

AI-Powered Chess Analysis for Skill Development: A Case Study Aligned with Sustainable Development Goal 4

Dinesh Rajan M¹, Dr. P. Durgadevi²

¹M.Tech Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India

²Assistant Professor (Senior Grade), Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India

Abstract

Chess rewards calculation, pattern recognition, and judgment under limited time. For many learners the usual route to improvement is a book, a coach, or a slow manual review of one game. A modern chess engine can inspect a recorded game and return a numerical evaluation of every move, yet that number is not, by itself, a lesson. The learner still has to decide whether the loss of evaluation came from the opening, from a missed tactic, from a calculation, or from the endgame.

This paper presents a case-study design for an AI-powered chess analysis system whose output is a skill profile rather than a raw engine trace. A game in Portable Game Notation is reconstructed move by move. A chess engine supplies a best move and an evaluation. Moves that change the evaluation beyond stated design bands are marked as critical and are placed in one of six skill areas. Repeated labels become a player profile, and the profile selects a kind of practice. The design is aligned primarily with Sustainable Development Goal 4, because the feedback can be read without a coach being present for every game.

The study specifies the pipeline, the design rules, and the tests a later implementation would have to pass. It does not report a completed program, a measured accuracy, or a learning gain from a user study. The charts in the paper are illustrations of how a profile could be shown. They are not experimental results.

Keywords: Chess Analysis, Stockfish, Skill Development, Personalized Learning, Sustainable Development Goal 4, Case Study.

1. Introduction

A recorded chess game is a complete trace of decisions. Each move can be checked for legality, placed on a board, and compared with a strong alternative. Engines built for this comparison are now widely available, and players already use them after games. The educational difficulty is interpretation. An evaluation graph shows where the position became worse. It does not name the skill that should be practiced next.

The question addressed by the case study is how engine evaluation can be turned into a small set of skill labels and then into a training suggestion. The proposed answer keeps the engine inside one module. A

separate classifier assigns a single skill area to each critical move. A profile stores those labels across games, and a recommendation table maps the weakest area to a type of exercise. The separation matters: the engine can be replaced without rewriting the learning rules, and a later statistical classifier can replace the rules without changing the report.

Six areas are used throughout: Opening, Calculation, Tactics, Positional Play, Prophylaxis, and Endgame. The areas are ordinary coaching language. They are not a claim that chess skill has been shown, by a factor analysis, to consist of exactly these six parts. They are a stable vocabulary so that a report written in March can be compared with a report written in June.

The remainder of the paper is organized as follows. Section 2 reviews the line of chess AI that leads from playing strength to human learning. Section 3 specifies the framework. Section 4 explains the illustrative figures and states what they are not. Section 5 lists the validation that would be required before any accuracy is quoted. Section 6 relates the design to Sustainable Development Goals 4, 9, and 10. Section 7 concludes.

2. Related Work

Computer chess first demonstrated that search and a hand-built evaluation could compete with the strongest human players. A later step was self-play reinforcement learning. Silver and colleagues showed that a single algorithm, trained by playing against itself, could reach a very high standard in chess, shogi, and Go [1]. That result is about playing strength. It does not describe how a club player should study the games that the program analyzes.

A different line of work models the human move rather than the strongest move. Maia was trained to predict moves made by players at particular skill levels, using chess as a model system for aligning a strong program with human behavior [2]. Prediction of a human move is useful evidence that skill level leaves a trace in the choice of move. It is not yet a coaching plan. A learner who is told which move a similarly rated player would choose still needs to know which weakness to practice.

Evidence on training with chess computers suggests that access to a strong program can improve performance, while also warning that the program does not replace a coach [3]. Commentary on human–AI chess analysis makes a related point: engines are already part of preparation and of post-game review, and the open question is how that review should be presented to a person [4]. Studies of errors under time pressure and positional ambiguity, and studies that predict human moves with skill-group models, show that mistakes are patterned rather than random [5], [6]. A pattern of mistakes can be counted. Counting is not the same as assigning the next exercise.

The case study sits in the gap between these lines. The engine remains the source of the evaluation, as in ordinary post-game analysis. The added layer is a skill label, a memory of labels for one player, and a training suggestion tied to that memory. Machine learning is left as a later replacement for the rule-based label, to be used only after a set of positions has been labeled by hand. Until then, the reason for a label can be read from the rule.

