Chess Cheating Detection Software

User Manual

Table of Contents:

About The Program

Motivation

The world of chess has gained a considerable amount of popularity over the last few years. Significant increase in numbers of new users was noted by the major chess playing websites like chess.com. With rising interest in the game, the issues regarding the world of chess have caught the eye of the general public. Such publicity was particularly noticeable in September 2022 when the world chess champion Magnus Carlsen forfeited the tournament after losing to Hans Niemann despite being a clear favourite. Many interpreted this behaviour of the world's best player as an implication that his opponent had cheated. That claim was later supported by the statement released by the champion. Because of considerable publicity and the high profile of the incident, a number of investigations were launched to find if there was cheating involved.

Although this was only one of many incidents of cheating accusations, the Carlsen Niemann incident publicly revealed a fundamental problem that chess has been facing since strong chess computer engines were created. Although such engines can easily be accessed for learning purposes, but their use during matches is prohibited by FIDE, national federations and major chess websites. Despite those bans there have been numerous instances of players using technology to cheat in games.

The issue of cheating in chess is particularly problematic in competitive events that include considerable money prizes, title norms and other rewards. Since the prices in top tournaments range in tens or even hundreds of thousands of dollars, cheating could be considered not only a fair-play violation, but also a serious criminal offence.

To ensure that chess remains fair and competitive, it is vital to take steps to discourage and prevent cheating and detect it if it is present.

The problem faced by the chess world is how to effectively detect cheating. While many anti-cheating measures and systems of cheat detection have been introduced, some influential chess figures question their effectiveness. Detection of chess cheating is particularly difficult, as often the suspects are not caught with direct physical evidence. Therefore, a number of sophisticated and diverse statistical methods have been developed to evaluate whether cheating occurred or not. It is crucial that such systems produce legitimate and undisputed results as they may have serious legal consequences.

The aim of the project is to provide a new non-secret chess cheating detection solution using innovative tools that could contribute to the verification process and add new value to the previously existing solutions. Consequently, the project could contribute to how chess players under suspicion are investigated and increase confidence in the results of such investigation.

How to use the program

Use this program to investigate games of players that you suspect may have used engine assistance. This program uses a machine learning algorithm trained on a large number of both high level human and engine games. The algorithm gives his prediction for engine assistance use for every viable move. By observing the amount of high predictions, the user can understand better if the assessed player used engine assistance. However the output of the program should not be considered sufficient evidence to deem the player guilty of fair play violation. It is recommended to conduct a thorough investigation including physical and other statistical measures before making such accusations.

Prerequisites

Function Requirements:

This project uses several technologies that need to be installed on your machine if you are running it locally.

Python

1. You must have python installed on your machine. Ideally 3.10.10 or newer. If you don't have it installed, you can download it for free from this website: https://www.python.org/downloads
2. There are also several python modules that you need for this program. They include:

- pandas
- numpy
- torch
- re
- subprocess
- chess

These can be installed in your command line by typing:

```
pip install 'name of the module"
```

Node.js

1. You must have Node.js installed on your machine. Ideally 16.14.2 or newer.
2. Once you have it installed, find the app.js file in the source. In its directory run the following commands

```
npm install node
```

```
npm install express
```

```
npm install cors
```

```
npm install path
```

```
npm install multer
```

```
node app.js
```

This should start the local server in your terminal. You don’t have to do anything more at this point. Just type Ctrl+c in your terminal once you want to deactivate it. Next time you want to run it, just navigate to the project directory and type the following command

```
node app.js
```

Browser and internet access

You need any modern browser to open the main_menu.html and interact with the program. You don’t need internet access for processing PGNs and FENs, but you will need it for PGN and FEN generation which is optional and runs on lichess.org.

