An Empirical Study of AttLSTM Neural Networks for Chess Move Prediction

MSc Research Project
Data Analytics

Ambar Balkrishna Pawar
Student ID: x21172641

School of Computing
National College of Ireland

Supervisor: Dr. Abubakr Siddig

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ambar Balkrishna Pawar
Student ID:	x21172641
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Abubakr Siddig
Submission Due Date:	14/08/2023
Project Title:	An Empirical Study of AttLSTM Neural Networks for Chess Move Prediction
Word Count:	8995
Page Count:	25

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ambar Balkrishna Pawar
Date:	17th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An Empirical Study of AttLSTM Neural Networks for Chess Move Prediction

Ambar Balkrishna Pawar
x21172641

Abstract

This research present a comprehensive exploration of advanced chess move prediction using deep learning techniques. Leveraging various Python libraries, including TensorFlow and Keras, the research meticulously construct and fine-tune an Attention-enhanced LSTM model for predicting subsequent chess moves based on historical game data. Through thorough evaluation, model showcases commendable accuracy in move prediction, achieving 87% on the test dataset. Employing graphical simulations, the research visually depict the capabilities of the model in generating chessboard states and predicting moves. Furthermore, research demonstrate the model's strategic capabilities by engaging it in chess matches against the renowned Stockfish engine. Impressively, model manages to secure 4 draws against Stockfish, a remarkable feat considering its status as one of the most powerful chess engines. This study also encompasses insights into the dataset utilized, which spans a diverse collection of chess games. Overall, this research contributes to the advancement of chess move prediction methodologies and underscores the potential of deep learning in complex board games.

1 Introduction

1.1 Motivation and Background

Chess, a game of strategy and intellectual prowess, has captivated players and researchers for centuries. Its roots can be followed to ancient India during the 6th century, and from that point forward, it has matured into a game of profound intricacy and richness. Over the course of time, remarkable players and iconic matches have contributed to the captivating charm and cultural importance of chess. Nonetheless, in the era of artificial intelligence and machine learning, chess has transformed into a fertile realm, also for exploration and innovation.

The existence of extensive datasets containing historical game records, along with the progress in machine learning techniques, has opened doors for creating models capable of anticipating moves and evaluating player capabilities with growing precision. This convergence of chess and machine learning presents unparalleled prospects to delve into the complexities of the game and expand the horizons of achievable outcomes.

The field of machine learning has witnessed remarkable advancements Gobet (2018), particularly in the domain of deep learning. Advanced models like AttLSTM have brought about a significant transformation in how chess is approached and studied. These models harness the capabilities of deep learning methods to anticipate moves and tactics, offering valuable perspectives into players' choices and unveiling fresh prospects for exploration in data analysis.

Predicting moves in chess is a challenging task due to the vast number of potential moves and the combinatorial explosion of possibilities. The subjective nature of evaluating board positions further complicates the prediction process Tesauro et al. (1995). Each player brings their unique skills and strategies to the game, resulting in varying evaluations of board positions. Developing a universally applicable prediction model that accommodates these diverse evaluations presents a significant challenge.

Additionally, meeting the computational demands to handle extensive data essential for predicting chess moves can be quite challenging. The need for efficient and specialized technology to handle the computational complexities of the game becomes apparent.

Given the difficulties posed by these issues, the aim of this study is to develop a reliable and efficient system for predicting chess moves. Taking into account the complexities and limitations of the game, the objective is to enhance chess players' performance and push forward the domain of data analysis. This research investigates machine learning methods, specifically focusing on LSTM networks, to uncover the connections among different elements of the chess game and forecast moves.

1.2 Research Question and Objectives

Precisely forecasting chess moves is captivating as a research challenge and carries practical benefits within the chess circle. Precise move prediction aids player training, bolsters game analysis, and fosters novel game variations. Hence, it's pivotal to examine the influence of different game aspects, like piece arrangements and remaining time, on move prediction accuracy.

To address this, the research question to explore is: **How can LSTM-based neural networks with attention mechanisms improve the accuracy of predicting chess moves and provide strategic insights in chess games?** This inquiry aims to unveil the key factors driving accurate predictions, while also identifying less impactful features. Moreover, the wide-ranging implications of precise move prediction in chess extend to various aspects within the chess community. Consequently, the investigation into accurate move prediction holds significant potential for advancing machine learning and data analysis applications.

The use of a distinct chess move dataset in this study presents several challenges for achieving accurate predictions. The dataset's considerable size and complexity necessitate a thorough analysis of potential constraints. To address these issues, this research employ the AttLSTM model, renowned for its aptitude in handling sequential data like chess moves. The research meticulously configure the model's architecture and hyperparameters to optimize its performance on the dataset. As highlighted by Schmidhuber et al. (1997), existing research suggests that the AttLSTM model surpasses alternative methods in terms of accuracy and efficiency, a notion further explored and supported in the literature review section of this paper. Nonetheless, identifying certain drawbacks associated with the AttLSTM approach, notably its susceptibility to overfitting and the requirement for substantial computational resources. In sum, study offers insights into the potential merits and limitations of this approach for chess move prediction, demonstrating the efficacy of the AttLSTM model in this context.

The outcomes of this investigation hold significant promise for the broader domain of artificial intelligence and machine learning, especially in the arena of sequence prediction. The establishment and utilization of the AttLSTM model for prognosticating chess moves can lay the groundwork for forthcoming inquiries spanning various areas where sequence prediction holds relevance. The model's capacity to accurately anticipate the subsequent move in a chess game could have implications for self-directed systems, natural language processing. Moreover, potential avenues for future research might delve into incorporating supplementary data sources such as player rankings, strategic chess openings, and move analysis.

The study is organized as follows: Section 2 reviews deep learning-based chess move prediction literature. Section 3 covers Research Methods & Specifications, including methodology, data preprocessing, and the proposed AttLSTM model. Section 4 details Design Specification for LSTM and Attention-based LSTM. Section 5 explains Implementation, encompassing data import, model creation, and training. Section 6 evaluates models across experiments, including loss, accuracy, single-move prediction, user vs. model simulations, and model vs. Stockfish. Discussions follow, leading to the Conclusion and Future Work, summarizing findings and suggesting future research paths.

2 Related Work

The aim of this section is connected to previous research on neural networks in the field of chess and other sectors. This segment is divided into four components:: 1) Predicting Chess Moves using Neural Networks 2) Introduction to Attention Mechanism in LSTM Models 3) Existing Research on Attention-based LSTM in Chess Move Prediction 4) Synthesis.

2.1 Predicting Chess Moves using Neural Networks

Chess move prediction is a challenging task that has seen significant advancements with the integration of neural network-based approaches. Researchers have explored various deep learning architectures to analyze chess positions and predict optimal moves. Among these designs, convolutional neural networks (CNNs) have become a favored selection because of their capability to apprehend spatial characteristics on the chessboard.

The paper by Oshri and Khandwala (2015) from Stanford University presents a novel approach to chess move prediction employing a three-tiered Convolutional Neural Network (CNN), the research address the challenge at hand. The problem is approached as a two-step classification task. Initially, a piece-selector CNN assesses viable white pieces for movement, while move-selector CNNs, corresponding to individual pieces, gauge possible move options. This approach effectively reduces the complex chess class spectrum by a square root, thereby elevating the efficiency of move prediction. The networks are trained on a dataset of 20,000 games and validated against a separate dataset. The best model for the piece selector achieves a validation accuracy of 38.3%, while move-selector networks show varying performance for different pieces. Notably, pieces that move locally outperform those with global mobility. The CNN-based AI is tested against the Sunfish Chess Engine, resulting in 26 draws out of 100 games played. The authors recommend that convolutional layers are beneficial for recognizing small, local tactics in chess, and suggest combining this approach with evaluation functions for more intelligent overall play.