3. Proposed Framework

Figure 1 shows the learning cycle that the framework is built to support. A recorded game is analyzed by the engine, critical decisions are placed in a skill area, and the player receives a practice task before the next game. The same cycle is the link to quality education: feedback does not wait for a scheduled lesson.

Figure 1: AI-Assisted Chess Learning Cycle and Alignment with the Sustainable Development Goals

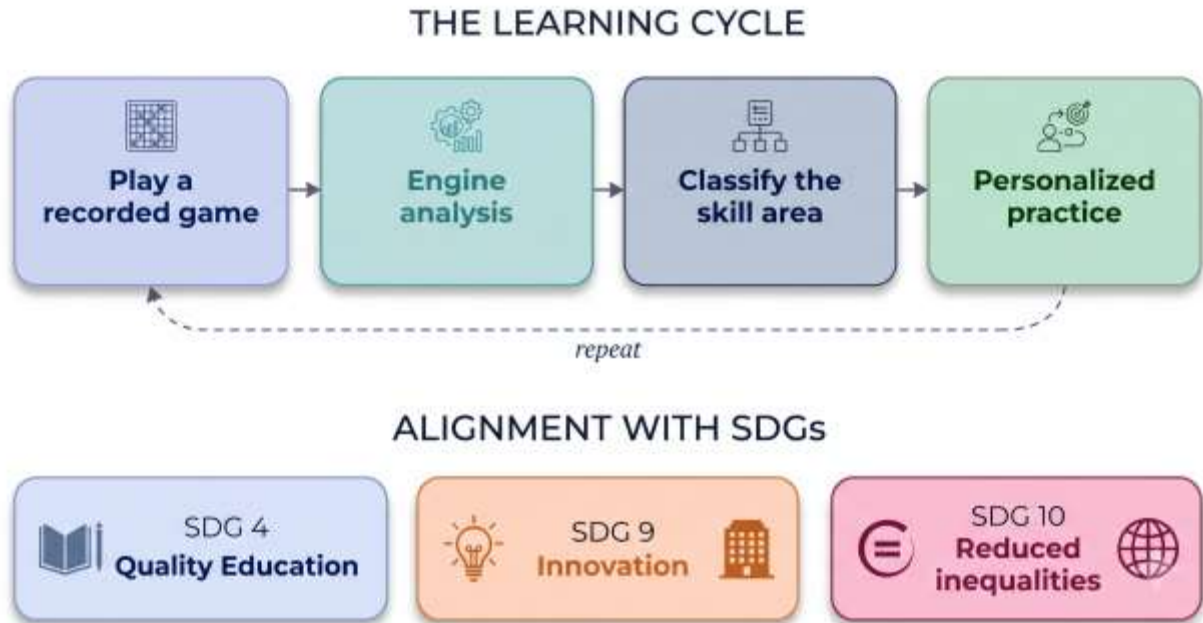
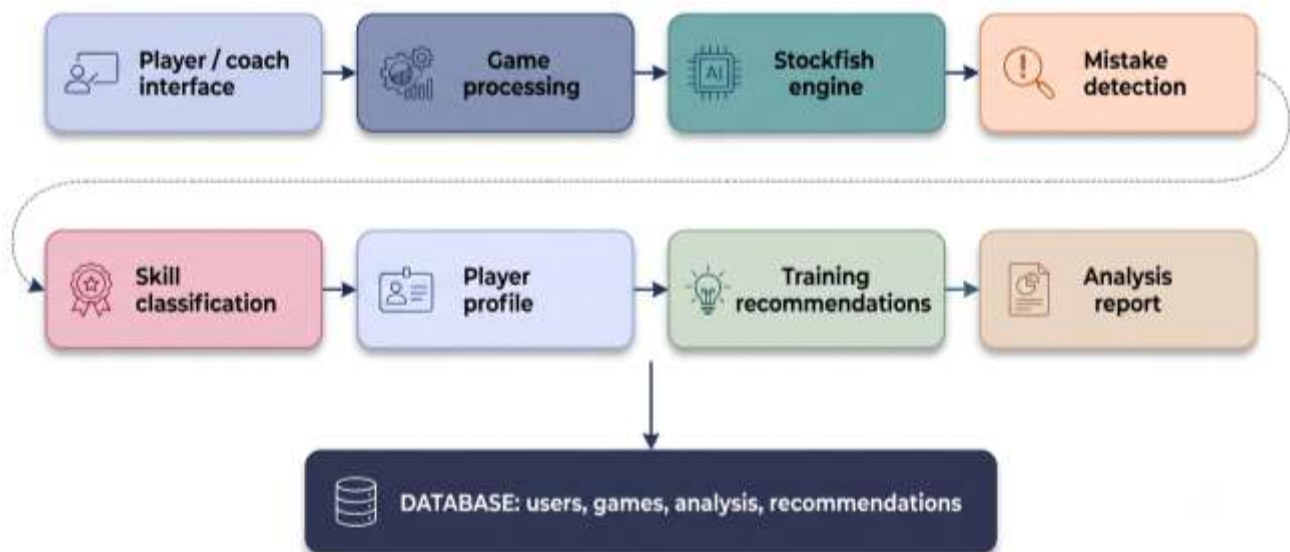


