Predicting Chess Game Winner After x Moves

***Abstract*—The game of chess has been greatly studied in the machine learning and artificial intelligence fields in order to either build chess playing AI's or predict game winners. In this paper, we aim to model human behavior in a game of chess in order to predict the winner at various points throughout the game. We hypothesize that the state of the board is able to convey information that a classifier can use to reliably predict the winner of the match. Through this move-dependent feature selection, which includes features that would be visible to a standard recreational player throughout the game, we show that the current state of the board possesses powerful information which we were able to use to make predictions during the games. Using a Random Forest Classifier, we obtained a 58% accuracy 30 moves before the end of the game, and a 70% accuracy 10 moves before the end of the game. The results are potentially globally useful to those involved in planning and pattern recognition games, as well as applications of these games.**

***Index Terms*—Prediction, Classification, Chess, Feature Extraction, Machine Learning, Random Forest**

I. Introduction

THE game of chess presents an interesting area for machine learning research, as it is a game primarily about planning and pattern recognition. The world's best chess players can plan ahead about 20 moves, incorporating the various piece movement patterns as they progress through a game. With a maximum of 32 pieces on the board, 64 checkered squares, and each piece having the ability to move between one and seven spaces, there are estimated around 10^{40} possible sensible games. With an average game length of 80 ply (40 moves) and each move having a large number of possible move combinations as the game progresses into a distinct path, each game and move evolves into one that has probably never been played before.

This paper investigates the degree of accuracy to which a chess game outcome can be determined at various points in the game based on features about the state of the board. In particular, the main interest of the paper is in modeling human chess player behavior move-by-move, rather than better understanding optimal strategy. Features about the state of the board that are evident to the average player, such as value of pieces left on the board (using standard chess piece values), play a critical role in planning ahead and predicting move patterns throughout a game and are used in the models in this paper.

Understanding how features about the state of the board relate to game outcome has applications in several different areas. For instance, such predictive models could be of interest to different stakeholders within the chess playing industry and contribute to game informational statistics. These predictions could also allow players of all skill levels to look back and analyze shifts in momentum through in-game statistics and help them better plan and recognize patterns in future games. In addition, with the rise of the sports betting industry, formulating live odds for chess games may be of interest to those betting, as well as contribute to increasing general interest in the game.

II. Related Works

While much of the past research on chess has focused on AI systems built to play the game at a high skill level, there also exists some research on predicting game outcomes. For example, "Online Prediction of Chess Match Result" discusses how the move based information obtained from the data-set for training could be used for feature extraction. By looking at games as a whole, padding was introduced to represent the games as having the same length. Additional features were used for each possible move in each possible game [1]. This paper predicts on a game-by-game basis (i.e. each game is fed into the model as training example).

Additionally, "Statistical Analysis on Result Prediction in Chess" looks at whether each move is a positive or negative move for each player using Naive Bayes classification and predicts winners based on their own point calculations. Using opening move sequences and move evaluation statistics as part of their features, they predicted game outcomes, including ties[3]. This paper predicted on a game-by-game basis as well.

After analyzing previous studies, we predicted on a move-by-move basis, taking into account the given state of the board at each move (described in section III below). This method incorporated move-specific features that varied greatly from previous research conducted on predicting chess game winners.

III. Proposed Method

A. Dataset Overview

The dataset we obtained was from Kaggle.com, and they obtained the data from the popular online chess site Lichess.org. It contains 20,058 games worth of data from the top 100 users on the site (at the time the dataset was created). The dataset has 16 features per game, including information on each player's opening strategy and game moves, as well as meta-information about the matches, such as the start time, end time, and player ratings [2].

B. Length of Games in the Dataset

The distribution of the length of the games in the dataset (by number of moves) can be seen in Fig. 1 below.
Some dataset statistics related to the length of games include:

Statistic	Value
Average length of game	59 moves
Median length of game	54 moves
Minimum length of game	1 move
Maximum length of game	349 moves
Standard deviation for the games	34 moves

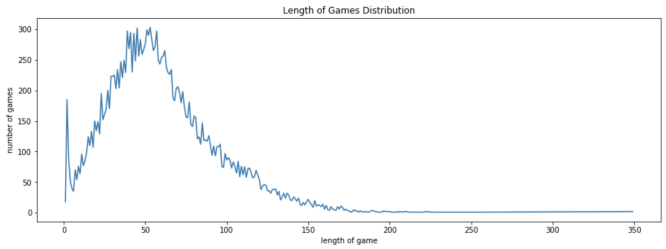

Fig. 1. Length of Games Distribution: This distribution is skewed right, with most games having just over 50 moves.

