# An online application for automated creation and optimization Elo rating systems validated with AUDL and NFL data from 2000-2022

Simon D. Weaver[*]

[*]University of Notre Dame, Notre Dame, IN, USA. sweaver4@nd.edu. (608) 213-1182

## 1. Abstract

Elo rating systems are widely used to predict the outcome of various sports and leagues. However, to generate an accurate system for any particular league, it is usually necessary for a user to create custom code and wrangle historical data. Here, an open-source online application is presented that takes a simple .csv file of game data from any sport or league, and automatically creates and optimizes an Elo rating system that can be used to predict the outcome of future games and analyze historical trends. In this manuscript, example models are created for the American Ultimate Disc League (AUDL) and the National Football League (NFL) and are used to accurately predict the outcomes of games in both leagues. The AUDL Elo model is the first of its kind, and the expected win percentages for the NFL Elo model compare favorably with the expected win percentages from a customized model generated by FiveThirtyEight. Importantly, this tool can be used without any knowledge of coding or advanced data handling and can be applied to any sport or league of interest, lowering the accessibility barrier of sports data analysis.

## 2. Keywords

Elo Ratings, Automation, Web Application, AUDL, Prediction, Optimization

# 3. Introduction

Elo rating systems have been used since the sixties to predict the outcomes of sporting events. First proposed by Arpad Elo (Elo, 1967), the method was designed to compare the skill level of chess players and provide a method to predict the outcome of a game between any two players. In an Elo system, the important factor in determining the outcome of a match is the difference in ratings between the two players. At the conclusion of a match, the rating of each player is adjusted based on the result, as well as the pre-match predicted result, and the updated value is used to predict the outcome of the subsequent match (Glickman, 1995).

In the decades since it was first proposed, Elo-based systems have been introduced for many sports and leagues, including the soccer (Hvattum and Arntzen, 2010), the NBA (Fischer-Baum and Silver, 2022), tennis (Vaughan Williams *et al.,* 2021), and many more. These systems have been used both in an unofficial capacity by amateur sports analysts, as well as in an official capacity to determine seeding for tournaments (Stefani, 2011). To create an accurate Elo model for any given sport or league, there are a variety of factors that must be considered, which has resulted in a family of Elo-based rating systems that follow the same mathematical models but differ in various parameters and corrections. For example, FiveThirtyEight, a highly regarded website affiliated with ESPN and the New York Times, has created a complicated Elo model for predicting the outcome of NFL games that considers the distance traveled by the away team and the skill level and injury status of quarterbacks, among other factors (Silver, Boice and Paine, 2022).

The complexity of these models, as well as the continual updating of Elo scores because of game outcomes makes the creation of an Elo model a non-trivial task. Despite their proven effectiveness as a prediction tool (Barrow *et al.,* 2013) it is difficult to develop and validate an Elo system for any given league or sport without coding a custom program to do the work. There are several examples of this type of coding available as blog posts online (Cunningham, 2021; SpreadsheetSolving, 2016). While these posts do an excellent job describing what goes into an Elo model, it is still up to an interested user to create their own using either a spreadsheet template (which is difficult to scale to larger leagues), or by coding it in a data science language such as python.

To lower the accessibility threshold of creating an Elo model for any given sport or league, this manuscript presents an online application that will generate a mathematical Elo model, and automatically optimize several important Elo parameters to find a model that most accurately predicts the outcomes of games for a particular dataset. Here, I have taken example data from the American Ultimate Disc League (AUDL) and the National Football League (NFL) and generated accurate Elo models for both the historical analysis of these leagues and the prediction of future games and seasons. Importantly the AUDL does not currently have a publicly available prediction model, which shows the ability of this application to easily generate novel rating systems for under-studied or amateur leagues. By comparing the results of the NFL Elo model generated here to the custom FiveThirtyEight NFL model, this manuscript shows that the application presented can take simple historical data and generate an accurate prediction tool that compares favorably with highly regarded systems.

# 4. Methods

## 4.1 The Tool:

The Elo ratings tool is a shiny app (Chang *et al.,* 2021) written in R (R Core Team, 2022) and fully open source. It is available hosted online (https://sweaver4.shinyapps.io/EloCreationOptimization/) and can also be downloaded and run locally by following the instructions on the GitHub repository (https://github.com/weaversd/Elo_shiny_app).

Practical instructions for using the tool can be found on the documentation page of the app. Briefly, The input files are a .csv file of historical game data, as well as an optional .csv file of future games to predict. The outputs are a variety of plots showing game trends from the historical data (**Figure 2**), Elo parameter optimization (**Figure 3**), historical Elo trends (**Figure 5**), and Elo accuracy (**Figure 4**), as well as .csv files containing the predicted results and historical win percentages for the imported games. The plots are generated with the *plotly* package (Sievert, 2020) which allows for the user to zoom in and out on plots, hover over data points for more information, as well as export the graphics as high-resolution .png files.

## 4.2 Elo Model Parameters:

### *4.2.1 The Elo equation:*

An Elo model consists of a closed point system, where each team or player has a rating, and at the outcome of each game, the loser's rating decreases by some amount and the winner's raking increases by some amount (Elo, 1978). The magnitude of these changes is what determines the model. The ratings of any two teams can also be compared and used to predict an outcome for any game in terms of winning percentages for each team. How accurately the model predicts these winning percentages determines how good a model is.