Figure 2 places the software between the player and that cycle. The interface accepts a game and later shows a report. It does not speak to the engine. Game processing checks the file and rebuilds the board. The engine module returns a best move and a score. Mistake detection, skill classification, the player profile, and the training recommendation are separate boxes. A database stores users, games, analyses, and recommendations so that a later game can be compared with an earlier one.

Figure 2: Architecture of the Proposed Chess Analysis System



3.1 Game Intake and Position Reconstruction

The input is a standard chess game in Portable Game Notation. Headers may carry the names, the ratings, the event, the date, and the result. The move text is checked one ply at a time. A ply is a single move by one player. An illegal move stops the analysis and is reported. Variants other than standard chess are outside the first design, even when the file resembles Portable Game Notation.

Reconstruction starts from the initial position and applies the moves in order. For each ply the framework keeps the position before the move and the position after it. Castling rights, en passant, and promotion are delegated to a chess library rather than reimplemented [7]. The before-position is what the engine is asked to search. The after-position is searched again once the played move has been made.

3.2 Engine Analysis and Evaluation Change

The intended engine is Stockfish, addressed through the Universal Chess Interface [8]. The application sends the position in Forsyth–Edwards Notation and a search limit, then reads the best move and the centipawn score. One hundred centipawns correspond to about one pawn on the engine scale. The depth is a parameter. The same position searched at two depths may receive two scores, so any later test must record the depth beside the score. The Laws of Chess remain the authority for what a legal move is [9]. The engine is not asked to explain why a person chose a move.

Let E_1 be the evaluation before the player's move and E_2 the evaluation after it. Both are written from the side that has just moved. The change is

$$\Delta E = E_2 - E_1 \quad (1)$$

A negative value means that the position became worse for the player who has just moved. Mate scores are converted to a large centipawn magnitude before the subtraction, so that a missed mate is not treated as a smaller event than the loss of a pawn. Only a drop that crosses a design band is marked as critical. Public chess sites use related bands, and those bands are not identical and can change. Table 1 is therefore a design rule of this case study. It is not a statement of any website's current cut-off, and it is not a threshold fitted to a dataset.

Table 1: Design Bands for the Change in Evaluation

Label	Drop for the player to move	Use in the report
Inaccuracy	50 to 99 centipawns	Shown in the game review, low training priority
Mistake	100 to 199 centipawns	Counted in the skill profile
Blunder	200 centipawns or more	Highest priority inside that game

3.3 Skill Classification and the Player Profile

Each critical move receives one primary label. When several labels are possible, a fixed priority prevents the same move from being counted twice: Tactics, then Calculation, then Endgame, then Opening, then Prophylaxis, then Positional Play. The priority is a bookkeeping rule. It does not say that a tactical miss is always more important, for the player's development, than an endgame error.

The opening window is the first twelve full moves, unless the queens have already been exchanged. An endgame, for this design, is a position in which the queens are off and each side has at most one rook, or in which the remaining material, excluding the kings, is at most thirteen points on the usual scale where a pawn is one point. A tactical label is supported by a concrete motif such as a fork, a pin, a skewer, a

discovered attack, or a hanging piece. A prophylactic label is used when the engine's best move met a direct threat and the played move did not. These boundaries are design choices that make the label stable. They are not claims about the true end of every opening.

