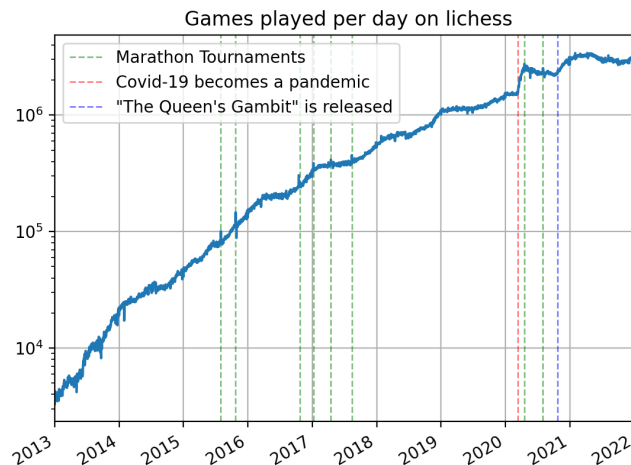# Machine Learning in Chess

Isaac Cheng*

Abstract
(100-200 words)

I certify that all material in this dissertation which is not my own work has been identified.

Signature: _____

* isaaccheng9@gmail.com

Fig. 1. Popularity of Lichess in Number of Games Played Daily [3]

1. Introduction

Over the past decade, chess has become increasingly significant in popular culture. The rise of full-time streamers on platforms like Twitch has made chess a mainstream form of entertainment. Sponsors have invested substantial sums of money into chess, with platforms such as chess.com incentivising streamers to integrate the game into their schedules. The onset of the COVID-19 pandemic further encouraged this, with people exploring digital forms of entertainment due to lockdown mandates from governments across the globe. 'The Queen's Gambit' was released in 2020, serving as the most poignant example of chess in popular culture yet. People started to take a greater interest in the game, and the number of games played daily on online platforms like chess.com and Lichess has skyrocketed.

The study of chess has great social importance, as chess games involve cognition and human behaviour that we can apply more generally. For example, social learning theory is a crucial aspect of chess strategy - players building their chess fundamentals learn from the same material and are thus likely to develop homogeneous habits. People have researched the impact of social learning in the game of Go [1], but not for chess.

Strategic board games are good candidates for artificial intelligence research due to their fixed environment. Other games like Go have also undergone extensive studies [2]. Chess is an apt game for data mining and machine learning - the games are well-structured, with a defined search space and comprehensive databases of historical chess games. Literature on chess strategy is centuries-old, but the terabyte-scale exploration of games has only caught on recently. Professional players have started to improve their game using analytics tools like Stockfish. In this project, I will apply data mining and machine learning tools to millions of chess games to understand patterns in how people play chess.

2. Preliminary Research

2.1. Growth in the Popularity of Chess

While there has been continuous growth in the popularity of chess over the last decade, a significant spike occurred with the pandemic and the release of 'The Queen's Gambit'. Some people have researched Lichess databases to demonstrate this, such as in the chessOpeningStats repository (see figure 1) [3]. However, there is a lack of formal academic research on this topic - the closest is a thesis that investigates the influence of Beth and their qualitative effects on how viewers in the Netherlands and United States view chess [4], but this does not provide quantifiable results.

2.2. History of Computer Chess

Since the first digital computer was switched on 1945, programmers have sought to build a machine that could defeat the world chess champion [5]. In 1948, Alan Turing worked with David Champernowne (an economist and mathematician) to create a machine routine for playing chess, dubbed the 'Turochamp' [6]. They did not have a machine, so they simulated the machine's behaviour by hand with a pen and pencil [6]. Champernowne reported that his wife, who was a beginner at chess, took on the machine and lost [7]. A year later in 1949, Claude Shannon wrote a paper that described a computer that could play chess - his idea was to focus on analysing the seven most likely moves from the current position, and branching from each of these positions into the next seven most likely replies from the opposition, and so forth until the computer runs out of memory [8]. The first chess program was eventually implemented in November 1951 by Dietrich Prinz. However, unlike Turing's Turochamp, Prinz's program could not complete a full game of chess and used exhaustive search to play moves rather than a heuristic [7].

The first computer to beat a world champion was IBM's Deep Blue, which defeated Garry Kasparov over a six-game match in 1997 [9]. Deep Blue was a supercomputer that was able to perform 200 million calculations per second [10]. It was the first computer to use a heuristic search algorithm, which is a method of searching for a solution to a problem by evaluating possible solutions and choosing the best one. By this point, chess engines were stronger than human players, yet they still lacked a deep understanding of the game. Deep Blue's victory was accredited to its brute-force ability - its over-reliance on this was demonstrated in the first game of their match, as it sacrificed Kasparov's sacrifice of a rook for a bishop, only to lose 16 moves later [10].

