UNIVERSIDADE DO ESTADO DA BAHIA (UNEB)
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA, CAMPUS I
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GUILHERME FRANÇA DE SOUZA

INTERPRETAÇÃO DE LANCES REALIZADOS POR ENGINES DE XADREZ UTILIZANDO INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL

SALVADOR
2024

GUILHERME FRANÇA DE SOUZA

INTERPRETAÇÃO DE LANCES REALIZADOS POR ENGINES DE XADREZ UTILIZANDO INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do Departamento de Ciências Exatas e da Terra (DCET) - Campus I, da Universidade do Estado da Bahia (UNEB), como requisito à obtenção do grau de bacharel em Sistemas de Informação.
Área de concentração: Ciências da Computação.

Orientador: Prof. Dr. Ernesto de Souza Massa Neto

SALVADOR
2024

TERMO DE ANUÊNCIA DO ORIENTADOR

Declaro para os devidos fins que li e revisei este trabalho e atesto sua qualidade como resultado final desta monografia. Confirmo que o referencial teórico apresentado é completo e suficiente para fundamentar os objetivos propostos e que a metodologia científica utilizada e os resultados finais são consistentes e com qualidade suficiente para submissão à banca examinadora final do Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação.

Prof. Dr. Ernesto de Souza Massa Neto
Orientador

GUILHERME FRANÇA DE SOUZA

INTERPRETAÇÃO DE LANCES REALIZADOS POR ENGINES DE XADREZ UTILIZANDO INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do Departamento de Ciências Exatas e da Terra (DCET) - Campus I, da Universidade do Estado da Bahia (UNEB), como requisito à obtenção do grau de bacharel em Sistemas de Informação. **Área de concentração:** Ciências da Computação.

Aprovada em: .

BANCA EXAMINADORA

Prof. Dr. Ernesto de Souza Massa Neto
Orientador

Prof. Dr. Nome Completo da Pessoa
Examinador interno (DCET-I/UNEB)

Prof. Dra. Nome Completo da Pessoa
Examinador interno (DCET-I/UNEB)

AGRADECIMENTOS

É com profunda gratidão que dedico esta seção de agradecimentos do meu Trabalho de Conclusão de Curso. Esta jornada acadêmica não teria sido possível sem o apoio e incentivo de pessoas incríveis que estiveram ao meu lado.

Agradeço primeiramente à minha família, por ser a minha base, pelo amor incondicional, e por sempre acreditarem no meu potencial. Vocês foram a força que impulsionou cada passo desta caminhada.

Ao meu orientador, agradeço pela paciência, sabedoria e orientação ao longo de todo o processo de pesquisa. Seu apoio foi fundamental para o desenvolvimento deste trabalho.

Aos professores que contribuíram com seus conhecimentos e experiências, a minha sincera gratidão. Cada aula, conselho e feedback foram fundamentais para o meu crescimento acadêmico.

Aos amigos e colegas de curso, pela troca de ideias, momentos de estudo e pelo apoio mútuo, meu muito obrigado. Compartilhamos desafios e conquistas que tornaram esta jornada ainda mais significativa.

Por fim, agradeço a todos que, de alguma forma, contribuíram para o sucesso deste trabalho. Cada palavra de encorajamento, cada gesto de apoio, foi fundamental para chegar até este momento.

Este TCC não é apenas um marco acadêmico, mas uma realização coletiva. A todos vocês, o meu mais sincero agradecimento.

"A educação é a arma mais poderosa que você pode usar para mudar o mundo."

(Nelson Mandela)

RESUMO

O presente trabalho visa utilizar técnicas de inteligência artificial explicável para trazer interpretabilidade das complexas engines de xadrez que funcionam com uma caixa preta. O objetivo é facilitar o entendimento de quais fatores são utilizados pela engine para tomar suas decisões. A abordagem adotada para alcançar esse objetivo segue a metodologia do Design Science Research (DSR), que se concentra na criação e avaliação de artefatos projetados para resolver problemas práticos. Inicialmente, foram selecionadas posições com diferentes temáticas, para realizar duelos entre o Stockfish com redes neurais (NNUE) ativadas e desativadas, limitando a profundidade das análises para enfatizar uma visão holística do tabuleiro.Os resultados preliminares já apontam que existe uma vantagem considerável entre engines que utilizam redes neurais para avaliar posições, sobre aquelas que ainda utilizam o método clássico.

Palavras-chave: XAI; Chess engine; explicabilidade; Xadrez.

ABSTRACT

The present work aims to use explainable artificial intelligence techniques to bring interpretability to complex chess engines that function as black boxes. The goal is to facilitate the understanding of which factors are used by the engine to make its decisions. The approach adopted to achieve this objective follows the Design Science Research (DSR) methodology, which focuses on the creation and evaluation of artifacts designed to solve practical problems. Initially, positions with different themes were selected to perform matches between Stockfish with neural networks (NNUE) enabled and disabled, limiting the depth of analysis to emphasize a holistic view of the board. Preliminary results already indicate that there is a considerable advantage for engines that use neural networks to evaluate positions over those that still use the classic method.

Key-words: XAI; Chess engine; explicabilidade; Xadrez.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Ao longo dos anos os seres humanos aprimoraram suas habilidades no xadrez, em um processo contínuo de aprendizado e absorção da sabedoria acumulada dos jogadores que vieram antes deles. Com o advento da tecnologia computacional, esse processo atingiu novos patamares de velocidade e profundidade. As engines de xadrez, impulsionadas pela crescente capacidade de processamento dos computadores e algoritmos cada vez mais modernos, contribuíram significativamente para esse avanço, adicionando constantemente novos conhecimentos e entendimentos ao jogo.

As engines de xadrez representam uma etapa significativa no desenvolvimento da inteligência artificial. Esses programas avaliam meticulosamente numerosas potenciais jogadas e selecionar consistentemente as mais vantajosas. Embora a origem das engines remonte ao meio do século XX, foi apenas no final da década de 1990 e início dos anos 2000 que eles alcançaram a paridade com os Grandes Mestres humanos (Østensen, 2016). Desde então, suas capacidades cresceram exponencialmente, superando as capacidades humanas de forma notável.

Atualmente, os melhores jogadores de xadrez dependem de computadores para analisar posições complexas, explorar variantes e estudar suas ideias. A fusão entre a intuição humana e o poder computacional proporciona uma abordagem altamente eficaz para o desenvolvimento das estratégias e técnicas mais avançadas no xadrez competitivo. Assim, a jornada de aprimoramento no xadrez continua, alimentada pela interação dinâmica entre a mente humana e a inteligência artificial.

As engines de xadrez são ferramentas que ajudam muito na preparação e treinamento de jogadores profissionais. Essas ferramentas são capazes de analisar posições complexas com muita profundidade e precisão. No entanto, compreender essas análises e utilizá-las para melhorar a capacidade de jogar xadrez não é uma tarefa fácil, confiar cegamente nos seus veredictos sem entender o porquê por trás de cada jogada não é nada produtivo, é importante reconhecer que sua interpretação exige tempo, paciência e uma compreensão sólida dos princípios do jogo.

O problema abordado neste trabalho é a dificuldade de compreensão das decisões tomadas pelas engines de xadrez.

Para que o jogador que utiliza essas engines com o intuito de aprimorar seu entendimento sobre o jogo possa realmente aprender, é necessário compreender todas as nuances das posições que influenciam as decisões da máquina. Caso contrário, o jogador não conseguirá aprender, ocorrendo uma retenção de conhecimento dentro daquela "caixa preta".

A motivação e justificativa deste estudo reside na busca pela inovação no contexto da análise enxadrística, visando proporcionar não apenas uma compreensão superficial das avaliações das posições, mas sim uma compreensão mais profunda de forma acessível para jogadores de todos os níveis. Pretende-se, assim, fornecer um arcabouço teórico para pesquisas futuras, e também, estabelecer diretrizes práticas para aprimorar a forma como as engines são abordadas e consequentemente facilitar o estudo de xadrez por intermédio delas. Em termos práticos, a pesquisa visa auxiliar os enxadristas a compreenderem melhor as recomendações das engines, oferecendo explicações sobre como as decisões tomadas por elas.

