The UEFA Champions League is a yearly international club soccer tournament featuring the best teams in Europe. Several machine learning models have been used to predict the results of sports games. Bunker and Thabtah [1] found that using an ANN with backpropagation performs with a slightly higher accuracy than expert predictions, beating ESPN sportscasters 75% to 63% accuracy when predicting NFL games. According to Zhang et al [2], RNNs can be unable to capture long-term dependencies. They recommend the use of LSTM, a variant of RNN, to get more accurate results. Yue et al [3] discuss the use of a Bayesian approach called GLICKO to predict strength of tennis players and therefore outcomes of tennis matches. In our particular use case, the unique format of the Champions League allows for more complex predictions of team knockout stage performance based on domestic league and group stage performance.

We want to create a model that accurately predicts the team knockout stage based on scores. Despite soccer being the most popular sport worldwide, more money is spent on analytics for the NBA and NFL in the United States. The field of predicting soccer matches is scarce. Additionally, since UEFA only has European teams, it receives less resources than other soccer events such as the World Cup. An accurate model could help teams allocate their resources, knowing which factors are important for a winning team. It can also allow a team to see where weaknesses lie, allowing them to improve on their deficiencies.

We obtained our data from [https://www.uefa.com/uefachampionsleague](https://www.uefa.com/uefachampionsleague/history/seasons/1991/). Using this resource, we collected data from 1981 to 2021 on club teams and how they finished (winner, runner-up, etc.), points scored by the winner of the game, and points scored against the winner of the game. We also took note of the location of the game, but we did not use that data in this round. After doing this, we cleaned the data, as some years used a point system similar to what is used in World Cup brackets. We also filled in some missing information and made the years uniform. Once we finished cleaning all the data, we moved on to pre-processing. In order to fit the data to our model, we manipulated the score data, changing it from points scored for each team to the point difference between teams. This allows our model to predict the point difference when given 2 teams without knowing the winner in advance. Then, we parsed the team names and converted them into numerical values that our model could understand.