The paper by David et al. (2017) introducing an inventive holistic learning approach for chess, harnessing the capabilities of deep neural networks. Utilizing a two-fold training strategy involving independent preliminary training and guided training without any prior chess rule insights, the deep neural network gains proficiency in recognizing intricate patterns from board positions and making informed choices for optimal positions. Operating solely on extensive datasets of chess games and without specialized chess knowledge, the resultant neural network, dubbed DeepChess, achieves a performance akin to that of a skilled chess player, aligning with contemporary manual feature-focused chess programs. This investigation represents a significant stride in machine learning-oriented methodologies for autonomous chess learning, highlighting the transformative potential of deep neural networks in achieving competitive chess-playing prowess.

In the paper by McGrath et al. (2022), investigate the insights gained from AlphaZero, a neural network-based engine that excels in chess through self-generated gameplay experience. Despite training without access to human games or guidance, the system demonstrates an ability to grasp ideas similar to those used by human chess players. The authors provide evidence through linear probes applied to AlphaZero's internal state, allowing them to quantify the representation of such concepts within the network. Additionally, a behavioral analysis of opening play, accompanied by insightful commentary from a former world chess champion, further enhances their understanding of AlphaZero's learning process. The study sheds light on the fascinating capacity of neural network models to develop sophisticated chess knowledge through self-training, making it a remarkable achievement in the domain of artificial intelligence and chess.

In the last few years, neural network models have displayed encouraging outcomes in the field of automated essay scoring. Previous studies have investigated the application of recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to analyze input essays and assign grades using a unified vector representation of the essay content. However, a direct comparison between the advantages of RNNs and CNNs has been lacking. Additionally, existing models fail to account for the varying contributions of various sections of the paper contribute to the overall evaluation. To address these gaps, this paper presents a hierarchical sentence-document model that leverages the attention mechanism to effectively allocate varying importance to words and sentences, the outcomes reveal that the suggested model surpasses earlier cutting-edge techniques. This underscores the efficacy of the attention mechanism in automating the evaluation of essays. (Dong et al.; 2017).

The paper by Wölflein and Arandjelović (2021) addresses the challenging problem of recognizing the arrangement of chess pieces depicted in an image of a chessboard, using computer vision techniques and deep learning. The authors highlight the importance of accurate chess piece recognition for facilitating automatic computer analysis and aiding amateur chess players in improving their games. To overcome the limitations of existing approaches, the study introduces a novel dataset generated from a 3D model, which is notably larger compared to existing datasets. The proposed chess recognition system integrates traditional computer vision techniques with two convolutional neural networks. This fusion results in a remarkable test set error rate of 0.23% per square, surpassing the current leading approach by a significant factor of 28. Additionally, the research introduces a few-shot transfer learning method that enables the system to adapt to new chess sets using only two photos of the initial position. This approach achieves an impressive per-square accuracy of 99.83%. The accessibility of the code, dataset, and trained models online further contributes to the wider research community.

2.2 Introduction to Attention Mechanism in LSTM Models

The attention mechanism is a valuable tool in sequence modeling. It boosts the abilities of LSTM models to grasp key patterns and relationships in sequences. When it comes to predicting chess moves, this mechanism becomes essential. It allows the model to concentrate on important moves and positions, factoring in the strategic impact and lasting connections of each move. This part offers an introduction to the attention mechanism's role in LSTM models for predicting chess moves, taking inspiration from its use in other fields.

The attention mechanism addresses the limitation of traditional LSTM models, which treat all elements in a sequence equally. On the other hand, attention mechanisms assign varying importance to elements within a sequence, enabling the model to concentrate on pertinent elements and skillfully grasp relationships throughout the sequence. By doing so, attention-based LSTM models can better understand the context and importance of each move in a chess game, leading to improved move predictions.

The paper presents a study that explores the sequential nature of chess moves made during matches and investigates the applicability of LSTM layers for sequence modeling. The authors propose a novel model architecture that combines LSTM layers with chess move and game metadata data types, aiming to achieve high classification accuracy. The results demonstrate that the model built solely with LSTM layers is effective in interpreting chess moves as sequences. Additionally, the study compares two chessboard representation methods, bitmap input, and algebraic input, to determine their relevance for neural network training. Surprisingly, better scores were obtained using the bitmap input, despite carrying less information theoretically than the algebraic input. The results offer valuable understandings regarding the appropriateness of LSTM models for anticipating chess moves and highlight how representing data plays a crucial role in achieving precise predictions.(Drezewskia and Wator; 2021)

The attention mechanism has gained prominence in the realm of deep learning, bringing about significant changes in domains like Natural Language Processing (NLP) and Computer Vision. Comprehensive guide by Prodip Hore and Sayan Chatterjee, the attention mechanism's significance and workings in deep learning algorithms are thoroughly explored. The paper provides valuable insights into how attention mechanisms have changed the way this research approach various tasks in NLP and Computer Vision. Additionally, the guide offers a practical implementation of the attention mechanism in Python, enabling researchers and practitioners to leverage this powerful technique in their own projects. The paper serves as an essential resource for anyone seeking to understand and utilize the attention mechanism in the realm of deep learning.(Hore and Chatterjee; 2023)

2.3 Existing Research on Attention-based LSTM in Chess Move Prediction

Existing research on attention-based LSTM in chess move prediction has expanded to explore the potential of natural language transformers in supporting generic strategic modeling for text-archived games. Notably, the work of Noever et al. (2020). presents the Chess Transformer, a fine-tuned model based on OpenAI's Generative Pre-trained Transformer (GPT-2) architecture. Through practice on 2.8 million Portable Game Notation chess games, the Chess Transformer demonstrates its ability to generate meaningful moves on a chessboard and develop complex gameplay strategies. The model optimizes weights for 774 million parameters, showcasing its capacity to produce plausible game formations akin to classic openings like English or the Slav Exchange. Additionally, the Chess Transformer features an encoder-decoder cycle with a unique 'attention' mechanism, enabling it to effectively prioritize relevant features and learn from vast amounts of textual inputs. The transformer's potential to filter illegal moves and engage in live play with a human-to-transformer interface further enhances its application in chess strategies. This research paves the way for future exploration of transformers in other strategy games, leveraging the power of attention mechanisms to capture complex rule syntax from player annotations.

In recent years, transformer-based language models have shown impressive advancements in different tasks related to understanding human language. Nonetheless, comprehending how reliable these models are in capturing the true meaning behind intricate language remains a puzzle. To tackle this issue, the current literature review delves into an innovative way of using language models in the realm of chess. Unlike regular language, chess notations provide a clear, structured, and predictable environment, which makes it a perfect arena to assess language models. The paper by Toshniwal et al. (2022) explores this concept and presents intriguing findings. Through experiments, the authors reveal that transformer language models, when sufficiently trained on move sequences, can effectively track chess piece positions and predict legal moves with high accuracy. Additionally, the study emphasizes the importance of giving training participants with access to board of state data, particularly in scenarios with limited training data. Moreover, the research highlights that the success of transformer language models in this chess testbed is contingent upon full attention, implying that approximating full attention leads to a notable performance drop. This testbed serves as a valuable benchmark for future investigations, offering insights into the creation and evaluation of transformer model languages for the purpose of predicting chess moves.

2.4 Synthesis

A thorough summary of the body of work on neural network-based chess move prediction as well as the importance of the attention process in LSTM models is provided in the literature review's synthesis. The reviewed papers showcase the advancements made in the domain of chess AI, highlighting the potential of deep learning techniques for understanding chess positions and making optimal move predictions.

The first set of papers explores the use of convolutional neural networks (CNNs) in chess move prediction. Oshri and Khandwala (2015) introduce a novel approach using a three-layer CNN to predict moves. They achieve promising results, especially for pieces with local mobility. David et al. (2017) present DeepChess, a comprehensive learning approach that, without any domain-specific expertise, produces grandmaster-level performance. These papers demonstrate the effectiveness of neural networks, particularly CNNs, in improving chess move prediction accuracy and overall performance.