A Inteligência Artificial Explicável (XAI, do inglês Explainable Artificial Intelligence) é uma subárea da inteligência artificial que se dedica a tornar as operações internas dos modelos de IA mais transparentes e compreensíveis para os seres humanos. À medida que os algoritmos de IA se tornam mais sofisticados e amplamente utilizados em diversas aplicações, desde diagnósticos médicos até sistemas financeiros @@@citar, a necessidade de entender como essas máquinas chegam às suas conclusões torna-se cada vez mais importante. A XAI aborda essa necessidade desenvolvendo métodos e técnicas que fornecem explicações claras e detalhadas sobre o funcionamento interno dos modelos de IA.

O objetivo deste trabalho é tornar mais acessível a compreensão da avaliação de posições no xadrez, realizada por engines, por meio do uso da inteligência artificial explicável, evidenciando de forma clara o impacto de nuances do tabuleiro do xadrez na avaliação dos lances sugeridos pelas engines.

Para atingir o objetivo é necessário desenvolver um modelo de avaliação estática que funcione como uma aproximação da função de avaliação do Stockfish para Aplicar técnicas de XAI ao modelo de avaliação estática com o intuito de analisar e interpretar as avaliações de posições no xadrez realizadas pelo modelo. Além disso, investigar e identificar os contextos em que as explicações das análises estáticas são proveitosas ou não, buscando compreender em que situações a interpretação das avaliações contribui para uma compreensão mais profunda dos fatores posicionais envolvidos.

A hipótese dessa pesquisa é que o uso de XAI poderá propiciar a compreensão das decisões e avaliações de engines de xadrez, permitindo que jogadores entendam melhor as razões por trás das análises e linhas sugeridas, compreendendo a leitura posicional que a máquina tem sobre o tabuleiro de forma mais clara.

Para a metodologia do trabalho, foram selecionadas 6 técnicas de XAI para serem aplicadas à avaliações estáticas em análises feitas pelo stockfish, a fim de extrair características do tabuleiro que contribuíram para a escolha da linha por parte do stockfish.

Antes da aplicação das técnicas de XAI, visando entender a importância da função de avaliação estática e da profundidade de busca na análise das engines, foi feita uma

seleção de posições com diferentes temáticas para realizar confrontos entre engines clássicas e de redes neurais, limitando a profundidade máxima de profundidade dos cálculos. Quando retiramos o fator da força bruta no contexto da análise das engines no tabuleiro, o poder de avaliação da posição como um todo e o entendimento das dinâmicas apresentadas nas situações de jogo ficam mais evidentes. Daí podemos extrair entender as diferenças no comportamento de uma engine com uma função de avaliação clássica e outra baseada em redes neurais, como a Rede Neural Eficientemente Atualizável (NNUE do inglês Efficiently updatable neural network).

É essencial reconhecer que o xadrez é um jogo dinâmico e, em várias situações, a avaliação completa de uma posição vai além de uma análise meramente posicional. Em casos onde existe uma sequência tática inevitável, como uma ameaça tática que resulta em ganho material substancial, essa sequência prevalecerá sobre a avaliação posicional, tornando a explicação das sequências de avaliações estáticas irrelevante naquele contexto. A avaliação completa de uma posição, portanto, combina uma análise estática com uma busca aprofundada, que simula o desenrolar das jogadas possíveis para capturar vantagens ou desvantagens materiais ou posicionais.

Com base no resultado dos duelos, foi possível observar situações onde a decisão do stockfish se deve mais à análise estática do que à profundidade, destacando momentos em que o uso de técnicas de XAI pode ser particularmente proveitoso. Assim, fundamentando a aplicação das técnicas de XAI, em posições se beneficiam de explicações das avaliações estáticas.

[FALAR BREVEMENTE SOBRE OS RESULTADOS OBTIDOS NA UTILIZAÇÃO DAS TÉCNICAS]

A estrutura da presente monografia foi planejada para fornecer uma análise abrangente e detalhada do problema de explicabilidade em engines de xadrez, abordando tanto a teoria quanto a prática envolvidas na pesquisa. A partir da fundamentação teórica, a monografia é organizada em várias seções, cada uma contribuindo para a construção do conhecimento necessário para compreender e resolver o problema identificado.

A Fundamentação Teórica é a primeira grande seção do trabalho e está dividida em três partes principais. A primeira parte oferece uma fundamenos sobre o xadrez, abordando sua história, regras básicas e estratégias comuns. A segunda parte da fundamentação teórica concentra-se no xadrez computacional, explorando a evolução das engines de xadrez desde os primeiros motores baseados em regras simples até as modernas engines que utilizam algoritmos de aprendizado profundo e redes neurais. São discutidos aspectos técnicos e funcionais dessas engines, incluindo como elas avaliam posições e tomam decisões. A terceira parte aborda a XAI, são apresentadas as principais técnicas de explicabilidade, seus princípios e aplicações, com ênfase naquelas que foram consideradas mais promissoras para aplicação no contexto das engines de xadrez.

A Metodologia é a próxima seção da monografia e descreve detalhadamente como o trabalho foi desenvolvido. Esta seção cobre a seleção e curadoria de posições temáticas de xadrez, a preparação do ambiente de desenvolvimento e a execução dos duelos entre engines com e sem redes neurais, e a coleta e catalogação dos dados das partidas geradas. Além disso, a metodologia aborda a aplicação das técnicas de explicabilidade selecionadas, descrevendo como serão conduzidas as análises comparativas e interpretativas das decisões das engines. Por fim, a metodologia também apresenta os resultados preliminares, onde são discutidos os achados iniciais do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 O Xadrez

O xadrez é um jogo de tabuleiro estratégico disputado entre dois jogadores. Cada jogador controla um exército de 16 peças, que incluem um rei, uma rainha, duas torres, dois bispos, dois cavalos e oito peões. O tabuleiro é composto por 64 casas, dispostas em um padrão de 8x8, alternando entre cores claras e escuras, as casas são nomeadas com a letra da coluna de “a” a “h” e o número da linha de 1 a 8. A disposição inicial das peças é sempre a mesma: as peças pesadas (torres, cavalos, bispos, rainha e rei) são colocadas na primeira linha, enquanto os peões ocupam a segunda linha.

O jogo se desenvolve para uma resolução quando as peças são estrategicamente movimentadas, ocorrendo capturas, quando uma peça do jogador move-se para a casa ocupada por uma peça do adversário, removendo-a do tabuleiro. Por exemplo, se um peão branco se move para a posição ocupada por um cavalo preto, o cavalo preto é capturado e retirado do tabuleiro. A única peça que não pode ser capturada é o rei, e o objetivo do jogo é colocar o rei adversário em uma posição onde ele esteja em xeque-mate, ou seja, em um ataque do qual não pode escapar.

2.1.1 Movimentação das peças

No xadrez, o rei é a peça mais crucial e seu movimento é limitado a uma casa em qualquer direção: vertical, horizontal ou diagonal. Existem regras especiais para o rei, como o roque, que é uma jogada estratégica onde ele troca de posição com uma das torres. Isso só pode ocorrer se nenhuma das peças envolvidas tiver sido movida anteriormente, se não houver peças entre elas e se o rei não estiver em xeque, nem puder passar por uma casa atacada durante o roque. A principal proibição que o rei enfrenta é não poder se mover para uma casa onde ele seria colocado em xeque. Durante o início e meio do jogo, o rei deve ser protegido, frequentemente através do roque para uma posição segura. No final do jogo, o rei torna-se uma peça ativa, ajudando a capturar peças adversárias e promovendo peões.