After each game, the new Elo rating (*R<sub>new</sub>*) is calculated for a team by taking the old Elo rating and adding a term that is calculated from a variety of factors which will be discussed below (**equation 1**).

$$R_{new} = R_{old} + K * MOV * (W - W_e) \qquad (1)$$

The actual outcome of the game is considered with the *W* term, where *W* is 1 if the team won, 0 if the team lost, and 0.5 if the game was a draw. In this way, the new Elo rating will be greater than the old rating if the team won, and less than the old rating if the team lost. $W_e$ is the pre-game win expectancy for that team, which is calculated using **equations 2-3** depending on whether the team was home or away.

$$W_{eHome} = \frac{1}{10^{\left(\frac{D-\delta r}{400}\right)} + 1} \qquad (2)$$

$$W_{eAway} = \frac{1}{10^{\left(\frac{D+\delta r}{400}\right)} + 1} \qquad (3)$$

In these two equations, *D* is a term that is a correction for the possibility of a tie (see **equation 8**) and δ*r* is the difference in Elo scores, adjusted for with a home field advantage factor (**equation 5**). The draw expectancy can then be calculated using the win expectancies of both teams (**equation 4**).

$$D_e = 1 - \left(W_{eAway} + W_{eHome}\right) \qquad (4)$$

$$\delta r = R_{Home} - R_{Home} + HFA \qquad (5)$$

To optimize an Elo model, the combination of parameters that best predicts overall outcome of games must be found. The following four parameters (K value (*K*), margin of victory multiplier (*MOVx)*, home field advantage (*HFA*), and regression between seasons (*REG*)) have the most influence on the success of a model, and so can be optimized in the app for each dataset using a semi-intelligent brute-force method (described in detail below). Each parameter's influence on the model is described in the following sections.

### *4.2.2 K value:*

K value is the most discussed and arguably the most important factor that goes into an Elo model. This value is simply a coefficient that is multiplied by the difference in expected outcome and actual outcome, which determines by what magnitude the Elo rating of each team or player shifts (**equation 1**). Higher K values mean that the model is more sensitive to individual outcomes and more volatile, whereas low K values mean that it takes longer for teams to shift in relative rating, and the model is less sensitive to individual games. It is possible to set a K value that scales by the skill level of the individual players, but for ease and simplicity in this app K is confined to a single value for all teams that can be optimized for each dataset.

### 4.2.3 Margin of victory:

The margin of victory considers the difference in final score between the two teams or players. This is important because generally the margin of victory is correlated with how much better the winner is than the loser. The margin of victory factor (MOV) is multiplied by the K value (**equation 1**) to determine how much the Elo scores of each player change due to the outcome of any particular game. MOV can be calculated with **equation 6**, which is taken directly from the FiveThirtyEight NFL Elo ratings calculation (Silver, Boice and Paine, 2022), with the addition of the *MOVx* term. This term was added to increase the versatility of the overall *MOV* multiplier so that it could be appropriately applied to different sports where the same absolute point differential may have very different implications. For example, a point differential of 3 in basketball may signify a close victory, whereas that same point differential in soccer probably means that the game was not very close.

$$MOV = MOVx * \left[\frac{2.2}{(\delta r * 0.001) + 2.2} * \ln(PD + 1)\right] \quad (6)$$

In the *MOV* equation, *PD* is the absolute value of the home final score minus the away final score or point differential. A higher *PD* means a higher *MOV*. However, the point differential is wrapped within a logarithmic term so that larger point differentials have diminishing returns on the change in Elo scores, as can be seen in **Figure 1**. This is to reduce the effect that massive victories have on the ratings. For example, winning by 50 is only marginally more impressive than winning by 40, but winning by 20 is substantially more impressive than winning by 10. The *MOV* equation also takes into effect the difference between Elo scores of the two teams (δ*r*). This is to reduce the impact of much better teams blowing out much worse teams. For example, a large win of equivalent magnitude is more impressive when the teams had similar Elo ratings, than when the winning team had a much higher Elo rating than the losing team (**Figure 1A**).

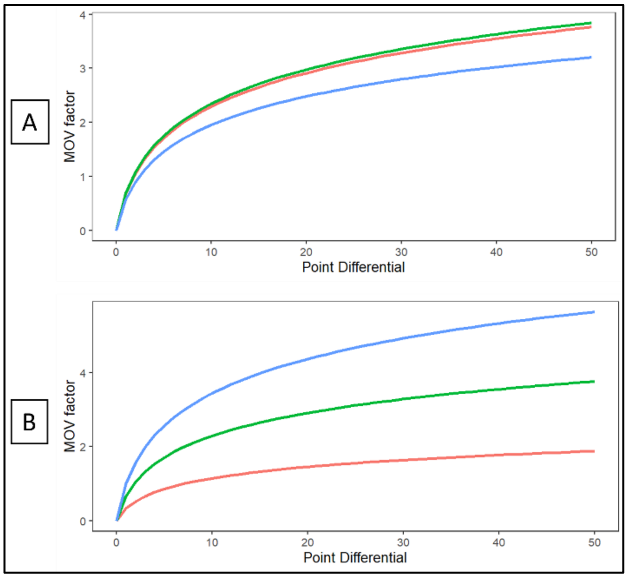

***Figure 1: Margin of Victory parameter visualization.*** *(A) The MOV factor used to adjust Elo scores vs the point differential at three different Elo differences: 100 (orange), 50 (green), and 500 (blue). MOVx is held constant at 1. (B) The MOV factor used to adjust Elo scores vs the point differential at three different MOVx values: 1 (green), 0.5 (orange), and 1.5 (blue).*