How to run the app:

To run the app you need to have installed all of the above requirements. When that is done go to your terminal, navigate to the project directory, run

```
node app.js
```

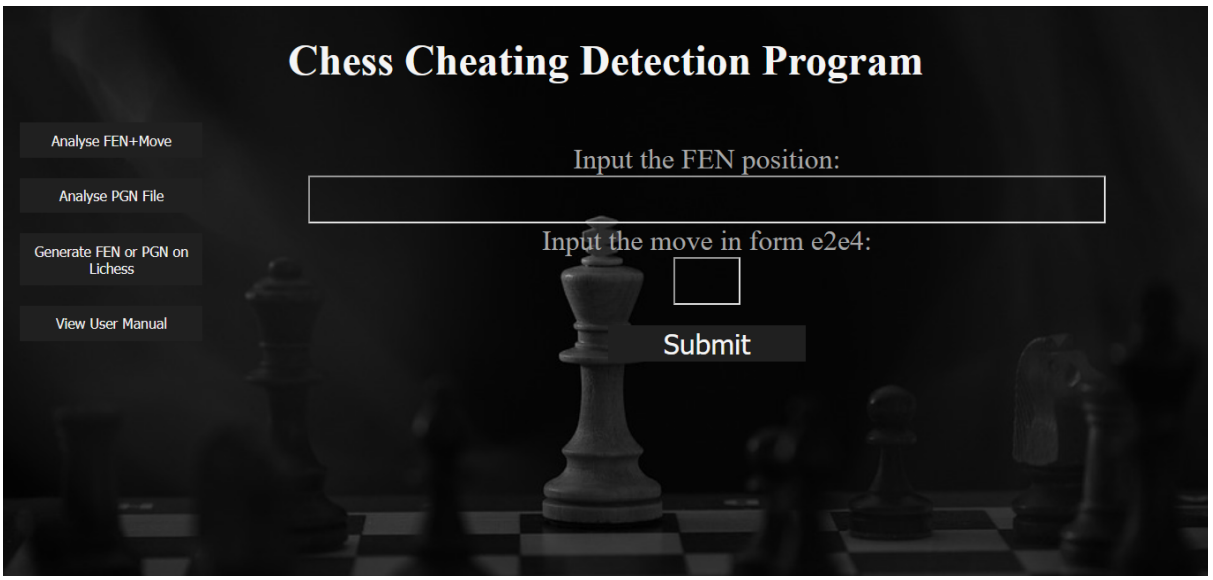
That should start the local server in your terminal. Do not close the terminal for as long as you want to use this session of the program.

Once you have done this open the file main_menu.html in your browser. The program should be ready for use.

