JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AlphaZero Vs StockFish – A Review of Chess Engines

[1]Om Kshirsagar, [2]Aman Maurya, [3]Fatima Shaikh
[1]Student, Department of IT, Jai Hind College (Autonomous), Mumbai
[2]Student, Department of IT, Jai Hind College (Autonomous), Mumbai
[3]Assistant Professor, Department of IT, Jai Hind College (Autonomous), Mumbai
kshirsagar.om38018@gmail.com, maurya.aman14305@gmail.com, fatima.shaikh@jaihindcollege.edu.in

Abstract: One of the biggest domains for Artificial intelligence is the domain of games. The algorithms for chess mostly include various search techniques, chess-specific adaptation, and specially developed algorithms by the experts in chess. Recently Alpha Zero - a recent development has performed exceptionally. AlphaZero is an algorithm based on reinforced learning i.e., a self-learning, observing, and self- rewarding-based algorithm. The program has learned everything it knows by playing- evaluating-rewarding itself. The leading program in chess - stockfish which used alpha-beta search - was one of the strongest. In a 100-game match, AlphaZero defeated StockFish. In this paper, we will compare AlphaZero with Stockfish by looking at the algorithms used, games played, and the decision process of each program**.**

IndexTerms **- Artificial Intelligence, reinforcement learning, Monte-Carlo Tree Search, AlphaZero, Deep Mind, StockFish, Computer Chess, neural network, alpha- beta, chess shogi, AlphaZero stockfish.**

I. INTRODUCTION

Chess is the most popular and longest- studied game in which many algorithms have been tested. Most of the algorithms, programs developed rely on the knowledge gained from the games played by great players in history.

The developer developed many algorithms to win the games. Among them, two programs are Stockfish and AlphaZero. The authors have compared these two algorithms.

Marco Costalba, Joona Kiiski, Gary Linscott, Tord Romstad programmed Stockfish. StockFish uses the "alpha-beta algorithm" for their gameplay. Before 2017 Stockfish was the most winning game engine. But in the December 2017 tournament, the Stockfish algorithm lost the title against AlphaZero.

AlphaZero algorithm was developed by Deep Mind. AlphaZero replaced the traditionally used gameplay with his deep neural network and tabula rasa reinforcement learning algorithm. AlphaZero won 28 games against Stockfish. AlphaZero inferences some possible moves of the opponents and playsits best move which is best suited for a given state.

II. LITERATURE REVIEW

2.1 Evaluation of Strategy by AlphaZeroand StockFish on Chess Game - SreeragSR (1)

StockFish is an open-source chess engine developed by Marco Costalba, Joona Kiiski, Gary Linscott, and Tord Romstad. It was the most winning chess engine until December 2017 when AlphaZero, developed by DeepMind. In an event where 100 matches were played between the two, AlphaZero won 28 out of 100 games while drawing the rest.

This is a significant event because AlphaZero works on self-acquired knowledge unlike StockFish with relies on the knowledge provided by human experience. This shows the advancements in the field of AI and ML.

The strategy of AlphaZero as black:

1. Importance to pawn structures.
2. Exchanging important pieces only after setting up pawn structure. Thishelps in going offensive from the defensive.
3. Early and same side casting.
4. Early movement of the king when attacking implying the forward calculations.
5. Exchanging pieces and blocking theopponents' pieces from activation.

The strategy of AlphaZero as white:

1. Clearing the central of the broadhelps in attacking.
2. Placing Bishop on G2 when castlingkind side, helps in attacking anddefending.
3. Clearing D and E files for Rook to attack.

4. Breaking the pawn structure nearthe opposite king.

2.2 Mastering Chess and Shogi by Self- Play with a General Reinforcement Learning Algorithm (2)

StockFish - 2016 Top Chess Engine Championship (TCEC) world champion, isone of the strongest programs. It analyses the games based on various factors such asposition, material point values, amongothers. It then applies a function that evaluates the value of the current state 's'. To calculate the next optimal/suboptimal move, it uses a minimax search with alpha- beta pruning. This solution was a very domain-specific solution. It relied heavily on human knowledge.