Table 2: Skill Areas and a Typical Problem

Skill area	Typical problem recognized by the rule
Opening	Early loss of king safety or neglect of development
Calculation	A concrete variation is miscounted and no simple motif explains it
Tactics	A fork, pin, or other immediate combination is missed
Positional play	Piece placement is poor without an immediate tactical cost
Prophylaxis	A clear plan or one-move threat by the opponent is not met
Endgame	Technique fails in a simplified position

Labels from several games of the same user are stored as a profile. An area that often contains critical moves is treated as a weakness. An area that rarely does so is treated as a relative strength. The profile is not a rating and it is not an evaluation of the player by the engine. If an area has too little evidence, the report should say so rather than invent a score.

3.4 Recommendations

The recommendation names the skill area, a kind of exercise, and a priority. Calculation is paired with candidate-move practice. Tactics are paired with motif puzzles. Positional play is paired with model games. Prophylaxis is paired with exercises that ask for the opponent's threat to be named. The endgame is paired with technical positions. The opening is paired with a repertoire and with model games. The mapping is pedagogical. It does not promise a change in rating.

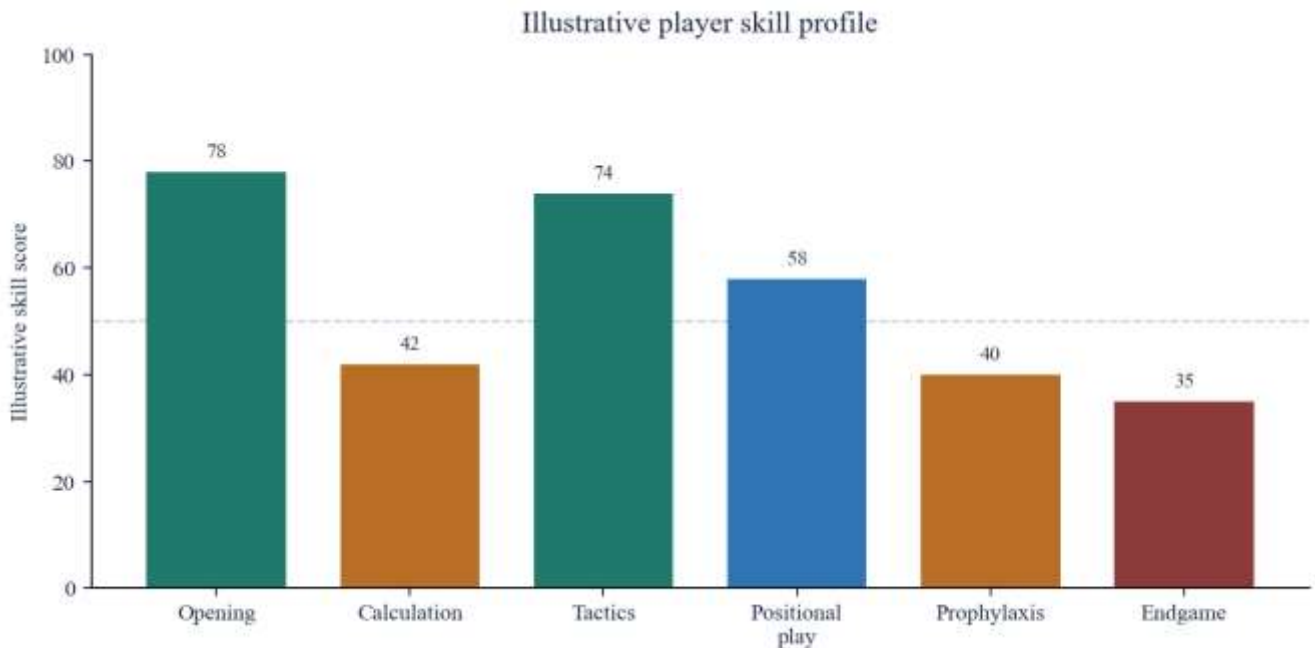
The report is ordered so that the skill summary comes before the engine lines. A reader can see what to practice without first reading every variation. Each suggestion must point back to at least one stored critical move. A suggestion that cannot be traced to a move is not emitted.

4. Illustrative Observation Design

No game collection was scored for this paper, and no engine log is presented as data. Section 4 therefore shows how the profile would be drawn, using one constructed example that matches a prose description: Opening good, Tactics good, Positional play moderate, Calculation in need of improvement, Prophylaxis in need of improvement, and Endgame weak.

