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

1. Introduction

The use of computers in chess has been a popular topic of study over the past century. Computers are able to perform tasks sequentially incredibly quickly, they generally do not make mistakes, and they have a large capacity for storing information quickly in the form of secondary storage devices. In chess, this means they are able to perform move calculations at orders of magnitudes faster than humans, and they can store large amounts of information about chess games.

Strategic board games are good candidates for artificial intelligence research due to their fixed environment. Other games like Go have also undergone extensive studies [1]. 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, and platforms like Lichess and Chess.com have integrated such tools to make them more accessible to everyone.

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 [2], but not for chess. 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. Early 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 [3]. In 1948, Alan Turing worked with David Champernowne (an economist and mathematician) to create a machine routine for playing chess, dubbed the 'Turochamp' [4]. They did not have a machine, so they simulated the machine's behaviour by hand with a pen and pencil [4]. Champernowne reported that his wife, who was a beginner at chess, took on the machine and lost [4]. 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 [5]. 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 [4].

The first computer to beat a world champion was IBM's Deep Blue, which defeated Garry Kasparov over a six-game match in 1997 [6]. Deep Blue was a supercomputer that was able to perform 200 million calculations per second [7]. 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 [7].

2.2. Modern Developments in Computer Chess

Stockfish is currently the strongest chess engine in the world [8]. It is used by many professional players to analyse their games, with Chess.com and Lichess also using Stockfish to provide game analysis on their platforms. It is a free, open-source chess engine that started with the open-source Glaurung engine, which was authored by Tord Romstad. The first version, Stockfish 1.0, was released in November 2008 when Marco Costalba took over the Glaurung 2.1 code [9]. It started to gain prominence from their Stockfish 3 release in 2013, in which they started using a distributed testing framework called Fishtest [10]. This was a web application written in Python that allowed volunteers to donate CPU time to test improvements to the engine. They used Fishtest to test whether changes to the game-playing code demonstrate statistically significant improvements to the engine, using the chi-squared test to verify tests on the framework. Twelve months after the inception of Fishtest, Stockfish skyrocketed to the top of chess engine rating lists, gaining 120 Elo points [11]. As of November 2022, users have contributed over 9580 years of CPU time to the project, playing over 5.5 billion chess games [12]. Stockfish uses the alpha-beta pruning search algorithm, which improves minimax search [13] by avoiding variations that will never be reached in optimal play because either player will redirect the game [14]. Due to the number of possible moves at each position, it is computationally infeasible to search every tree subgraph until the end of the game, so Stockfish finishes its search once it reaches a certain depth. This is measured in ply, which measures the distance between the root node and the leaf nodes [14]. Stockfish uses iterative deepening to incrementally increase the depth of its search tree [15]. Despite this, it does not necessarily search all the moves in this search space – Stockfish uses heuristics to search promising variations in more depth than normal, and less promising variations in less depth than normal [14]. The engine uses forward pruning to reduce the search space via futility pruning – if the evaluation of a position is significantly worse than the player's alternative, it prunes the position's children early [14]. It also uses reduction techniques, which involves searching tree subgraphs to lower depths rather than completely omitting them from the search [14], such as late move reductions – this assumes that the engine checks better moves earlier, so it searches later moves with less depth in the move tree [16]. Once Stockfish's search algorithm arrives at a leaf node, it applies a heuristic evaluation function to determine whether it favours white or black [14]. Since Stockfish 11, it uses NNUE (Efficiently Updatable Neural Networks) evaluation [17], a neural network that is trained to predict the output of the classic Stockfish evaluation function using basic inputs using the previous evaluations of millions of positions at moderate search depth [18]. NNUE's architecture is optimised for speed on CPU machines, and it uses a shallow, fully connected network with four hidden layers [19]. The input layer of binary features describes the position of the white and black king relative to the white and black pieces, and dimensionality reduction is performed by passing the board feature vector through two hidden layers and an output layer – this results in a final scalar value for position evaluation [14].