To remove the dependency on human knowledge which is borrowed, self- learning programs/algorithms are now being developed. Among many, AlphaZero is one. It is a general algorithm that can playgames like Chess, Go, Shogi, etc by just knowing the rules for each game. This is byimplementing reinforcement learning techniques. The Methodology and implementation are further visited in this paper.

2.3 One Giant step for a chess-playingMachine by Steven Strogatz (3)

AlphaZero, a reinforced learning algorithm that had mastered chess, shogi, go in a few hours. The author transitions from Deep Blue to Stockfish and Komodo then to AlphaZero.

All the Engines before AlphaZero played in a robotic style, playing a move that was 'programmed' to be best as per principles which were refined over decades of human grandmasters experience. They played like a machine. Their attacks were brutal and they played materialistically, lacking insight.

He describes AlphaZero's style as a romantic, attacking style. AlphaZero finished with 28 victories and 72 draws. without losing a single game out of 100 games.

The author then talks about the application of the same principle, on which AlphaZero works, on other domains like medical, mathematics, etc. But one of the biggest shortcomings is that the knowledge of such AI cannot be taught/expressed to humans.

The author goes on by hypothesizing a future generation of AlphaZero as 'AlphaInfinity'. How it would be capable of producing stunning proofs that would demonstrate why a theory was correct.

Finally, he talks about a time when a human would not be able to keep up to the machine's speeds.

2.4 AlphaZero – What's Missing? IvanBratko (4)

AlphaZero, the successor of AlphaGo and AlphaGo Zero, learned chess, Go, shogi in less than 24 hours better than any human or computer player by self-play. It employs a deep neural network in conju nction with Monte-Carlo Tree Search. The application of MTCS in chess differs from the applications of Alpha-Beta search in other sophisticated chess engines.

The limitation to the AlphaZero is that the domain it was applied to is smaller than in real life. The rules and constraints are known. Hence it could generate a large number of training data from which AlphaZero learned. This is not very real life where the data may not be available or very costly to get.

The gameplay of AlphaZero was similar to grandmasters, thinking positionally. Even though there is a vast difference between the number of states calculated by AlphaZero and a human grandmaster, the moves played by AlphaZero are human- like. The MCTS could be the reason because of which, AlphaZero plays positionally as it only searches in selective variations but deeper. Hence it was able to play a move by which the position after 20 moves was advantageous.

All this is well but one of the biggest things that AlphaZero and in turn many ML methods lack is the ability to express the learned knowledge in a symbolic or human- readable form. AlphaZero may have a lot of chess knowledge that is not discovered by humans but cannot be known by humans except by analysis of the games.

III. METHODOLOGY:

3.1 AlphaZero:

The AlphaZero algorithm is a more generalized version of the AlphaGo Zero. AlphaZero utilizes a deep neural network (p, v) = fø(s). It accepts a board position as input and returns a vector of move probabilities p with components Pa = Pr(a | s) for each action a, as well as a scalar value v estimating the expected outcome z from position s, E[z | s] (2). AlphaZero discovers these move probabilities and their estimate values exclusively through self-play. These are then used to guide its search (2).

AlphaZero uses the Monte-Carlo tree search algorithm. In every search, it stimulates a series of games with itself-play that starts with tree root to the leaf. In each simulation, it chooses a move of 'a' for state 's' where a has a low visit count, a high move probability, and a high value (averaged overleaf states of stimulations that choose 'a' from 's') according to the present neural network. This search then returns a vector of the probability distribution of the moves. The probability depends on the visit counts at the root state selected proportionally or greedily (2). Here greedily means that the algorithm won't explore other moves if it knows a move has already been proven best. Proportionally means it will, with a small chance, select an unexplored move.