[WRITE EXPLANATION OF STOCKFISH]

Recently, researchers at DeepMind have developed their machine-learning algorithms to master chess alongside other strategic games such as shogi and Go with their AlphaZero machine-learning algorithm [11]. It was based on their recent AlphaGo Zero program, which achieved superhuman performance using an algorithm based solely on reinforcement learning, with no human data, guidance, or domain knowledge beyond the rules of Go [12]. Likewise, AlphaZero started with no knowledge of chess (nor any of the other strategy games) beyond the basic rules and became the best chess player (human or computer) in the world within hours of learning by playing millions of games [10]. AlphaZero's chess abilities were tested by playing against Stockfish, a chess engine that is considered to be the strongest chess engine in the world. AlphaZero was able to beat Stockfish 8-0 in a match of 8 games [11]. AlphaZero's victory was attributed to its ability to learn from its mistakes, which is a key aspect of machine learning. It implements a general-purpose Monte-Carlo tree search algorithm, which simulates random games to find the best move [13]. It also uses a neural network to evaluate the board position, which is a technique that is used in many machine learning algorithms.

[WRITE EXPLANATION OF RECENT SCANDAL REGARDING HANS NIEMANN]

2.3. Piece Valuation

2.4. Social Learning Theory

3. Project Specification

3.1. Requirements Analysis

Machine learning in chess covers a broad range of topics, so it is crucial to scope design and development systematically. We will start exploring simple, wide-spanning areas to build a baseline – this may include answering questions such as:

- What are the most common chess openings and their success rates?

- What are the most popular chess openings over time?

- What is the distribution of ELO ratings on a given online platform?

Then, we will build up to more complex questions using previous findings. Questions may include:

- What are the best counters against particular openings?

- To what degree is social learning theory evident in chess?

- How good are existing metrics for predicting win probability in chess games?

It is also important to outline what will be out of the project's scope. There is a vast range of chess game variants like Chess960 (also known as Fischer random chess), King of the Hill, and Antichess [CITATION NEEDED]. However, they would all introduce significant overhead to our research as they change the fundamental rules and theories of the game. We will not include them in the project to avoid scope creep and maintain a strong focus on the standard variant. In addition, we will not be creating a chess bot that plays games - we will instead focus on generating insights to understand patterns in chess strategy.

3.2. Design

A large data set is paramount to performing machine learning effectively. We will be using Lichess as our data source. They provide an open database of standard-rated games played on lichess.org every month in PGN (Portable Game Notation) format [CITATION NEEDED], which is the standard for recording chess games [CITATION NEEDED]. Lichess is one of the two mainstream platforms for playing chess online (alongside chess.com), which means they provide a large sample to analyse each month – October 2022 includes over 90 million games. While it would be ideal to obtain games from chess.com, this is infeasible as they do not provide an open database like Lichess. The file sizes for Lichess' monthly data sets are huge, measuring at 30 GB for the compressed October 2022 download, and 214 GB when uncompressed.

We will use Python 3.11.0 because Python the de facto standard for data science [CITATION NEEDED] and that is the latest version at the time of writing. The python-chess library has emerged as the most popular for PGN parsing alongside other features – it has over 1,900 stars on GitHub and is maintained regularly [CITATION NEEDED]. It will accompany the Python Stockfish library, the best and most popular chess engine available to the public [CITATION NEEDED] – this enables us to integrate the Stockfish engine with Python to analyse positions and generate insights. In addition, we will use industry-standard Python libraries such as seaborn for graph visualisations, pandas for data frames, and NumPy for mathematical analysis.

The computational analysis will take place on my personal laptop, which is an M1 Max MacBook Pro 16 (2021) with 32 GB of RAM and 1 TB of SSD storage. If the lack of storage space becomes an issue when analysing multiple months' worth of data sets, we could look at using high-speed external solid state drives or using another machine. The latter would be a high-performance desktop computer running a Linux distribution provided by Chico Camargo, which could provide further benefits such as more powerful GPU accelerated computation.

3.3. Development

Downloading the data is a straightforward process. As aforementioned, we will download monthly games from the Lichess Open Database (provided in compressed form), extract the games from the PGN file, and load them using python-chess. This library will enable us to read the game and navigate to the part we want to analyse. We will start by navigating to the final position of the games. Then, we will get the FEN (Forsyth-Edwards Notation) string, the standard notation for describing a chess game's board position [CITATION NEEDED].