Figure 3 draws that example as six bars: Opening 78, Calculation 42, Tactics 74, Positional play 58, Prophylaxis 40, and Endgame 35. In the drawing, a bar at or above 70 is read as good, a bar from 50 to 69 as moderate, and a bar below 50 as in need of improvement. The dashed line at 50 is only a visual guide. The numbers were chosen so that the drawing and the prose say the same thing. They are not averages from any player's games.

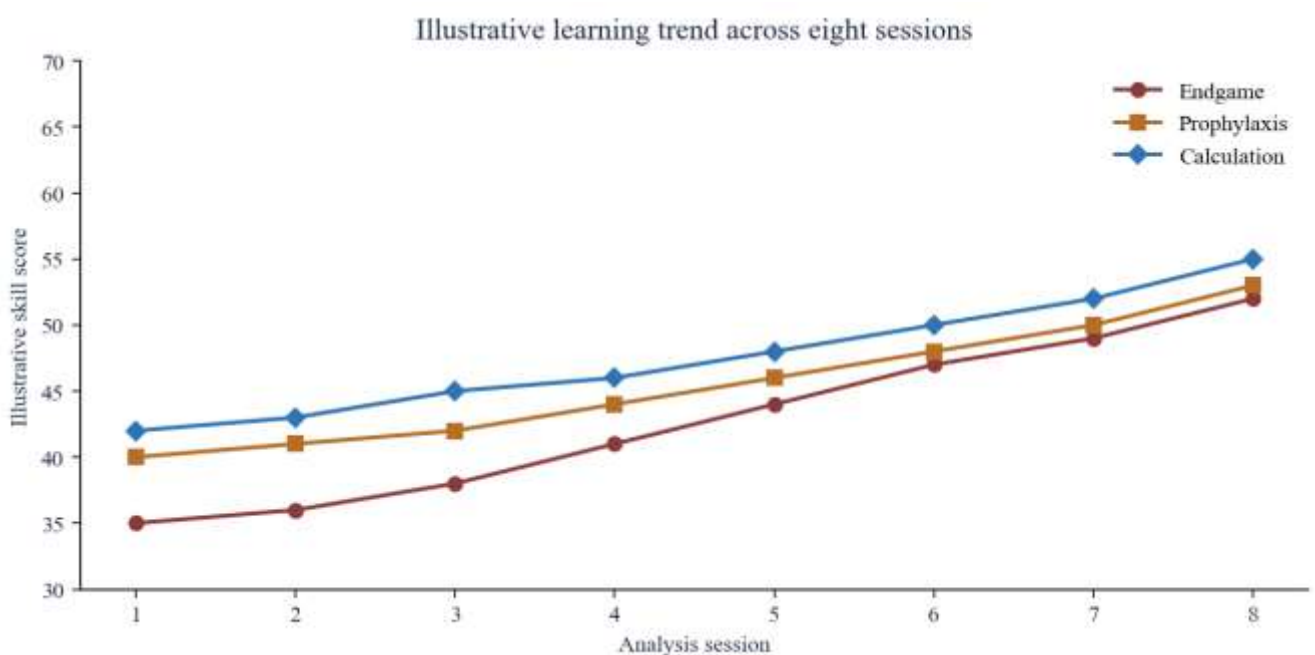
Figure 3: Illustrative Player Skill Profile



Illustrative example only. Not a measured experimental result.

Figure 4 draws the three weaker areas across eight imaginary sessions. Endgame moves from 35 to 52, Prophylaxis from 40 to 53, and Calculation from 42 to 55. The lines rise because the figure is a picture of a progress screen. They do not record eight meetings with a learner. A third display, not repeated here as a measured chart, can show the share of critical moves in the same example: Calculation 27, Endgame 22, Tactics 18, Prophylaxis 16, Positional play 9, and Opening 8. Those shares add to 100 for the sake of the drawing. They were not counted in a database.

Figure 4: Illustrative Learning Trend Across Eight Sessions



Illustrative example only. Not a measured experimental result.

Read together, the figures say that opening and tactics would be treated as relative strengths in this example, and that calculation, prophylaxis, and the endgame would be the training priorities. That reading belongs to the chosen numbers. It is not a finding about chess players in general. When an implementation exists, measured profiles should replace these figures rather than be placed beside them.