As torres, que movem-se em linha reta na horizontal ou vertical por qualquer número de casas, também são peças poderosas, especialmente em linhas abertas e colunas. Elas exercem pressão sobre peças adversárias e controlam grandes áreas do tabuleiro, sendo particularmente poderosas no final do jogo. Nesse estágio, as torres são cruciais para apoiar a promoção de peões e criar ameaças de mate. Além disso, as torres participam do roque com o rei, uma jogada importante para a segurança do rei e a mobilidade da torre.

Os bispos movem-se qualquer número de casas em uma direção diagonal e cada um deles permanece na cor de casa em que começou (clara ou escura) ao longo do jogo. Os bispos são mais eficazes em diagonais longas e abertas, onde podem controlar múltiplas casas de uma só vez. Eles são ideais para criar ameaças a longa distância e controlar pontos-chave no tabuleiro. Quando em pares, os bispos podem cobrir tanto as diagonais claras quanto as escuras, formando uma poderosa força de ataque e defesa.

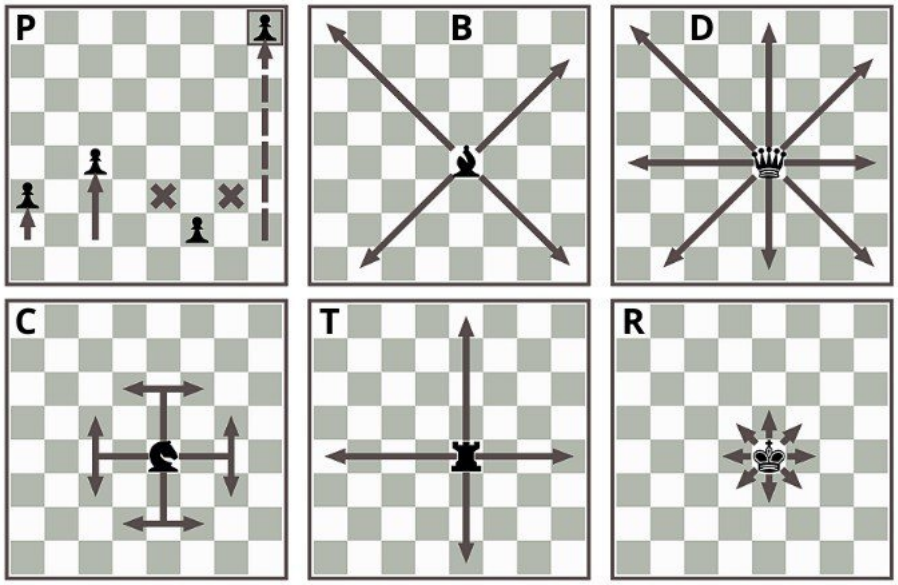
Os cavalos movem-se em um padrão de "L", o que significa que se movem duas casas em uma direção (horizontal ou vertical) e uma casa em uma direção perpendicular, ou vice-versa. O cavalo é a única peça que pode saltar sobre outras peças, o que lhe dá uma vantagem única no jogo. Os cavalos são ideais para atacar peças adversárias em posições onde outras peças não podem alcançá-las facilmente. Eles são especialmente poderosos em posições fechadas e complexas, onde sua habilidade de saltar sobre peças pode criar ameaças inesperadas. Os cavalos são eficazes em controlar casas centrais e em criar ataques duplos.

A rainha, sendo a peça mais poderosa do tabuleiro, combina os movimentos da torre e do bispo, podendo mover-se qualquer número de casas em linha reta na horizontal, vertical ou diagonal. Essa versatilidade permite que a rainha controle uma vasta área do tabuleiro, tornando-a eficaz para ataques diretos ao rei adversário, capturas de peças importantes e controle estratégico de linhas e diagonais. No meio-jogo, sua mobilidade cria múltiplas ameaças simultâneas, facilitando a pressão sobre o adversário.

Os peões são talvez as peças mais subestimadas no xadrez, mas sua importância não deve ser minimizada. Eles movem-se apenas para frente, uma casa por vez, mas na sua jogada inicial, podem mover-se duas casas. A captura pelos peões é feita movendo-se uma casa na diagonal.

Existem várias regras especiais que envolvem os peões, como a promoção, que ocorre quando um peão atinge a última linha do tabuleiro do adversário e pode ser promovido a qualquer outra peça, exceto um rei. Outra regra especial é a captura en passant, que significa "de passagem"em francês, essa regra permite a um peão capturar outro peão que tenha avançado duas casas na sua jogada inicial e que esteja posicionado ao lado do peão que executará a captura; essa captura só pode ser feita imediatamente no lance seguinte após o avanço do peão adversário. Os peões são fundamentais para a estrutura do jogo. Eles controlam o centro, protegem peças mais valiosas e são valiosos no final do jogo devido à sua capacidade de promoção. Os peões são mais eficazes quando avançados em conjunto, formando cadeias de peões difíceis de penetrar pelo adversário.

Figura 1 – Representação da movimentação das peças do xadrez, identificadas pelas suas iniciais.



Fonte: Significados, 2002

2.1.2 Fases do jogo

O xadrez é tradicionalmente dividido em três fases principais: abertura, meio-jogo e final. Cada fase possui suas próprias estratégias e táticas, que são fundamentais para o desenvolvimento da partida. A abertura é a fase inicial do jogo, que se concentra no desenvolvimento das peças e no controle do centro do tabuleiro. Durante a abertura, os jogadores buscam posicionar suas peças de maneira eficiente, preparando-se para a transição ao meio-jogo. A abertura é crítica, pois uma boa posição inicial pode proporcionar vantagens táticas e estratégicas ao longo da partida. Movimentos como o avanço dos peões centrais (e4, e5, d4, d5) e o desenvolvimento de cavalos e bispos são comuns nesta fase.

O meio-jogo começa quando as peças estão desenvolvidas e o controle do centro está estabelecido. Nesta fase, os jogadores buscam executar táticas como ataques diretos ao rei adversário, criação de fraquezas na posição do oponente e captura de peças valiosas. Estratégias como a criação de uma estrutura de peões sólida e a coordenação entre peças são essenciais para o sucesso. O meio-jogo é muitas vezes o palco das batalhas mais intensas, onde erros táticos podem levar a consequências significativas. O final do jogo ocorre quando restam poucas peças no tabuleiro, e o foco se desloca para a promoção dos peões e a captura do rei adversário. A habilidade de coordenar as peças restantes e utilizar a força do rei é crucial.

No final, os peões ganham maior importância, pois sua promoção pode decidir o resultado da partida. Estratégias como a oposição de reis e a criação de peões passados são fundamentais para a vitória nesta fase.

Uma partida de xadrez pode terminar em vitória para um lado e consequentemente

derrota para o outro, ou empate entre os dois jogadores. Começando pelas condições de vitória, existem três formas de terminar o jogo, o xeque-mate, o abandono e o fim do tempo.

O xeque-mate é uma posição onde o rei adversário está sob ataque e não possui movimentos legais para sair do xeque. Um xeque-mate resulta em vitória para o jogador que o realiza.

O abandono, um jogador pode desistir do jogo a qualquer momento, resultando em vitória para o oponente. Isso geralmente ocorre quando o jogador percebe que está em uma posição irremediavelmente desfavorável.

O tempo, no xadrez competitivo, os jogadores têm um tempo limitado para completar seus movimentos. Se o tempo de um jogador se esgotar, ele perde a partida, a menos que o adversário não tenha material suficiente para dar xeque-mate.

O jogo pode terminar empatado por afogamento, repetição de posição, pela regra dos 50 movimentos, por insuficiência de material ou por acordo mútuo entre os jogadores.

O afogamento ocorre quando um jogador não possui lances legais para fazer e ele não está em xeque, dessa forma o jogo não pode continuar, resultando em empate. Se a mesma posição ocorre três vezes no tabuleiro, com os mesmos movimentos possíveis, o jogo pode ser declarado empatado por repetição de posição.