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 [20]. 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 [21]. 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 [7]. AlphaZero's chess abilities were tested by playing against Stockfish. AlphaZero was able to beat Stockfish 8-0 in a match of 8 games [20]. 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 (MCTS) algorithm rather than minimax, which Stockfish uses [22]. MCTS is a heuristic search algorithm based on the idea is that when a player selects their actions randomly, very little can be inferred from a single game, but a good strategy can be found by simulating a multitude of random games [23]. It expands the search tree based on random sampling of the search space and performs playouts, where each game is played to the end via random move selection. The results of each game is used to weight the nodes in the game tree to make better nodes more likely to be chosen in the future – after some number of simulations, the child node with the most samples is chosen

[23]. Whereas minimax and alpha-beta pruning algorithms are serial in nature, MCTS is parallel, as the random sampling of the search space can be performed in parallel [22]. Hence, MCTS can use batching on GPUs to speed up the search process, which is fundamental to AlphaZero's success.

Stockfish and AlphaZero represent the current state of the art in computer chess. While Deep-Mind have not open sourced AlphaZero, a project 'LCZero' was born as an attempt to reproduce it using crowd computing [14]. This predominantly uses the same search and evaluation techniques as AlphaZero, but with some improvements that improve the strength of the engine. Stockfish's strength lies in its search, whereas LCZero's strength is in its evaluation. This has led to performance comparisons between Stockfish and LCZero using Plaskett's Puzzle, a chess endgame study known to be an anti-engine puzzle. Stockfish searched through nearly 1.9 billion positions using its minimax algorithm, and its domain-specific search optimisations helped it to reach the optimal solution. However, LCZero's search algorithm was unable to find the optimal solution as it chose the wrong lines to search deeply [14]. Part of this can be attributed to the fundamental differences between the engines – LCZero was built from zero knowledge of the game except the basic rules of chess, whereas Stockfish uses chess-specific search heuristics. Thus, Stockfish is stronger than LCZero, but LCZero is more generalisable and can be applied to other strategy games like shogi and Go.

2.3. Relevance of Computer Chess

The patterns that computer chess engines like Stockfish and AlphaZero learn are similar to those that humans learn. This is because the search algorithms used by these engines are based on the same principles as those used by humans. Stockfish's futility pruning method is similar to the way humans prune search trees – if a move is unlikely to be optimal, then it is unlikely to be optimal in the future, and so the search can be pruned. Meanwhile, AlphaZero's MCTS algorithm is similar to how humans use their memory of historical games to make decisions in the future – if they recognise a similar position to one they have seen before, they can use their memory of the outcome of that game to influence their decision. If they achieved a positive outcome from their previous move, they are more likely to make the same move again, whereas they are likely to change their strategy if they achieved a negative outcome.

## 2.4. Practical Use of Chess Databases

Most recently, computer chess has come under scrutiny in chess tournaments, as US Grandmaster Hans Niemann was involved in a cheating scandal when he defeated reigning five-time World Chess Champion Magnus Carlsen in the third round of the 2022 Sinquefield Cup [24]. Carlsen withdrew from the cup the next day on 5 September 2022, posting a tweet implying an accusation that Niemann was cheating over the board, an accusation that Niemann denied [25]. In a follow-up tweet on 26 September 2022, Carlsen accused Niemann of cheating, stating his belief that Niemann cheated more often and more recently than he publicly admitted [26]. Chess.com conducted an investigation into Niemann's games and banned the player from their platform, publishing a report on 4 October 2022 alleging that Niemann likely received illegal assistance in more than 100 online games, as recent as 2020, including contests in which prize money was on the line [27]. They used a variety of analytics tools designed to detect cheating, including tools that compared the moves he made to the moves recommended by the top chess engines [27]. An independent investigation was conducted by Kenneth Regan, a professor of computer science at the University of Buffalo, whom is paid by The International Chess Federation (FIDE) to monitor tournaments [28]. He also concluded that Niemann was likely cheating, based on the results of his proprietary cheat detection software that calculates the odds that a player of a given skill level can make a given set of moves [28]. Similar to part of Chess.com's investigation, Regan's software compared Niemann's moves to the moves recommended by the top chess engines and used a statistical model to output a z-score representing the degree of similarity between the player and the engine [28]. Hans Niemann filed a $100 million defamation lawsuit against Magnus Carlsen and Chess.com amongst others for 'colluding to blacklist' Niemann from the chess world [29]. The case is ongoing at the time of writing in November 2022, but the scandal has highlighted the influence that computer chess has in the modern game, both in terms of its ability to cheat and its ability to analyse games for investigation purposes.