The parameter ø is initially a random value that is perfected over time during training by self-play. After each game finishes in training, the terminal state (final position as per rules) St is evaluated for the utility of the game. For each player, the utility can be

-1 for losing, 0 for the draw, 1 for winning. The ø is then updated so that the predicted value is as close as possible to the terminal utility.

In AlphaZero, only one neural network is maintained throughout, in contrast to its counterpart AlphaGo Zero, where many neural networks are employed and the best player is maintained. The single network is continually updated in AlphaZero. For each iteration, the parameters are only tweaked as per the last iteration.

3.2 Domain Knowledge Given:

1. The neural network design is tailored to the board's grid pattern and piece movement.
2. AlphaZero is equipped withcomplete knowledge of the gamerules. These are used during MCTSto simulate the positions that come from a sequence of moves, to decidegame termination, and to score any iterations that reach a terminal state.
3. The rules are also used to encode theinput planes (castling, repetition, no-progress) and output planes (how pieces move, promotions, andpiece drop in shogi).

4. To scale the exploration noise, the average number of legal moves is employed.
5. Chess and shogi games that went b eyond a certain number of steps (decided by usual game duration) wereended and awarded a drew result; Go games were terminated and sco red using Tromp-Taylor procedures, as in prior work.

3.3MCTS Search:

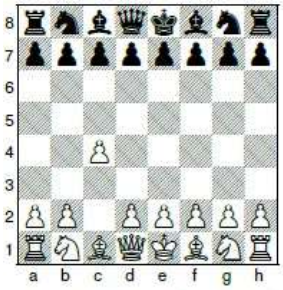
AlphaZero uses MCTS search as opposed to the traditional alpha-beta search, the supposed best for this domain. AlphaZero searches for 80 thousand chess positions per second and 40 thousand shog i positions per second, compared to StockF ish's 70 million and Elmo's 35 million. Al phaZero's deep neural network compensates by focusing more on the important selective variations despite the numbers. This is a more human-like approach as humans tend to calculate moves that seem most important.

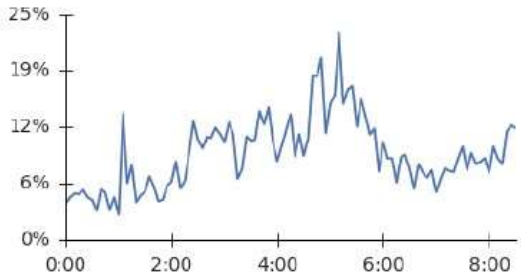
It is important to mention that many other programs tried to implement MCTS search to the traditional approach. Many have tried to apply alpha-beta search to neural networks but were slower to compete with the handcrafted evaluation functions. The fact that AlphaZero uses nonlinear function approximation which is based on deep neural networks helps for the MCTS Search which negates the approximation errors by the function.

3.4 Chess knowledge acquired by self-learning:

AlphaZero player is the topmost played openings played by human players which were discovered by itself. They were frequently played by it during the training (2). During the tournament evaluation, it was observed that AlphaZero convincingly defeated Stockfish, which implies mastery over them.

