

## RAPPORT DE STAGE

BERTHAUT Alexandre 2A GP

Lieu du stage : Cercle d'Echecs de Monte-Carlo (C.E.M.C) - 5 rue des oliviers 98000 MONACO

Date du stage : Du 19/06/23 au 17/08/23

Tuteur de stage : Rapaire Jean-Michel – [cemc@libello.com](mailto:cemc@libello.com)



### Remerciements :

Je tiens à exprimer mes plus vifs remerciements envers mon tuteur de stage, Jean-Michel Rapaire, pour m'avoir offert l'opportunité exceptionnelle de réaliser un stage enrichissant de deux mois au sein du Cercle d'Échecs de Monte-Carlo. Je tiens également à souligner son soutien constant, et sa bienveillance tout au long de cette expérience, d'autant plus que ce stage m'a permis de travailler dans un domaine qui me passionne.

### Résumé du rapport :

Pendant mon stage, j'ai été chargé d'accomplir diverses activités en lien avec les objectifs de ma formation. J'ai participé activement à la gestion de la base de données des parties du club en utilisant Chess Base. Mon rôle comprenait également l'analyse approfondie des parties jouées par les membres du club en utilisant le deep learning, ce qui m'a permis de découvrir des schémas de jeu, et de comprendre les limites entre l'humain et l'ordinateur.

J'ai travaillé sur l'analyse des performances des joueurs en fonction de leur historique de parties, en examinant des facteurs tels que le temps pris pour effectuer un coup et en recherchant des corrélations avec des modèles de jeu établis. J'ai également eu l'opportunité de détecter les faiblesses et les points forts des joueurs dans différentes phases du jeu (ouverture, milieu de partie, finale) en utilisant des calculs probabilistes basés sur les données de la base.

Tout au long de mon stage, j'ai développé des compétences essentielles telles que le traitement de données, le deep learning, l'optimisation et l'innovation. Mon expérience m'a permis d'acquérir une compréhension approfondie de l'analyse des parties d'échecs, de la détection de schémas, de l'optimisation des performances et de l'application de méthodes innovantes pour améliorer les compétences des joueurs.

During my internship, I was tasked with various activities aligned with the objectives of my training. I actively participated in managing the club's game database using Chess Base. My role also encompassed conducting in-depth analysis of games played by club members using deep learning, which allowed me to uncover gameplay patterns and understand the boundaries between human and computer play.

I worked on analyzing players' performances based on their game history, examining factors such as time taken for moves and seeking correlations with established gameplay patterns. I also had the opportunity to identify players' weaknesses and strengths across different game phases (opening, middlegame, endgame) by utilizing probabilistic calculations based on database information.

Throughout my internship, I honed essential skills including data processing, deep learning, optimization, and innovation. My experience provided me with a profound understanding of chess game analysis, pattern detection, performance optimization, and the application of innovative methods to enhance players' skills.

#### Glossaire – abréviations :

**1-C.E.M.C** : Cercle d'Echecs de Monte-Carlo

**2-NNUE** : "NNUE" est l'acronyme de "Efficiently Updatable Neural Networks", ce qui signifie en français "Réseaux de Neurones Mis à Jour de Manière Efficace". Ce terme désigne une méthode spécifique d'utilisation de réseaux de neurones artificiels pour l'évaluation de positions aux échecs.

**3-Chess Base** : Base de données répertoriant les parties d'échecs.

**4-Stockfish** : Stockfish est l'un des moteurs d'échecs les plus puissants et populaires au monde.

**5-Cœur** : Un cœur est une unité de traitement indépendante au sein d'un processeur (CPU) qui peut exécuter des tâches en parallèle. Le CPU peut être composé de plusieurs cœurs, et les logiciels d'échecs utilisent souvent la puissance de traitement multithread (utilisation de plusieurs cœurs) des processeurs pour accélérer la recherche et l'évaluation des positions dans l'arbre de jeu.

**6-kN/s** : kilo Nodes per second - C'est une unité de mesure de la vitesse de calcul utilisée pour évaluer les performances des moteurs d'échecs et d'autres programmes d'intelligence artificielle qui effectuent des recherches dans des arbres de jeu.

**7-Nœud** : Un "nœud" dans le contexte des échecs représente une position spécifique dans l'arbre de jeu, c'est-à-dire une configuration particulière du plateau d'échecs à un certain stade de la partie. La recherche dans l'arbre de jeu implique l'exploration de différentes variations de mouvements à partir de chaque nœud.

**8-CPU** : Central Processing Unit - Le CPU est le composant central d'un ordinateur ou d'un système informatique. Il s'agit de la partie du matériel qui exécute les instructions et les calculs nécessaires pour faire fonctionner un programme ou un système d'exploitation.

#### Table des Matières :

- I. Introduction générale
- II. Prise en main de la mission
- III. Première problématique
- IV. Deuxième problématique
- V. Conclusion
- VI. Bilan du stage
- VII. Bibliographie

## **I. Introduction générale**

Le C.E.M.C (1) est l'un des meilleurs clubs d'échecs de la Côte d'Azur composé de 66 licenciés. Le président du club est Jean-Michel Rapaire qui est également joueur d'échecs et mon tuteur de stage, ainsi que deux professeurs d'échecs qui encadrent les élèves au sein du club. A mon arrivée, le président du club m'a listé les tâches à accomplir tout au long de mon stage : tout d'abord je devais faire des recherches sur les avancées des moteurs d'échecs notamment avec l'arrivée récente du deep learning. Dans un second temps, j'ai dû comprendre les différences avec les anciens logiciels qui n'utilisaient pas le deep learning et de voir les avantages de ces nouveaux moteurs utilisant le réseau de neurone NNUE (2). Enfin, je devais utiliser ces nouveaux outils afin d'améliorer les performances des joueurs du club en fonction de leur historique de parties stockées dans la base de données.

## **II. Prise en main de la mission**

J'avais à ma disposition l'ordinateur du club avec notamment le logiciel Chess Base (3) permettant de gérer, analyser et étudier des parties d'échecs. Après une semaine, je maîtrisais les fonctionnalités du logiciel tels que : gestion de base de données, analyse de parties, préparation à l'ouverture, recherche de parties, création de rapports...

Dans un second temps, je me suis intéressé aux études qui ont été faites sur l'intelligence artificielle aux échecs. Pour ce faire, je suis d'abord remonté aux prémisses de l'informatique avec l'étude de Claude Shannon intitulé « *Programming a Computer for Playing Chess* » (\*1). Puis je me suis appuyé notamment sur la plateforme en ligne Arxiv (\*2) qui héberge des prépublications académiques dans divers domaines de la science, des mathématiques, de l'informatique et de la recherche en général. C'est un service où les chercheurs peuvent soumettre et partager leurs articles scientifiques. A travers ces articles scientifiques très récents, j'ai pu approfondir davantage les réseaux neuronaux pour le jeu d'échecs ainsi que les rouages de l'apprentissage profond. Parallèlement à cela, j'ai voulu comprendre la différence entre le deep learning et le machine learning. (\*3)

Ensuite, j'ai voulu m'intéresser concrètement aux différents moteurs d'échecs utilisant le réseau de neurone NNUE (2) : principalement *Stockfish* (4), accessible en open source sur le site « abrok.eu » (\*4). J'ai donc pu télécharger les dernières versions de Stockfish et les comparer aux anciennes versions n'utilisant pas le deep learning, notamment en comparant leur vitesse de calcul, leur profondeur de calcul, et leur évaluation en fonction d'une position d'échecs.

Par ailleurs, j'ai trouvé sur le site « *gitHub* » (\*5), qui est une plateforme en ligne qui facilite le partage de code source entre les développeurs de logiciels, l'accès au code de Stockfish(4) NNUE(2) qui est codé en C++ pour sa base de moteur d'échecs traditionnel, qui comprend l'algorithme alpha-bêta avec élagage, et en C pour les parties spécifiques du réseau de neurones NNUE (2). Le réseau de neurones lui-même est basé sur des structures de réseau et des opérations mathématiques qui sont souvent implémentées en utilisant des bibliothèques d'algèbre linéaire et de calcul tensoriel. Ainsi, par curiosité j'ai pu plonger dans les méandres du code, et comprendre la façon dont ce réseau de neurones peut être codé : structure du réseau, calculs matriciels, fonction d'activation, propagation avant, calcul de la perte, rétropropagation, apprentissage, incorporation dans Stockfish (4).

### **III. Première problématique**

Mon encadrant de stage m'a demandé dans un premier lieu de répondre à la problématique suivante : « Dans quelle mesure le réseau de neurone de Stockfish (4) NNUE (2) est-il meilleur que son homologue Stockfish sans réseau de neurone ? »

Tout d'abord, je vais rappeler quelques faits historiques pour mieux cerner la problématique : en 1997 Garry Kasparov – le meilleur joueur d'échecs du monde de son époque – a été battu par le superordinateur *Deep Blue* créé par IBM (International Business Machines Corporation) constitué de 32 cœurs (5) capables d'évaluer 200 millions de positions par seconde. Ce match avait marqué l'humanité car c'était la première fois qu'une machine battait un humain à un jeu qui représente depuis des siècles l'intelligence humaine ou du moins une forme d'intelligence que l'on croyait propre à l'Homme. En effet, on pensait que les échecs étaient basés sur des concepts stratégiques et donc qu'une « boîte de conserve », comme l'a appelé Kasparov (se sentant humilié dans son égo), ne pouvait pas rivaliser et raisonner comme les meilleurs grands maîtres. Cependant les échecs sont bien plus concrets que ce que l'on pensait et sont basés sur du calcul brut, et c'est justement la force de *Deep Blue* qui possède une vitesse de calcul (kN/s) (6) titanique lui permettant de parcourir un arbre de possibilités, et de position donnée (nœud) (7) en un temps très court. Nous détaillerons plus tard comment l'ordinateur évalue les positions à la fin des ramifications qu'il calcule.

Au début du 21<sup>ème</sup> siècle, les ordinateurs sont devenus de plus en plus forts notamment grâce aux progrès scientifiques sur les CPU (8). L'humain ne rivalisant plus avec les ordinateurs depuis Kasparov, il existe maintenant des compétitions entre ordinateurs pour connaître quel est le meilleur ordinateur pour ensuite s'en servir comme un outil afin d'améliorer notre jeu, ce qui sera le sujet de notre seconde problématique. En 2017, un nouveau match éclata et provoqua encore plus de stupeur ; les opposants sont Stockfish (4) (le moteur d'échecs le plus puissant basé comme son ancêtre Deep Blue sur le calcul brut) contre Alpha Zero, un programme d'intelligence artificielle développé par DeepMind (filiale de Google) utilisant le deep learning pour jouer aux échecs. Pour remettre dans le contexte, il faut savoir que AlphaZero avait été entraîné pendant quelques jours en jouant contre lui-même, sans avoir accès aux bases de données d'ouvertures humaines ni aux parties d'autres programmes d'échecs. Les résultats du match ont été étonnantes. AlphaZero a dominé Stockfish avec un score écrasant : AlphaZero a remporté 28 parties, perdu 0 partie et 72 parties se sont terminées par une égalité. Ces résultats étaient particulièrement impressionnantes car AlphaZero avait battu l'un des meilleurs programmes d'échecs traditionnels du monde sans avoir eu besoin d'une base de connaissances humaines préalable.

Ce match a donc attiré l'attention de la communauté scientifique et des joueurs d'échecs du monde entier. Les performances d'AlphaZero ont montré le potentiel du deep learning pour créer des programmes capables d'apprendre et de maîtriser des domaines complexes sans une connaissance humaine antérieure. Les stratégies d'AlphaZero étaient souvent nouvelles et novatrices, remettant en question certaines des idées établies dans les ouvertures et les styles de jeu.

En 2023, le dernier Stockfish (4) utilise, en plus de sa force brute, un réseau de neurone appelé NNUE (2) lui permettant de surclasser son ancien homologue qui lui ne possédait pas de réseau de neurone, ce qui nous ramène à notre problématique de comprendre concrètement la différence dans l'évaluation des positions d'échecs entre ces deux moteurs dont un qui n'utilise pas le deep learning. A noter que le terme "réseau de neurones" et le terme "deep learning" sont liés, mais ils ne sont pas tout à fait interchangeables. La distinction rigoureuse entre les deux concepts est qu'un réseau de neurones est une structure de calcul basée sur le fonctionnement des neurones biologiques, tandis que le deep learning est une approche spécifique de l'apprentissage automatique qui utilise des réseaux de

neurones profonds pour modéliser et résoudre des problèmes complexes. Le deep learning est l'une des applications les plus puissantes et répandues des réseaux de neurones, mais les deux termes ne sont pas exactement synonymes.

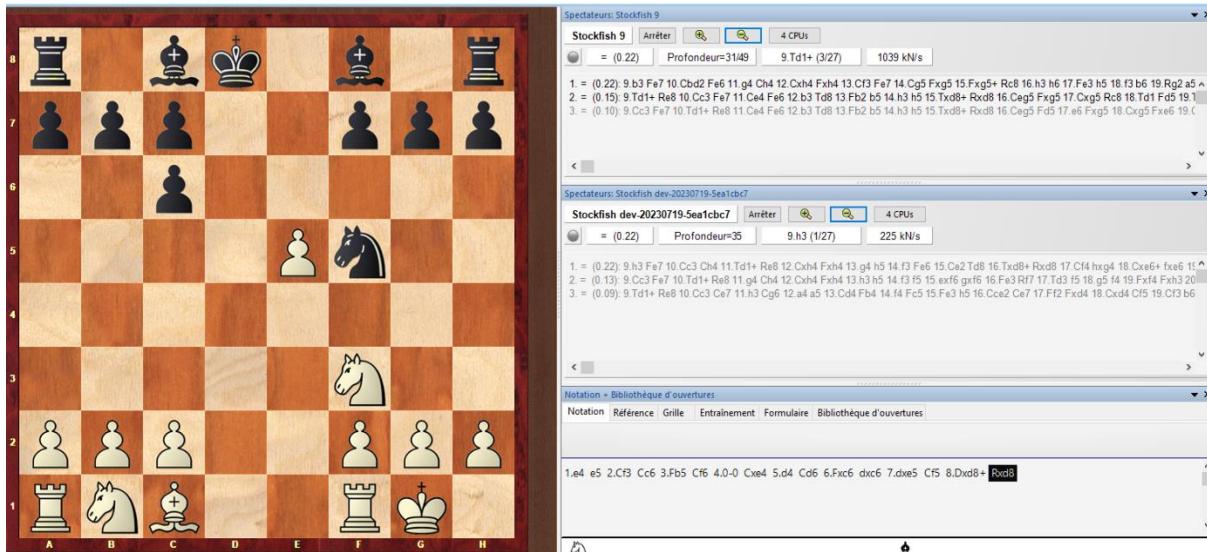
Pour répondre à la problématique, j'ai dû dans un premier temps m'intéresser à comment les moteurs évaluent les positions quand ils parcourrent l'arbre des possibilités, car finalement c'est cela la clef : en effet, lorsque les moteurs parcourrent les ramifications des possibilités, malgré le fait qu'ils aient une forte puissance de calcul, il arrive à un moment donné que le moteur ne puisse pas aller plus en profondeur dans l'arbre de jeu (ou du moins très lentement), dû aux nombres de positions possibles qui en découlent qui tendent vite vers « l'infini ». En effet, selon Shannon il existe  $10^{120}$  parties d'échecs possibles dépassant le nombre d'atomes dans l'univers observable ( $10^{80}$ ), ce qui rend la recherche exhaustive de toutes ces positions totalement irréalisables en termes de temps de calcul. Le moteur est donc obligé de s'arrêter à un moment dans l'arbre de jeu et évaluer la position à chaque fin de variantes et de rejeter les moins bonnes pour garder la position avec la meilleure évaluation.

Stockfish (4) utilise une évaluation heuristique impliquant l'utilisation de critères approximatifs pour évaluer la force ou l'avantage d'une position. Voici quelques exemples d'évaluations heuristiques courantes :

1. Valeur des pièces : Une évaluation heuristique peut attribuer des valeurs numériques fixes à chaque type de pièce (pion, cavalier, fou, etc.) en fonction de leur force perçue. Par exemple, les pions peuvent valoir 1 point, les cavaliers et les fous 3 points, les tours 5 points, et la reine 9 points. Cela peut donner une estimation approximative de l'équilibre matériel.
2. Mobilité des pièces : Une évaluation heuristique peut prendre en compte le nombre de cases accessibles par chaque pièce et attribuer une valeur plus élevée aux pièces ayant plus de mobilité.
3. Structure de pions : L'évaluation peut inclure des critères relatifs à la structure des pions, tels que l'isolement des pions, les pions doublés, etc.
4. Sécurité du roi : L'évaluation peut inclure des critères liés à la sécurité du roi, comme la présence de pièces adverses menaçantes à proximité.
5. Contrôle du centre : Une évaluation heuristique peut donner plus de poids aux pièces qui contrôlent le centre du plateau.
6. Développement : Les pièces développées en début de partie peuvent recevoir une valeur accrue dans l'évaluation.

Il est important de noter que les évaluations heuristiques sont des approximations et peuvent être biaisées. L'avantage de Stockfish NNUE est qu'il utilise une combinaison d'évaluations heuristiques et de réseaux de neurones, pour obtenir des évaluations plus précises et équilibrées.

J'ai donc voulu comparer les deux moteurs dans des positions données et voir les évaluations par les deux modules dans chaque cas. J'ai abouti à environ la même évaluation sur un échantillon de 100 positions différentes après quelques coups. Prenons ici l'exemple de la défense Berlinoise qui résulte après la position ci-dessous :



### Défense Berlinoise en fonction de l'évaluation des 2 moteurs

Tout d'abord, on remarque que les coups proposés par les moteurs sont identiques et ont quasiment la même évaluation : Cc3 (Cavalier c3) est estimé à 0.10 pour Stockfish et à 0.13 pour Stockfish NNUE. Pour ne pas perdre nos lecteurs, 0.10 signifie que les blancs ont l'avantage d'un dixième de pion, en effet comme nous l'avons évoqué plus haut dans l'évaluation heuristique de la valeur des pièces, un pion vaut 1 point. Cet avantage est donc microscopiquement ridicule et donc on peut légitimement estimer que la position est égale selon les deux modules. Deuxièmement, nous pouvons noter que la profondeur des deux moteurs est identique, c'est-à-dire que l'arbre des possibilités est allé à environ 35 coups plus loin que la position présente sur l'échiquier. En revanche, il y a une grosse différence au niveau du nombre de milliers de nœuds (7) explorés par seconde (kN/s) (6) ; pour Stockfish on obtient une vitesse de calcul de 1039 (kN/s) tandis que pour Stockfish NNUE on n'a que 225 (kN/s). Il est nécessaire de préciser que les moteurs tournent tout deux sur 4 CPUs (8).

On peut donc se demander pourquoi Stockfish NNUE a besoin d'explorer moins de nœuds que son homologue pour obtenir la même évaluation finale. Stockfish utilise la force brute en combinaison avec des techniques d'élagage pour réduire l'espace de recherche. Cela signifie que certaines branches de l'arbre de recherche sont éliminées dès le départ si elles semblent peu prometteuses. Les techniques d'élagage, comme l'algorithme alpha-bêta, permettent de réduire considérablement le nombre de positions à évaluer, rendant la recherche plus efficace.

Selon moi, cette différence de vitesse de calcul peut être due à la méthode de Monte-Carlo (technique d'échantillonnage stochastique afin d'estimer les résultats probables d'une action en simulant un grand nombre de scénarios aléatoires) qui lorsqu'elle est combinée avec un réseau de neurones, peut permettre à Stockfish NNUE d'évaluer rapidement et de manière approximative les positions en effectuant un petit nombre de simulations aléatoires tout en s'appuyant sur les informations fournies par le réseau de neurones pour guider ces simulations. Cela peut réduire la nécessité d'explorer de manière exhaustive de grandes quantités de positions, ce qui peut contribuer à une réduction du nombre total de nœuds parcourus tout en maintenant une évaluation précise.

Dans un second temps, j'ai voulu aller plus loin pour comprendre dans quel type de position Stockfish NNUE allait prendre le dessus sur Stockfish sans réseau de neurones. Pour cela, j'ai proposé à ces deux différents moteurs des positions qui nécessitaient moins de calcul brut mais plus de concept stratégique, et de compréhension de la position. Nous allons nous attarder sur le problème suivant :



### Problème d'échecs – Trait au blanc

Ce problème est issu du livre intitulé « Les échecs artistiques » où les positions données permettent de stimuler l'imagination et la perspicacité des joueurs d'échecs. J'ai moi-même essayé de résoudre ce problème il y a plusieurs années, et j'avais trouvé la solution au bout de 20 minutes de réflexion. Pour expliquer de manière simple l'enjeu de la position (j'insiste volontairement sur l'aspect technique de la position pour justement bien comprendre comment fonctionne le réseau de neurones NNUE afin de le comparer aux réseaux de neurone humain) : les blancs semblent totalement perdants, ici ils ont un retard matériel, de plus le cavalier risque d'être perdu et la dame noire va se faufile dans le camp blanc et faire de gros dégâts. Un joueur d'échecs aguerri repère tout de suite qu'il est possible de faire une forteresse en fermant la position. Pour cela il faut sacrifier le Cavalier en g4 qui fait une fourchette royale (attaquant le roi et la dame) forçant les noirs à recapturer avec le pion en h5, et les blancs continuent par pion d4 échec forçant le roi à bouger. On semble sauver avec les blancs car on a réussi à fermer la position mais la dame noire peut encore se faufile en passant par a8 puis la case a6. C'est là que toute la beauté artistique (beauté propre aux yeux de l'homme !) de ce problème intervient : il faut réussir à imaginer de « s'auto pater » (le pat aux échecs correspond à une situation dans laquelle un joueur ne peut pas effectuer de coup légal sans mettre son roi en échec, et cela aboutit à une partie nulle) : pour cela, il faut bouger la tour en h1 et ramener le roi blanc en g1 de sorte que quand la dame arrivera sur la case a6 afin de pénétrer le camp blanc il y aura pat car les blancs ne pourront pas jouer de coup légal car aucune pièce ni pions ne peut bouger, et donc étant donné que l'on ne peut passer son tour aux échecs, la partie se conclut par match nul.

Maintenant, voyons ce qu'ont nos deux turbines dans le ventre ! Tout d'abord Stockfish 9 n'utilisant pas de réseau de neurones est totalement noyé dans la complexité de la position malgré le fait qu'il tourne pleine puissance (4 CPUs (8)), une vitesse de calcul à 1491 kN/s (6) et une profondeur de 37 coups (soit 37 demi-coups blancs et 37 demi-coups noirs) : en effet, il donne une évaluation à -3,22 (cela signifie que les blancs sont totalement perdants) et son coup proposé n'est pas le coup de la solution.

Nous pouvons nous demander comment un moteur d'échecs qui est bien plus fort que tous les grands maîtres du jeu, ne trouve pas la solution alors qu'un joueur aguerri va trouver la solution en se creusant les méninges. De plus, la solution ne demande pas d'aller très loin dans l'arbre des possibilités mais seulement une profondeur de 10 coups suffit. Le coup qui est proposé par Stockfish 9 ne fait que repousser le problème et va conduire à une perte encore plus importante qui est celle de perdre la partie. C'est ce qu'on appelle l'effet d'horizon : cela correspond à la limitation de la profondeur de

recherche lors de l'analyse des positions, ce qui peut entraîner des erreurs d'évaluation en raison de l'incapacité à anticiper les conséquences à long terme des coups.

Maintenant intéressons-nous à Stockfish NNUE : je n'ai volontairement utilisé qu'un CPU au lieu de 4 afin qu'il soit encore plus handicapé par rapport à son homologue, et on remarque que sa vitesse de calcul (102 kN/s) est plus de 10 fois inférieure à celle de Stockfish 9 et sa profondeur est à peu près équivalente (31 de profondeur). Ce dernier propose la bonne suite de coup de la solution mais donne une évaluation de -1.66 donc moins importante que celle de Stockfish 9 mais qui reste quand même conséquente sur l'avantage des noirs de gagner la partie, alors que ce n'est pas le cas, l'évaluation devant être de 0, étant donné que si les blancs jouent les bons coups la position est égale.

J'ai ensuite voulu mettre la position avec les bons premiers coups de la combinaison pour voir si l'évaluation des modules changent.



Position après la bonne suite de coups en fonction des évaluations des moteurs

J'ai fait tourner les moteurs à plein régime (pendant plusieurs dizaines de minutes) avec une profondeur de 65 coups. Du côté de Stockfish 9, l'évaluation s'est même encore plus emballé donnant les noirs encore plus gagnants que précédemment (-7.04), et concernant Stockfish NNUE, l'évaluation est à peu près similaire (-1.83). Il est intéressant de noter que je n'affiche pas seulement la première ligne de l'ordinateur mais également les autres coups proposés afin de comparer les évaluations des différentes lignes proposées par les modules. On remarque donc que sur les 3 autres coups proposés par les moteurs, l'évaluation est strictement la même. J'ai donc demandé à un grand maître du club son expertise. Il m'a expliqué que si le moteur affichait la même évaluation sur toutes les lignes, cela signifie qu'il y a de grandes chances qu'on soit en présence d'une forteresse. Le terme "forteresse" aux échecs fait référence à une configuration particulière de pièces sur l'échiquier qui rend très difficile, voire impossible, pour l'adversaire de progresser et de gagner la partie, même s'il a un avantage matériel ou positionnel. Une forteresse se caractérise par le fait qu'aucun coup ne permet à l'adversaire de faire des progrès significatifs en dépit de ses efforts pour créer des menaces.

Finalement on peut en déduire que même si Stockfish NNUE ne donne pas la bonne évaluation précise (0.00), il trouve tout de même les meilleurs coups qui mènent à la résolution du problème tandis que Stockfish 9 (qui n'utilise pas de réseaux de neurones) ne trouve pas le meilleur coup de la position et donne une évaluation erratique de la position qui varie de manière incohérente, imprévisible, entre -3.22 à -7.04 en quelques coups. En d'autres termes, l'évaluation ne suit pas de tendances claires et présente des fluctuations importantes sans raison apparente.

En résumé, la différence majeure entre Stockfish et Stockfish avec NNUE réside dans la méthode d'évaluation des positions. Stockfish avec NNUE utilise un modèle d'apprentissage automatique profond qui apprend à partir de données d'entraînement pour fournir des évaluations plus précises et plus sophistiquées des positions d'échecs, tandis que Stockfish traditionnel repose davantage sur des heuristiques et des règles d'évaluation traditionnelles, ne lui permettant pas de résoudre des positions basées sur de la compréhension stratégique à long terme.

En conclusion, Stockfish NNUE surclasse son homologue sans réseau de neurones dans les positions stratégiques complexes tout en calculant moins de nœuds (7) par seconde, notamment grâce à la méthode de Monte-Carlo combinée à l'apprentissage profond permettant d'élaguer rapidement l'arbre des possibilités tout en fournissant une évaluation précise des positions. Comme nous venons de le voir, la force brute de Stockfish ne suffit pas dans certains types de positions, en effet en plus de la profondeur de calcul il est nécessaire d'avoir une bonne évaluation des positions parcourues. L'objectif ultime serait d'obtenir des évaluations plus précises et fiables, ce qui améliorerait la qualité des décisions prises par les moteurs d'échecs. En effet, même si Stockfish NNUE surpassé de loin le niveau des meilleurs joueurs d'échecs du monde, il existe encore des positions spécifiques où les moteurs d'échecs ont leur limite ; certes ils vont trouver les meilleurs coups mais ils ne vont pas fournir l'évaluation correcte.

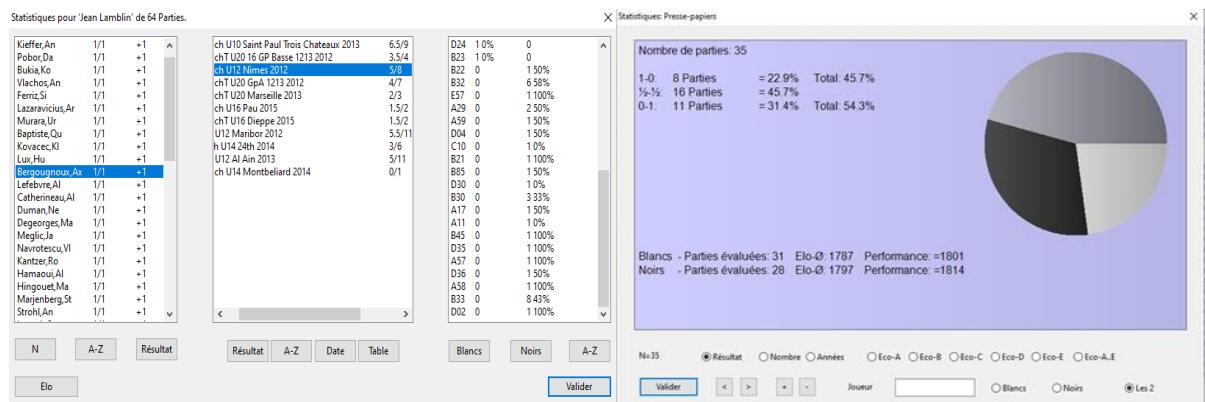
Pour aller plus loin sur la question, on peut se demander si les moteurs d'échecs à l'avenir seront encore plus puissants et permettront de trouver la « vérité » de ce jeu millénaire, c'est-à-dire de résoudre le jeu d'échecs, je m'explique : en théorie des jeux, le théorème de Zermelo stipule que dans tout jeu tour à tour, fini, à deux joueurs, à information parfaite, et sans hasard, l'un des deux joueurs gagne ou fait nulle, quelle que soit la manière dont l'adversaire joue. En 1950, le théoricien de l'information Claude Shannon décrit une méthode théorique afin de résoudre le jeu d'échecs ; cela repose sur un processus hypothétique où l'on explore toutes les possibilités de coups à partir de la position initiale du jeu, en considérant également tous les coups potentiels de l'adversaire et ainsi de suite jusqu'à la fin de la partie. À chaque étape, chaque variante se termine par une victoire, une défaite ou une égalité. En remontant à rebours à partir de la fin de la partie, il serait possible de déterminer si une victoire est forcée, si la position de départ mène à une égalité ou si elle est perdante en suivant un jeu parfait. Mais la procédure théorique de Shannon se heurte à plusieurs problèmes ; tout d'abord, il est important de noter que même avec un ordinateur quantique, explorer toutes les  $10^{120}$  parties d'échecs possibles restent un défi monumental, de surcroît même si on arrivait à générer l'ensemble complet de l'arbre du jeu d'échecs, cela ne signifie pas nécessairement qu'on aurait résolu le jeu d'échecs. En effet la résolution du jeu d'échecs nécessite de trouver la meilleure séquence de coups dans chaque position, ce qui va au-delà de simplement énumérer toutes les variations. Il faudrait des preuves mathématiques pour déterminer les meilleurs coups dans chaque position et avoir la certitude que la partie a été jouée parfaitement du côté blanc et noir. La question philosophique se pose ; on sait que nul n'est parfait mais la machine peut-elle jouer parfaitement, le débat est ouvert.

En outre, selon les mathématiciens, la complexité inhérente du jeu d'échecs fait que sa résolution semble être un problème insoluble. En tant qu'adepte du roi des jeux, je pense que le jeu d'échecs est dans sa nature intrinsèque un jeu qui mène à la partie nulle sur des coups parfaits des deux camps car il y a beaucoup trop de ressources défensives pour le camp noir comme par exemple le pat, l'insuffisance matériel, l'échec perpétuel, les forteresses...

## IV. Deuxième problématique

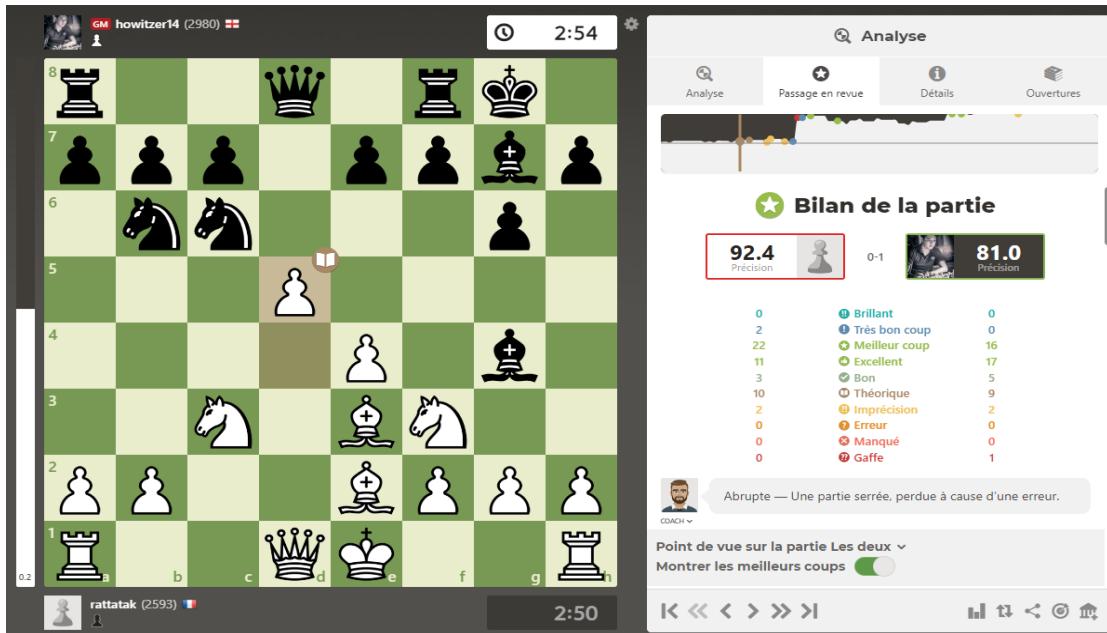
Comme je l'ai évoqué dans la première partie, nous allons maintenant nous intéresser, à la demande de mon tuteur Jean-Michel Rapaire, comment les modules d'échecs basés sur un réseau de neurones tels que Stockfish (4) NNUE (2) peuvent aider les joueurs d'échecs à améliorer, optimiser leur performance. En effet, après la défaite de Garry Kasparov contre le superordinateur Deep Blue comme évoqué plus haut, il expliqua qu'en réalité l'ordinateur ne jouait pas au même jeu que l'Homme car la machine n'a pas d'émotions et donc ne subissait pas la dimension psychologique du jeu, par conséquent les moteurs d'échecs ne doivent pas être nos ennemis mais plutôt un outil afin d'améliorer notre manière de jouer. En effet, depuis l'avènement des moteurs d'échecs depuis deux décennies, il n'y a jamais eu autant de forts grands maîtres, et de jeunes joueurs de plus en plus précoces atteignant les sommets. Les raisons d'un tel essor de prodige de l'ère de l'informatique sont dues à l'accessibilité aux bases de données tels que Chess Base (3)(\*6) répertoriant des millions de parties d'échecs jouées à travers le monde entier permettant d'étudier les tendances, les stratégies et les modèles de jeu des meilleurs joueurs du monde. De plus, les logiciels d'analyse ont permis aux joueurs de revenir sur leurs parties et d'identifier leurs erreurs et leurs points faibles. En outre les analyses assistées par ordinateur ont favorisé une approche plus scientifique du jeu d'échecs, avec des méthodes de calcul plus rigoureuses et une compréhension plus approfondie des aspects tactiques et stratégiques.

Mon tuteur de stage m'a demandé tout d'abord de faire une analyse approfondie sur le joueur d'échecs Jean Lamblin, membre du C.E.M.C (1) qui participe régulièrement pour le club à des compétitions au niveau international. Dans un premier temps, j'ai importé les parties de ce joueur dans un dossier à l'aide du logiciel Chess Base (3). J'ai pu ensuite établir des statistiques sur ses résultats avec les pièces blanches/noires et en fonction des ouvertures jouées.



### Statistique des parties de Jean Lamblin

Dans un second temps, j'ai réalisé une étude plus approfondie de chaque partie en utilisant l'analyse de Stockfish(4) NNUE(2). Nous allons nous intéresser dans ce rapport à la partie suivante afin d'établir les forces et faiblesses de ce joueur à travers les différentes phases de la partie (ouverture, milieu de jeu, finale).



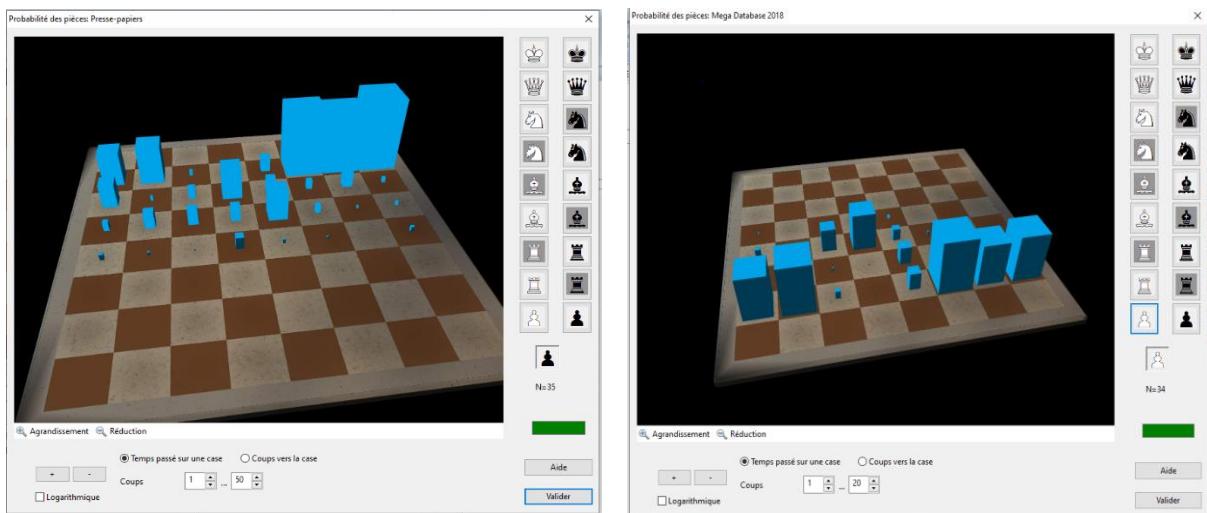
### Partie de Jean Lamblin contre le grand maître Davil Howell

On peut voir à droite d'après le « Bilan de la partie » que les blancs ont été précis à 92.4 et les noirs à 81.0. La précision lors des parties d'échecs fait référence à la justesse et à l'exactitude des coups joués par un joueur. Un joueur est considéré comme précis lorsqu'il prend des décisions qui correspondent aux meilleurs coups possibles dans une position donnée, selon l'analyse et les recommandations des moteurs d'échecs et des normes établies par la théorie échiquier. La précision est mesurée en pourcentage et peut être calculée en comparant les coups joués par un joueur avec les coups recommandés par un moteur d'échecs. Par exemple, dans la partie ci-dessus Jean Lamblin a joué 92.4% de coups conformes aux suggestions optimales, on peut donc dire que sa précision est de 92.4. La recherche de la précision est un objectif important pour les joueurs d'échecs, car elle contribue à éviter les erreurs coûteuses et à maximiser les chances de succès. Les joueurs de haut niveau s'efforcent généralement d'améliorer leur précision en étudiant les parties, en analysant les positions critiques et en développant leur compréhension stratégique et tactique du jeu.

En revanche, la précision ne suffit pas toujours à remporter la partie, comme d'après une fameuse citation "celui qui gagne est celui qui commet l'avant-dernière erreur". L'expression souligne également le caractère dynamique et changeant du jeu d'échecs, où l'avantage peut basculer d'un côté comme de l'autre, en effet « un mauvais coup peut ruiner 40 bons coups ». C'est comme si à chaque coup il y avait une épée de Damoclès au-dessus de la tête des joueurs, il ne faut donc jamais considérer la partie comme gagnée ou perdue. Dans la partie à laquelle nous nous intéressons opposant Jean Lamblin à David Howell, on peut souligner que malgré le fait que les blancs ont été plus précis que leur adversaire, cela n'a pas suffit à remporter la partie. L'avantage de l'analyse avec Stockfish NNUE (2) permet de repérer facilement le moment où la partie a basculé. Je ne vais pas rentrer sciemment en détail dans les aspects techniques échiquiers car cela n'aurait aucun intérêt pour les non joueurs d'échecs qui lisent ce rapport. Cependant je vais résumer la partie en 3 phases : tout d'abord dans l'ouverture j'ai remarqué par rapport au temps de réflexion des deux joueurs que les blancs connaissaient plus loin la théorie des ouvertures que leur adversaire, ce qui donne toujours un ascendant pour le camp qui répète ce qu'il a déjà vu car il connaît plus loin la théorie. De plus j'ai pu, grâce à la base de données Chess Base, estimer qu'après le 10<sup>ème</sup> coup blanc d5 (voir capture d'écran de la partie ci-dessus), les blancs ont 51% de chance de gain basé sur un échantillon de 527 parties à partir du coup 10.d5. Dans le milieu de partie, Jean Lamblin a dominé son adversaire en étant plus affûté tactiquement que son

adversaire. Et en finale les noirs ont su profiter d'une erreur des blancs pour conclure la partie. Une partie ne permet évidemment pas d'établir des conclusions sur les points faibles et point forts d'un joueur. C'est pour cette raison que j'ai dû analyser, à l'aide de Stockfish (4) NNUE (2), plus d'une cinquantaine de parties issues de la base de données afin d'obtenir des résultats qualitatifs. Après une analyse détaillée, il en ressort que Jean Lamblin est très précis dans le calcul dans les positions dynamiques, en revanche lorsqu'il faut jouer des finales de tours techniques, il est plus dur pour ce joueur d'être précis. En travaillant les finales en général Jean Lamblin a la possibilité de conclure plus de parties ou de tenir des finales inférieures.

Par ailleurs, j'ai étudié plus en détail le début de partie de Jean Lamblin : grâce à Chess Base (3) j'ai pu afficher la probabilité de présence des pièces sur chaque case de l'échiquier.



### Probabilité de présence des pions de Jean Lamblin

Cet outil est très efficace pour avoir les tendances d'un joueur dans les différentes phases du jeu, et donc de pouvoir établir des patterns. Pour pouvoir ensuite les comparer aux moteurs d'échecs les plus puissants et s'inspirer de leur schéma de jeu afin d'améliorer les capacités des joueurs. En effet les récents moteurs d'échecs utilisant le deep learning ont une approche du jeu plus humaine (sûrement due aux réseaux de neurones NNUE(2)), leur façon de jouer est menée par des concepts stratégiques à long terme. En analysant de nombreuses parties jouées par Stockfish NNUE contre d'autres ordinateurs durant le « Computer Chess Championship » sur le site chess.com (\*7), j'ai pu remarquer plusieurs tendances de Stockfish NNUE ; la première est qu'il aime prendre de l'espace afin d'étouffer son adversaire, deuxièmement l'activité des pièces pour lui est très importante il n'hésite pas à même sacrifier du matériel afin de dominer positionnellement son adversaire. En effet comme nous avons évoqué plus haut, les modules qui n'utilisent pas de réseaux de neurones évaluent les positions grâce à des heuristiques, comme par exemple la valeur des pièces, mais en réalité la valeur d'une pièce est relative au développement de la partie.

Les ordinateurs les plus puissants ont donc beaucoup de choses à nous apprendre et en analysant avec eux notamment les ouvertures, on peut trouver des nouvelles idées qui n'ont jamais été jouées auparavant. Il existe des parties dites par correspondance : chaque joueur joue depuis chez lui et étudie la position à l'aide d'un moteur d'échecs pendant des heures afin de trouver le meilleur coup, en effet un très bon joueur d'échecs assisté par un moteur d'échecs est plus fort que le moteur lui-même, d'où l'intérêt des parties par correspondance.

Cependant lors des parties entre humains nous n'avons pas accès à Stockfish (4) NNUE (2) ce qui change l'approche à adopter lorsqu'on cherche des idées avec l'ordinateur depuis chez nous. Je m'explique : imaginons que l'on trouve une nouvelle idée dans l'ouverture italienne et le moteur nous donne une évaluation de 0.50, on obtient ensuite cette nouvelle position lors d'une partie, et après quelques coups on se rend compte qu'en pratique la position n'est pas du tout claire et que si l'on ne joue pas de manière précise on peut tout de suite se retrouver perdant. En effet, comme le précise le champion du monde Magnus Carlsen quand l'ordinateur nous affiche une évaluation dite « théorique », il ne nous dit pas pour qui c'est plus facile à jouer en « pratique », en particulier il peut être très désagréable devant l'échiquier de devoir trouver la seule ligne et cela pendant plusieurs coups afin d'obtenir un petit avantage, alors qu'une imprécision de notre côté nous coûterait la partie. L'ordinateur quant à lui n'a pas ce problème puisqu'il trouve les meilleurs coups. C'est ce que le grand maître français Etienne Bacrot appelle le prix d'un coup, c'est-à-dire qu'un coup chez les blancs n'a pas la même valeur que chez les noirs par exemple. Le travail des seconds (personnes qui assistent les joueurs de haut niveau lors des tournois) est de justement comprendre, analyser, expliciter les idées, les plans d'une position afin d'aider le joueur à bien comprendre une position clef avec ses tenants et ses aboutissants, tout en vérifiant que la position n'est pas trop compliquée à jouer en pratique pour son joueur et pose des problèmes pratiques à l'adversaire c'est surtout ce dernier point qui est important. En effet, j'ai remarqué que plus on faisait tourner l'ordinateur longtemps et en profondeur, plus les évaluations des lignes tendent vers l'égalité (0.0), car comme je l'ai dit plus haut le jeu par sa nature est intrinsèquement nul, il est donc mathématiquement impossible de trouver une idée dans une l'ouverture qui va réfuter les échecs, c'est pourquoi les seconds cherchent des petites idées qui posent des problèmes pratiques à l'adversaire. On comprend donc mieux pourquoi l'intelligence humaine et l'intelligence artificielle sont plus fortes ensemble car s'il suffisait de recopier les coups du moteur pour prendre l'avantage, les joueurs du top niveau n'auraient pas besoin de seconds. De manière plus évidente, l'ordinateur montre juste des coups tandis que l'humain montre des plans.

Pour conclure afin d'optimiser les performances des joueurs d'échecs, les moteurs utilisant le deep learning tels que Stockfish (4) NNUE (2) sont un très bon outil afin d'ouvrir de nouvelles perspectives, mais il faut garder un œil humain et pratique dans la manière de jouer car ce qui est évident pour un moteur qui calcule des milliers de nœuds par seconde (kN/s)(6) ne l'est pas forcément pour un humain qui est soumis en plus de cela au stress, à la fatigue, à la psychologie du jeu...

## **V. Conclusion**

Depuis des siècles, la manière de jouer aux échecs a constamment évolué, à l'époque de Napoléon avec le style romantique caractérisé par une approche audacieuse et aventureuse du jeu, puis depuis le début du 20<sup>ème</sup> siècle jusqu'à aujourd'hui nous sommes dans l'époque moderne caractérisée par une approche plus scientifique du jeu (théorie des ouvertures, principes, stratégie/ tactique, techniques en finale, aspect statique/dynamique...). Et au début du 21<sup>ème</sup> avec l'avènement des moteurs d'échecs, et des bases de données telles que Chess Base(3), la manière de jouer s'est uniformisée à l'échelle internationale même si les joueurs peuvent avoir des styles différents. Depuis 5 ans, des nouveaux programmes d'intelligence artificielle basés sur le deep learning, tels que AlphaZero, Stockfish (4) NNUE (2), ont vu le jour, ce qui a permis encore de repousser les limites du niveau de jeu : comme nous avons vu à travers la première problématique, la puissance de calcul associée au réseau de neurones NNUE (2) et à la méthode de Monte-Carlo permet d'obtenir des évaluations précises des positions tout en ayant une vitesse de calcul (kN/s)(6) inférieure à son homologue Stockfish n'utilisant pas de deep learning. On peut supposer qu'à l'avenir les moteurs d'échecs vont être encore plus puissants

notamment grâce aux progrès sur les processeurs permettant aux moteurs d'explorer plus de variantes, d'analyser plus en profondeur, tout en étant assistés par un réseau de neurones profonds afin d'affiner les évaluations des positions. D'ici plusieurs années, les ordinateurs quantiques peuvent également repousser encore plus les limites du niveau de jeu des moteurs. Cependant comme nous avons vu à travers la seconde problématique, il ne suffit pas de recopier les meilleurs coups affichés par les modules pour améliorer le niveau de jeu des joueurs d'échecs, en effet il y a une différence entre la valeur théorique d'une position évaluée par un moteur et son aspect pratique. C'est pour cette raison qu'il faut comprendre les enjeux de la position, analyser avec un œil humain les coups suggérés par les moteurs afin d'optimiser notre manière d'appréhender les positions. En conclusion, l'intelligence artificielle n'est qu'un outil pour améliorer les capacités des humains, elle ne peut en aucun cas remplacer l'intelligence humaine même si elle restera toujours plus rapide, plus précise, plus endurante que l'Homme sur les tâches lui incombeant.

## **VI. Bilan du stage**

Au cours de mon stage, j'ai eu l'opportunité de plonger dans le monde professionnel en intégrant le club d'échecs et en travaillant sur des projets concrets. J'ai acquis une compréhension approfondie de l'importance de gérer efficacement une base de données de parties d'échecs à l'aide de Chess Base, ce qui a renforcé mes compétences en traitement de données et en organisation de l'information. J'ai également exploré la pertinence du deep learning dans l'analyse des parties des joueurs du club, ce qui m'a permis de lier mes connaissances en intelligence artificielle avec une application pratique.

Domaine	Connaissances et Compétences Acquises
Traitement de données	Gestion optimisée de bases de données et extraction d'informations pertinentes.
Statistiques et Prédictions	Application de méthodes statistiques pour identifier des corrélations dans les performances des joueurs.
Intelligence Artificielle	Compréhension approfondie des concepts de deep learning appliqués aux échecs.
Optimisation	Utilisation de modèles pour prévoir les gains potentiels et optimiser les performances dans les tournois

En somme grâce à ce stage, j'ai pu découvrir comment les compétences en traitement de données, en programmation et en intelligence artificielle peuvent être appliquées dans un contexte professionnel. J'ai également appris à travailler en équipe, à communiquer efficacement avec les membres du club et à répondre aux besoins spécifiques des joueurs. Cette expérience renforce mon intérêt pour l'intelligence artificielle et me motive à explorer davantage les domaines de l'analyse de données et de l'optimisation dans ma future carrière d'ingénieur.

## Bibliographie :

- (\*1) [http://archive.computerhistory.org/projects/chess/related\\_materials/text/2-0%20and%202-1.Programming\\_a\\_computer\\_for\\_playing\\_chess.shannon/2-0%20and%202-1.Programming\\_a\\_computer\\_for\\_playing\\_chess.shannon.062303002.pdf](http://archive.computerhistory.org/projects/chess/related_materials/text/2-0%20and%202-1.Programming_a_computer_for_playing_chess.shannon/2-0%20and%202-1.Programming_a_computer_for_playing_chess.shannon.062303002.pdf)
- (\*2) [https://arxiv.org/search/?query=chess+NNUE&searchtype=all&abstracts=show&order=-announced\\_date\\_first&size=50](https://arxiv.org/search/?query=chess+NNUE&searchtype=all&abstracts=show&order=-announced_date_first&size=50)
- (\*3) <https://www.ionos.com/digitalguide/online-marketing/search-engine-marketing/deep-learning-vs-machine-learning/>
- (\*4) <https://abrok.eu/stockfish/>
- (\*5) <https://github.com/official-stockfish/Stockfish/wiki>
- (\*6) <https://fr.chessbase.com/>
- (\*7) <https://www.chess.com/computer-chess-championship#event=ccc21-blitz-semifinals&game=337>
- (\*8) <https://tests.stockfishchess.org/tests/view/64d147845b17f7c21c0dd86c>
- (\*9) <https://stockfishchess.org/>
- (\*10) <https://ipmancChess.yolasite.com/>
- (\*11) <https://github.com/official-stockfish/Stockfish/wiki/Regression-Tests>
- (\*12) <https://www.openingtree.com/>