The actual factor that is optimized in the app is the margin of victory multiplier (*MOVx*), which is the coefficient that is multiplied by the entire rest of the MOV equation. Depending on the sport or league, the magnitude of victory may have a larger or smaller overall effect on the ratings of the teams, and so achieving an optimal *MOVx* is important for creating a good Elo model. The resulting *MOV* multipliers resulting from three different *MOVx* values can be seen in **Figure 1B**.

### 4.2.4 Home field advantage:

Home field advantage is a factor in almost every sport but can vary in its influence over the game. There are some sports such as baseball, where the home team has an advantage based on the actual rules of the game. In other cases, home field advantage is imparted by travel, familiarity with the environment and weather, the influence of the crowd, and perhaps officiating (Bilalić, Gula and Vaci, 2021). Because of these variations, the amount of impact that home field advantage (or *HFA*) has on the Elo rating system can be optimized from dataset to dataset. This value influences the effective difference in Elo scores between two teams by adding to the differential if the home team is better or subtracting from the differential if the home team is worse (**equation 5**).

### *4.2.5 Regression between seasons:*

In most leagues, there is shuffling of personnel that occurs between seasons, which tends to introduce parity. To account for this, the Elo scores of each team regress towards the starting Elo score between seasons. The amount of regression is a parameter that can be optimized league to league. The new Elo score for each team after a season is complete and before the next season can be calculated with **equation 7**, where *REG* is the percent regression. For example, for each team to regress 20% after each season, *REG* would be set to 0.2.

$$R_{new} = R_{old} - [(R_{old} - 1400) * REG] \qquad (7)$$

### *4.2.6 Adjusting for ties:*

Adjusting for ties can be difficult when using an Elo rating system. In most use cases, the expected win percentages that are calculated for each team add up to 100%, leaving no room for a predicted draw. This isn't an issue for many sports where ties are rare or impossible, but because this tool is designed to be flexible and used with many different datasets, the possibility of a tie must be considered. To do this, a historical tie-rate is calculated ($D_{rate}$), which is the percent of games in the dataset that ended in a draw. It is then assumed that future games will follow this same pattern. The *draw* factor, or *D*, is then calculated using **equation 8** (Craylton, 2015), and substituted into **equations 2-3**.

$$D = 400 * \log_{10}\left(-\frac{D_{rate} + 1}{D_{rate} - 1}\right) \qquad (8)$$

With these considerations, the sum of the home and away win expectancies will be less than one, if $D_{rate}$ is greater than zero. Then for any given game, the Draw expectancy can be calculated with **equation 4**.

### *4.2.7 Other parameters:*

At the beginning of any dataset, the Elo score for each team is set to an arbitrary default, which for this application is 1400 (Commonly used starting values vary between 1000 and 2000). Because the difference between Elo scores is the important parameter in determining winning likelihood, the absolute value of any Elo score doesn't matter. Because each team starts out at the default score, the confidence in predicted win percentage for the first games in any dataset is relatively low. For this reason, the first several results are often not included in the evaluation of an Elo model. In all the analyses presented in this manuscript, the first 10% of games were not considered in goodness-of-fit. This value can be set by the user in the app but in practice, the results change little when altering this value.

Another parameter that can be adjusted is an 'importance multiplier', which assigns more value to games of higher magnitude. In this app, this value is considered a playoff multiplier, and the overall change in Elo scores after a given match is multiplied by a user specified value if the game was a playoff. In practice, the higher the playoff multiplier, the worse the model does at predicting future outcomes. However, it does align with our notion that playoff games are more important, and it results in teams that win the championships often being ranked highest after that season, which is sometimes a desirable outcome.

The scale, or Elo denominator is another arbitrary parameter, almost always set at 400 (as it is in this application). This value changes how much 'power' a specific difference between two players has in determining winning outcome. However, the value of the Elo denominator within any particular Elo model does not matter. In his original publication (Elo, 1967), Elo chose 400 so that the resulting ratings for chess players would be on the same order of magnitude as the previous system by Harkness (Soltis, 1993).

## 4.3 Optimization Method:

The optimization of the Elo parameters is fully automated in the application and uses a semi-intelligent iterated brute-force technique. The input for this optimization is a range of values for each of the four optimization parameters described above. The application then creates 5 datapoints for each parameter evenly spaced throughout this range (minimum, maximum, and the 25th, 50th, and 75th percentile values). It then uses every possible combination of these Elo parameters (625 possibilities), and taking the historical data provided, calculates a win percentage probability for each team in each game, updating the Elo score for each team based on the actual result of each game. Following this analysis, a value called the scaled residual score is calculated for each combination of Elo parameters. This value comes from a calibration plot as described by FiveThirtyEight  (Boice and Wezerek, 2021).