The subsequent paper by McGrath et al. (2022) delves into the knowledge acquisition of AlphaZero, a neural network engine trained through self-play. It highlights the network's ability study chess strategies comparable to those used by professional players, providing insights into the fascinating capacity of neural network models to develop sophisticated chess knowledge through self-training.

The second part of the literature review introduces the attention mechanism in LSTM models. It discusses the attention mechanism's significance in sequence modeling and its applicability to chess move prediction. The reviewed works emphasize how the attention mechanism enables LSTM models to focus on relevant moves and positions, capturing long-term dependencies and strategic implications in chess games.

The guide by Hore and Chatterjee (2023) provides a comprehensive exploration of the attention mechanism's workings in deep learning algorithms. It serves as an essential resource for researchers and practitioners interested in implementing attention mechanisms in their projects.

Lastly, the work by Noever et al. (2020) presents the Chess Transformer, a fine-tuned model based on the GPT-2 architecture, capable of generating meaningful moves and developing complex strategies in chess. This research opens up possibilities for exploring transformers in other strategy games, harnessing the power of attention mechanisms to learn from vast amounts of textual inputs and prioritize relevant features.

Furthermore, the literature review explores the paper by Toshniwal et al. (2022), which uses chess as a testbed to evaluate the state tracking abilities of transformer language models. The study reveals that, with sufficient training, transformer models can accurately track piece positions and predict legal moves. It also underscores the importance of board state information during training and the impact of full attention on the models' performance in this chess testbed.

The synthesis highlights neural networks and attention mechanisms in chess AI, signaling deep learning's potential for future research.

Table 1: Comparative Table of Limitations in Existing Work

Aspect	Existing Work Limitations	Implications for Your Study
Predicting Chess Moves using Neural Networks	Model Performance: Varied performance for different pieces; dependency on local mobility for higher accuracy	Investigate methods to improve model consistency and handle pieces with global mobility.
	Data Dependency: Training on a limited dataset (20,000 games); potential overfitting	Explore larger and more diverse datasets for improved generalization.
	Evaluation Metric: Focusing on validation accuracy alone; limited insights into strategic gameplay	Introduce additional evaluation metrics considering strategic gameplay elements.
Introduction to Attention Mechanism in LSTM Models	Application Scope: Limited discussion on attention mechanism in LSTM models applied to chess	Extend the discussion to encompass various applications of attention mechanisms in chess AI.
	Performance Metrics: Few insights into how the attention mechanism affects model performance	Investigate the impact of different attention mechanisms on chess move prediction accuracy.
Existing Research on Attention-based LSTM in Chess Move Prediction	Model Generalization: Specific focus on fine-tuned transformer models; may not generalize to other architectures	Explore the general applicability of attention-based LSTM models beyond transformers.
	Lack of Comparative Analysis: Limited comparison between transformer-based models and other LSTM architectures	Conduct a comprehensive comparative analysis to identify the most effective model for chess move prediction.
Overall Limitations of Existing Work	Limited Attention to Chess-specific Features: Most studies lack detailed analysis of chess-specific features and strategies	Investigate the incorporation of chess-specific knowledge into neural network models.
	Lack of Real-time Play Evaluation: Few studies evaluate models in real-time chess games	Assess the model's performance in real-time chess gameplay scenarios.

3 Methodology

The process of utilizing machine learning's potential in the field of chess move prediction begins. A well designed architecture must be used to reveal the complex patterns hidden inside a dataset. In order to uncover the secrets buried within the data, this thesis investigates sequential iterative approaches, as seen in Figure 1. A dataset is first loaded into a pre-processing and transformation, and then a rigorous model is trained and tuned. The model is then meticulously monitored and evaluated to guarantee peak performance.

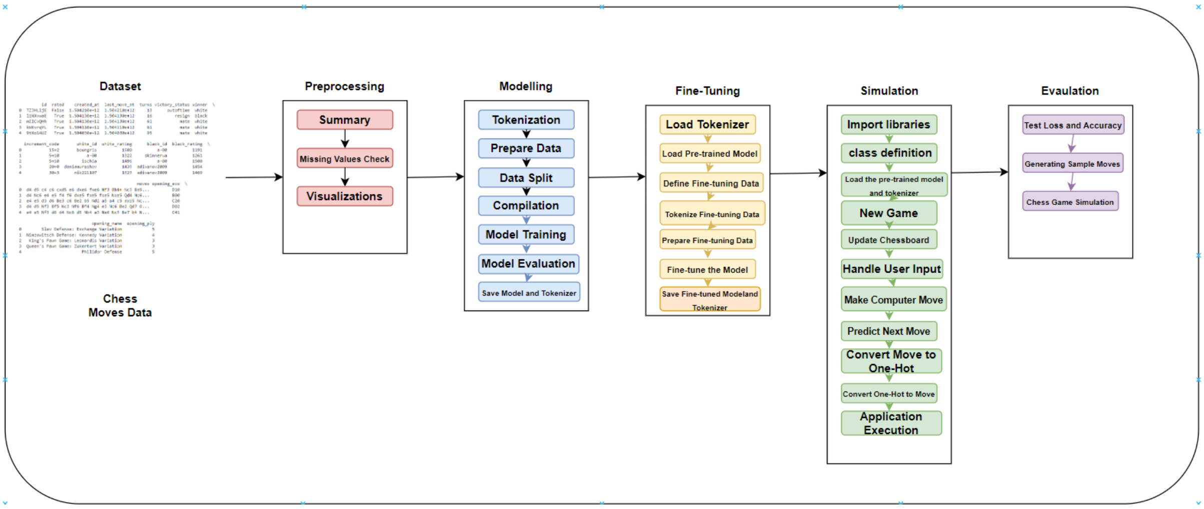

Figure 1: Methodology

3.1 Data Description

Just over 20,000 games from Lichess.org users make up the dataset used for chess move prediction. Inclusions include player IDs, ratings, start and finish timings, Number of moves in conventional chess notation, game status, winner, turn count, opening eco, opening name, as well as and opening ply (the overall number of moves throughout the opening phase). It also includes IDs for players and ratings.

This extensive dataset provides a wealth of knowledge for chess pattern research and data analysis. The influence of meta-factors on game results, the connection between openings and triumphs for players playing as white or black, and the strategic ramifications of winning moves are just a few of the topics that researchers might explore. The dataset is a useful tool for AI and data science enthusiasts interested in improving chess move prediction algorithms and learning more about the strategic gaming analysis appreciating to its wide range of data.[1]

3.2 Data Pre-processing and Transformation

In Figure 2 the overall flow of the data, Preparing the dataset for chess move prediction in this study required significant data translation and preprocessing. The dataset un-

[1] https://www.kaggle.com/datasets/datasnaek/chess

derwent a careful preparation workflow to verify that it was appropriate for the machine learning model. The dataset included game information and movements in conventional chess notation. To ensure data quality and completeness, the initial stage involves looking for missing values and displaying rating distributions. The number of rounds in each game was then carefully examined, giving us insights into the distribution of game lengths and enabling us to make wise choices about data transformation. The research's