5. Planned Validation

Validation is specified so that a later implementation can be tested without treating Section 4 as an oracle. Four checks are distinct. The file check accepts a legal game and rejects an illegal one before the engine is called. The reconstruction check compares the board after each move with the rules of chess. The engine check compares the best move and the score with the same engine at the same depth. The label check compares the skill area with a label agreed in advance for a small set of positions. A pass on the engine check does not imply a pass on the label check.

Boundary values of the bands in Table 1, namely 50, 100, and 200 centipawns, belong in the unit tests of the detector. The recommendation test uses a profile in which one area is weak and the others are strong, and it expects the exercise that the mapping assigns to the weak area. Until a labeled set exists, the classifier may be reviewed by walking through the priority rule, but it cannot be given an accuracy, a precision, or a recall. No such figure is claimed in this paper.

Two limits will remain even after those tests. The engine score depends on depth, on the engine version, and on the hardware. The player's reason for a move is not present in the game file. A label describes the position under the stated rules. It does not describe a thought that was not recorded.

6. Alignment with the Sustainable Development Goals

The primary alignment is with Sustainable Development Goal 4, quality education [10]. The framework turns a learner's own game into feedback that names a weakness and a form of practice. The loop can be repeated when a teacher is available and when a teacher is not. The claim is about access to feedback. It is not a measured gain in rating or in a school grade.

Sustainable Development Goal 9, industry, innovation, and infrastructure, is relevant because the pipeline is a documented use of an open chess engine and of ordinary software interfaces [11]. Sustainable Development Goal 10, reduced inequalities, is relevant because the path from a game file to a report does not require a paid coach for every game, and because the intended components are open source [12]. Table 3 states these links as properties of the design.

Table 3: Contribution to the Sustainable Development Goals

Goal	What the design provides
Goal 4, Quality education	Skill-labeled feedback and a practice task from the learner's own game
Goal 9, Innovation	A documented analysis pipeline with a replaceable engine module
Goal 10, Reduced inequalities	A route to digital review that does not require continuous expert access

7. Conclusion

An engine evaluation is a strong signal and a weak lesson. The case study specifies a way to keep the signal and to add the lesson. A recorded game is reconstructed, searched, and reduced to critical moves

by the change in Equation (1). Each critical move receives one skill label under a published priority. Labels accumulate in a profile, and the profile selects a kind of practice. The engine, the classifier, and the report are separate, so that each can be tested and later replaced.

What has not been done is equally part of the result. There is no completed deployment, no counted corpus, and no user study. The bars and the trend lines are illustrations of a screen, drawn from a single constructed profile. Future work is a web implementation, import from online platforms, a labeled set for a learned classifier, and clock-aware analysis when the game file contains times. Any number produced by that work should replace the illustrations in Section 4. Until then, the contribution of the paper is the framework: a critical-move test, six skill areas, a profile, and a training suggestion, aligned with quality education.

References

1. Silver, D., Hubert, T., Schrittwieser, J., et al., A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, *Science*, 2018.
2. McIlroy-Young, R., Sen, S., Kleinberg, J., Anderson, A., Aligning superhuman AI with human behavior: Chess as a model system, *Proceedings of the ACM Conference*, related Maia research.
3. Gaessler, F., Piezunka, H., Training with AI: Evidence from chess computers, *Strategic Management Journal*, 2023.
4. Rubin, Y., A new paradigm: Chess AI in game analysis, *Google Research*.
5. Cüvitoğlu, A., Interpretable machine learning analysis of nonlinear error amplification under time pressure and positional ambiguity in elite blitz chess, *Scientific Reports*, 2026.
6. Zhong, D., Huang, D., Greenberg, C., Predicting human chess moves: An AI assisted analysis of chess games using skill-group specific n-gram language models, *arXiv*, 2025.
7. Python-Chess, Documentation and project resources.
8. Stockfish Chess Engine, Documentation and official project resources.
9. FIDE, Laws of Chess and official chess regulations.
10. United Nations, Sustainable Development Goals, Goal 4: Quality education.
11. United Nations, Sustainable Development Goals, Goal 9: Industry, innovation and infrastructure.
12. United Nations, Sustainable Development Goals, Goal 10: Reduced inequalities.