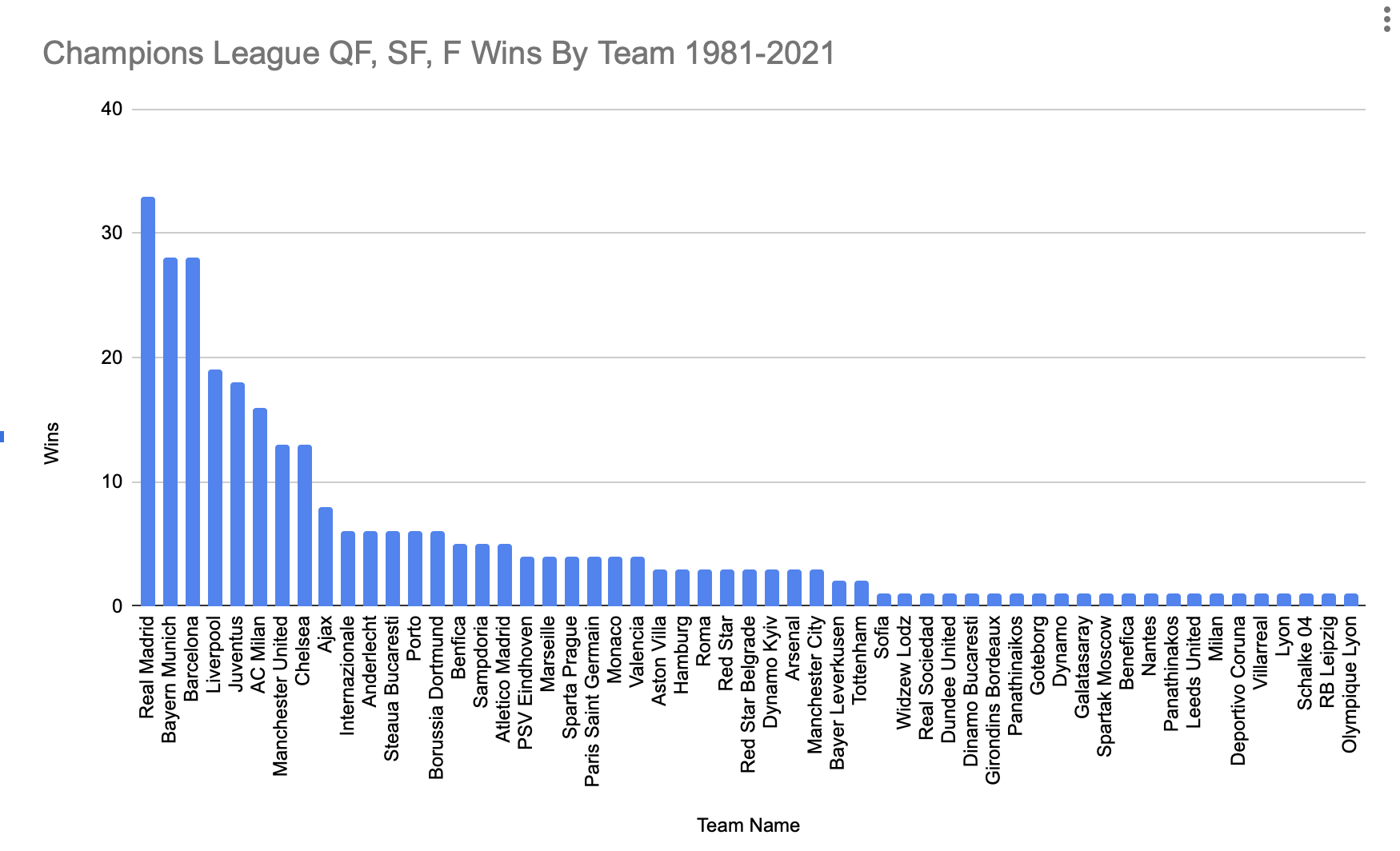
For our first model, we plan to implement a Recurrent Neural Network (RNN). RNNs excel at time series analysis (predicting future results based on past data) because they are able to capture temporal dependencies. This method of implementation should be able to provide significant outputs based on our training data. In order to implement the RNN, we can utilize the TensorFlow library. The library has pre-existing architecture for implementing a simple RNN, GRU, and LSTM, so we can alter the type of neural network we’re using in future implementations without significantly altering the code or data. This allows us to get initial results in our first iteration while maintaining flexibility for later iterations.

In our first model, we predicted the score differential primarily through past games. We took a random sample from the data, splitting it into testing and training data. Our mean squared error on our training set was 1.0521, with the loss per epoch being around 1.2 every iteration. The model yielded similar results on the testing data. In future iterations, we hope to use the most recent years as our testing data and past years for training.

Based on the statistics obtained, it does seem that the past performance does have a strong indication on how a team will perform in the future. This is demonstrated through a plot of the original data, as the same teams tended to perform well overtime. As shown in Figure 1, the same teams tend to appear every year in the tournament, and those same teams tend to win.

However, the model does need improvements. Currently, the model is able to predict the score differential when two teams are inputted, but the score is typically within two goals. We hope to improve the accuracy with future iterations and additional features.

In our next iteration of the project, there are several additional factors we want to consider. This includes looking into the amount of players that are returning from the previous season, the distance each team travels to the stadium, and individual games from the season. We also want to work with finding a good weight for the years, so more recent seasons are more relevant to the prediction of future seasons. Additionally, we located a site that contained country coefficients based on their wins, losses, and points, so we may consider using this factor to make predictions.

**Figure 1**

**Literature Cited**

[1] R. P. Bunker and F. Thabtah, “A machine learning framework for sport result prediction,”

*Applied Computing and Informatics*, vol. 15, no. 1, pp. 27–33, 2019.

doi:10.1016/j.aci.2017.09.005

[2] Q. Zhang *et al.*, “Sports match prediction model for training and exercise using

attention-based LSTM network,” *Digital Communications and Networks*, vol. 8, no. 4,

pp. 508–515, 2022. doi:10.1016/j.dcan.2021.08.008

[3] J. C. Yue, E. P. Chou, M.-H. Hsieh, and L.-C. Hsiao, “A study of forecasting tennis matches

via the GLICKO model,” *PLOS ONE*, vol. 17, no. 4, Apr. 2022.

doi:10.1371/journal.pone.0266838