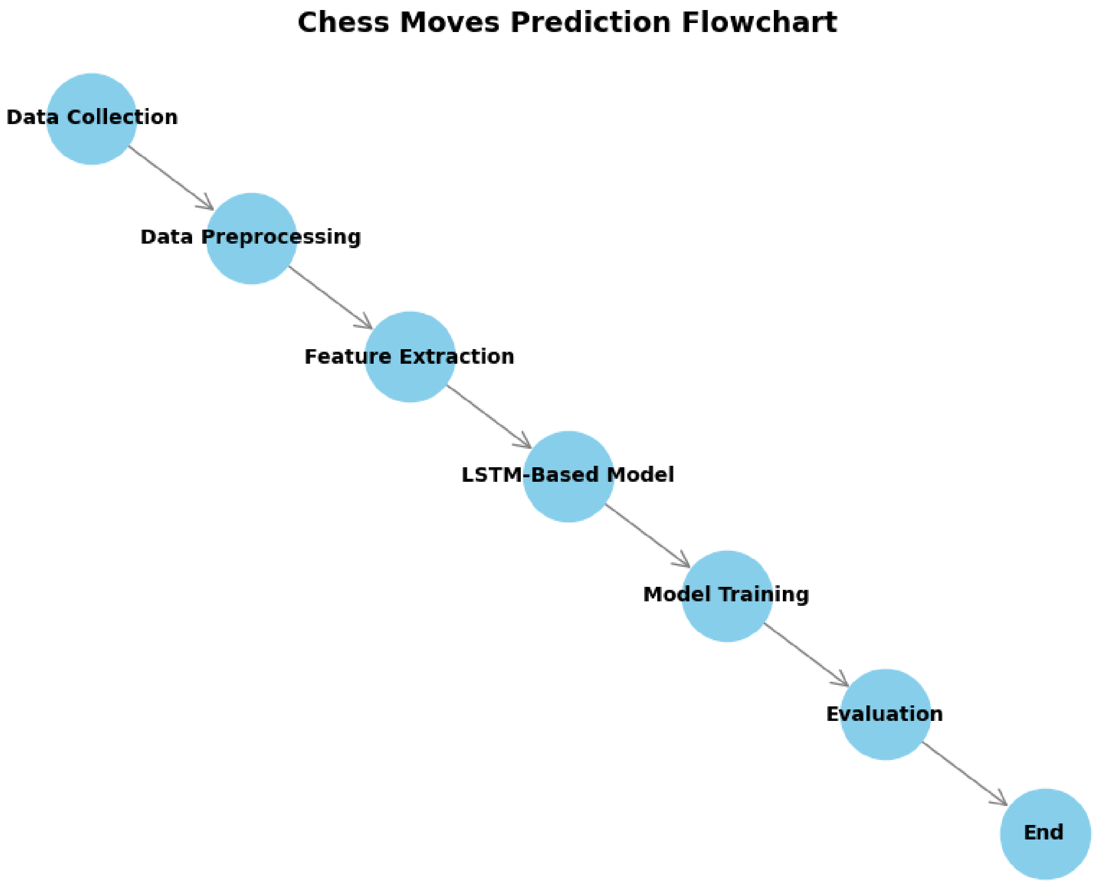

Figure 2: Flowchart

key component was the feature extraction procedure. Successfully encoded the movements by expertly tokenizing the written chess moves into numerical sequences, setting the groundwork for the LSTM-based model with attention mechanisms. This change gave the model the ability to recognize the complex sequential patterns found in chess moves, enabling it to make wise predictions. To fully evaluate the effectiveness of the model, Additionally spliting the dataset into sets for training and testing. This procedure was crucial for obtaining accurate and impartial test findings.

Overall, molding the dataset into a manner appropriate for the machine learning model required significant data preparation and manipulation. Preparing the groundwork for the LSTM-based model to forecast chess moves with amazing accuracy by streamlining the data and extracting useful characteristics, creating new opportunities for strategic gaming analysis and the investigation of AI-driven chess tactics. Enable to harness the power of neural networks and harvest priceless insights from the quantity of data included in the dataset by combining data preprocessing and feature extraction.

3.3 Feature Extraction

A crucial step in converting the raw chess move data into useful inputs for the LSTM-based model is feature extraction. The two main components of the strategy are move encodings and positional characteristics.

3.3.1 Move Encodings

A one-hot vector encoding approach is used to represent chess moves. A binary vector identifying the source and target squares is used to represent each motion. For instance, "e2e4" denotes the "e2" source and "e4" target squares and corresponds to a vector with "1" at locations 63 and 52.

3.3.2 Positional Features

Board positions are converted into numerical sequences using the FEN notation. These sequences record binary values for square occupancy. LSTM model can identify patterns and tactical considerations appreciating to the padding of the data for homogeneous input dimensions. (David et al.; 2017)

The LSTM model with attention receives structured datasets of move encodings and positional characteristics as input from this combined technique. It gives program the ability to comprehend the complex chess dynamics, spanning textual and numerical representations for precise move predictions.

3.4 Modelling

In an effort to promote a deep grasp of chess move prediction, this study employed an LSTM architecture with attentional enhancement. The ability of LSTMs to identify sequential correlations is widely recognized. have demonstrated astounding effectiveness in tasks involving sequence modeling (Schmidhuber et al.; 1997). These pieces naturally counteract the vanishing gradient issue, enabling the capture of complex, multiple-move chess strategy.

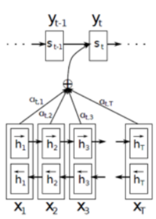

Figure 3: Understanding the Attention Mechanism

For each input motion, the Bidirectional LSTM used here generates a number of annotations, as seen in Figure 3. Including attention processes into LSTM model under the direction of the groundbreaking research of Vaswani et al. (2017) and Britz et al. (2017). This dynamic attention improved model's capacity to give diverse input sequence components varied degrees of significance, simulating the complex human discernment process while examining critical board positions. This architectural improvement shed light on the model's logic for making decisions, which was crucial in the effort to interpret strategic insights.

3.5 Evaluation Criteria

The LSTM-based chess move prediction model's validity depends on a wide range of assessment metrics, each carefully selected to assess particular facets of the model's performance. The criteria used to evaluate this section explains how the model can properly represent intricate chess move sequences as well as emulate strategic human games.

3.5.1 Loss Function and Training Performance

The LSTM-based chess move prediction model is built on a foundation of loss functions and training results. The sparse categorical cross-entropy loss function, a basic mathematical concept, is used extensively throughout this journey. This process, written as:

$$L = -\sum_{i=1}^{C} y_i \log(p_i)$$

where y_i signifies the true label and p_i denotes the predicted probability for class i, methodically directs the model as it improves prediction precision. In order to decrease this divergence over subsequent training epochs, the model is driven to quantify the difference between predicted probability and true labels.

Each epoch in the training performance domain turns into a canvas where the model's parameters are creatively altered. The loss decreases with each gradient descent iteration, demonstrating the model's improvement in understanding complex chess move patterns. The model's internal representations get tuned to the strategic subtleties as it navigates the complexities of the training dataset, which is supported by the rising accuracy measure.

3.5.2 Accuracy and Model Prowess

Through the lens of accuracy, the story of the model's efficacy is clearly revealed. This crucial indicator perfectly encapsulates how good at anticipating future chess moves the model is. The effectively simple formula for accuracy is as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

It is the proportion of forecasts that accurately reflect actual gameplay actions. High accuracy indicates that the model is adept at understanding how nuanced actions and tactics interact.

In addition, the F1 score, a skillful balance of recall and accuracy, highlights the model's abilities. The F1 score is calculated as follows:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

This score highlights how well the algorithm captures strategic nuance, combining tactical expertise from accuracy with strategic adaptability from recall. An improved F1 score results in a model that reflects a cogent and perceptive gaming forecast and resonates with the complex transitions of a chess game.

4 Design Specification

This research go into the exact design requirements of the models used in the chess move prediction study in this section. The baseline model, the Attention-enhanced LSTM (AttLSTM), and the Long Short-Term Memory (LSTM) model is further employed for the assessment method, are presented together with their architectural designs and essential elements.