A10: English Opening

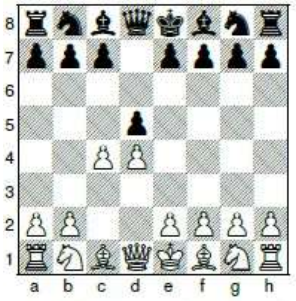


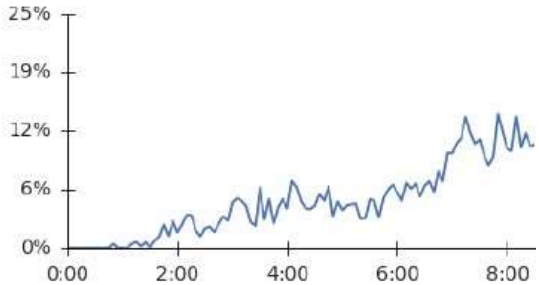


w 20/30/0, b 8/40/2

1...e5 g3 d5 cxd5 ♘f6 ♗g2 ♘xd5 ♘f3

D06: Queens Gambit

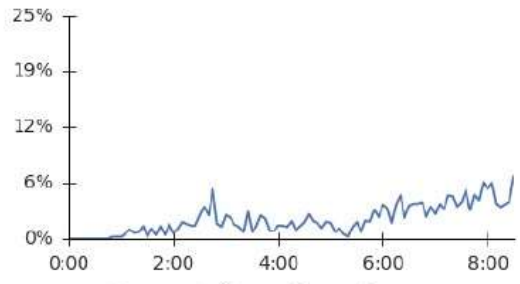




w 16/34/0, b 1/47/2

2...c6 ♘c3 ♘f6 ♘f3 a6 g3 c4 a4

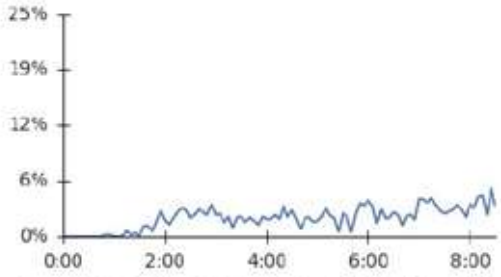
A46: Queens Pawn Game



w 24/26/0, b 3/47/0

2...d5 c4 e6 ♘c3 ♗e7 ♗f4 O-O e3

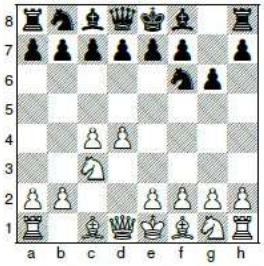
E00: Queens Pawn Game

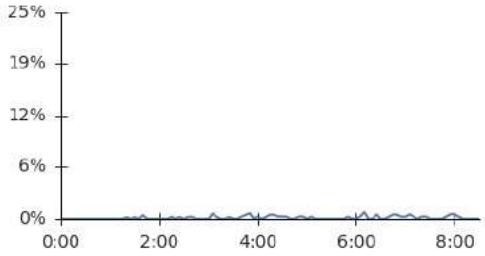


w 17/33/0, b 5/44/1

3.♘f3 d5 ♘c3 ♗b4 ♗g5 h6 ♕a4 ♘c6

E61: Kings Indian Defence

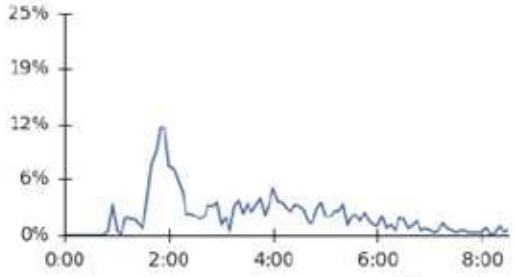




w 16/34/0, b 0/48/2

3...d5 cxd5 ♘xd5 e4 ♘xc3 bxc3 ♗g7 ♗e3

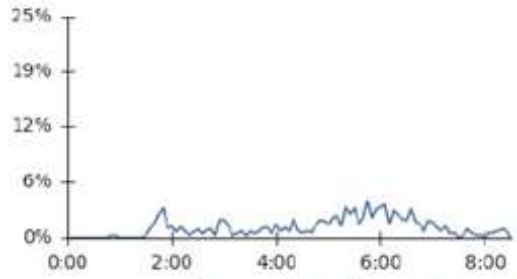
C00: French Defence



w 39/11/0, b 4/46/0

3.♘c3 ♘f6 e5 ♘d7 f4 c5 ♘f3 ♗e7

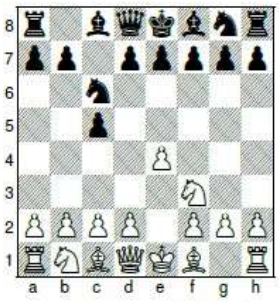
B50: Sicilian Defence

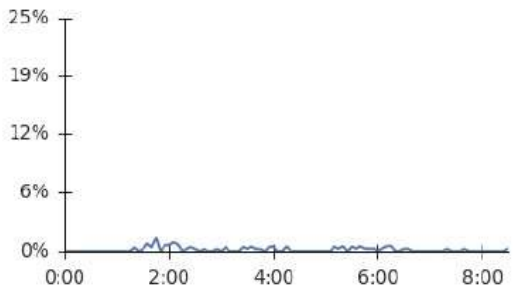


w 17/32/1, b 4/43/3

3.d4 cxd4 ♘xd4 ♘f6 ♘c3 a6 f3 e5

B30: Sicilian Defence





w 11/39/0, b 3/46/1

3.♗b5 e6 O-O ♘e7 ♖e1 a6 ♗f1 d5

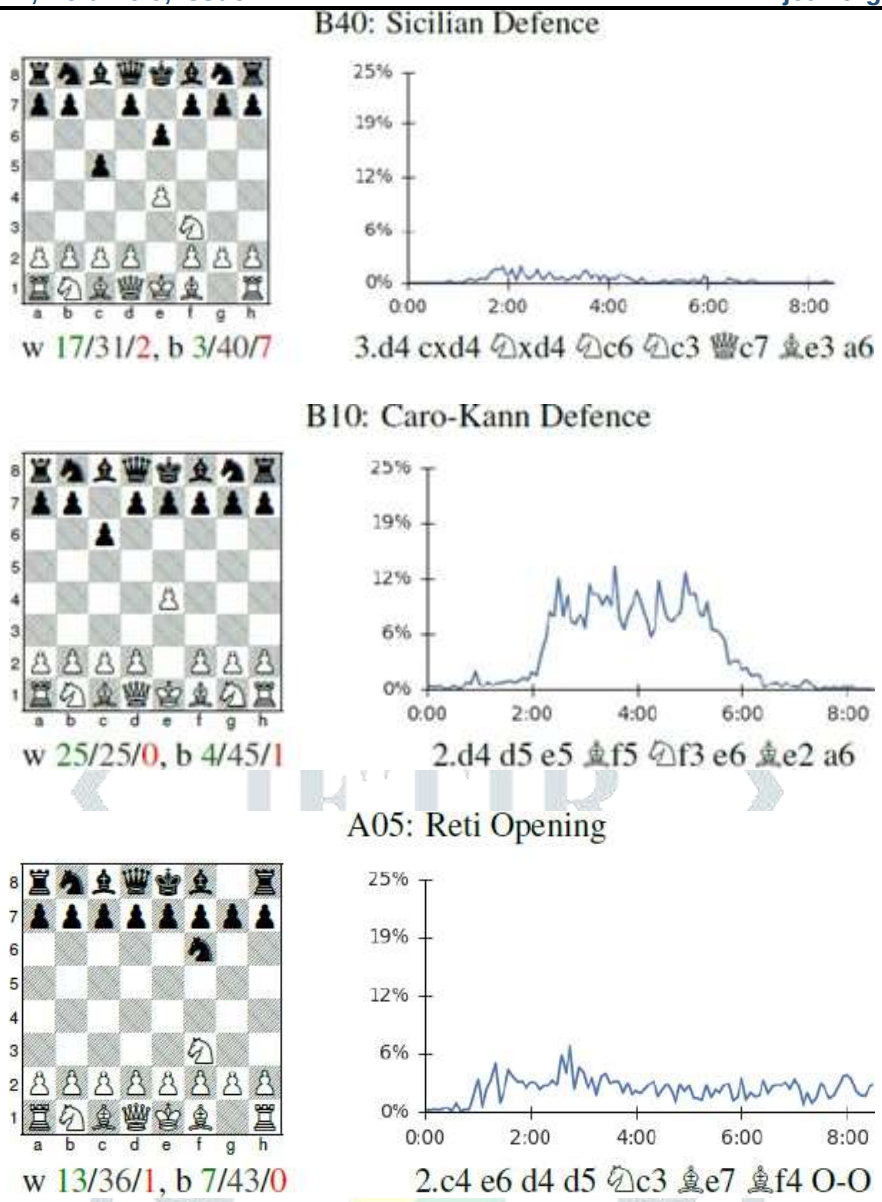


Figure 1: Analysis of most popular human openings.