2.5. Piece Valuation

2.6. 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). 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 [30], but this does not provide quantifiable results.

Fig. 1. Popularity of Lichess in Number of Games Played Daily [31]
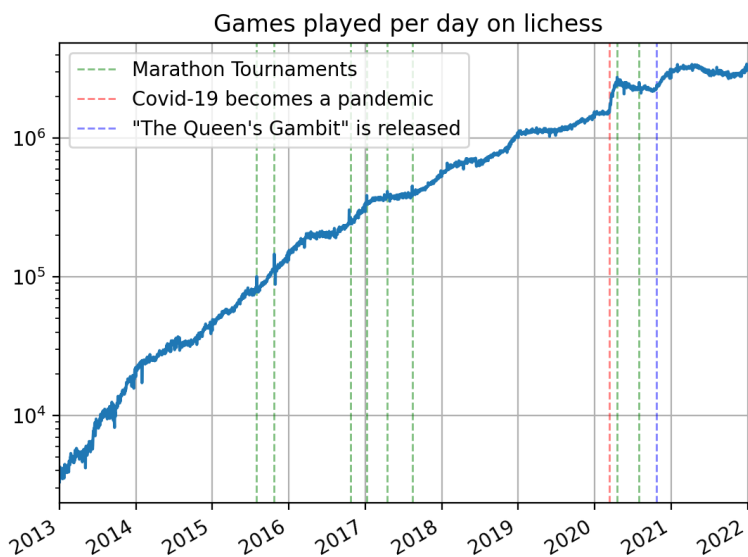
3. Project Aims and Objectives

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:
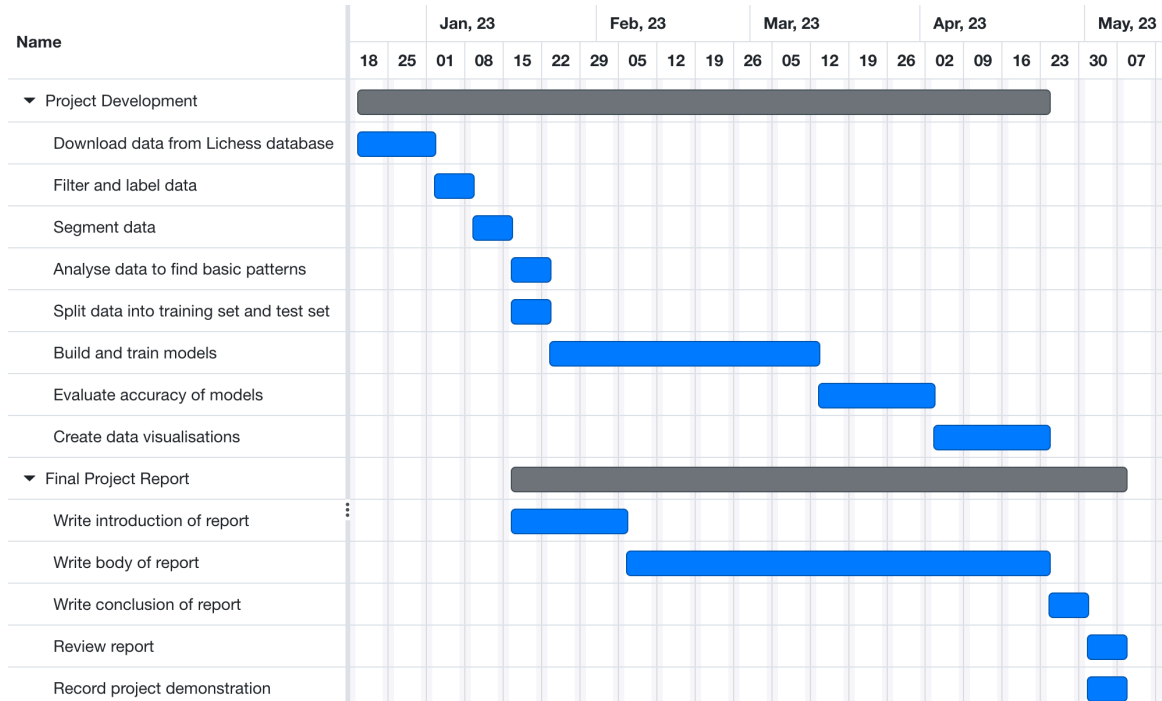
- What are the best counters against particular openings?
- To what degree is social learning theory evident in chess?
- How good are existing metrics for predicting win probability in chess games?

