\_advantage set at 20). This means that that my strength\_of\_victory function needs to be refined to better provide more accurate results.

I performed a similar test for the home\_advantage (K set at 16). A graph with blue dots

Description automatically generated

Figure 4. Tuning Home Advantage

As we can see in this graph, the home\_advantage does change the accuracy of the model quite a bit. I didn’t want to overfit the model to the home\_advantage, so by looking at other sources for a home\_advantage, I decided to use an advantage of 32 ELO points.

Now that our model has been somewhat fitted to our data, we can look at the accuracy of the model in particular. The table below shows the probability for the home team to win, along with the accuracy of the prediction and the sample size to give a better idea of how accurate the model was in each situation. Looking at these results, the model is extremely accurate in extreme cases (< 20% and > 80% ) but struggles in the middle performing worse than 50% between (40,50]. Out of 2776 matches, this model correctly predicts 1900 matches or 68.44 %. This is lower than other prediction models (538 NFL ~75%), but my hope is that with a better strength of victory function and incorporating other data such as advanced stats the model can improve. To take a closer look at each team’s progress over a season, I recommend at looking at the plotly graph in final.ipynb.

|  |  |  |
| --- | --- | --- |
| Probability Bin | **Accuracy (%)** | **Sample Size** |
| (0,10] | 88.89 | 18 |
| (10,20] | 87.12 | 163 |
| (20,30] | 72.83 | 254 |
| (30,40] | 56.81 | 345 |
| (40,50] | 44.54 | 366 |
| (50,60] | 54.77 | 524 |
| (60,70] | 70.11 | 435 |
| (70,80] | 86.71 | 316 |
| (80,90] | 92.76 | 290 |
| (90,100] | 96.92 | 65 |

## Betting

I struggled to find accurate odds as most companies don’t track historical odds for handball. I did manage to find odds on oddsportal[[2]](#footnote-2) and tried to scrape the odds but they’ve made their website extremely difficult to scrape. Just to test the model and see if it was profitable, I added 1x2 odds for roughly 30 matches in the 2022-2023 season. 1x2 odds refer to the odds for team 1 winning, the odds for a draw, and the odds for team 2 winning. The 3-way money line is more profitable to bet on if you have a better idea of what the winning team is rather than the 2-way money line or betting on the spread. I did manage to find more historical odds for games, but just haven’t integrated them into my dataset as of yet. (will be done soon within the next week).[[3]](#footnote-3)

### Basics of the Betting Strategy

In each season there are 34 weeks, which means each team plays 34 matches equating to roughly 306 matches in a season. Based on my model, I would rather not predict a tie as any match between teams with roughly equal ELO’s would be predicted as a tie. Looking at the probability bins, this means potentially all matches between (40, 60] could be predicted as tie which is 890/2776 = 32%. Given that we are not predicting a tie, the max prediction accuracy of our model is 100-9%, 91%. This leaves roughly 270 matches in a season to predict.

The betting strategy for handball matches initially involved a prediction model that calculated the probabilities of outcomes. This included the likelihood of home and away teams winning and also the odds for a draw. However, it was identified that the model performed suboptimally when predicting matches with a home win probability between 40% and 60%. The predictive accuracy dropped significantly in this range, which could lead to potential losses if used as a basis for betting.

Upon reflection, this was primarily due to the high competitive balance that is typical in handball matches. The teams have a relatively similar level of skills, and matches often hinge on various unpredictable factors, such as players' performance on the day, slight injuries, or even emotional state of the team. This led to an inherent unpredictability when the model was trying to forecast the result in this specific probability range, leading to its poor performance.

To mitigate this issue and to enhance the effectiveness of the betting strategy, a decision was made to eliminate this range from consideration. The model was then optimized to only bet on situations where the home win probability was either less than 40% or greater than 60%.

The core hypothesis of this strategy was that there are inherent biases present in how oddsmakers set their odds. These biases can manifest in several ways, such as underestimating the potential of an underdog or overvaluing the popularity of a team. My betting model was designed to exploit these biases to maximize potential returns. The model will place bets when the odds provided by the oddsmakers do not align with the calculated probabilities. In this case, the model places a bet only if the expected value, based on the probability and the potential return, is positive.

This strategy, while seemingly simplistic, has the advantage of focusing on matches where the model has the highest predictive accuracy and avoiding matches with an inherent level of uncertainty. It also seeks to exploit any biases in the odds-making process, thus optimizing the potential returns from betting.

### Accuracy of Betting

As of right now, there is a framework for deciding when to bet, whenever the difference between my model and the odds is greater than a threshold, but I haven’t looked too closely at this as I’m still waiting to get better data. Doing some initial analysis, it does seem like this does seem to work as some favorites are underrepresented in odds.

# Next Steps

The current iteration of the handball betting model operates by placing a fixed bet amount on each selected match. However, a potential area of improvement for the model lies in the customization of the bet amounts based on the model's confidence in the predicted outcome. Introducing a dynamic betting system is a potential next step for enhancing the model's effectiveness and efficiency. The modified system could place bets ranging from 1-5 units, with the bet size proportionate to the model's confidence in a team's victory.

Implementing such a strategy requires developing a mechanism within the model to gauge the 'certainty' of its predictions. The level of certainty could be derived from factors like the margin of the calculated winning probability over the threshold, the recent performance of the teams, and the statistical variance of the teams' historical data. By doing this, the model can make more informed betting decisions and optimize the return on investment.

Another promising area of improvement is incorporating spread betting into the strategy. Currently, the model focuses exclusively on 3-way money line bets (home win, away win, or draw). However, in closely contested matches, betting on the points spread rather than the outright winner could provide more profitable opportunities. For example, if the model predicts a close game but with one team slightly favored, a spread bet could be a more beneficial strategy.

In the case of spread betting, the model will have to predict not just the winner but also the margin of victory. This requires a different set of predictive algorithms and a more granular analysis of the teams' performance data. It also demands an understanding of how oddsmakers set their spreads and how these can be exploited.

1. [↑](#footnote-ref-1)
2. <https://www.oddsportal.com/handball/germany/bundesliga/results/> [↑](#footnote-ref-2)
3. <https://betsapi.com/l/515/Germany-Bundesliga> [↑](#footnote-ref-3)