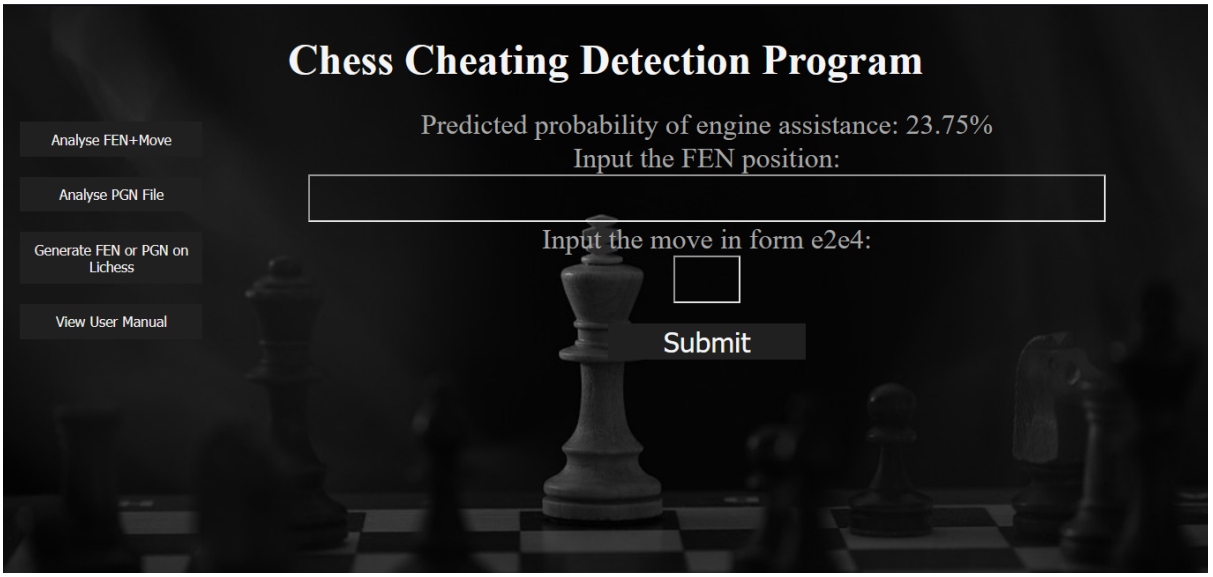
Overview of Functions

Analyse FEN+Move

This function lets the user submit a FEN position and a move made by the player for an assessment. The client will algorithm on a server to process the data and return an answer in the form of probability prediction of engine assistance for this move. Upon the selection of the option the user is prompted to input the FEN string and the move in form of a starting square and destination square as shown in a figure below:

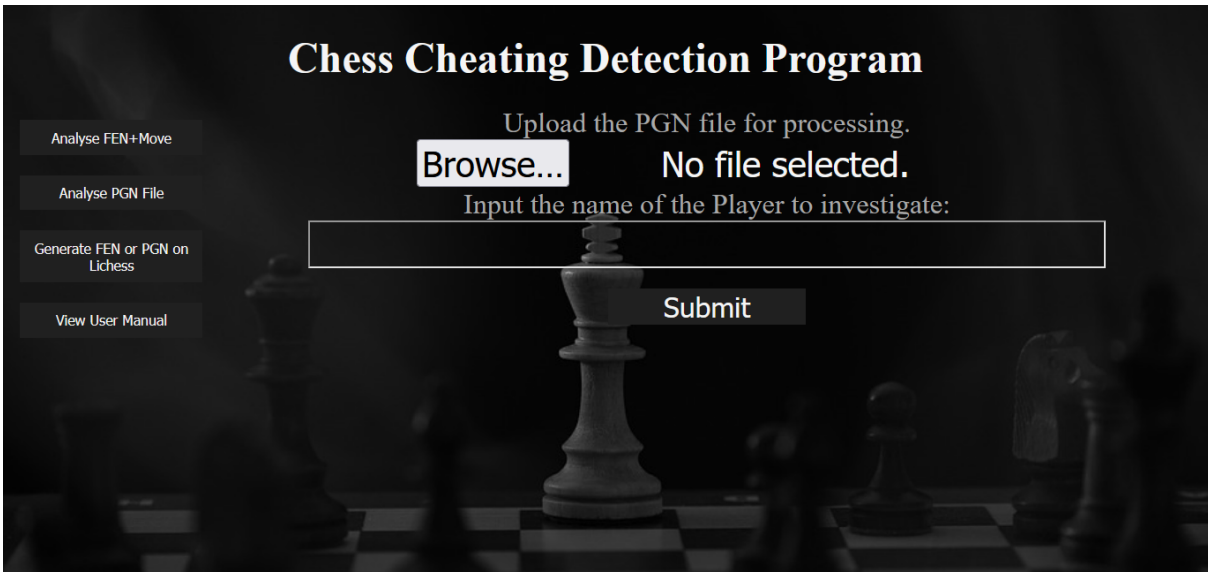


Once the user submits, the data is passed on to a local server, which runs a backend_fen.py python script for input validation and then the neural network for evaluating the likelihood of engine assistance being used for such a move. Once the script is executed, the server sends a message back to the client containing its evaluation or the error message if the data supplied by the user is in the wrong format. That message is then displayed in the user interface as shown in the picture below. It must be noted however that this function in itself is not a strong tool for recognising cheating in games as it only analyses one position instead of the entire game.