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 [32]. However, they would all introduce significant overhead to our research as they change the fundamental rules and theories of the game. In addition, the sample sizes for these games is much smaller – there were only 337,940 games played on Lichess in October 2022 [33]. We will not include

them in the project to avoid scope creep and to 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.

4. Project Management

4.1. Overview of Plan

Name	Jan, 23					Feb, 23				Mar, 23				Apr, 23				May, 23	
	18	25	01	08	15	22	29	05	12	19	26	05	12	19	26	02	09	16	23	30	07

(Gantt chart)

- ▼ Project Development
  - Download data from Lichess database
  - Filter and label data
  - Segment data
  - Analyse data to find basic patterns
  - Split data into training set and test set
  - Build and train models
  - Evaluate accuracy of models
  - Create data visualisations
- ▼ Final Project Report
  - Write introduction of report
  - Write body of report
  - Write conclusion of report
  - Review report
  - Record project demonstration

4.2. Planning Our Approach

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 [33], which is the standard for recording chess games [34]. 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 92 million standard rated games.

   We will use Python 3.11.0 because Python is the de facto standard for data science [35] 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 [36]. It will accompany the Python Stockfish library, the best and most popular chess engine available to the public [9] – 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.

4.3. Project Risks

There are very few risks in the project. The main risk is that the data set is not large enough to provide meaningful insights. However, this is unlikely to be an issue as Lichess provides a large data set every month, and there is already a history of monthly data from January 2013 onwards. The file sizes for Lichess' latest monthly data sets are huge, measuring at 30 GB for the compressed October 2022 download, and 214 GB when uncompressed. The computational analysis for this project 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 solution would involve 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 to streamline our process.

Another risk is that the data set is not representative of the entire chess community. However, this is unlikely to be an issue as 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. Lichess is a free platform, which means that it is accessible to a wide range of people. While it would be ideal to also obtain games from Chess.com to cover the two main platforms, this is infeasible as they do not provide an open database like Lichess.

4.4. Legal and Ethical Issues

There are a few legal and ethical considerations in the project, but none of them are of real concern. A potential legal issue could arise from Stockfish, which is licensed under the GNU General Public License (GPL) v3.0 [37]. This means that we must make the source code for our project available to the public, which is not an issue as we will be using GitHub to host our project.

Furthermore, there may be concerns about the ethics of using machine learning to analyse chess games. However, this is not an issue as the data set is publicly available and does not contain any personal information. While the data set that we will analyse contains information about specific chess games and the usernames of the players, it does not contain any personally identifiable information (PII) such as names, addresses, or phone numbers. The output of our project will only concern chess strategy rather than the strategy or habits of individual players, and it will consist of highly aggregated data to find general trends. As our project will not create a chess bot that plays games, there is no risk of it being used to cheat in chess tournaments, which is a common concern with machine learning in chess.

4.5. Downloading the Data

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 [38].

4.6. Preparing the Data

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 [14]. It measures white's advantage, where a centipawn equals 1/100 of a pawn [39]. 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. 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 label the data according to how the game ended. This is important because the most popular game modes in Lichess are timed (Bullet/Blitz/Rapid) as opposed to untimed (Classical) – as of 24 November 2022, there were over 350,000 Bullet players, 700,000 Blitz players, and 400,000 Rapid players as compared with around 45,000 Classical players in the past week [32]. In timed games, a player loses the game when they run out of time regardless of their position. It is a particularly common situation in Bullet games that a player loses on time despite having an advantageous position on the board – players are usually only given one or two minutes over the span of the game to make their moves, which poses a distinct challenge. This is an example of the types of games we may want to filter out, as they add unnecessary noise to the data.