4.1 Models

4.1.1 Long-Short Term Memory

The goal of chess move prediction is LSTM model-based: Long Short-Term Memory. Given its skill in identifying sequential patterns in data, this recurrent neural network design is a great choice for simulating the complex dynamics of chess plays. An embedding layer in the design converts the tokenized chess moves into continuous vector representations. This layer feeds into a 64-unit LSTM layer that is intended to maintain long-range dependencies and strategic knowledge during movements. In order to forecast the probability distribution across the vocabulary of the following moves, the LSTM output is then fed through a TimeDistributed Dense layer with a softmax activation function. Figure 4 provides an illustration of this architecture.

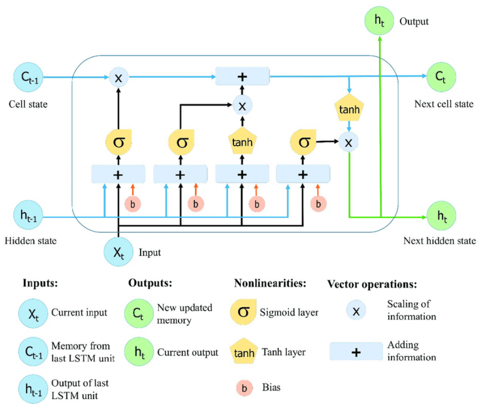

Figure 4: Architecture of the LSTM Model for Chess Move Prediction.

4.1.2 Attention-enhanced LSTM (AttLSTM) Model

To further improve the model's comprehension of crucial plays and tactical intricacy, offering the Attention-enhanced LSTM (AttLSTM) model, building on the LSTM architecture. The AttLSTM adds an Attention mechanism which allows the model to focus on specific motions and their contextual importance while making predictions. An LSTM layer is applied after the embedding layer has processed the input sequence. The model dynamically assesses the significance of each move's contribution to the forecast when an Attention layer is added, which is where the magic happens. The TimeDistributed Dense layer uses this attention-weighted output to anticipate the subsequent movements. The inclusion of Attention enables the model to make predictions that are more perceptive and contextually aware. Figure 5 shows the arrangement of the architecture.

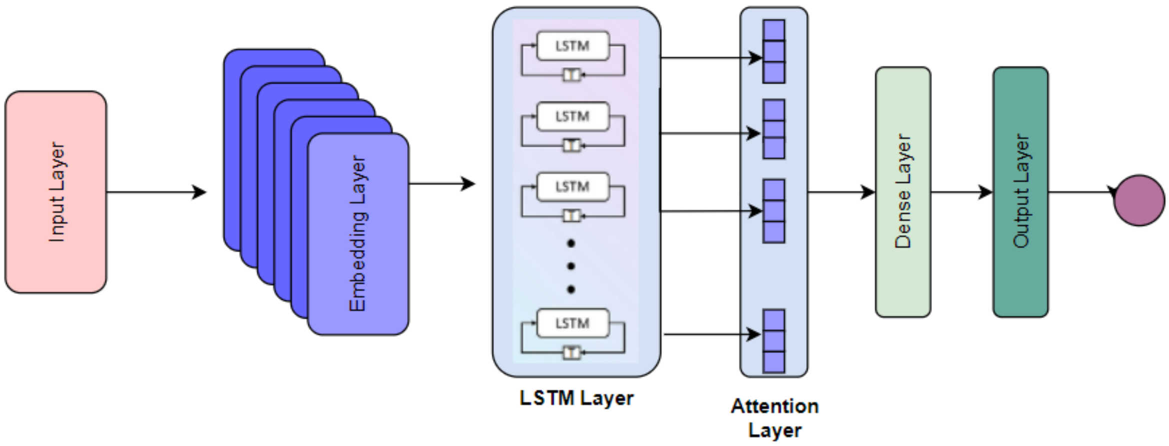

Figure 5: Architecture of the Attention Based LSTM Model for Chess Move Prediction.

5 Implementation

The section below contains information on the implementation:

5.1 Computational Details

In order to develop, train, and evaluate the suggested chess move prediction model, usage of a variety of computational tools and resources during the research's implementation phase. Utilization of a hybrid strategy, combining Google Colab and Jupyter Notebook environments, to expedite execution and take use of powerful hardware capabilities. The flexible Google Colab framework made it possible to run Python programs in a cloud-based setting equipped with GPUs and TPUs. This proved to be quite helpful, especially when training complex neural network models that required a lot of processing power, like Long Short-Term Memory (LSTM) model using attention mechanisms. Additionally, the collaborative and interactive features of Google Colab made it easier for the study team to share code and debug together. In the meantime, usage of Jupyter Notebook for thorough performance measurement and gameplay analysis. In the research for a graphical chess interface that took use of its interactive capabilities and allowed us to play against the powerful Stockfish chess engine in addition to the implemented model was able to evaluate the model's actions due to the real-time visualization component, which also gave more insights into how the model makes strategic decisions.

5.2 Importing the Dataset

The dataset was intentionally imported through storage in Google Drive, making it easier for Python code to directly integrate it into the Google Colab environment. Before being exposed to perceptive visualizations, this dataset underwent meticulous preparation. These visualizations provided a variety of insights on player dynamics and game results by incorporating elements like Rating Distribution by Player Color, Win Distribution by Player Color, and White and Black Player Rating Distribution by Game Status which is seen in Figure 6, Figure 7, and Figure 8 respectively. Figure 9 shows the Average Number of Turns by Opening Name also provided insight on opening tactics. These visualizations served as a springboard for the development of the LSTM and Attention-based LSTM (AttLSTM) models, which were methodically built.

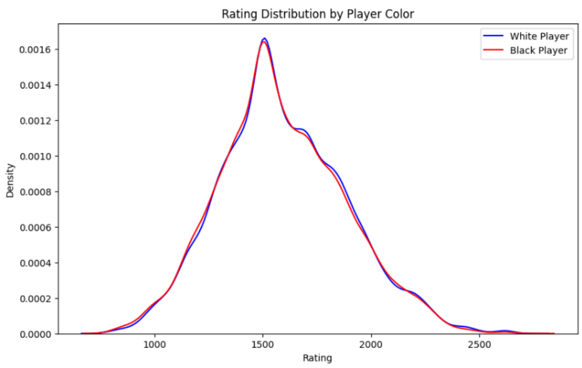

Figure 6: Rating Distribution by Player Color

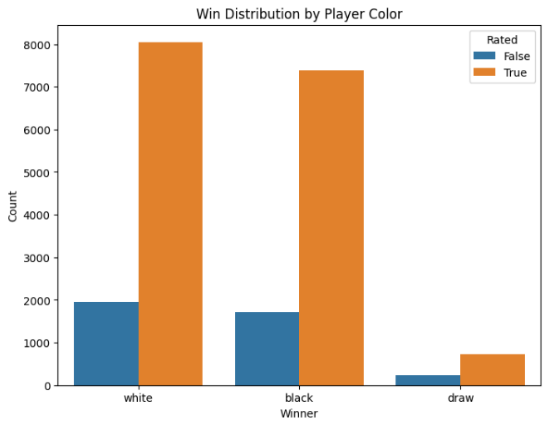

Figure 7: Win Distribution by Player Color

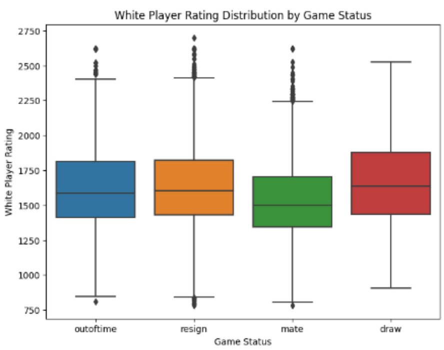

Figure 8: White Player Rating Distribution by Game Status

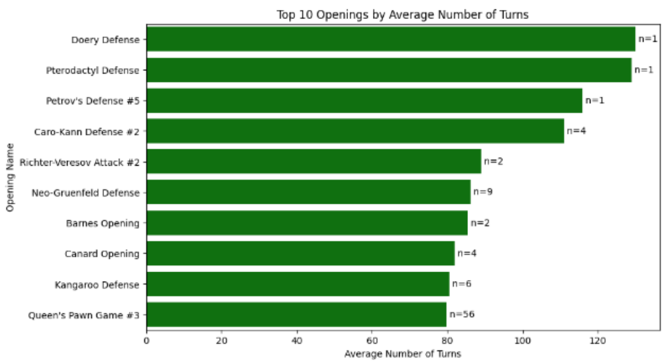

Figure 9: Average Number of Turns by Opening Name

5.3 LSTM Implementation

The development and training of the LSTM-based chess move prediction model were part of the implementation phase. This stage entailed a series of thoroughly planned methods to convert the processed dataset into a solid model that could predict moves with accuracy. The first stage was utilizing the Keras library's Tokenizer class to tokenize the chess moves. This procedure made it easier to create a word index mapping that converted each distinct move into an associated integer value.