Based on this distribution and the above statistics, we chose to consider only games with more than 30 moves and only consider moves after the 30th move. This is because the early moves in the game introduce a significant amount of noise, given that early in the game no player has a significant advantage.

C. Feature Extraction and Selection

We extracted features per individual move, as compared to the entire game, from the data-set and used it to train our model. The move notation and meaning are given in Appendix C, and the piece symbols are given in Appendix A. These features encapsulate the state of chess board at the i^{th} move of the game. The features are divided into 4 main categories, namely, check, total value of pieces left, accessible areas of the board and value of vulnerable pieces. These categories for features represent the important parameters which affect the game of chess, as explained below.

1) ***Check****:* This feature states whether the i^{th} move in the game checked the opponent player's king.

2) ***Total value of pieces left****:* This feature is equal to the sum of piece values of the pieces that the opposing player has remaining on the board. The standard chess piece provided in Appendix B are used to calculate the total value of pieces left on the board. For example, if for the white player there are 3 pawns, 2 bishops, a rook, and the queen on the board then the total value of pieces left after the black player's turn at the end of the i^{th} move would be $3 \cdot 1 + 2 \cdot 3 + 5 \cdot 1 + 9 \cdot 1 = 23$.

3) ***Value of vulnerable pieces****:* This feature also uses the piece value system referenced in the above subsection. However, this feature represents the cumulative value of all of a players pieces that are able to be captured by the opposing player in their next move. We chose to represent this as two different features, one for the value of white pieces that were vulnerable, and one that represents the same information for black. This is shown in Fig. 2 for the white pieces and Fig. 3 for the black pieces.

4) ***Accessible areas of the board****:* This feature represents how many unique squares of the board were available for each player to move into on their next turn. We once again chose to represent this as two different features, one for white and one for black. The distribution for the white pieces is given in Fig. 4, and the distribution for the black pieces is given in Fig. 5.

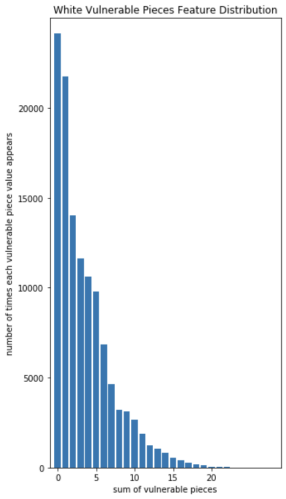

Fig. 2. White Vulnerable Piece Distribution: There is a negative relationship between the sum of the white vulnerable pieces and the number of times each sum appears on the board.

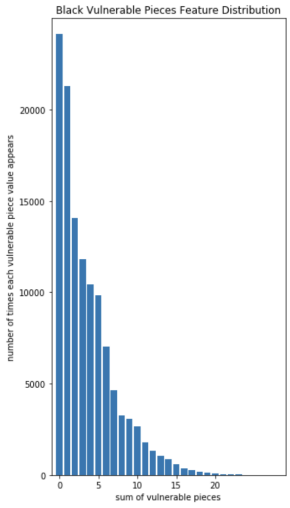

Fig. 3. Black Vulnerable Piece Distribution: There is a negative relationship between the sum of the black vulnerable pieces and the number of times each sum appears on the board.

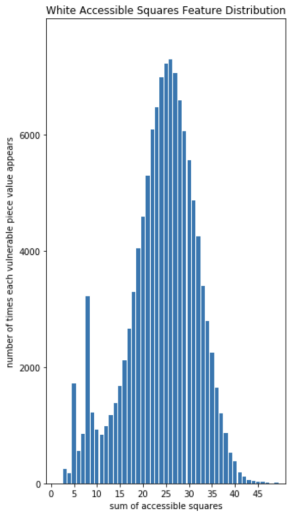

Fig. 4. White Accessible Squares Distribution: The sum of the white accessible squares is approximately normally distributed with the number of times each vulnerable piece value appears, with a two major spikes towards the left of the graph.