Next, we will segment the data, as patterns could be found in different subsets of games based on the game mode. Games with shorter time limits in general may have different patterns to others, as time management skills become a more significant factor. For example, this factor would be more significant in a 5+0 Blitz game compared to a 10+0 Rapid game. 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 when they may in fact be in a winning position.

4.7. Analysing the Data

Once the data has been prepared, we will analyse the data to find patterns and investigate our questions. The basic patterns that we find will provide the foundation for our investigations and may not require machine learning, whereas investigating the 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.

To ensure that our machine learning models will be able to generalise, we will split the data into a training set and a test set. A common approach is to use the large majority as training data, typically 80% of the data. The rest will be used as the testing data. We will use the training data to train our model, and the testing data to evaluate its performance. This will allow us to determine whether our model is overfitting or underfitting the data. If it is overfitting, we will need to reduce the complexity of the model, and if it is underfitting, we will need to increase it.

4.8. Evaluation

We will also use cross-validation to ensure that our model generalises well to new data. This will involve splitting the data into k folds, where k is a number that we choose. We will then train the model k times, each time using a different fold as the testing data and the rest as the training data. In the end, we will average the results to get a more accurate estimate of the model's performance.

As this project focuses on using machine learning to analyse chess games, we will evaluate the performance of our models using metrics that are relevant to machine learning. This starts with the confusion matrix, which counts the number of true positives, true negatives, false positives, and false negatives. Based on the confusion matrix, we will measure accuracy, precision, recall, F1 score, and receiving operating character (ROC) curve to evaluate the performance of our models.

- Accuracy is the proportion of correct predictions, defined as: $\frac{\text{number of correct predictions}}{\text{number of data points}}$

- Precision is the proportion of positive predictions that are correct, defined as:
$$\frac{\text{number of true positives}}{\text{number of true positives + number of false positives}}$$

- Recall is the proportion of positive cases that are correctly identified, defined as:
$$\frac{\text{number of true positives}}{\text{number of true positives + number of false negatives}}$$

- F1 score is the harmonic mean (reciprocal of the average of reciprocals) of precision and recall, defined as: $2 * \frac{\text{precision * recall}}{\text{precision + recall}}$

- Receiver operating characteristic (ROC) curve visualises the trade-off between the true positive rate and the false positive rate. This will allow us to see how well our model performs at different classifications thresholds.

4.9. Visualising our Results

At the end, 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. This will allow us to see which features are more important in determining the outcome of a game.

5. Conclusion

To conclude, this paper critically reviewed the existing uses of machine learning in chess, the challenges that arises from applying machine learning in chess, and the potential applications of machine learning in chess. We also reviewed the increasing influence of chess in popular culture and how the study of chess has great social importance, with links to social learning theory. Thus, we formed a hypothesis that machine learning can be used to analyse chess games to identify patterns in how people play chess. We proposed a project that will study large samples of rated games on Lichess to explore areas like the degree to which social learning theory is evident in chess, and how good the existing metrics are for predicting the outcome of a game.

---

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

[2] 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.

[3] W. Hartston, "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-bright-as-you-1318781.html, 1996. Date Accessed: 22 November 2022.

[4] 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.

[5] 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.

[6] 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.

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

[8] "Computer Chess Rating Lists: November 2022." https://computerchess.org.uk/ccrl/4040/rating_list_all.html, 2022. Date Accessed: 22 November 2022.

[9] "About Stockfish." https://stockfishchess.org/about/, 2022. Date Accessed: 22 November 2022.

[10] "Fishtest Distributed Testing Framework." https://www.talkchess.com/forum3/viewtopic.php?start=0&t=47885, 2022. Date Accessed: 22 November 2022.

[11] "Computer Chess Rating Lists: May 2014." http://archive.today/OSMBi, 2014. Date Accessed: 23 November 2022.

[12] "Users | Stockfish Testing." https://tests.stockfishchess.org/users, 2022. Date Accessed: 23 November 2022.

[13] J. v. Neumann, Zur theorie der gesellschaftsspiele *Mathematische annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[14] S. Maharaj, N. Polson, and A. Turk, Chess AI: competing paradigms for machine intelligence *Entropy*, vol. 24, no. 4, p. 550, 2022.