In the above figure, each image shows one of the most popular open. The image contains the ECO code and its name. The graph plots the proportion of games during self-training where the opening was used, with reference to training time. The win/draw/loss results of 100 game AlphaZero vs. Stockfish matches as either white (w) or black (b) of AlphaZero is reported below of each. The main variation played by AlphaZero is mentioned.

3.5 StockFish:

StockFish on the other hand containsspecially handcrafted features that are used to categorize a position. Features are common chess concepts like the point value of each piece, etc. Each feature is then given a weight, which is manual as per the criticality of the feature along with automatic tuning. A linear combination function is then applied to the position to evaluate it. This assessment is utilized in positions that are 'silent,' that is, there are no pending captures or checks. In specific situations, some domain-specific “quiescence search” is used before the evaluation function (2).

Minimax search is then responsible to get the actual final evaluation of the position by evaluating each leaf node. To limit the actual positions to be searched alpha-beta pruning is applied to exclude a branch. Cuts are made in various situations. Following are a few of the situations:

1. If the branch is deviating a lot fromthe current variation.
2. Using the concept of aspiration windows (5), Null move (6), futilitypruning (7).

Alpha-Beta is observed that the efficiency relies heavily on the order of the moves considered. To tackle this, a shallow search is completed before a deep search iteratively.

The transposition table (8) stores the values of positions that are encountered through different branches. Opening Book (9) assists to play moves at the beginning of thegame. Endgame Tablesbase (10) helps to quickly decide moves in the endgame where fewer pieces are remaining on board.

3.6 Comparison between AlphaZero andthe Top engines of respective games:

In a stimulated comparison, the AlphaZeroalgorithm was trained in chess, shogi, Go. The settings, network architecture, and hyper-parameters used were the same for all games except if any changes were specifically required by a game. Separate instances of AlphaZero for each game. AlphaZero was trained for 3 days (2).