Se durante 50 movimentos consecutivos de ambos os jogadores nenhuma peça for capturada e nenhum peão for movido, qualquer jogador pode solicitar um empate.Se nenhum dos jogadores tiver material suficiente para dar xeque-mate, o jogo é declarado empatado. Por exemplo, se restam apenas reis no tabuleiro, não é possível dar xeque-mate.Por fim, no empate por acordo mútuo, os jogadores podem concordar em empatar a partida a qualquer momento.

2.1.3 Princípios estratégicos

O xadrez, um dos jogos de estratégia mais complexos e desafiadores, baseia-se em princípios fundamentais que orientam os jogadores em sua busca pela vitória. Entre os principais princípios estratégicos, destacam-se o controle do centro, a atividade das peças e a segurança do rei. Cada um desses aspectos desempenha um papel crucial na construção de uma partida bem-sucedida, exigindo atenção e habilidade tanto dos iniciantes quanto dos enxadristas experientes.

O controle do centro é um dos conceitos mais importantes e primordiais no xadrez. As casas centrais do tabuleiro (d4, d5, e4, e5) são posições estratégicas que permitem o máximo de mobilidade para as peças. Dominar essas casas possibilita um melhor desenvolvimento das peças, além de restringir os movimentos do adversário. Ao controlar

o centro, o jogador assegura uma base sólida para suas operações ofensivas e defensivas, facilitando a coordenação das peças e a criação de oportunidades táticas. Assim, o controle central é frequentemente estabelecido nas primeiras jogadas, com peões avançando para o centro e peças menores sendo desenvolvidas para apoiar essa configuração.

A atividade das peças é outro princípio estratégico essencial. Peças ativas são aquelas que ocupam posições onde têm o maior impacto no jogo, seja ameaçando peças adversárias, controlando espaços críticos ou preparando ataques. Uma peça ativa exerce mais influência sobre o tabuleiro do que uma peça passiva, que está restrita em seus movimentos e alcance. Portanto, jogadores habilidosos procuram desenvolver suas peças rapidamente, colocando-as em posições onde possam contribuir efetivamente para o plano estratégico geral. A movimentação ativa de peças como bispos, cavalos, torres e a dama pode desequilibrar a posição do adversário e criar oportunidades para ganhos materiais ou vantagens posicionais decisivas.

A segurança do rei é o terceiro pilar da estratégia no xadrez. Manter o rei protegido é vital, pois sua captura resulta na perda imediata do jogo. O roque é uma das primeiras medidas que os jogadores tomam para garantir a segurança do rei, movendo-o para um canto do tabuleiro e colocando-o atrás de uma fileira de peões. Além do roque, é crucial evitar debilitar a estrutura de peões ao redor do rei, pois buracos ou fraquezas podem ser explorados pelo adversário para lançar um ataque mortal. A segurança do rei envolve tanto medidas preventivas quanto reativas, como fortalecer a defesa ao redor dele e estar preparado para reagir a ataques diretos.

[CITAR]

2.2 O Xadrez Computacional

O xadrez computacional refere-se ao uso de computadores para jogar e analisar o jogo de xadrez. Desde os primeiros experimentos com inteligência artificial nos anos 1950, o campo evoluiu significativamente, resultando em programas capazes de derrotar os melhores jogadores humanos.

As engines de xadrez mais tradicionais, utilizam funções de avaliação complexas e algoritmos de busca inteligentes para encontrar o melhor movimento possível. (Rodríguez, 2022) A potência dessas engines também está diretamente relacionada à quantidade e capacidade de processamento das unidades centrais de processamento (CPUs) disponíveis no dispositivo em que estão sendo executadas. Quanto mais poderosas e numerosas as CPUs, mais forte se torna a engine.

Um dos componentes fundamentais dessas engines é o gerador de movimentos, cuja implementação depende fortemente da representação do tabuleiro de xadrez. Existem dois tipos de geração de movimentos: a geração de movimentos pseudo-legais e a geração de

movimentos legais. Na geração de movimentos pseudo-legais, as peças obedecem às regras normais de movimento, mas o movimento não é verificado antecipadamente para garantir, por exemplo, que não deixará o rei em xeque. Já na geração de movimentos legais, apenas os movimentos legais são criados, o que pode tornar o processo mais demorado devido à necessidade de verificar se o rei não será deixado em xeque após o movimento.

Outro módulo fundamental de uma engine é a função de avaliação, e a principal melhoria nas engines de xadrez modernas geralmente está relacionada à alteração dessa função. (Rodríguez, 2022)

A função de avaliação recebe como entrada uma posição de xadrez e produz como saída um número. Se esse número resulta em uma avaliação de 0, significa que a posição é igual para ambos os jogadores. Quanto maior o número positivo, mais vantagem para as brancas; quanto menor o número negativo, mais vantagem para as pretas. A função de avaliação considera diversos dados para gerar essa avaliação, que são oriundos de teorias já estabelecidas pelos seres humanos, os quais são codificados para simular o pensamento de um grande mestre do xadrez.

As engines mais modernas, não utilizam mais a força bruta na busca por soluções no xadrez, essa abordagem se torna impraticável devido ao número enorme de possíveis jogos que teriam que ser avaliados, estimado em pelo menos 10^{120}, conhecido hoje como o "número de Shannon". Mesmo se cada átomo no universo fosse capaz de calcular uma única avaliação estática a cada nanossegundo desde o nascimento do universo, ainda assim estaríamos a mais de 20 ordens de magnitude de resolver o xadrez. Portanto, qualquer tentativa de resolver o xadrez usando somente força bruta é considerada fútil.(Klosowski, 2019)

A impossibilidade prática de avaliar todas as possibilidades de jogadas destaca a necessidade de abordagens mais inteligentes e eficientes na construção de engines de xadrez.

Segundo (Vrzina, 2023) o mini-max é a ideia central por trás da função de busca em uma engine de xadrez, este algoritmo procura determinar a melhor jogada, assumindo que o oponente também sempre fará a melhor jogada possível. Para isso, o mini-max explora todas as jogadas possíveis até uma certa profundidade. Nas folhas da árvore de busca, a posição é avaliada por uma função de avaliação. Cada posição é associada a uma jogada que a criou, permitindo que o jogador amigável escolha uma jogada que maximize sua pontuação de posição, enquanto o oponente busca minimizar essa pontuação.

O algoritmo alpha-beta é uma técnica de busca em profundidade que poda ramos não promissores da árvore de busca mais cedo, melhorando assim a eficiência da busca. Cada posição no jogo é considerada como a raiz da árvore de busca, e os movimentos legais para cada lado criam os nós da próxima camada. Quanto mais tempo disponível,

mais profunda a árvore de busca pode ser processada, levando a uma melhor qualidade geral do jogo. Nas folhas da árvore de busca, uma função de avaliação é aplicada para determinar a qualidade das posições. (David; Netanyahu; Wolf, 2016)

A poda, ou prune é utilizada para melhorar o desempenho desses algoritmos, priorizando a avaliação das jogadas mais favoráveis primeiro. Quando uma jogada é mostrada como resultando em uma avaliação pior do que uma jogada anteriormente examinada, o prune é aplicado, evitando a exploração adicional das ramificações da árvore de busca. (Brange, 2021)

Devido à complexidade do xadrez, ainda não foi possível resolver completamente o xadrez, no entanto, os pesquisadores conseguiram resolver problemas de finais de jogo com um número reduzido de peças. Essas soluções incluem informações de jogo perfeito, como o resultado ótimo do minimax, o número de movimentos para vencer ou perder, para todos os estados legais do jogo contendo de 3 a 7 peças no tabuleiro. Esses resultados são armazenados em bancos de dados chamados tablebases de finais de jogo. As tablebases fornecem informações precisas sobre finais de jogo, permitindo que as engines de xadrez tomem decisões mais informadas em situações específicas. Além disso, o artigo destaca a complexidade envolvida na geração e no armazenamento dessas tablebases, incluindo o desafio de equilibrar compactação e velocidade de consulta. (Haque; Wei; Müller, 2022)