To calculate the scaled residual score, each game is placed in a bin based on the home team's expected win percentage, with a bin width of 5%. Then the actual win percentage of games within each bin is calculated and plotted against the expected win percentage of that bin. The difference between actual win percentage and expected win percentage for each bin is multiplied by the number of games within that bin and divided by the total number of games. The absolute values of these scaled values are summed to produce an overall score, where a lower value is better, and a higher number is worse. If a model perfectly predicted each game, the expected win percentages would equal the actual win percentages, and the score would be zero.

For each possible permutation of Elo parameters, this scaled residual score is calculated, and the permutations are ordered from best score to worst. Then the top *n* permutations (a user specified value with a default of 10) are taken, and the median value for each parameter is chosen as the optimal value from the possible five options. This value becomes the 'center' for the next iteration, and the application creates a new range for each Elo parameter that is a certain percent smaller than the previous range based on a user specified value called convergence multiplier (default 0.5). A new minimum, maximum, and three intermediate percentiles are created based on these values for each Elo parameter, and the process is repeated for a user-specified number of iterations. This results in a convergence to an optimal value for each Elo parameter that can be used in a custom Elo model for the dataset provided. Overall, this method can be thought of as creating a 4-dimensional heatmap with each optimizable parameter as one of the axes. Then the best scoring position from this heatmap becomes the center of a new heatmap, with more nuanced values chosen for each axis.

### 4.4 Data:

NFL data was downloaded on 10/6/2022 from FiveThirtyEight.com as a .csv file (https://github.com/FiveThirtyEight/data/tree/master/nfl-elo). The format of the csv file was adjusted to match the format for the Elo predictor in Microsoft Excel, and only games belonging to seasons 2000-2021 were considered.

AUDL data was scraped from the AUDL website (https://theaudl.com) on 9/25/2022 using code deposited in the GitHub for this project. The format of the .csv file was adjusted to match the format for the Elo predictor in Microsoft Excel, and games from 2014-2022 were considered.

# 5. Results and Discussion

## 5.1 Use Case – American Ultimate Disc League 2014-2022

To demonstrate the utility of generating and optimizing an Elo ratings system for an under-analyzed league, game data from the American Ultimate Disc League (AUDL) was used from 2014-2022. The Elo model was generated with historical data from 2014-2021, and the games from 2022 were predicted using this model, and then compared to the actual results from 2022.

### *5.1.1 Game analysis:*

The Elo generating app outputs a variety of plots to evaluate the trends in historical games (**Figure 2**) in the ‘Game Stats’ tab. Histograms of the final scores for each game as well as the margins of victory are produced. In leagues like the AUDL, where goals are worth one point each, the final score distribution is relatively gaussian (**Figure 2A**) and the margin of victory distribution follows an exponential decay from one (**Figure 2B**). In other leagues and sports, such as the NFL, scores are worth different numbers of points (touchdowns are 7, field goals are 3, etc). This causes the final score and margin of victory histograms to display sport-specific trends (**SI Figure 1**).

A home field advantage can also be seen in the final score distributions of the home team vs the away team, where the home team score distribution is shifted approximately 2 points higher than the away team distribution (**Figure 2C**). Games where the away team won also have a margin of victory distribution that is skewed lower than the margin of victory distribution in games where the home team won (**Figure 2D**). This initial game analysis also produces a table of all the teams in a dataset with their W/L/D records calculated, along with point differentials and win percentages for easy analysis.

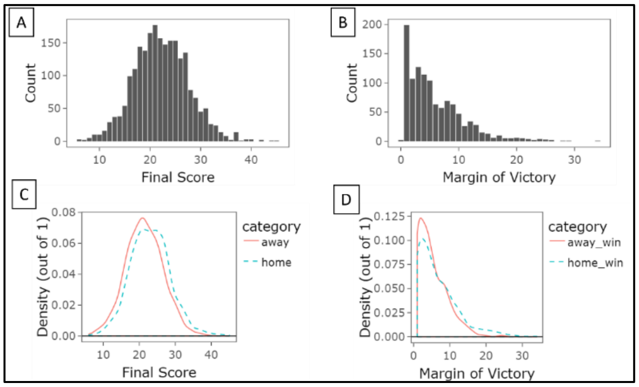

***Figure 2: Margin of Victory and Final Score Histograms from AUDL game data 2014-2021*** *(A) Histogram of Final scores. (B) Histogram of margin of victory. (C) Density plot of final scores for home and away teams. (D) Density plot of margin of victory in games where the home team won vs games where the away team won. All figures produced by the Elo app, with legends left in for clarity.*

### 5.1.2 Elo Optimization:

The Elo optimization of the 2014-2021 AUDL seasons (1124 total games) took approximately 50 minutes with the application using 8 iterations (~6 minutes per iteration) on a PC with 128 GB of installed RAM. There are certainly more efficient methods to optimize multidimensional variables, such as Design of Experiment (DoE) workflows using response surface methodology (RSM) or neural networks (Elfghi, 2016). However, these methods can vary significantly in their effectiveness, and require more nuanced optimization. It is very difficult to repeatedly obtain useful results with RSM for example without a knowledgeable human supervising the process. For these reasons, a brute force approach was utilized which, while less efficient, is more predictable and robust to a wide variety of datasets and requires minimal supervision.