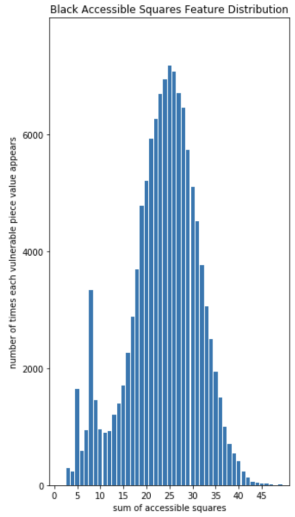

Fig. 5. Black Accessible Squares Distribution: The sum of the black accessible squares is approximately normally distributed with the number of times each vulnerable piece value appears, with a two major spikes towards the left of the graph. As expected, it looks very similar to Fig. 4 on the left.

D. Implementation of Features

Given these features, we calculated the value of each for each game for each move. To do so, we used string parsing to transform the move information (see Appendix C) into matrix representations of the board. The first representation keeps track of the color of each piece and the second representation keeps track of the name of each piece (rook, bishop, etc). We updated these matrices after each move to encapsulate the state of the board at that specific move and then calculated the features based on the matrices for feeding into the models. The set of six features formed a training example for each move of each game, corresponding to a binary label of the game outcome, as defined below.

Since each move captures the state of the board at that given point in time, the moves do not have to be passed in sequentially as individual moves as the game progresses.

E. Labels

For game prediction, we used a binary label for whether the player who just moved ultimately wins the game. For instance, if it is white's turn and white ultimately wins the game, this label is 1 and if black wins, this label is 0. This means that for any given game, the labels alternate [010101...] or [101010...] across all of the moves of that game based on whether or not white wins the game.

F. Training and Evaluation

In developing a model for predicting chess match outcome at various stages of the game, we tried training several different models with the above discussed features. We expected Random Forest models to perform best for this task, but also trained a Naive Bayes model since a Naive Bayes model was used in "Statistical Analysis on Result Prediction in Chess"[3].

For each of the models, we designated 80% of the matches as "train matches" and the other 20% as "test matches". For each of the test matches, we considered the final 30 moves of each game, meaning that we had $6562 * 0.8 * 30 = 157470$ training examples. For each of the test games however we only considered one move, and since our baseline for the final accuracy was 65.6% as obtained in "Online Prediction of Chess Match Result"[1], which was obtained 10 moves before the end of the game, we also used the accuracy on the 10^{th} to last move for model comparison.

After deciding on Random Forest, we then ran the model 26 times, each time measuring accuracy at a different move within the final 30, starting from the 30^{th} move prior to the end of the game, up until the 4^{th} move from the end, due to the fact that our aim was to determine the prediction accuracy at different stages of the game. Thus, for each iteration of the model, we had $6562 * 0.2 = 1312$ test examples.

G. Results

Using these models and feature extraction method, we were able to predict the outcome of the games 10 moves before the end of the game with better accuracy than our baseline accuracy of 65.6% [1].

The following table provides training and test accuracy for models tested. In particular, the reported training and test accuracies are the percentage of games correctly predicted 10 moves before the end of the game.

Model	Training Accuracy	Test Accuracy
Naive Bayes	68.6	64.8
Random Forest	78.4	69.6

Accuracies for the Models Tested: The Random Forest Model gave a final test average of almost 70%, which occurred 10 moves before the end of the game.

As hypothesized, the Random Forest model performed the best out of models tried. 10 moves before the end of the game, we were able to predict the winner with accuracy of 70%. Comparing this to the baseline found in "Online Prediction of Chess Match Result," we were able to get a higher accuracy 10 moves before the game as compared to their value of 65.6%. This is shown in Fig. 6 below.

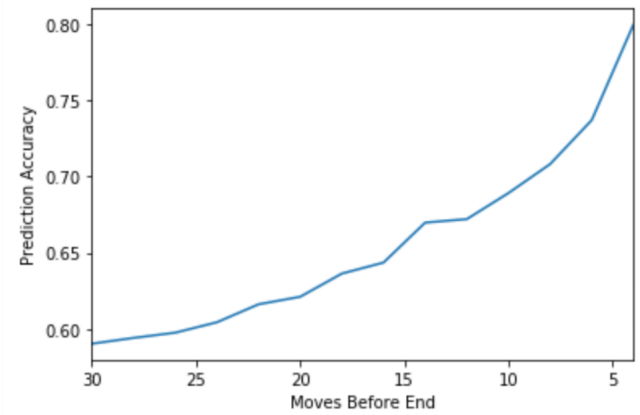

Fig. 6. Accuracy x Moves Before the End of the Game: As the games progress, the winner predictions get more and more accurate. At 30 moves before the end of the game, the winner can be predicted with an accuracy of 58%. At 10 moves before the end of the game, the winner can be predicted with an accuracy of 70%.