The dataset was then altered to fit the architecture of the model of sequence to sequence. The order of the inputs and the target sequence were two separate sections of the move sequences. The goal sequence comprised movements beginning with the second move, however the input sequence only contained moves up to the next-to-last move. These sequences were padded to make sure that their lengths remained constant, enabling the LSTM model to process them.

The dataset was then divided among testing and training sets using the train-test split method. This division made it feasible to objectively assess the model's effectiveness on data. In-order to build the LSTM model, a sequential architecture was used. The motion indices were transformed into dense vectors by an embedding layer first, and then the move sequences' sequential patterns were captured by an LSTM layer. A softmax activation was used by the output layer to forecast the following move for each time step while being encapsulated in a TimeDistributed wrapper.

5.3.1 Training LSTM

The model for training was constructed using a sparse categorical cross-entropy loss function as well as the Adam optimizer. Accuracy was the criteria for evaluation. The model's weights were repeatedly adjusted depending on the computed loss and the training procedure was carried out across 10 epochs with a batch size of 32 to enhance its prediction skills. During training, the model was verified against the testing set, giving information on how well it generalized.

For later evaluations and assessments of the model's skill in foretelling chess moves, this implementation step served as a crucial cornerstone. The resulting LSTM-based model is evidence of the intricate integration of machine learning techniques. with domain-specific knowledge, and it holds the promise of opening up new chess move prediction possibilities.

5.4 Baseline : Attention based LSTM Implementation

The application portion of this work saw a major development with the practical realization of the Attention-based LSTM model. This procedure included a number of carefully planned procedures designed to develop, improve, and maximize the model's predictive power. Preparing the dataset as part of the initial step was done in a manner similar to that used for the LSTM model. The dataset's chess move sequences were encoded into a structured manner using a strict tokenization strategy using the Tokenizer class, simplifying further analysis.

The well-known train-test split method was then used to divide the data into training and testing sets, and input-output pairs were created. The attention mechanism was integrated into the architecture, which was distinct from the Attention-based LSTM model and permitted the model's improved emphasis on important move sequences during prediction. A dynamic model was carefully put together from the Input Layer, embedding layer, LSTM layer, attention layer, Dense Layer as well as the output layer to capture complex move patterns and forecast following movements with accuracy.

5.4.1 Training and Fine-tuning AttLSTM Model

Training was started using the compiled model architecture to evaluate the model's effectiveness and tune its predicted accuracy. Multiple epochs and batch iterations were used throughout the training phase to fine-tune the model's internal parameters. Throughout the training phase, the model's loss and accuracy measures were carefully tracked, which allowed for a thorough grasp of its convergence and predictive power.

An important aspect of this method is the fine-tuning stage, which aimed to make the model's predictions even better. The model was improved to include a wider range of chess scenarios using a variety of extra move sequences. To conform to the enlarged output format, the model's architecture was recompiled, and its loss was calculated using categorical cross-entropy.

The model was given time to adapt to the expanded dataset and improve the accuracy of its predictions during the fine-tuning phase, which was carried out across a number of epochs. Cutting-edge machine learning methods and domain-specific knowledge come together in this careful integration of the attention-based architecture and rigorous fine-tuning procedure. As a result, an Attention-based LSTM model that can capture tactical complexity and anticipate upcoming moves with astounding precision has the potential to transform chess move prediction.

6 Evaluation

The following tests included simulating chess games in which the proposed AttLSTM model was tested by competing against the Stockfish chess engine and a human opponent. These experiments' results were examined using the evaluation criteria listed in section 3.5.

6.1 Experiment 1

This experiment's objective is to assess the loss and accuracy metrics of the LSTM and AttLSTM models. The research seek to untangle the complexities of their predicted performance by thorough testing and analysis, expanding knowledge and improving their forecasting skills.

6.1.1 LSTM Model Results

In Table 2 the performance of the LSTM model across 10 epochs is showed encouraging development, with a accuracy improvement and a reduction in loss. The initial loss of the model was 1.7906, and its accuracy was 0.8277. The model obtained a validation loss of around 0.8460 and a validation accuracy of roughly 84.93% in the last epoch during training. The model performed well on the test data, with a final test accuracy of 85% and a final test loss of 83%. The model's capacity to pick up new information and adjust to the chess move prediction job is highlighted by this rising trend.

Table 2: LSTM Model Results

Epoch	Training Loss	Training Accuracy
1	1.7906	0.8277
2	1.0503	0.8347
3	1.0012	0.8372
4	0.9713	0.8384
5	0.9396	0.8404
6	0.9081	0.8425
7	0.8823	0.8445
8	0.8604	0.8467
9	0.8410	0.8487
10	0.8460	0.8493

Test Loss: 0.8351 **Test Accuracy:** 0.8503

6.1.2 Attention based LSTM Model Results

In Table 3, Ten epochs in order for training the AttLSTM model were employed. Precision and loss were tracked throughout training. The initial loss of the model was 1.7856, and its accuracy was 0.8247. The model's loss dropped and its accuracy rose as training went on. The accuracy was greatly increased to 0.8717 by the last epoch, while the loss had been cut in half to 0.7604. The model regularly outperformed the test data, with a final test accuracy of 87% and a final test loss of 76%. These findings show that the AttLSTM model successfully learned during the training period and exhibited predictive skills.

Table 3: AttLSTM Model Results

Epoch	Training Loss	Training Accuracy
1	1.7856	0.8247
2	1.1498	0.8293
3	1.0800	0.8306
4	1.0398	0.8334
5	1.0147	0.8347
6	0.9895	0.8354
7	0.9545	0.8374
8	0.9030	0.8434
9	0.8402	0.8558
10	0.7765	0.8685

Test Loss: 0.7604 **Test Accuracy:** 0.8717

It is clear from comparing the outputs of the two models that the AttLSTM model performed better than the conventional LSTM. The AttLSTM model demonstrated a notable reduction in loss and a large gain in accuracy despite identical training epochs. The AttLSTM model beat the LSTM model, showing a final test accuracy of 87% and a test loss of 76% as opposed to an accuracy of 84.93% and a validation loss of around

84%. These results highlight how well the AttLSTM model's prediction abilities may be improved by including attention processes.

6.2 Experiment 2

In Experiment 2, the research used the python code to investigate the model's capacity for dynamic prediction. The algorithm that has been put in place enables interactive move prediction from a specified beginning sequence. The model uses the researcher's actions to dynamically forecast the next movements. Tokenizing utilizing the model that was trained to anticipate the input sequence as well as the following move are both steps in the prediction process. The results show how the model can adapt dynamically to different move sequences by making a rough forecast for the following move based on the input.