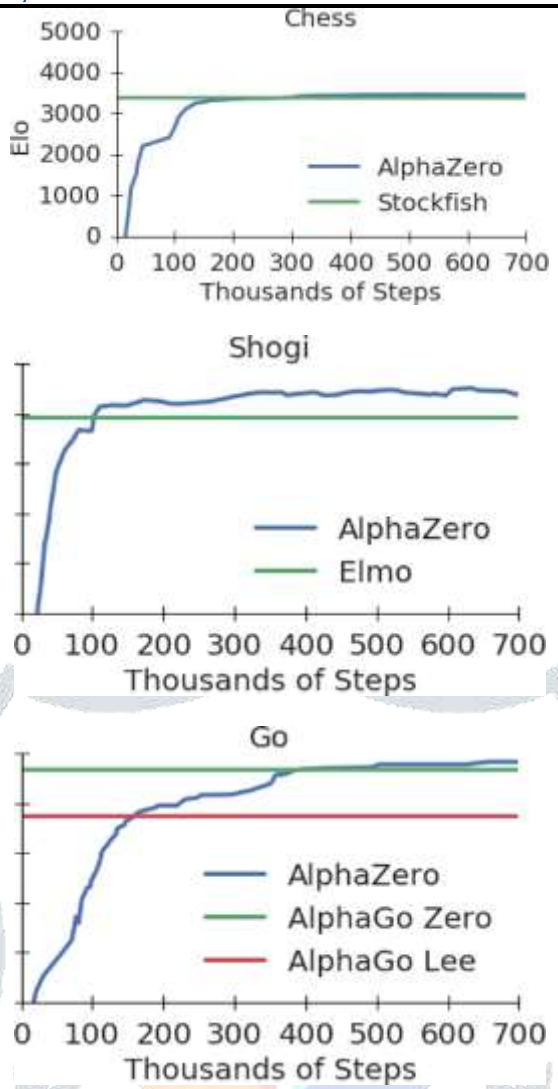


Figure 2: Training curve.

Figure 2 shows the ELO for 700,000 steps.Elo ratings were calculated from evaluationgames between different algorithms when given a second for each move.

a. AlphaZero compared with StockFish inchess.
b. AlphaZero compared Elmo in Shogi.
c. AlphaZero compared to AlphaGo Leeand AlphaGo Zer. (2)

3.7 Hardware Used:

AlphaZero: single machine with 4 TPUs (2).
StockFish: 64 threads and a hash size of 1GB with the highest level of skills

Game Settings:

- 100 games at tournament timecontrols (2)
- One minute per move (2).The results were:

Gam e	Whit e	Blac k	Wi n	Dra w	Los s
Ches s	Alph aZer o	Stoc kfish	25	25	0
	Stoc kfish	Alph aZer o	3	47	0
shog i	Alph aZer o	Elmo	43	2	5
	Elmo	Alph aZer o	47	0	3
Go	Alph aZer o	AG0 3-day	31		19
	AG0 3-day	Alph aZer o	29		21

Tournament evaluation of AlphaZero in chess, shogi, andGo from AlphaZero's perspective

IV. CONCLUSION:

In this paper, we looked at AlphaZero andStockFish, two chess-playing programs that use different approaches to play the game. AlphaZero has mastered the game by self-play and surpassed all the humansand computers. This shows the power of reinforcement learning, which given the basic knowledge can then train itself to achieve a task. Although it seems great there are some drawbacks to it. The performance of AlphaZero was because the rules and states in the games are morepredictable. It also lacks knowledge representation i.e., cannot explain its knowledge to humans. This implementation can still be powerful and be used in various scenarios where the goal is definitive and data is available fortraining. Examples can include medical implementation where a disease can be detected by examining a CT scan.

V. REFERENCES:

[1] Sreerag SR "Evaluation of Strategy by AlphaZero and StockFish on Chess Game" Krishi Sanskriti Publications Volume 5, Issue 5; July-September, 2018, pp. 443-444 [1]

[2] David Silver and Thomas Hubert and Julian Schrittwieser and Ioannis Antonoglou and Matthew Lai and Arthur Guez and Marc Lanctot and Laurent Sifre and Dharshan Kumaran and Thore Graepel and Timothy Lillicrap and Karen Simonyan and Demis Hassabis "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm" arXiv:1712.01815v1. [2]

[3] Strogatz, S. (2018, December 26). One Giant Step for a chess-playing Machine. The New York Times. https://nyti.ms/2Rjetd3

[4] Bratko, Ivan (2018). AlphaZero – What's Missing? Informatica 42 (2018) 7–11. https://www.informatica.si/index.php/informatica/article/download/22 26/1153 [4]

[5] https://www.chessprogramming.or g/Aspiration_Windows [5]

[6] https://en.wikipedia.org/wiki/Null_ move [6]

[7] https://www.chessprogramming.or g/Futility_Pruning [7]

[8] https://www.chessprogramming.or g/Transposition_Table [8]

[9] https://www.chessprogramming.or g/Opening_Book [9]

[10] https://www.chessprogramming.or g/Endgame_Tablebases [10]