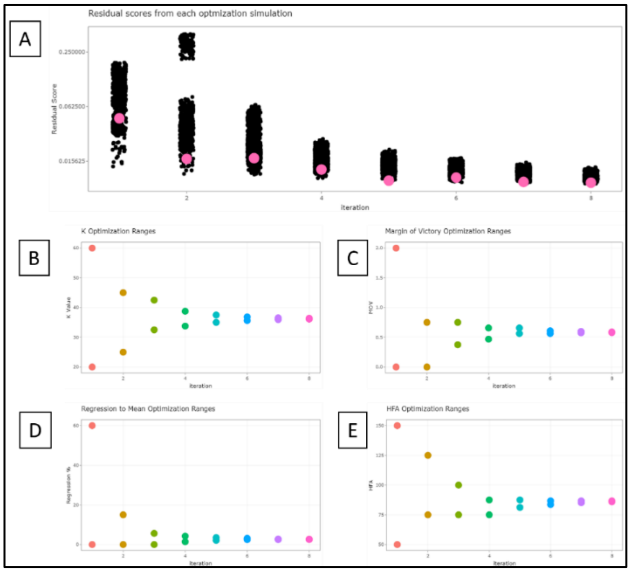

***Figure 3: Elo Optimization Results for AUDL data 2014-2021.*** *(A) Residual calibration score of the optimization trials vs trial iteration, showing an improvement of score over each iteration, with the center value for the next iteration shown in pink. (B-E) The ranges (minimum and maximum values plotted) trialed for the Elo optimization parameters vs trial iteration as follows: (B) K value, (C) margin of victory multiplier, (D) % regression to the mean between seasons, and (E) home field advantage.*

The results from the optimization can be seen in **Figure 3**. The Elo model is optimized using the absolute scaled residual score as a metric, where a lower score is better with zero being the best score. This value corresponds to the scaled absolute percent difference between expected win percentage and actual win percentage in calibration plots such as those seen in **Figure 4A-B**. **Figure 3A** shows the residual score of each replicate tried for each iteration (1-8 in this case) during the optimization as black points. In later iterations, the score improves, and the residual scores also begin clustering, indicating convergence on ideal parameters. The pink dots in this plot represent the median top value which is used for the center of the following iteration. **Figure 3B-E** show the ranges of values used in each iteration for the four optimization parameters. These also show convergence on ideal conditions over the iterations.

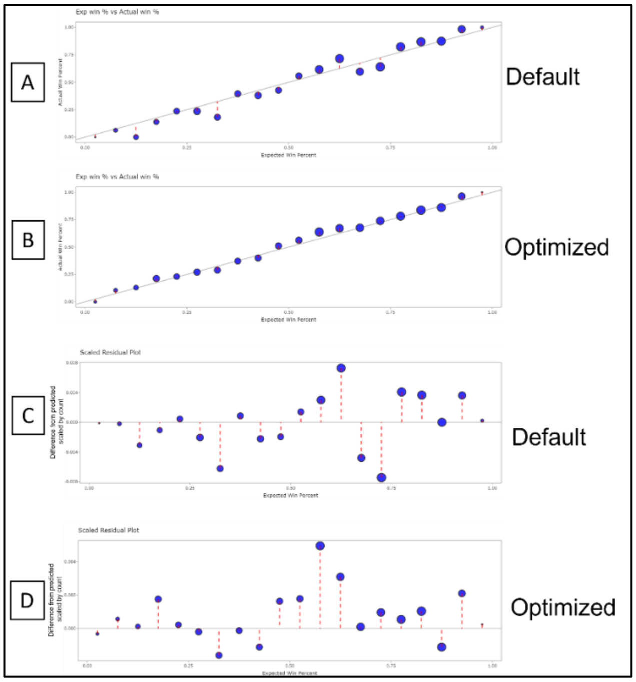

***Figure 4: Expected win vs actual win % calibration improves with Elo parameter optimization.*** *(A-B) Calibration curves of expected win percent vs actual win percent with games grouped into bins of 5% width before(A) and after (B) optimization. Points are scaled to the number of games in each bin, and the difference from a perfect 1:1 correlation is shown with red dotted lines. (C-D) the magnitude of the residuals from the y=x line in A&B plotted against the expected win percent in each bin, with each magnitude scaled by the number of games in each bin before (C) and after (D) optimization. Note the decrease in y-axis scale after optimization.*

The improvement in calibration score is evident when comparing the calibration curves and scaled residual score plots for the model before and after optimization (**Figure 4**). The absolute scaled residual score improved from 0.054 with the default values to 0.024 after optimization, a more than two-fold improvement. Importantly, the default values are still intelligently chosen numbers for a standard Elo model, so starting with a completely random Elo model would show even more stark improvement. For example, an Elo model with the default K value and % Regression between seasons but disregarding home field advantage and margin of victory produces a score of 0.108, four times worse than the optimized model. The optimization is achieved here with almost no human input. In fact, in this case there was no alteration of the default ranges for Elo parameters. This is extremely important in the accessibility, robustness, and broad application of this Elo optimization app.