The output log, as seen in Figure 10 below, illustrates how recursive the prediction process is:

```
Enter the sequence of moves: b1a3
1/1 [==============================] - 1s 641ms/step
The predicted next move is: e7e5
Enter the sequence of moves: c2c4
1/1 [==============================] - 0s 64ms/step
The predicted next move is: g8f6
Enter the sequence of moves: h2h4
1/1 [==============================] - 0s 33ms/step
The predicted next move is: b8c6
Enter the sequence of moves: e2e4
1/1 [==============================] - 0s 41ms/step
The predicted next move is: b7b6
Enter the sequence of moves: b2b3
1/1 [==============================] - 0s 33ms/step
The predicted next move is: g8h6
Enter the sequence of moves: d2d3
1/1 [==============================] - 0s 35ms/step
The predicted next move is: g8f6
Enter the sequence of moves: f2f3
1/1 [==============================] - 0s 40ms/step
The predicted next move is: c7c5
```

Figure 10: Dynamic Prediction of the next move

The AttLSTM model's predictive skills in the setting of dynamic chess move prediction are further demonstrated in this experiment, which also shows the model's interactive aspect by enabling researchers to enter a sequence of moves and obtain anticipated following moves.

6.3 Experiment 3

In Experiment 3, a graphical chess interface was built so that participants may challenge the AttLSTM model's move predictions. The code creates a platform where users may make movements, and the model reacts with its projected moves using the PyQt5 library. The interface uses a visually appealing chessboard layout to demonstrate the dynamic nature of the model's predictions. With each move, the chessboard is rendered as well as the learned model for move prediction is integrated into the given code.

The stages for implementation are as follows:

- A text input box and an empty chessboard are present when the interface first starts up.

- The user enters their move using the conventional notation for algebra (for example, "e2e4").
- The user's move is carried out on the chessboard after pressing the Enter key.
- Based on the revised board position and taking into account both players' actions, the model forecasts its next move.
- On the board, the model's move is put into action.
- The user and the model alternate roles as this back-and-forth interaction continues.

The graphical interface's capacity to graphically depict the board and interact with user input defines it. A model move is triggered by each user move, making the game of chess fluid and real-time.

After 10 matches versus the model, the performance evaluation yielded a mixed bag of results. The model was able to draw three times, win three times, and lose four times out of the 10 games that were played. These outcomes highlight the AttLSTM model's aggressive and adaptable behavior in an interactive chess game. The range of outcomes the complexity of the game's mechanics and the model's ability to choose a course of action depending on the changing board state.

Potential mistakes are also taken into consideration during implementation. To ensure respect to accepted chess rules, the interface informs the user to try again if they submit an invalid move rather than continuing the game.

Figure 11 & Figure 12 shows the graphical chess game:

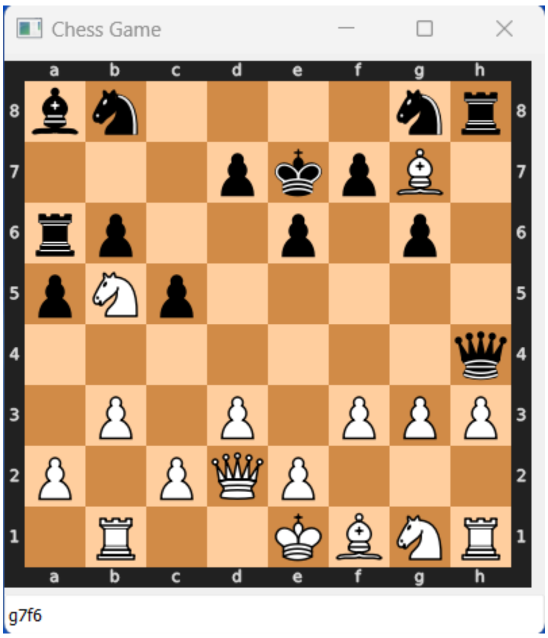

Figure 11: User's Move (g7f6)

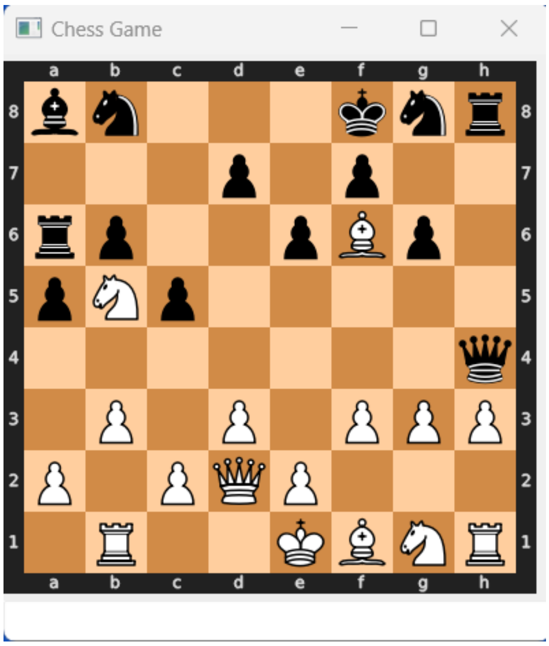

Figure 12: Model's Move (e7f8)

This experiment demonstrates the AttLSTM model's capacity to predict chess moves dynamically within a game setting, and it serves as an example of how machine learning models may be integrated into interactive applications.

6.4 Experiment 4

In this experiment, the research compared trained model to Stockfish Engine, one of the most powerful chess engines, in order to evaluate its performance. The objective was to assess the model's performance in a difficult setting and compare it against a powerful opponent.

The research involves creating a thorough testing framework for this experiment using the Stockfish Engine and the Python Chess module. In this project, development of a unique ChessGame class that smoothly combines the model with the Stockfish Engine. By drawing the chessboard and enabling user input, this class initializes the game interface. The interaction between the model and the Stockfish Engine, where each takes turns making movements, is the basis of the experiment.

The research started assessment by loading pretrained model and tokenizer into the Stockfish Engine. Then, to ensure there were enough data points for analysis, setting up a loop to coordinate 50 games. The crucial actions in each iteration were as follows:

- As the white player, model performed a move utilizing the attention-based LSTM, which has undergone rigorous fine-tuning. By simulating a variety of alternative actions and outcomes, this program automatically chooses the most advantageous approach, greatly increasing decision-making.

- The research verified that the game was over after the model's move. If not, the black player, the Stockfish Engine, was instructed to reply with a move of its own.

- Up to the end of the game, these other maneuvers were performed. The research used the Stockfish Engine to analyse the positions and choose the moves during the game, which helped to make it competitive and interesting.

- The research kept track of each game's final score, whether it was a victory, defeat, or a tie, and added up the results for analysis.

This experiment produced encouraging and illuminating findings. In Figure 13 derives that despite the Stockfish Engine's fearsome reputation, the model was able to earn draws in four of the 50 games played. The created games demonstrate the model's capacity to stand its ground against a top-tier opponent, despite the fact that were unable to score outright victories. This success highlights model's ability in difficult decision-making situations and its capacity for chessboard strategy.