The most important feature in making these predictions was the value of pieces remaining on the board that belong to the opposing player. The full feature importance breakdown can be seen in Fig. 7.

IV. Conclusion

A. Summary

In summary, by considering additional features about the current state of the board beyond just the progression of moves, we were able to improve accuracy of prediction compared to other papers read. This suggests that there are features about the state of the board beyond just the series of moves that are important in understand which player is ahead in the match

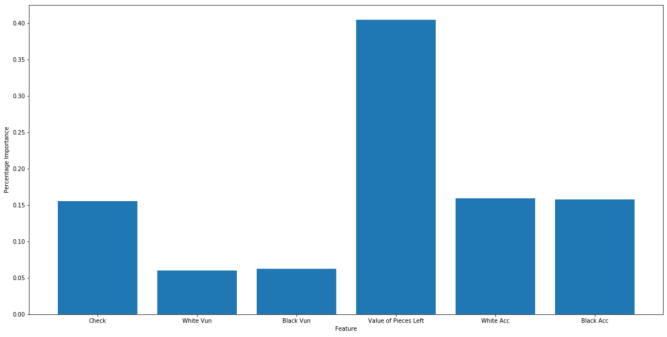

Fig. 7. Feature Importance: It can be seen that the value of pieces remaining on the board was the most important feature in the model, contributing 40% to the model. The sum of the white and black vulnerable piece the least important feature in the model, contributing only 6% to the model.

and will ultimately win the game. Most importantly, the sum of the value of pieces remaining features contributed the most to the accuracy of our model. Applications of these results can have an impact in chess playing, learning, and predicting.

B. Next Steps

Several possible next steps could be taken as further research to this paper. First, additional features could be investigated to improve accuracy. For example, the impact of particular pieces or combinations of pieces (i.e. relative position of knights to king) could be considered.

Another possible area for further research would be comparing accuracy scores for prediction across various player rating categories to better understand the impact of skill level of player on ability to accurately predict the winner of a game.

Lastly, we could apply similar feature extraction methods and models to other strategy board games such as Go, Quarto, Othello, etc. to see if measures of the current state of the board are able to accurately predict the winner of those games, as well.

ACKNOWLEDGMENTS

We would like to thank Professor Dan Roth and the teaching assistants for CIS519 Fall 2019 for providing the education, guidance, and feedback necessary for the completion of this project.

APPENDIX A
PIECE NOTATION AND VALUES

Using standard chess piece notation, the following shows how chess pieces are symbolized within this paper:

Piece	Symbol
Pawn	has no piece symbol
Knight	N
Bishop	B
Rook	R
Queen	Q
King	K

APPENDIX B
PIECE VALUES

Standard Chess piece values used in used in this paper.

Piece	Value
Pawn	1
Knight	3
Bishop	3
Rook	5
Queen	9
King	Has no value assigned (Without the King, the game is over)

APPENDIX C
CHESS MOVE NOTATION

Below is an example of a chess game, with several moves annotated as descriptions (white always moves first, and moves alternate):

d4 d5 e3 e6 c4 a6 a3 dxc4 Bxc4 b5 Bd3 Bb7 Nf3 c5 O-O h5 Ne5 Qd5 Be2 Qxg2

d4: With no piece symbol, this means the pawn moved from their starting square to d4.

dxc4: The x means capture, so the pawn from d5 captured the pawn on c4.

Bd3: The Bishop moved to d3. Bishops starting position can be traced back following the move-by-move game list.

Nf3: The knight moved from its starting position to f3.

O-O: King-side castle for white player.

Qxg2#: The Queen captured a piece on g2, and subsequently put the opposing King in checkmate, symbolized by the #.

REFERENCES

[1] Mohammad M. Masud et al., "Online Prediction of Chess Match Result." In: *Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science* (July 2015), pp. 525–537. DOI:10.1007/978-3-319-18038-0_41.

[2] Mitchell J.Chess Game Dataset (Lichess).https://www.kaggle.com/datasnaek/chess. Sept. 2017

[3] Paras Lehana et al. "Statistical Analysis on Result Prediction in Chess". In: *International Journal of Information Engineering and Electronic Business* (July 2018). DOI:10.5815/ijieeb.2018.04.04.