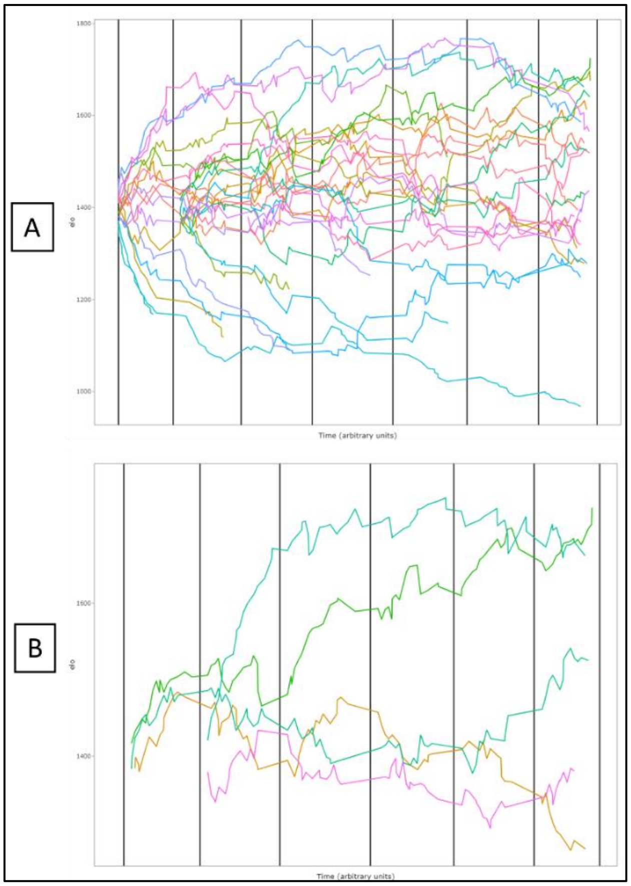

***Figure 5: Elo scores over time (2014-2021) using optimized Elo parameters.*** *(A) All teams in the AUDL overlaid, each in a different color. (B) Teams from the current AUDL South Division. In both plots the vertical black lines indicate the divisions between seasons. The plots in the app are interactive, allowing users to select specific traces to display while hiding others, and allowing the user to zoom in or mouse over data to see more*

*information in pop-up boxes. Legends are shown in the app but removed here for clarity.*

The result of the optimization process is a set of Elo parameters that can both be used to predict future games, as well as look at historical trends of team performance. The app produces a plot of Elo score for each team in the dataset over time using the optimized Elo parameters. Examples of these plots can be seen in **Figure 5**. The plots in the app are interactive, allowing users to select specific traces to display while hiding others (**Figure 5B**), and allowing the user to zoom in or mouse over data to see more information in pop-up boxes. More examples using NFL data are shown in **SI Figure 2**.

### *5.1.3 Predicted Results and Evaluation:*

To evaluate the Elo Model developed based on the 2014-2021 AUDL seasons, the 2022 AUDL season was predicted using the optimized Elo parameters. The model predicted the correct outcome in 74% of 2022 games, a reasonable percentage. The scaled residual score for the 2022 games with the optimized parameters was 0.084, whereas the same value with the default parameters was 0.115; evidence that the optimization improved the predictive power of the Elo model (**Figure 6**). This value is higher than the scaled residual score for the 2014-2021 data which is to be expected for several reasons, primarily because the 2014-2021 score is for data that the model was trained on, while the 2022 score comes from experimental data.

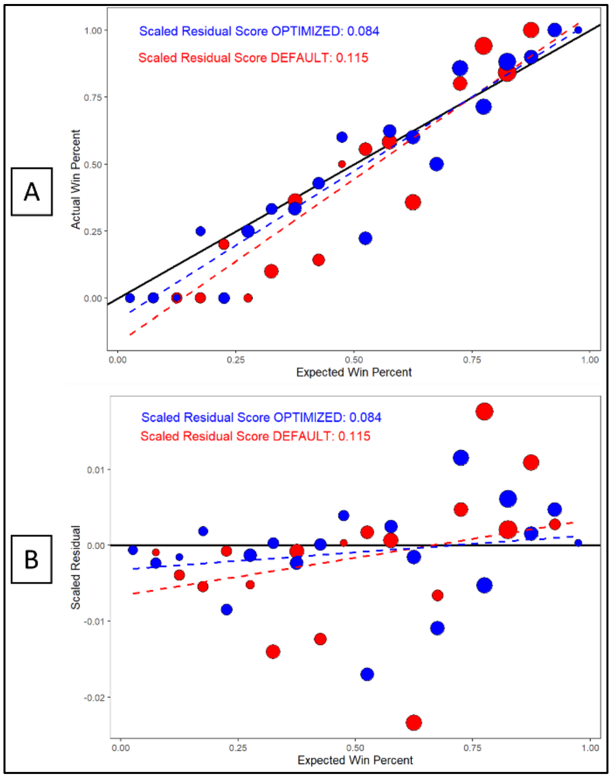

***Figure 6: 2022 AUDL season predicted results compared to actual results.*** *Blue data is after Elo optimization, and Red data is predictions with default values. (A) Calibration curves of expected win percent vs actual win percent with games grouped into bins of 5% width Points are scaled to the number of games in each bin. Black line is at y=x, and dotted lines are best fits for the data. (B) the magnitude of the residuals from the y=x line in A plotted against the expected win percent in each bin, with each magnitude scaled by the number of games in the bin.*

The predicted win percentage based on the model for each AUDL team in the 2022 regular season was compared to the actual win percentage for each team, and the Pearson correlation coefficient between these two variables was 0.75 (**Figure 7**). While not a perfect correlation, this is evidence that the optimized model has reasonable predictive power for future games, a relevant feature in fields such as sports gambling. Additionally, this evaluation provides an excellent way to determine teams that over- or under-achieved during a given time, which can be used to evaluate the performance of personnel and guide administrative decision making. For example, during the 2022 AUDL season the Legion, Nitro, and Radicals were the most underachieving teams, whereas the Sol, Phoenix, and Summit overachieved.

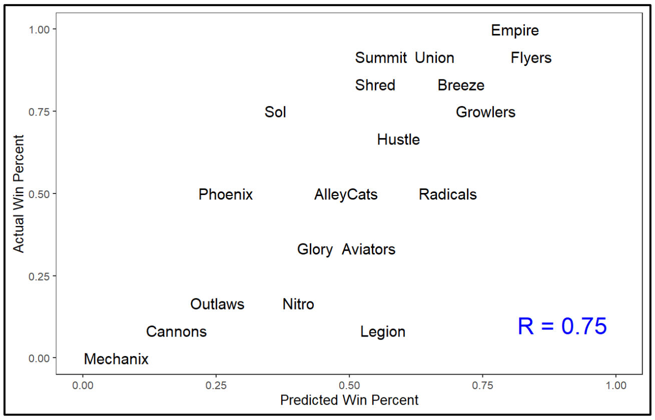

***Figure 7: Predicted win percentages for 2022 AUDL regular season vs actual win percentages. Pearson Correlation Coefficient shown.***

## 5.2 Comparison to other tools:

To evaluate how the application described in this manuscript performs compared to other tools, the NFL predictions for the 2021 season from FiveThirtyEight were compared to the 2021 season predictions generated from an optimized Elo model in the app. The app was trained on NFL data from 2000-2020 with the default optimization settings, and the resulting Elo parameters were used in a model to predict all the games from the 2021 season. Plotting the predicted winning percentage for the home team in each game vs the equivalent predicted winning percentage from the FiveThirtyEight model shows excellent correlation between the two models (**Figure 8A**), with an $R^2$ value of 0.99 and a slope of 1.026 for the best fit line (A perfect match would give a slope of 1.000). This shows that the optimized Elo model is providing win percentages that are very close to the FiveThirtyEight win percentages, which come from a model that is custom tailored to the NFL.

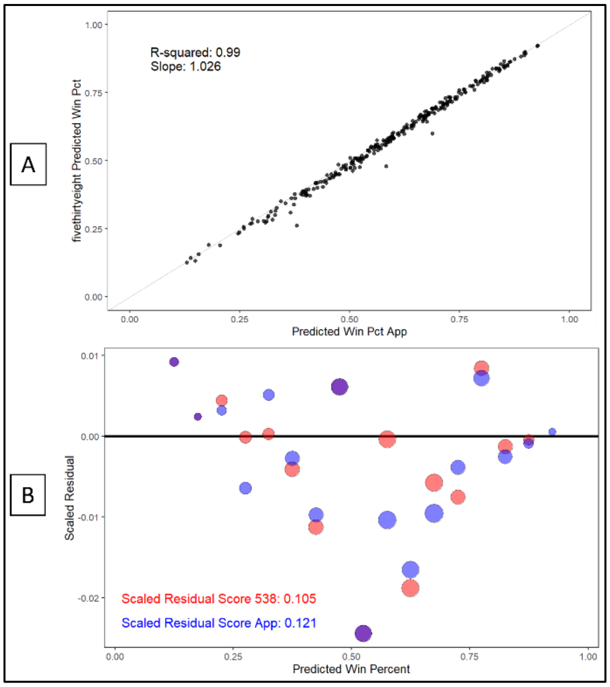

***Figure 8: FiveThirtyEight NFL predictions vs App Elo predictions for the 2021 NFL season.*** *(A) Each game plotted by the predicted win percentage for the home team from FiveThirtyEight's model (y-axis) vs the Elo model generated by the app (x-axis). (B) Residual plot for the calibration plot between expected win percentage and actual win percentage scaled to the number of games in each bin for FiveThirtyEight predictions (red) and the app predictions (blue).*

The scaled residual score for the FiveThirtyEight predictions is slightly better than the scaled residual score for the app generated model (0.121 vs 0.105) as can be seen in **Figure 8B**. However, this difference is marginal, and a close look at the plot shows that there are several bins where the app generated Elo model slightly outperforms the FiveThirtyEight model. Importantly, the FiveThirtyEight model is generated with a significant amount of optimization specifically for the NFL, with substantial human oversight. On the other hand, the model generated with this Elo app was created generically with almost no human input. Overall, this application appears to be comparable to a highly regarded custom prediction tool and can generate Elo models without any need for the user to code, perform complicated data wrangling, or even know much about the Elo system.