Once we have the position, we will analyse it using the Stockfish library. It provides information from the game, including the centipawn score. Stockfish computes the centipawn score using static evaluation of the board position and applying the mini-max algorithm and alpha-beta pruning [CITATION NEEDED]. It measures a white's advantage, where a centipawn equals 1/100 of a pawn [CITATION NEEDED]. Hence, a score of 100 means that white has a pawn's worth of advantage, whereas a score of -100 indicates a pawn's advantage for black. If a player has a significant advantage, Stockfish may find a forced mate - this means that one player can checkmate the other, regardless of the other player's moves [CITATION NEEDED]. In this case, it will represent the score with either '#x' or '#-x', where x is the number of moves from a checkmate - the former means white can checkmate black, and the latter means black can checkmate white. Note that this also includes 0 if the game ended due to a checkmate occurring on that move.

Before performing data analysis on the results, we will filter and segment the data according to how the game ended – this is important because most rated Lichess games are games with a time limit [CITATION NEEDED]. In these games, a player loses the game regardless of their position when they run out of time. It is a common situation that a player loses the game on time despite having an advantageous position on the board [CITATION NEEDED] – this is an example of the types of games we may want to filter out, as they may add unnecessary noise to the data. On a similar note, games with shorter time limits may have different patterns to others, as time management skills become a more significant factor. We will also differentiate between games that ended from checkmate and games where a player resigned - the former may be more indicative, as a player resigning is effectively extrapolating the rest of the game and deciding that they will lose.

Next, we will analyse the data to investigate our questions. The simple questions will provide the foundation for our investigations and may not require machine learning, whereas the more complex questions that form the bulk of our project will involve it. For example, identifying the degree to which social learning theory is evident in chess may become a feature importance problem. A reasonable hypothesis is that social learning theory takes place to a greater degree for popular openings, as there is substantially more educational material for them. Thus, we could use feature selection methods such as information gain and variance thresholding to identify the features of games containing less popular openings.

Finally, we will create visualisations of our results to demonstrate them in a manner that is easier to understand. We will plot a bar chart to show the most popular chess openings by percentage and their win rates, as well as a bar chart to demonstrate the distribution of ELO ratings in Lichess. Meanwhile, a line graph will show the most popular chess openings over time. These visualisations for the results of our simple questions will help readers by providing context for further studies on more complex topics. Using the same example as the previous paragraph, we could plot bar charts to compare the importance of different features of games containing less popular openings.

3.4. Evaluation

4. Conclusion

[1] B. A. Beheim, C. Thigpen, and R. Mcelreath, Strategic social learning and the population dynamics of human behavior: The game of Go *Evolution and Human Behavior*, vol. 35, no. 5, pp. 351–357, 2014.

[2] M. Müller, Computer go *Artificial Intelligence*, vol. 134, no. 1-2, pp. 145–179, 2002.

[3] Paul556, "Games played per day on lichess." `https://github.com/Paul566/chessOpeningStats`, 2022. Date Accessed: 15 November 2022.

[4] F. L. Lowie, How big is the Netflix effect? The Queen's Gambit: The influence of Beth on the increase in chess numbers Master's thesis, 2021.

[5] "It calculates 1 billion chess moves every second, but it's still not as bright as you." `https://www.independent.co.uk/news/uk/it-calculates-1-billion-chess-moves-every-second-but-it-s-still-not-as-l html`, 1996. Date Accessed: 22 November 2022.

[6] A. Turing, Alan Turing 1912-1954

[7] B. J. Copeland and D. Proudfoot, Turing and the computer *Alan Turing's Automatic Computing Engine: The Master Codebreaker's Struggle to Build the Modern Computer, Oxford University Press, Oxford*, pp. 107–148, 2005.

[8] C. E. Shannon, XXII. Programming a computer for playing chess *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950.

[9] Y. Seirawan, H. A. Simon, and T. Munakata, The implications of kasparov vs. deep blue *Communications of the ACM*, vol. 40, no. 8, pp. 21–25, 1997.

[10] S. Strogatz, One giant step for a chess-playing machine *New York Times*, pp. 1–6, 2018.

[11] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran,

T. Graepel, *et al.*, A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[12] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, Mastering the game of go without human knowledge *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[13] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, Mastering chess and shogi by self-play with a general reinforcement learning algorithm *arXiv preprint arXiv:1712.01815*, 2017.