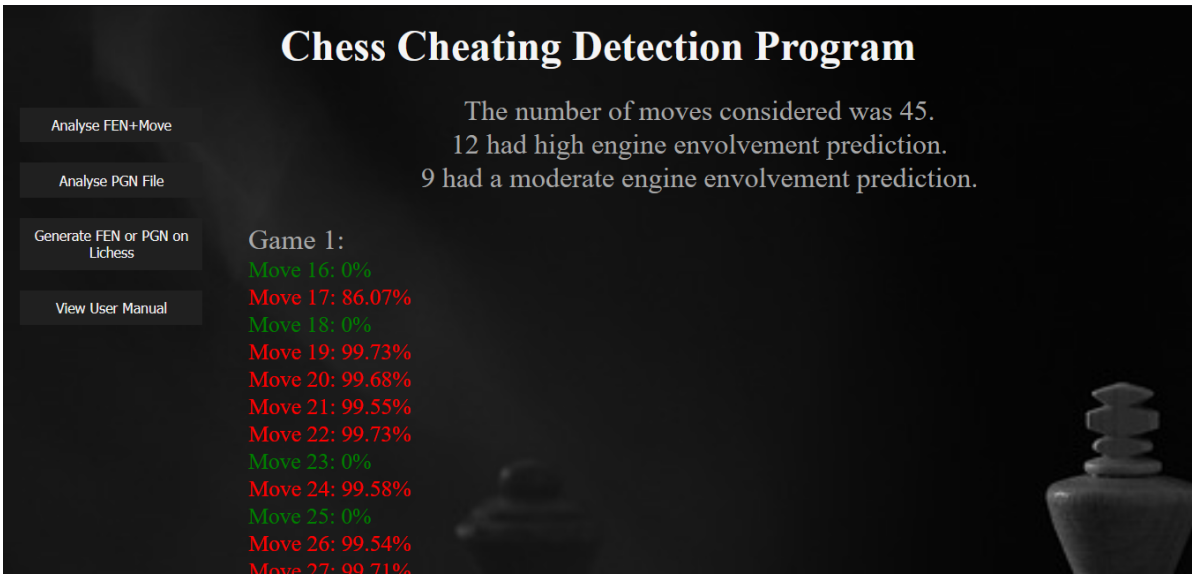
Analyse PGN file

This function is the main feature of the web application and the main goal of the project. This function lets the user submit a PGN file containing at least one chess game and a name of a player to investigate as shown in the picture below.



The file and the player's name are then submitted to a server, which executes the python script backend_pgn.py. The script first validates the input by checking if the pgn file contains any unwanted text, if the player is not found in the pgn and if there are any illegal moves in the game notation. If any of these errors occur, the server will send an appropriate error message to the client. Assuming the file was validated without errors, the python script will go to move 16 of each game and start running leela zero on the positions and then forward the output to the neural network that will return its judgement on all the player's choices.

Once that process is complete, the server sends an array containing all the judgements to the client. Once the client receives the response, it produces a report. The report consists of general information on a number of moves analysed and the number of moves highlighted as very likely to be engine generated and ones that were moderately likely to be engine generated as well as detailed colour highlighted values for each of the analysed moves made by the player in each of the games. Once completed, that report is displayed in the user interface as shown in the picture below.



Generate FEN or PGN on Lichess

The users may want to analyse games that are not currently in digital form. The application facilitates generating the PGNs and FENs by providing a direct link to a free, open-source chess website lichess.org and its analysis tool. The lichess analysis tool opens in a new window upon clicking the button. The user is then able to use a tool to input the game on a digital chessboard and generate its PGN or FEN that they can then submit for the program for analysis. A snapshot of lichess.org analysis tool with generated FEN and PGN can be seen in the picture below

View User Manual

This function will preview the pdf document containing the user manual in the application interface. It will provide the descriptions of all the features and guidelines on how to interpret the results. The result of this function is shown in the picture below: