Detecting Fair Play Violations in Chess Using Neural Networks*

Maksim Iavich[1,†], Zura Kevanishvili[1,*,†]

[1] Caucasus University, 31 Kvichia St, Tbilisi, 0102, Georgia

Abstract

This study addresses the challenge of distinguishing between human and computer-generated play in chess, crucial for ensuring the integrity and fairness of both online and tournament play. As unauthorized computer assistance becomes increasingly sophisticated, we utilize sequential neural networks to analyze a vast dataset of chess games, employing both traditional engines, such as Stockfish and Leela, and innovative neural networks like Maia and its individual sub-models. This analysis incorporates centipawn deviation metrics to gauge departures from typical computer strategies, Maia's insights into human and idiosyncratic playstyles, and an evaluation of time distribution for moves. Our method extends by considering the strategic implications of move sequences and the consistency of play under varying game conditions, enhancing our understanding of the nuanced differences between human and AI play. Remarkably, our algorithm achieves approximately 98% accuracy in identifying the use of chess engines, offering a significant advancement in efforts to maintain the game's integrity.

To further validate our findings, we conducted cross-validation with a separate dataset, confirming the robustness of our model. We also explored the algorithm's applicability to detecting AI assistance in other board games, suggesting its potential for broader use. The research highlights the critical role of machine learning in combating digital cheating, emphasizing the need for continuous adaptation of detection methods to keep pace with evolving technologies. Additionally, our findings point to the importance of developing ethical guidelines for the use of AI in games, ensuring a fair and level playing field for all participants. Lastly, by publishing our methodology and the criteria for AI detection, we aim to foster an open dialogue within the gaming community and among developers, promoting transparency and collaboration in the fight against cheating.

Keywords

Chess, Cheating, AI, Neural Networks

1. Introduction

The rivalry between humans and computers has been ever-present in the history of chess. As early as the late 1700s, there were claims of a machine named “The Turk” that supposedly surpassed humans in the game of chess. This automaton was in the form of a human-sized dummy seated behind a large wooden cabinet, its interior crammed with a series of gears, cranks, and levers resembling the innards of a clock. The dummy, wearing a turban and colorful robes, carried a pipe in one hand. The robot defeated many famous chess masters at the time and even famous individuals like Napoleon Bonaparte in 1809. The French emperor reportedly attempted a few illegal Chess, with its rich history spanning centuries, has witnessed a fascinating interplay between human intellect and artificial intelligence.

* IVUS2024: Information Society and University Studies 2024, May 17, Kaunas, Lithuania

[1,*] Corresponding author

[†] These author contributed equally.

miavich@cu.edu.ge (M. Iavich); zukakevanishvili@gmail.com (Z. Kevanishvili)

0000-0002-3109-7971 (M. Iavich); 0009-0005-9693-6560 (Z. Kevanishvili)

The rivalry between humans and computers in the realm of chess has been a subject of intrigue and study, dating back to the late 1700s with the emergence of the legendary automaton known as "The Turk." "The Turk," a mechanical marvel disguised as a chess-playing machine, captivated audiences with its seemingly superhuman abilities. However, behind its intricate gears and levers lay a concealed human operator, underscoring the ingenuity of combining technology with human skill to simulate machine intelligence.

Fast forward to 1997, a watershed moment in the annals of chess history, marked by the iconic match between world champion Garry Kasparov and IBM's Deep Blue. In a nine-day battle of wits, Deep Blue emerged victorious, heralding a new era where computers demonstrated their supremacy over even the most formidable human opponents. The subsequent evolution of chess engines, fueled by advances in computational power and algorithmic sophistication, has seen AI entities like Stockfish, Leela Chess Zero, and Alpha Zero ascend to unprecedented levels of strength. These engines, capable of analyzing vast numbers of possible moves in mere seconds, have reshaped the landscape of competitive chess, outstripping human players in sheer calculative prowess.

However, this era of technological dominance has not been without its challenges. Recent controversies surrounding suspected instances of cheating in high-stakes chess tournaments, such as the Sinquefield Cup incident involving Hans Niemann and world champion Magnus Carlsen, have underscored the need for robust mechanisms to detect and deter unethical behavior. In response to these challenges, our research endeavors to develop a cutting-edge neural network-based approach tailored for investigating suspicious player performances in competitive chess settings.

By harnessing a combination of traditional metrics, such as time distribution and centi-pawn loss, and innovative evaluation algorithms, our aim is to enhance the efficacy and accuracy of cheating detection methodologies. Through rigorous experimentation and analysis, we seek to elucidate the intricate interplay between human cognition and machine intelligence in the context of chess. By leveraging the power of artificial neural networks and sophisticated data analytics techniques, we endeavor to contribute towards the advancement of fair play and integrity in the ever-evolving world of competitive chess.

2. Literature Review

The study by [1] introduces Maia, a novel adaptation of AlphaZero tailored for human chess replication. Maia represents an innovative attempt to bridge the gap between AI and human chess players, aiming to enhance human-AI interaction and collaboration. By optimizing AlphaZero's architecture for human-like gameplay, Maia seeks to provide more interpretable and accessible AI models for human players. However, despite its potential, the study lacks rigorous empirical validation and comparative analysis against existing methods, raising questions about its effectiveness and generalizability. Furthermore, the oversight of fair play violations in competitive chess settings is a notable limitation, as maintaining integrity and fairness is paramount in such environments.

Similarly, [2] delves into the development of AI frameworks that adapt to individual human behavior, offering a glimpse into the potential of personalized AI models in chess. By leveraging open-source adaptations of AlphaZero trained on human players, the study showcases significant advancements in predicting specific players' actions and performing stylometry. These personalized AI models demonstrate promise in capturing human decision-making nuances, potentially enhancing human-AI collaboration and intuitiveness. However, the study's reliance on a narrow range of experimental data and lack of comprehensive validation hinder its broader applicability and reliability in real-world chess competitions. Addressing the pervasive issue of cheating in online chess, [3] proposes a data-driven approach to cheat detection through large-scale statistical analysis of human and computer decision-making tendencies. The study sheds light on the challenges posed by AI-assisted cheating and underscores the importance of transparent and effective cheat detection mechanisms. By identifying 'centaurs' (computer-assisted human players) and distinguishing them from genuine players, the proposed approach offers a promising avenue for enhancing fairness and trust in online chess platforms. However, limitations in access to comprehensive data from major chess websites and potential oversights in detecting sophisticated cheating strategies warrant further investigation and refinement of the proposed methodology.

In contrast, [4] critically examines existing cheat detection methods, highlighting the inherent risks of false positives and advocating for greater transparency in cheat detection processes. The study's

emphasis on the need for cautious and evidence-based approaches in leveling cheating allegations resonates with broader discussions on maintaining integrity and trust in competitive chess environments. However, the study's theoretical critique lacks empirical validation and may oversimplify the complexities of cheating behaviors, necessitating a more nuanced and comprehensive approach to cheat detection research.

The paper [5] explores the allocation of time in decision-making processes, offering insights into the cognitive factors influencing decision-making in chess. By examining the relationship between decision time and perceived complexity and value of options, the study provides valuable implications for understanding human decision-making strategies. However, the study's experimental design limitations and applicability to competitive chess settings warrant further scrutiny and empirical validation to fully elucidate its implications for AI-assisted chess and human-AI collaboration. The authors of paper [6] introduce Irwin, a successor to the cheat detection server "cheatnet," with a focus on online chess. Irwin is described as a secluded computer program designed with a streamlined approach utilizing Stockfish for game analysis. Key components include the 'modules/irwin' section, where TensorFlow learning and application take place, and details on 'Env.py' for interaction with Lichess and database handlers, as well as 'main.py' for accessing the Lichess API and conducting game analysis. This advancement in cheat detection technology represents a significant step in ensuring fair play in online chess.

In contrast, the research presented in [7] delves into the fundamental nature of chess as a game of perfect information, emphasizing its purely mental aspect. The authors explore the possibility of scientifically isolating and predicting well-known chess playing styles such as Aggressive, Positional, Strong, and Cautious, using game data. Through preprocessing chess notations to extract features and utilizing chess engines for analysis, the study examines data from thousands of games, identifies common playing style patterns, and develops prediction models. The findings contribute to a more systematic understanding of chess strategies, revealing both distinct styles and areas of overlap.

Moving beyond chess, [8] offers a comprehensive examination of Benford's Law, which observes that lower digits occur more frequently than higher digits in various data sets. The book explores the theoretical underpinnings, origins, and applications of Benford's Law, particularly in fraud detection. It discusses how fraudsters often overlook this statistical pattern, providing opportunities for detecting financial fraud. The book introduces a new algorithm capable of detecting fraud, even when perpetrators are aware of Benford's Law, by leveraging a subtle pattern within the Benford pattern itself, adding a new dimension to data analysis in the legal domain.

In the realm of steganalysis, [9] investigates the application of deep learning techniques to address the challenge of detecting hidden messages in images of varying dimensions. The study categorizes existing neural network architectures and reveals inconsistencies in performance across different image sizes. The goal is to adapt deep learning models for improved performance invariance and test these models on concealed images, contributing to advancements in steganalysis.

Similarly, [10] presents computer software designed to detect cheating in chess by analyzing engine relationships and metrics in game data, available at PGN-Spy on GitHub. The authors stress the importance of proper statistical analysis against relevant benchmarks and recommend analyzing games from elite grandmasters for pattern comparisons. The software includes various settings for game analysis and post-analysis filter options to aid in identifying potential engine-assisted play in chess games, serving as a valuable tool in upholding the integrity of online chess competitions.

Lastly, in [11], Rafael Leite's analysis investigates the relationship between chess ratings and performance metrics ACPL and STDCPL, focusing on Hans Niemann's controversial rise in the chess world. By comparing thousands of game moves with Stockfish 15 engine suggestions, Leite identifies a near-perfect linear relationship between players' ratings and their performance metrics. However, Niemann's metrics suggest a playing quality of 2500-2550, despite his 2700 rating, indicating discrepancies with expected performance levels. This study proposes a novel approach for evaluating chess players, potentially useful for identifying anomalies and enhancing fairness in player ratings.

3. Parameters in Dataset

As it was mentioned above the goal of the paper is to create the Neural network for analyzing the chess games. For this the dataset must be created. Based on our research we chose the following parameters for our dataset.

Centipawn loss (CPL) is a key component of the training parameters for the final neural network that will assess the fairness of a chess game. The key criterion for identifying engine users and centaurs-players that mix their human inventiveness with the computing capacity of chess engines - is CPL, which stands for a tenth of a pawn.

As was previously mentioned, human chess players are no longer as good as chess engines, hence human analyses of moves and strategies are no longer as meaningful. Moves and game strategies are frequently indicated in engine analysis with positive and negative values, respectively, indicating an advantage for white or black. That said, these concepts hold true if a game plays out as the engine predicts. On the other hand, the perceived advantage may decrease or even change in the opponent's favor if a player performs less-than-ideal plays. The reason for this variation in engine evaluation is that people are not able to play as flawlessly as computers, which frequently leads to less-than-ideal moves. The centipawn loss is the difference between the positional evaluation of the position had the optimal move been made and the move that the player made as evaluated by the engine. A low CPL score suggests that the user's gameplay is similar to that of a supercomputer, perhaps exposing a pure engine user. However, additional characteristics need to be taken into account in order to effectively identify centaurs, which are powerful chess players who can, not only execute great moves on their own, but also consciously choose the second or third-best plays recommended by the engine in order to avoid direct
linkage with engine assessments.

However, CPL is not a monolith; its calculation can vary significantly across different chess engines. This variation is important because different engines, including as Stockfish, Leela Chess Zero (Leela), and AlphaZero, have distinct algorithms and may favor alternative moves in the same situation. We can greatly improve the accuracy of the neural network by getting CPL from several sources, like Stockfish, Leela, and AlphaZero. With this method, the network can more accurately detect cheaters and even determine which particular engine they may be running.

Our approach also considers the association between player strength and centipawn loss as a crucial parameter. Based on empirical information, there appears to be a strong inverse relationship: the average centipawn loss reduces as player rating, which is a proxy for strength, increases. Higher ranked players appear to follow a pattern that more closely resembles the plays that an engine would suggest. On the other hand, situations in which players display a centipawn loss that is much smaller than their rating would indicate—particularly when paired with other questionable elements—may result in a definitive finding of fair play breaches.

Time is a crucial component of the algorithm as well as the game; to determine whether a game is fair, the neural network takes into account how long each player takes to move. When playing online, people who consistently make moves at regular intervals are a major red flag for possible cheating. Advanced tactics to avoid this detection, however, include timing your moves differently, which could cause a discernible difference in your comprehension of the game and your capacity to figure out intricate plans.

A crucial factor included in the building of the neural network for chess game analysis is the divergence from both the typical human play and the distinct style of each individual player, as represented by Maia neural networks. This dual-focused statistic evaluates how a player deviates from their own past playing patterns as well as how their movements vary from the more general tendencies seen in human chess play. With such a thorough approach, the algorithm can identify gaming irregularities that might point to cheating. Through analyzing departures from expected performance on two fronts against the collective norms of human players and against a player's established style— the network is better able to pinpoint situations in which a player's decisions may be influenced by outside help.

The parameters for the dataset were collected from ficsgames.org and chessbase.com. From the lichess.org were collected the parameters in order to train Maia AI. The parameter received from Maia AI was added to the main dataset. In whole 25000 games were gathered. The final size of the dataset was 2.5 GB. The dataset included the games of Grand Masters, Cheaters and personal games.

4. The offered Neural Network

The neural network architecture consists of an input layer, hidden layers (combining convolutional and recurrent layers), and an output layer. The size of the input layer depends on the chosen representation of chess data, such as the FEN string length. Activation functions, like ReLU for convolutional layers and tanh or sigmoid for recurrent layers, are employed to introduce non-linearity. For binary classification (cheating vs. fair play), the output layer comprises one neuron with a sigmoid activation function.

The binary cross-entropy loss function is used to measure the difference between predicted probabilities and true labels. The Adam optimizer, known for adaptive learning rates, is employed for

optimizing the model. The learning rate and batch size are hyperparameters that can be tuned during experimentation for effective model convergence and resource utilization.

The data for the training was collected from open source databases of lichess, chessbase and ficschess. More specifically these bases offered us a vast quantity of games both by humans and engines allowing us to further analyze them and extract the needed parameters from them. The games are downloaded as .pgn, by utilizing a python script we are able to process these games and evaluate each move with the engines and neural networks of our choice and then write the results in an excel file that will later be used for our sequential model to evaluate the fairness of the model.

Here is offered the pseudo code of the neural network:

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import
train_test_split from sklearn import metrics
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.ensemble import
RandomForestClassifier # Load dataset
df = pd.read_csv('work.csv', encoding="utf8", errors='ignore')
# Data preprocessing
df.columns = [...]
encode_numeric_zscore(df, ...)
encode_text_dummy(df, ...)
# Split features and labels
x_columns = df.columns.drop('Final')
x = df[x_columns].values
dummies = pd.get_dummies(df['Final'])
y = dummies.values
# Split into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25,
random_state=42) # Build and compile the neural network model
model = Sequential()
model.add(Dense(10, input_dim=x.shape[1], activation='relu'))
model.add(Dense(50, input_dim=x.shape[1], activation='relu'))
model.add(Dense(10, input_dim=x.shape[1], activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
model.add(Dense(y.shape[1], activation='softmax'))
model.compile(loss='categorical_crossentropy',
optimizer='adam')
monitor = EarlyStopping(monitor='val_loss', min_delta=1e-3, patience=5, verbose=1, mode='auto',
restore_best_weights=True)
# Train the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), callbacks=[monitor], verbose=2,
epochs=10)
# Evaluate the model on test
data pred =
model.predict(x_test)  pred =
np.argmax(pred, axis=1)
y_eval = np.argmax(y_test, axis=1)
score = metrics.accuracy_score(y_eval, pred)
print("Validation score:", score)
# Load test dataset
df_test = pd.read_csv('work_test.csv', encoding="utf8", errors='ignore')
# Data preprocessing for test data
df_test.columns = [...]
encode_numeric_zscore(df_test, ...)
```

```
encode_text_dummy(df_test, ...)
# Make predictions on the test data
x_test[0] = df_test.values[-1]
pred = model.predict(x_test)
pred = np.argmax(pred, axis=1)
y_eval = np.argmax(y_test, axis=1)
score = metrics.accuracy_score(y_eval, pred)
print("Validation score on test data:", score)
print(pred[0])
```

The neural network delivered commendable accuracy, showcasing its robust capabilities. Following an extensive training period of 150 epochs, the validation score reached 0.98. This underscores the model's proficiency in distinguishing between human and computer engine play, attaining a high level of precision. The attained accuracy of 0.98 not only reflects the model's success in capturing nuanced patterns but also holds significant implications for real-world applications. The ability to discern between human and computer-generated moves is essential, particularly in detecting unfair play or optimizing user experiences in chess gameplay. This accomplishment underscores the efficacy of the model's architecture and training strategy, setting the stage for further exploration into interpretability and fine-tuning to maximize its potential.

5. Conclusion

Our study's achievement of a 98% accuracy rate in distinguishing between human and computer-generated chess play underscores the effectiveness of using sequential neural networks, centipawn loss (CPL), insights from Maia neural networks, and time distribution analysis as primary parameters. CPL metrics, highlighting deviations from optimal play, alongside Maia's predictions of human-like moves, offer a nuanced approach to identifying potential computer assistance. Time distribution analysis further refines this by examining the pacing of moves, which can indicate unnatural consistency suggestive of engine use.

However, a notable limitation of our research is the exclusion of centaur games, where players leverage chess engines for assistance. These hybrid matches represent a significant challenge for detection methodologies, as they blend human creativity with computer precision. Including centaur games in future analyses could dramatically enhance our algorithm's ability to discern between solely human play and that augmented by artificial intelligence.

In essence, our study's high success rate demonstrates the potential of integrating diverse analytical tools to combat digital cheating in chess. Yet, the evolution of our detection methods must continue, especially to accommodate the complexities introduced by centaur games. Addressing this gap promises not only to bolster the integrity of chess competitions but also to advance our understanding of the intricate interplay between human ingenuity and computational power in the realm of competitive sports.

6. Future Plans

In our next steps, we plan to significantly diversify the tools and methods used to scrutinize chess matches by bringing in a larger variety of chess analysis software. This strategy aims to go beyond merely spotting unauthorized software aid to pinpointing the exact software variant in use. By incorporating a mix of well-established and cutting-edge chess analysis technologies, we intend to refine our approach to reveal specific signatures that different software imprints on the games it affects. Tailoring our approach to detect distinctive features of each software type will enhance our grasp on the digital nuances affecting chess.

To enrich our analysis, acquiring data that includes games where players have combined forces with chess software is paramount. Such data will be pivotal for refining and testing the improved detection capabilities. The distinctive blend of human strategic thinking and software precision in these games presents complex challenges that our current setup may not fully address. By integrating these games into our analysis, we aim to significantly improve the accuracy and adaptability of our model, making it a stronger safeguard for the integrity of chess competitions.

Exploring the professional trajectories of players who may have utilized software assistance is also on our agenda. This exploration will illuminate the broader implications of software assistance on the competitive scene and player development. This in-depth investigation will not only aid in fine-tuning our detection mechanisms but also foster a broader dialogue on ethical standards within chess and potentially beyond. Through these comprehensive strategies, we aim to advance the detection and understanding of digital interference in chess, ensuring a fair and equitable competitive environment for all.

7. References

[1] Petrick, Thomas, et al. "Building a Chess AI with Deep Learning.", 2020

[2] McIlroy-Young, Reid, et al. "Learning models of individual behavior in chess." Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022.

[3] Laarhoven, Thijs, and Aditya Ponukumati. "Towards Transparent Cheat Detection in Online Chess: An Application of Human and Computer Decision-Making Preferences." International Conference on Computers and Games. Cham: Springer Nature Switzerland, 2022.

[4] Barnes, David J., and Julio Hernandez-Castro. "On the limits of engine analysis for cheating detection in chess." Computers & Security 48 (2015): 58-73.

[5] Chabris, Christopher F., et al. "The allocation of time in decision-making." Journal of the European Economic Association 7.2-3 (2009): 628-637.

[6] Irwin - the protector of lichess from all chess players, villainous, 2018, URL: https://github.com/clarkerubber/irwin

[7] Jayasekara, M. G. P. B. Classification of Chess Games and Players by Styles Using Game Data. Diss. 2021.

[8] Kossovsky, Alex Ely. Benford's law: theory, the general law of relative quantities, and forensic fraud detection applications. Vol. 3. World Scientific, 2014.

[9] Planolles, Kévin, Marc Chaumont, and Frédéric Comby. "A study on the invariance in security whatever the dimension of images for the steganalysis by deep-learning." ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023.

[10] Chess game analyser for detection of cheating, 2017, URL: https://github.com/MGleason1/PGN-Spy

[11] Leite, Rafael V., and Anderson VC de Oliveira. "Expected Human Performance Behavior in Chess Using Centipawn Loss Analysis." International Conference on Human-Computer Interaction. Cham: Springer Nature Switzerland, 2023.