# 6. Conclusions

This manuscript presents an Elo model generation application that can take generic game data from any sport or league, and generate an accurate, optimized Elo model that can be used for future game prediction and historical analysis. The models generated with this tool consider margin of victory, home field advantage, and regression between seasons, and they compare favorably with state-of-the-art prediction tools that have been tailored to specific sports and leagues. This app is continually under development and will continue to incorporate new features. Some of the planned improvements include the ability to optimize simultaneous but independent Elo systems for different divisions within one dataset (two different models for the American and National league in the MLB for instance), the option to have a moving K value based on the Elo score of each team, and the potential to use different optimization metrics. This tool should be of use to analysts interested in predicting future seasons of their sports, and especially those who are interested in under-analyzed sports without custom tools available. Additionally, administrative personal may be interested in looking at performance over time, and how their team has over- or under-achieved during different stretches. It is my hope that data analysts and amateur sport enthusiasts will enjoy experimenting with this application and will use it to answer interesting questions about their league or sport of interest. Comments, suggestions, and bugs can be opened as issues on the GitHub repository for this project.

# 7. References

Barrow, D., Drayer, I., Elliott, P., Gaut, G. and Osting, B. (2013) 'Ranking rankings: an empirical comparison of the predictive power of sports ranking methods', 9(2), pp. 187-202. doi: 10.1515/jqas-2013-0013.

Bilalić, M., Gula, B. and Vaci, N. (2021) 'Home advantage mediated (HAM) by referee bias and team performance during covid', *Scientific Reports,* 11(1), pp. 21558. doi: 10.1038/s41598-021-00784-8.

Boice, J. and Wezerek, G. (2021) *How Good Are FiveThirtyEight Forecasts?* Available at: https://projects.fivethirtyeight.com/checking-our-work/ (Accessed: Oct 17, 2022).

Chang, W., Cheng, J., Allaire, J.J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A. and Borges, B. (2021) *shiny: Web Application Framework for R*.

Craylton. (2015) 'Soccer Wizard: Predicting draws with the Elo model', *Soccer Wizard,* Sunday, 5 April. Available at: http://soccerwizardbetting.blogspot.com/2015/04/predicting-draws-with-elo-model-part-2.html (Accessed: Oct 17, 2022).

Cunningham, D. (2021) *Developing a Generalized Elo Rating System for Multiplayer Games.* Available at: https://towardsdatascience.com/developing-a-generalized-elo-rating-system-for-multiplayer-games-b9b495e87802 (Accessed: Oct 17, 2022).

Elfghi, F.M. (2016) 'A hybrid statistical approach for modeling and optimization of RON: A comparative study and combined application of response surface methodology (RSM) and artificial neural network (ANN) based on design of experiment (DOE)', *Chemical Engineering Research and Design,* 113, pp. 264-272. doi: https://doi.org/10.1016/j.cherd.2016.05.023.

Elo, A.E. (1978) *The rating of chessplayers, past and present* Arco Pub.

Elo, A.E. (1967) 'The Proposed USCF Rating System, Its Development, Theory, and Applications', *Chess Life,* XXII(8), pp. 242-247.

Fischer-Baum, R. and Silver, N. (2022) *The Complete History Of The NBA.* Available at: https://projects.fivethirtyeight.com/complete-history-of-the-nba/ (Accessed: Oct 18, 2022).

Glickman, M.E. (1995) 'A comprehensive guide to chess ratings', *American Chess Journal,* 3(1), pp. 59-102.

Hvattum, L.M. and Arntzen, H. (2010) 'Using ELO ratings for match result prediction in association football', *International Journal of Forecasting,* 26(3), pp. 460-470. doi: https://doi.org/10.1016/j.ijforecast.2009.10.002.

R Core Team (2022) *R: A language and environment for statistical computing* Vienna, Austria: R Foundation for Statistical Computing.

Sievert, C. (2020) *Interactive web-based data visualization with R, plotly, and shiny* Chapman and Hall/CRC.

Silver, N., Boice, J. and Paine, N. (2022) *How Our NFL Predictions Work.* Available at: https://fivethirtyeight.com/methodology/how-our-nfl-predictions-work/ (Accessed: Oct 7, 2022).

Soltis, A. (1993) 'What's your Elo?', *Chess Life,* 48(7), pp. 19-20.

SpreadsheetSolving. (2016) 'Build a sports league power ratings spreadsheet', *SpreadsheetSolving,* . Available at: https://spreadsheetsolving.com/sports_league_power_ratings/ (Accessed: Oct 17, 2022).

Stefani, R. (2011) 'The Methodology of Officially Recognized International Sports Rating Systems', 7(4). doi: 10.2202/1559-0410.1347.

Vaughan Williams, L., Liu, C., Dixon, L. and Gerrard, H. (2021) 'How well do Elo-based ratings predict professional tennis matches?', 17(2), pp. 91-105. doi: 10.1515/jqas-2019-0110.