[15] A. D. De Groot, Thought and choice in chess in *Thought and Choice in Chess*, De Gruyter Mouton, 2014.

[16] D. Levy, D. Broughton, and M. Taylor, The SEX algorithm in computer chess *ICGA Journal*, vol. 12, no. 1, pp. 10–21, 1989.

[17] Y. Nasu, Efficiently updatable neural-network-based evaluation functions for computer shogi *The 28th World Computer Shogi Championship Appeal Document*, 2018.

[18] "Introducing NNUE Evaluation." https://stockfishchess.org/blog/2020/introducing-nnue-evaluation/, 2022. Date Accessed: 25 November 2022.

[19] S. Narayanan, "A Brief Guide to Stockfish NNUE." https://saumikn.com/blog/a-brief-guide-to-stockfish-nnue/, 2022. Date Accessed: 25 November 2022.

[20] 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.

[21] 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.

[22] 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.

[23] G. Chaslot, S. Bakkes, I. Szita, and P. Spronck, Monte-carlo tree search: A new framework for game ai in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 4, pp. 216–217, 2008.

[24] C. McGourty, "Sinquefield Cup 3: Niemann beats Carlsen to cross 2700." https://chess24.com/en/read/news/sinquefield-cup-3-niemann-beats-carlsen-to-cross-2700, 2022. Date Accessed: 23 November 2022.

[25] A. Beaton and J. Robinson, "Chess Is in Chaos Over Suspicion That a Player Cheated Against Magnus Carlsen." https://www.wsj.com/articles/magnus-carlsen-hans-niemann-chess-cheating-scandal-11662644458, 2022. Date Accessed: 23 November 2022.

[26] S. Jenkins, "When chess is hard and cheating is easy, the next move is complicated." https://www.washingtonpost.com/sports/2022/09/27/magnus-carlsen-hans-niemann-chess-cheating-controversy, 2022. Date Accessed: 23 November 2022.

[27] A. Beaton and J. Robinson, "Chess Investigation Finds That U.S Grandmaster 'L'ikely Cheated' More Than 100 Times." https://www.wsj.com/articles/chess-cheating-hans-niemann-report-magnus-carlsen-11664911524, 2022. Date Accessed: 23 November 2022.

[28] C. Beam, "A Cheating Scandal Has Rocked the Chess World. The 'Chess Detective' Is on the Case." https://time.com/6227677/magnus-carlsen-hans-niemann-kenneth-regan-chess-scandal/, 2022. Date Accessed: 23 November 2022.

[29] A. Beaton and J. Robinson, "Hans Niemann Files $100 Million Lawsuit Against Magnus Carlsen, Chess.com Over Cheating Allegations." https://www.wsj.com/articles/chess-cheating-hans-niemann-magnus-carlsen-lawsuit-11666291319, 2022. Date Accessed: 23 November 2022.

[30] 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, Utrecht University, 2021.

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

[32] "Weekly Blitz Rating Distribution – Lichess." https://lichess.org/stat/rating/distribution/blitz, 2022. Date Accessed: 24 November 2022.

[33] "lichess.org open database." https://database.lichess.org/, 2022. Date Accessed: 24 November 2022.

[34] "Portable Game Notation." https://www.mark-weeks.com/aboutcom/aa02j12.htm, 2022. Date Accessed: 24 November 2022.

[35] K. Basel, "How Did Python Become the Language of Choice for Data Science?." https://www.netguru.com/blog/python-for-data-science, 2022. Date Accessed: 24 November 2022.

[36] N. Fiekas, "python-chess: a chess library for Python." https://github.com/niklasf/python-chess, 2022. Date Accessed: 24 November 2022.

[37] "Stockfish GitHub Repository." https://github.com/official-stockfish/Stockfish, 2022. Date Accessed: 22 November 2022.

[38] S. Edwards, "Portable Game Notation Specification and Implementation Guide." https://ia902908.us.archive.org/26/items/pgn-standard-1994-03-12/PGN_standard_1994-03-12.txt, 2022. Date Ac-

cessed: 24 November 2022.

[39] "Centipawn." https://chess.fandom.com/wiki/Centipawn, 2022. Date Accessed: 24 November 2022.