6.5 Discussion

In this section, deeper dive into the research results obtained from a variety of experiments and case studies conducted to carefully evaluate the effectiveness of the suggested AttLSTM model during the field of dynamic chess move prediction. Even while thorough study produced encouraging results, it also made us reflect critically on the experimental design, methods, and consequences in the context of more general extant studies.

```
Game 30: 0-1
Game 31: 0-1
Game 32: 0-1
Game 33: 0-1
Game 34: 0-1
Game 35: 0-1
Game 36: 0-0
Game 37: 0-1
Game 38: 0-1
Game 39: 0-1
Game 40: 0-1
Game 41: 0-1
Game 42: 0-1
Game 43: 0-1
Game 44: 0-1
Game 45: 0-1
Game 46: 0-1
Game 47: 0-1
Game 48: 0-1
Game 49: 0-0
Game 50: 0-1
Total Model Wins: 0
Total Stockfish Wins: 46
Draws: 4
```

Figure 13: StockFish VS AttLSTM Model Results

6.5.1 Evaluation of AttLSTM Model

According to subsection 6.1 examination of the AttLSTM model, there is substantial evidence that it performs better than its traditional LSTM. The AttLSTM's noticeable loss reduction and significant accuracy improvements highlight the model's potential effectiveness. Even though these findings support the original theories, it is important to be aware of the experiment's constraints. The training and testing data sample sizes might be increased, adding more variety to the game scenarios and improving the generalizability of the model Silver et al. (2017). In order to find the ideal configuration for achieving even more reliable performance, this experiment's future iterations may take into account varying the hyperparameters and architectural configurations.(Bergstra and Bengio; 2012)

6.5.2 Dynamic Chess Move Prediction

The dynamic prediction experiment (subsection 6.2) provided an engrossing look into the flexibility and possible real-world applications of the AttLSTM model. Although the model's ability to predict chess moves dynamically showed promise, further research is necessary to fully realize this potential. It would be beneficial to test the model in different scenarios to gauge how well it performs Incorporate more intricate and dynamic game states that simulate situations that closely resemble real-world games. Comparing the model's ability to negotiate complex and changing chess positions to sequential decision-making literature Silver et al. (2017), further research would help to get a more in-depth grasp of the model's strategic aptitude.

6.5.3 Confrontation with Stockfish Engine

The experiment in subsection 6.4 that pitted the AttLSTM model against the powerful Stockfish Engine demonstrated the model's tenacity by attaining draws against a top-tier foe. The chess domain's intricacy and the need for more sophistication in the model's strategic decision-making processes are highlighted by the modest results in terms of outright triumphs. A more thorough examination of the AttLSTM model's performance versus a difficult opponent should be possible by expanding the evaluation to additional games and taking other time limits into account (Simonsson; 2023). Additionally, investigating hybrid strategies that combine well-established chess algorithms with the model's predictive skills may open the door to improved gaming and strategic decision-making.

7 Conclusion and Future Work

The purpose of the paper was to investigate the possibility of examining the potential of the suggested AttLSTM model for dynamic chess move prediction. The research extensively investigated the model's capabilities through a series of tests and case studies, assessing its performance against well-known chess engines and human-like interaction scenarios. In order to capture the core of the journey and its results, the research will briefly recap study question, aims, and completed work.

7.1 Recapitulation of Research Endeavors

The effectiveness of the AttLSTM model in predicting dynamic chess moves was main research goal. The research sought to understand the intricacies of its strategic flexibility and forecasting strength. The results of the efforts revealed a comprehensive comprehension of the behavior of the AttLSTM model in diverse experimental configurations.

7.2 Success and Key Findings

The research have had a measurable amount of success in answering the study question and achieving goals. With reduced loss and greater accuracy, the AttLSTM model beat traditional LSTM models with a significant advantage because to its attention-based design. This highlights how attention processes may improve one's capacity for prediction.

The dynamic prediction tests demonstrated the flexibility of the AttLSTM model in interactive, real-time contexts. Its skill at dynamically predicting chess moves positions it for use in gaming and decision support systems.

The confrontation with the Stockfish Engine also demonstrated the model's fortitude and skill in making strategic decisions. The AttLSTM model's skill at navigating complex strategic areas was highlighted by the drew results versus this difficult opponent.

7.3 Implications and Limitations

The findings have implications that go beyond chess prediction. The AttLSTM model's demonstrated capabilities points to its possible use in a range of sequential decision-making tasks, from gaming to financial prediction and beyond. The experiments did, however, also show certain limits. More comprehensive and varied training data, precise hyperparameter tweaking, and hybrid strategies that combine the model's advantages with tried-and-true procedures might all improve the model's performance further.

7.4 Future Work and Commercialization

There are promising potential to expand and enhance the research's results in the future. This study has demonstrated the AttLSTM model's skill in predicting dynamic chess moves, however there is still need for more research.

Integrating cutting-edge tactics like the Monte Carlo algorithm is one appealing option. This method, which is renowned for simulating many possibilities through random sampling, might improve the decision-making of the AttLSTM model. Developing intricate strategies and outwit adversaries like the Stockfish Engine by fusing learnt patterns with Monte Carlo's thorough methodology. The use of reinforcement learning strategies also presents an attractive opportunity. Enable to create a system that dynamically adjusted its predictions depending on the changing game state by fusing reinforcement learning with the AttLSTM model. This can result in an AI opponent who not only anticipates plays but also picks up on and adapts to different playing styles.

Moving toward commercialization, the potential of the AttLSTM model offers up several possibilities. Users might have an immersive and engaging experience, appreciating to its incorporation with online chess platforms, which would increase engagement and learning. Its use in educational settings also shows great potential for developing a dynamic learning environment that adapts to students' achievements and difficulties.

7.5 Final Thoughts

As the end of research, it is abundantly evident that the AttLSTM model's promise is not limited to chess prediction. The work has culminated a deeper comprehension of the advantages and disadvantages of AI-assisted strategic decision-making. The inquiry has only just begun, despite the fact that the work has yielded important findings. The research may enhance the AttLSTM model's capabilities through future work by utilizing cutting-edge algorithms and learning strategies, advancing both scholarly research and real-world applications.

Acknowledgement

I want to profess my sincere thanks to Dr. Abubakr Siddig for the tremendous advice as well as the assistance he has provided, which has substantially enhanced this research. I'm also grateful for providing me with this opportunity and a welcoming learning atmosphere at the National College of Ireland.

References

Bergstra, J. and Bengio, Y. (2012).
URL: *https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf*

Britz, D., Goldie, A., Luong, M. and Le, Q. V. (2017). Massive exploration of neural machine translation architectures, *CoRR* **abs/1703.03906**.
URL: *http://arxiv.org/abs/1703.03906*

David, E., Netanyahu, N. S. and Wolf, L. (2017). Deepchess: End-to-end deep neural network for automatic learning in chess, *CoRR* **abs/1711.09667**.
URL: *http://arxiv.org/abs/1711.09667*

Dong, F., Zhang, Y. and Yang, J. (2017). Attention-based recurrent convolutional neural network for automatic essay scoring, *Conference on Computational Natural Language Learning* pp. 152–162.

Drezewskia, R. and Wator, G. (2021). Chess as sequential data in a chess match outcome prediction using deep learning with various chessboard representations., *Procedia Computer Science* .

Gobet, F. (2018). *The psychology of chess*, Routledge.

Hore, P. and Chatterjee, S. (2023). A comprehensive guide to attention mechanism in deep learning for everyone.
URL: *https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning/*

McGrath, T., Kapishnikov, A., Kramnik, V. and Tomašev, N. (2022). Acquisition of chess knowledge in alphazero — pnas.
URL: *https://www.pnas.org/doi/10.1073/pnas.2206625119*

Noever, D., Ciolino, M. and Kalin, J. (2020). The chess transformer: Mastering play using generative language models.
URL: *https://arxiv.org/abs/2008.04057*

Oshri, B. and Khandwala, N. (2015). Predicting moves in chess using convolutional neural networks, *stanford.edu* .

Schmidhuber, J., Hochreiter, S. et al. (1997). Long short-term memory, *Neural Comput* **9**(8): 1735–1780.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D. (2017). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, *arXiv preprint arXiv:1712.01815* .

Simonsson, J. (2023).
URL: *https://oru.diva-portal.org/smash/get/diva2:1779339/FULLTEXT01.pdf*

Tesauro, G. et al. (1995). Temporal difference learning and td-gammon, *Communications of the ACM* **38**(3): 58–68.

Toshniwal, S., Wiseman, S., Livescu, K. and Gimpel, K. (2022).
URL: *https://ojs.aaai.org/index.php/AAAI/article/view/21390/21139*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention is all you need.

Wölflein, G. and Arandjelović, O. (2021). Determining chess game state from an image, *Journal of Imaging* **7**(6).
URL: *https://www.mdpi.com/2313-433X/7/6/94*