Outro sistema muito citado na literatura é o algoritmo de busca Monte Carlo Tree Search. o MCTS é uma abordagem para resolver problemas de busca computacionalmente intensivos, como o xadrez. No contexto computacional, "Monte Carlo"significa que algo arbitrário acontece. Em xadrez, um módulo que utiliza MCTS puro avaliará a posição gerando uma sequência de movimentos diferentes a partir da posição dada de maneira arbitrária e calculando a média dos resultados finais (vitória/empate/derrota) que gera. (Rodríguez, 2022)

O (David; Netanyahu; Wolf, 2016) explora o uso de redes neurais em engines de xadrez, destacando um método inovador que utiliza redes neurais profundas para aprender uma função de avaliação a partir do zero, sem incorporar as regras do jogo ou utilizar características manualmente extraídas. Em vez disso, o sistema é treinado de ponta a ponta em um grande conjunto de dados de posições de xadrez.

Em (Lemley *et al.*, 2018) vemos a abordagem da prática de engines de xadrez jogarem contra si mesmas para treinar e melhorar seu desempenho, destacando o exemplo notável do Alpha Zero. O Alpha Zero, utilizando uma Rede Neural Recorrente (RNN), aprendeu a jogar xadrez sem conhecimento prévio e aprimorou suas habilidades após cada partida por meio de autojogo, conseguindo superar engines de xadrez que utilizavam abordagens mais clássicas. A prática de “autojogo” permite que a engine explore uma ampla variedade de situações e estratégias, adaptando-se e melhorando continuamente seu desempenho ao enfrentar seus próprios movimentos. Isso não só demonstra o poder

das técnicas de aprendizado de máquina no contexto do xadrez, mas também destaca a importância do "autojogo" como uma abordagem eficaz para o treinamento de engines de xadrez de alto nível.

O Stockfish é uma das mais poderosas engines de xadrez de código aberto atualmente disponíveis. Em 8 das 10 últimas competições TCEC, o Stockfish saiu vitorioso, evidenciando sua superioridade. Uma das características fundamentais do funcionamento do Stockfish é o uso de uma busca alfa-beta para encontrar o melhor movimento possível em uma posição de jogo. Além disso, o artigo explica que o Stockfish avalia cada posição do jogo atribuindo um único valor a ela. Inicialmente, essa avaliação era baseada em heurísticas desenvolvidas por especialistas em xadrez, utilizando características pré-definidas manualmente. (Haque; Wei; Müller, 2022)

O artigo (Rodríguez, 2022) fornece uma visão detalhada de como funciona o Leela Chess Zero (Lc0), uma adaptação do Leela Zero específica para o xadrez. Ele explica que o Lc0 é um esforço distribuído para reproduzir os resultados do AlphaZero no xadrez. Assim como o AlphaZero, o Lc0 utiliza uma arquitetura de rede neural de duas cabeças (política e valor) e o mesmo algoritmo de busca. Surpreendentemente, em 2020, o Lc0 superou a força de jogo publicada do AlphaZero no xadrez. O artigo destaca adições como a cabeça de saídas auxiliares para prever o número de movimentos restantes no jogo e a probabilidade de vitória, empate ou derrota. Além disso, a descrição das diferentes redes geradas pelo Lc0 e os métodos de treinamento utilizados são fundamentais para entender como o Lc0 alcançou seu desempenho de jogo. O artigo também explora o funcionamento do AlphaZero, uma versão mais genérica do AlphaGo Zero que alcançou níveis superhumanos de desempenho em jogos como xadrez e shogi, além de Go. Diferentemente do AlphaGo, o AlphaZero inicia seu aprendizado a partir de jogadas aleatórias, sem nenhum conhecimento humano, por meio de aprendizado por reforço em autojogo. Ele simplifica e melhora o AlphaGo ao usar apenas a posição do tabuleiro como entrada e treinar uma única rede com duas saídas para política e valor. O AlphaZero não leva em consideração a simetria, uma vez que não é aplicável ao xadrez e ao shogi. Além disso, ao contrário do AlphaGo Zero, que esperava a conclusão de uma iteração para avaliar contra a melhor rede anterior, o AlphaZero mantém apenas uma rede que é atualizada continuamente.

O artigo (Vrzina, 2023) discute a incorporação do Efficiently Updatable Neural Network (NNUE) pelo Stockfish. O NNUE permitiu que o Stockfish e outros motores de xadrez alcançassem melhorias significativas em seu desempenho, como evidenciado por um salto de 100 pontos de Elo no caso específico do Stockfish. Essa melhoria foi a mais significativa experimentada pelo Stockfish nos últimos 7 anos. A capacidade do NNUE de aprender e incorporar dados de forma eficiente permitiu um avanço substancial no desempenho do Stockfish, demonstrando como o aprendizado de máquina pode impulsionar o desenvolvimento de engines de xadrez mais poderosas. Compreender a implementação

do NNUE no Stockfish é essencial para acompanhar os avanços tecnológicos no campo das engines de xadrez e pode orientar futuras pesquisas e desenvolvimentos para melhorar ainda mais o desempenho dessas engines.

2.2.1 NNUE, (Efficiently Updatable Neural Networks)

A NNUE foi inicialmente desenvolvida para o Shogi por Yu Nasu, integrada ao YaneuraOu por Motohiro Isozaki em maio de 2018, e posteriormente adaptada para o xadrez no motor Stockfish por Hisayori Nodai em junho de 2019 . A seguir, apresentamos uma visão geral dos princípios fundamentais que sustentam essa arquitetura neural e sua aplicabilidade. (disservin, 2024b)

Um dos pilares da NNUE é a minimização das entradas não-zero na rede neural. Isso significa que, à medida que a rede é escalada em tamanho, as entradas devem se tornar esparsas. Arquiteturas avançadas atuais apresentam uma esparsidade de entrada na ordem de 0,1%. Essa característica permite que a rede, mesmo sendo grande, possa ser avaliada de forma rápida e eficiente. A limitação do número de entradas não-zero é crucial para garantir um tempo de avaliação baixo em situações onde a rede precisa ser avaliada em sua totalidade. (disservin, 2024b)

Outro princípio importante é que as entradas devem mudar o mínimo possível entre avaliações subsequentes. No contexto de jogos de tabuleiro, isso significa que uma única jogada altera o estado do tabuleiro de maneira mínima. Esse princípio permite a atualização eficiente da rede, evitando a necessidade de reavaliá-la completamente a cada jogada. Embora não seja obrigatório para todas as implementações, esse conceito oferece uma melhoria mensurável na eficiência das atualizações da rede. (disservin, 2024b)

O terceiro princípio destaca a necessidade de uma rede simples o suficiente para facilitar a inferência de baixa precisão no domínio dos inteiros. Isso é essencial para alcançar um desempenho máximo em hardware comum, tornando o modelo especialmente adequado para inferência de baixa latência em CPUs. Este princípio é particularmente relevante para motores de xadrez convencionais, que exigem avaliações rápidas e frequentes. (disservin, 2024b)

A quantização, o processo de converter a representação do modelo de ponto flutuante para inteiro, é um aspecto fundamental para a eficiência das redes NNUE. Redes NNUE são projetadas para serem avaliadas rapidamente no domínio de inteiros de baixa precisão, utilizando ao máximo o desempenho disponível de CPUs modernas. A quantização introduz um erro que se acumula mais à medida que a profundidade da rede aumenta. No entanto, no caso das redes NNUE, que são relativamente rasas, esse erro é negligenciável. A quantização é crucial para alcançar avaliações de milhões de posições por segundo por thread, um uso extremo que requer soluções igualmente extremas. (disservin, 2024b)

Embora a NNUE tenha sido inicialmente aplicada a jogos de tabuleiro como Shogi e xadrez, seus princípios podem ser extrapolados para outros domínios onde atualizações frequentes e rápidas são necessárias. A capacidade de atualizar eficientemente apenas as partes alteradas da rede, em vez de reevaluar completamente, oferece uma vantagem significativa em termos de desempenho e eficiência. (disservin, 2024b)

A arquitetura NNUE destaca-se pela combinação de simplicidade, eficiência e capacidade de avaliação rápida, sendo particularmente valiosa em cenários que exigem avaliações constantes e rápidas, como em motores de jogos de tabuleiro. A aplicação dos princípios de minimização de entradas, atualização eficiente e inferência de baixa precisão posiciona a NNUE como uma solução robusta para desafios que requerem alta performance em tempo real. (disservin, 2024b)

2.3 Inteligência artificial explicável

A explicabilidade é crucial em áreas onde as decisões da IA têm impactos significativos, como medicina, finanças e justiça. Modelos explicáveis permitem aos usuários entender, confiar e agir com base nas recomendações da IA.

A crescente complexidade dos modelos de inteligência artificial (IA) trouxe consigo um desafio significativo: a necessidade de entender e interpretar como essas ferramentas tomam suas decisões. Em um mundo onde a IA é cada vez mais integrada a processos críticos de negócios, saúde, finanças e segurança, a capacidade de confiar nos sistemas de IA se tornou essencial. A XAI, surge como uma resposta a essa necessidade, fornecendo métodos e técnicas para tornar as operações dos modelos de IA mais transparentes e compreensíveis.

A importância da XAI vai além de simples curiosidade ou conveniência; Ela é crucial para garantir que os sistemas de IA sejam justos, éticos e responsabilizáveis. Em contextos onde as decisões têm impactos significativos, como diagnósticos médicos, concessão de créditos e segurança pública, compreender os fundamentos das decisões tomadas por modelos de IA não é apenas desejável, mas muitas vezes uma exigência legal e ética. A XAI permite que desenvolvedores e usuários identifiquem possíveis vieses nos modelos, compreendam as razões por trás das previsões e decisões, e ajustem os sistemas para melhorar seu desempenho e alinhamento com valores humanos. (Burkart; Huber, 2021)

Além disso, a transparência proporcionada pela XAI aumenta a confiança do público na utilização de tecnologias de IA, o que é fundamental para a sua aceitação e adoção em larga escala. Sem essa confiança, o potencial transformador da IA pode ser limitado pela relutância em confiar em sistemas que funcionam como "caixas-pretas". Portanto, a XAI não só promove um melhor entendimento e controle sobre as tecnologias de IA, mas também é um componente essencial para a evolução ética e responsável dessa área.

(Burkart; Huber, 2021)

Neste contexto, a XAI se destaca como uma área vital de pesquisa e desenvolvimento, buscando maneiras de desmistificar as operações internas dos modelos de IA e proporcionar explicações claras e úteis sobre como e por que as decisões são tomadas. Ao mergulhar nas técnicas e métodos da XAI, podemos não apenas aprimorar a tecnologia, mas também garantir que seu impacto na sociedade seja positivo, responsável e transparente.

No campo da Inteligência Artificial (IA), um dos desafios persistentes é a confusão criada pelo uso intercambiável dos termos "interpretabilidade"e "explicabilidade". Embora frequentemente considerados sinônimos, esses conceitos apresentam diferenças significativas e desempenham papéis distintos no entendimento e na interação com modelos de IA.

Interpretabilidade refere-se à capacidade de um modelo ser entendido de forma clara e direta por um observador humano, sem a necessidade de uma explicação adicional sobre seus mecanismos internos. Esta característica é intrínseca ao modelo e pode ser considerada uma forma de transparência. Um modelo interpretável permite que um ser humano compreenda a lógica subjacente a suas decisões apenas observando suas operações e resultados. É uma qualidade passiva, pois não requer esforços adicionais para gerar explicações detalhadas. Em essência, um modelo interpretável é intuitivo e suas decisões são autoexplicativas. (Burkart; Huber, 2021)

A interpretabilidade está associada à transparência imediata do modelo, permitindo que os usuários entendam como e por que as decisões são tomadas sem a necessidade de ferramentas ou técnicas complexas para elucidar os processos internos.

Modelos lineares ou de árvore de decisão são exemplos de modelos interpretáveis, onde a lógica de tomada de decisão pode ser seguida de maneira clara e direta por um observador humano.

Em setores críticos, como saúde ou finanças, a interpretabilidade é vital para que os especialistas confiem nos modelos e possam justificá-los facilmente para partes interessadas ou reguladores.

Explicabilidade, por outro lado, refere-se à capacidade de um modelo gerar explicações sobre seu funcionamento interno e as razões por trás de suas decisões. É uma característica ativa que envolve a criação de mecanismos ou ferramentas que ajudem a esclarecer como o modelo funciona e por que tomou determinadas decisões ou previsões. A explicabilidade fornece uma interface entre o modelo e os seres humanos, facilitando uma compreensão mais aprofundada das operações internas do modelo. (Burkart; Huber, 2021)

Explicabilidade é essencial para modelos mais complexos, como redes neurais profundas, onde a lógica de decisão não é imediatamente aparente e requer técnicas especializadas para ser elucidada.

Métodos como a análise de importância das características, técnicas de backpropagation para a visualização de redes neurais, ou a geração de explicações pós-hoc (como LIME e SHAP), são usados para tornar compreensível o comportamento dos modelos.

Em áreas como diagnóstico médico ou sistemas de recomendação, a explicabilidade permite que os usuários entendam as decisões complexas tomadas por modelos de IA, mesmo quando a lógica não é imediatamente aparente.

Enquanto a interpretabilidade é uma característica passiva que se refere à clareza inerente do modelo, a explicabilidade é uma característica ativa que envolve um esforço deliberado para gerar entendimento através de explicações adicionais.

Modelos interpretáveis permitem um entendimento direto e imediato, enquanto modelos explicáveis exigem ferramentas adicionais para fornecer contexto e detalhamento sobre as decisões tomadas.

A interpretabilidade é geralmente associada a modelos mais simples e transparentes, como regressões lineares ou árvores de decisão. Já a explicabilidade é crucial para modelos complexos, como redes neurais ou modelos de aprendizado profundo, onde a lógica de decisão não é intuitivamente clara. (Burkart; Huber, 2021)

A interpretabilidade é importante para garantir confiança imediata e compreensão fácil em aplicações onde a transparência é crucial. A explicabilidade, por sua vez, é fundamental para permitir que os usuários entendam e confiem em modelos complexos, fornecendo insights detalhados sobre como as decisões são feitas e justificadas.

Interpretabilidade é frequentemente preferida em setores onde a simplicidade e a transparência são essenciais, como em decisões legais ou em auditorias financeiras, onde as partes interessadas precisam entender rapidamente o raciocínio por trás das decisões.

Explicabilidade, por outro lado, é essencial em áreas onde modelos complexos são usados para tarefas como previsão de falhas em sistemas de engenharia ou na criação de diagnósticos médicos avançados, onde a compreensão detalhada das operações internas do modelo é necessária para garantir confiança e eficácia.

Essas distinções são fundamentais para entender as capacidades e limitações dos modelos de IA e para guiar a escolha da abordagem mais adequada para diferentes contextos e necessidades. Elas também são cruciais para promover uma interação mais ética e responsável com tecnologias avançadas de IA, garantindo que as decisões automatizadas sejam transparentes, justificáveis e compreensíveis para todos os envolvidos.

2.3.1 Técnicas de explicabilidade

Para aprofundar o conhecimento e aplicação dessa área crucial, é fundamental explorar as diversas técnicas de XAI disponíveis. A seguir, é apresentado um conjunto

abrangente de métodos que oferecem diferentes abordagens para tornar os modelos de IA mais transparentes e interpretáveis.

2.3.1.1 Gradientes

Os gradientes formam a base de muitos métodos de atribuição no aprendizado de máquina. Eles calculam o gradiente da saída do modelo em relação a cada recurso de entrada, fornecendo informações sobre como uma pequena alteração em um recurso de entrada afetaria a saída. Esse método se baseia na retropropagação para calcular os gradientes da camada de saída para a rede. Usada principalmente para dados contínuos, especialmente imagens, a atribuição baseada em gradiente requer acesso à arquitetura e aos pesos do modelo, o que a torna menos adequada para modelos black-box. Ao calcular gradientes para cada pixel ou recurso, ele destaca as áreas que mais influenciam a saída. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.2 Gradientes integrados

Com base no conceito de gradientes, os gradientes integrados abordam algumas limitações integrando gradientes ao longo de um caminho desde uma entrada de linha de base até a entrada real. A integral é aproximada pela soma dos gradientes calculados em intervalos entre a linha de base e a entrada. Esse método é adequado para vários tipos de dados, mas é comumente usado em tarefas de classificação de imagens e textos. Ele requer a seleção de uma entrada de linha de base significativa e calcula o gradiente médio da saída ao longo do caminho da linha de base até a entrada, fornecendo uma atribuição mais abrangente do que os gradientes simples. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.3 DeepLIFT (Recursos importantes de aprendizagem profunda)

O DeepLIFT oferece um método de atribuição baseado em comparação que compara a ativação de cada neurônio a uma ativação de referência. Usando uma passagem para trás semelhante à retropropagação, ele compara as ativações com uma referência em vez de calcular gradientes. Esse método é aplicável a uma ampla variedade de tipos de dados, incluindo imagens e genômica, e requer a definição de ativações de referência. Ele envolve um cálculo retroativo complexo para atribuir a contribuição de cada neurônio à saída. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.4 Guided BackPropagation

A retropropagação guiada é uma técnica de visualização que combina a retropropagação com a deconvolução para visualizar quais partes da entrada influenciam mais a

saída.Ela modifica o algoritmo de retropropagação padrão para propagar somente gradientes positivos, aumentando a clareza visual.Usada principalmente para dados de imagem, ela exige a modificação do algoritmo de retropropagação e pode ser implementada em várias arquiteturas de CNN, o que a torna uma ferramenta poderosa para a interpretabilidade visual. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.5 Redes Deconvolucionais (DeconvNets)

As redes deconvolucionais usam a convolução reversa para mapear as ativações de recursos de volta ao espaço de entrada, ajudando a visualizar quais padrões de entrada ativam determinados recursos. Esse método emprega camadas de deconvolução que revertem a operação de convolução, o que o torna particularmente útil para compreender os filtros e recursos aprendidos nas imagens. A construção de redes de deconvolução que correspondam à estrutura original da CNN é computacionalmente intensiva, mas fornece percepções profundas sobre o funcionamento interno do modelo. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.6 Class Activation Maps (CAM)

Os mapas de ativação de classe destacam regiões em uma imagem que são significativas para a classificação, projetando os pesos da camada de saída de volta para os mapas de recursos convolucionais para criar um mapa de calor.Projetado especificamente para dados de imagem, esse método exige a modificação da arquitetura da rede, principalmente das camadas finais, e é limitado a tipos específicos de redes.Ele fornece uma representação visual clara de quais partes de uma imagem contribuem mais para a decisão de classificação. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.7 Grad-CAM

Ampliando os recursos do CAM, o Grad-CAM oferece uma abordagem generalizada que funciona com qualquer rede neural convolucional sem exigir alterações arquitetônicas. Ele calcula os gradientes da pontuação da classe com relação aos mapas de recursos e os agrega para criar um mapa de localização. Adequado para qualquer CNN, independentemente da arquitetura, o Grad-CAM é usado principalmente para imagens e pode ser aplicado a modelos pré-treinados sem a necessidade de alterar a estrutura da rede. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.8 Grad-CAM++

O Grad-CAM++ aprimora o Grad-CAM ao lidar melhor com várias instâncias de um objeto em uma imagem, usando uma combinação ponderada de gradientes positivos para fornecer mapas de ativação de classe mais detalhados. Esse método é particularmente

eficaz para imagens com vários objetos ou instâncias. Ele exige mais esforço computacional em comparação com o Grad-CAM e envolve cálculos complexos de derivados parciais positivos para gerar mapas de localização refinados. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.9 Propagação de relevância em camadas (LRP)

A LRP (Layer-wise Relevance Propagation) distribui pontuações de relevância para cada recurso de entrada propagando a saída do modelo para trás por meio das camadas da rede. Ele utiliza um princípio de conservação em que a relevância de cada camada é distribuída para a camada anterior, mantendo a relevância total em todo o processo. Aplicável a vários tipos de dados, inclusive imagens e texto, o LRP exige a modificação da passagem para trás do modelo e pode ser computacionalmente intensivo devido à necessidade de propagação cuidadosa em cada camada. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.10 SmoothGrad

O SmoothGrad aprimora os métodos baseados em gradiente, calculando a média dos gradientes em várias cópias ruidosas da entrada para produzir mapas de sensibilidade mais claros. Ele adiciona ruído à entrada, calcula os gradientes e calcula a média deles para reduzir o ruído, tornando a interpretação visual dos gradientes mais robusta. Comumente usado para dados de imagem, o SmoothGrad envolve a geração de várias cópias perturbadas da entrada, aumentando o custo computacional, mas melhorando significativamente a clareza e a robustez da visualização do gradiente.(Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.11 RISE

O RISE gera mapas de saliência mascarando partes da entrada e observando as alterações nas previsões do modelo. Ele cria máscaras aleatórias, aplica-as à entrada e calcula uma combinação linear das saídas do modelo para gerar um mapa de saliência. Esse método independente de modelo é eficaz para qualquer tipo de dados e envolve várias passagens para frente com entradas mascaradas, o que o torna computacionalmente caro, mas amplamente aplicável a vários modelos de caixa preta. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.12 Explicações de diagnóstico de modelo interpretável local (LIME)

As explicações de diagnóstico de modelo interpretável local (LIME) explicam as previsões individuais ajustando modelos simples localmente em torno da instância. Ele perturba a instância de entrada, gera uma vizinhança de dados sintéticos e ajusta um

modelo interpretável a esses pontos de dados. Aplicável a qualquer tipo de dados, incluindo dados tabulares, de texto e de imagem, o LIME exige a seleção de parâmetros para a perturbação dos dados e o ajuste de um modelo local, o que pode afetar a estabilidade e a confiabilidade das explicações. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.13 Explicações aditivas de Shapley (SHAP)

As explicações aditivas de Shapley (SHAP) usam a teoria dos jogos para calcular as contribuições dos recursos com base nos valores de Shapley, fornecendo uma medida consistente da importância dos recursos. Ele atribui um valor justo a cada recurso, considerando todas as combinações possíveis de recursos. Aplicável a qualquer tipo de dados, o SHAP fornece explicações locais e globais, embora seja computacionalmente caro, pois exige a avaliação da contribuição de cada recurso em todos os subconjuntos possíveis. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.14 Anchors

As âncoras geram regras "se-então"de alta precisão que explicam as previsões, refinando iterativamente as regras candidatas até que uma âncora suficiente seja encontrada. Essa abordagem de baixo para cima garante que as regras sejam localmente suficientes para a previsão. Adequadas para qualquer tipo de dados e modelo, as âncoras fornecem explicações locais com alta precisão, mas envolvem um processo iterativo desafiador e exigente do ponto de vista computacional. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.15 Método das Explicações Contrastivas (CEM)

O Método de Explicações Contrastivas (CEM) identifica quais recursos precisam estar presentes e quais precisam estar ausentes para uma previsão específica. Ele usa a otimização para encontrar os recursos mínimos e suficientes para a previsão atual e os recursos ausentes que mudariam a previsão. Eficaz para explicar modelos complexos, o CEM destaca os recursos essenciais em vários tipos de dados, mas exige processos de otimização complexos, o que o torna computacionalmente intensivo e difícil de generalizar. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.16 Counterfactual Explanations

As explicações contrafactuais concentram-se na identificação das menores mudanças necessárias para alterar o resultado do modelo. Elas usam técnicas de otimização para encontrar as alterações mínimas de recursos que resultariam em uma previsão diferente, fornecendo insights acionáveis para a tomada de decisões. Adequadas para vários tipos de dados e modelos, as explicações contrafactuais envolvem otimização iterativa, que pode

ser cara do ponto de vista computacional e exigir um ajuste cuidadoso para garantir contrafactuais significativos. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.17 Partial Dependence Plots (PDP)

Os gráficos de dependência parcial (PDP) visualizam como um recurso afeta as previsões do modelo, calculando a média do efeito do recurso em outros recursos. Eles calculam o efeito marginal de um recurso calculando a média das saídas do modelo sobre os valores do recurso. Usados principalmente para dados tabulares com interações limitadas entre características, os PDPs são simples de gerar, mas podem não capturar interações complexas, exigindo uma interpretação cuidadosa. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.18 Individual Conditional Expectation (ICE) Plots

Os gráficos de expectativa condicional individual (ICE) estendem os PDPs mostrando o impacto de um recurso em cada instância individual, traçando a previsão do modelo como uma função de um único recurso enquanto mantém outros recursos constantes para cada instância. Úteis para dados tabulares com possíveis interações de recursos, os gráficos ICE fornecem uma visão mais detalhada dos efeitos dos recursos do que os PDPs, mas podem ser complexos de interpretar para grandes conjuntos de dados. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

2.3.1.19 Accumulated Local Effects (ALE) Plots

Os gráficos de efeitos locais acumulados (ALE) calculam o efeito médio de um recurso enquanto contabilizam dependências com outros recursos. Eles calculam as diferenças médias nas previsões calculando a média condicional de outros recursos, tornando-os mais precisos do que os PDPs para recursos correlacionados. Adequados para dados tabulares com recursos correlacionados, os gráficos ALE exigem uma implementação cuidadosa para levar em conta dependências condicionais e fornecer uma visão diferenciada dos impactos dos recursos. (Linardatos; Papastefanopoulos; Kotsiantis, 2020)

Para aprofundarmos a compreensão da Inteligência Artificial Explicável (XAI), é essencial analisar as diferentes abordagens que podem ser adotadas. Cada técnica oferece uma perspectiva única sobre como garantir a transparência e a interpretabilidade dos modelos de IA.

Os métodos de inteligência artificial explicável preenchem a lacuna entre os modelos de alto desempenho, porém opacos, e a necessidade humana de transparência e responsabilidade. Essa estrutura teórica explora várias dimensões ao longo das quais os métodos de XAI podem ser classificados, fornecendo uma compreensão estruturada de suas funcionalidades e aplicações.

2.3.2 Escopo da interpretação

2.3.2.1 Interpretabilidade local

Os métodos de interpretabilidade local são projetados para explicar as previsões individuais feitas por um modelo. Eles se concentram em entender por que uma decisão específica foi tomada para um determinado exemplo de dados de entrada, o que é particularmente valioso em domínios em que as decisões de casos individuais são críticas, como medicina personalizada ou aprovações de empréstimos financeiros. Por exemplo, o LIME (Local Interpretable Model-agnostic Explanations) perturba os dados de entrada em torno da instância de interesse e ajusta um modelo simples e interpretável para aproximar o limite de decisão local do modelo complexo. Isso ajuda a entender quais recursos foram mais influentes na decisão específica. Da mesma forma, o SHAP (Shapley Additive Explanations) atribui valores de importância a cada recurso para uma previsão específica, com base nos valores de Shapley da teoria dos jogos cooperativos, identificando assim a contribuição de cada recurso para a previsão. Além disso, as explicações contrafactuais identificam as mudanças mínimas necessárias nos recursos de entrada para obter uma mudança desejada na saída, ilustrando como um resultado pode ser alterado e fornecendo percepções acionáveis sobre o processo de tomada de decisão. (Burkart; Huber, 2021)

2.3.2.2 Interpretabilidade global

Por outro lado, os métodos de interpretabilidade global têm como objetivo fornecer uma compreensão holística do comportamento de um modelo em todas as instâncias de dados, oferecendo percepções sobre o funcionamento geral do modelo e destacando quais recursos geralmente influenciam suas previsões. Essa perspectiva global é fundamental para garantir que o modelo se comporte conforme o esperado em vários cenários. Por exemplo, os gráficos de dependência parcial (Partial Dependence Plots, PDP) visualizam o efeito médio de um recurso nas previsões do modelo, mostrando a relação geral entre o recurso e o resultado. Além disso, o SHAP pode agregar as importâncias dos recursos em várias instâncias para fornecer uma visão global do comportamento do modelo, ilustrando como os diferentes recursos contribuem para as previsões em média. Por fim, o Model-Agnostic Feature Importance avalia a importância de cada recurso para o desempenho geral do modelo, medindo o impacto nas previsões quando o recurso é variado, ajudando a entender quais recursos são mais importantes para o sucesso ou fracasso do modelo. (Burkart; Huber, 2021)

2.3.3 Momento da interpretação

2.3.3.1 Interpretabilidade Post-Hoc

Os métodos de interpretabilidade post-hoc são aplicados após o treinamento de um modelo, analisando as previsões do modelo para fornecer explicações sem alterar o próprio modelo. Esses métodos são versáteis e podem ser usados com qualquer modelo de aprendizado de máquina, inclusive modelos complexos de caixa preta, como redes neurais profundas. Por exemplo, os Gradientes e os Gradientes Integrados calculam o gradiente da saída em relação aos recursos de entrada para explicar a importância de cada recurso, oferecendo uma visão integrada ao caminho que ajuda a entender como cada recurso contribui para a previsão final. O LIME também gera explicações interpretáveis ao aproximar o limite de decisão local do modelo, fornecendo insights sobre o processo específico de tomada de decisão para instâncias individuais. Além disso, as redes deconvolucionais visualizam quais recursos nos dados de entrada ativam determinados neurônios em uma rede neural convolucional, ajudando a entender a função de diferentes partes da entrada na elaboração de previsões. (Burkart; Huber, 2021)

2.3.3.2 Interpretabilidade intrínseca

A interpretabilidade intrínseca refere-se a modelos que são inerentemente compreensíveis devido à sua estrutura ou design, oferecendo insights claros sobre seu processo de tomada de decisão sem a necessidade de técnicas adicionais de interpretação. Por exemplo, as árvores de decisão servem como uma explicação em si, com sua estrutura de árvore mostrando como as decisões são tomadas com base em divisões de recursos, e cada caminho da raiz para uma folha representando uma regra para fazer uma previsão. Os modelos lineares fornecem coeficientes que indicam diretamente o peso ou a importância de cada recurso na previsão do resultado, facilitando a visualização de como cada recurso afeta a previsão e a direção de seu impacto. Além disso, os modelos baseados em regras usam regras "se-então"para tomar decisões, em que cada regra fornece uma explicação clara e transparente de como a decisão foi tomada, o que os torna particularmente úteis em domínios em que as decisões precisam ser facilmente compreendidas e justificadas. (Burkart; Huber, 2021)

2.3.4 Especificidade do modelo

2.3.4.1 Interpretabilidade agnóstica de modelo

Os métodos agnósticos de modelo são projetados para serem aplicáveis a qualquer tipo de modelo de aprendizado de máquina, independentemente de sua estrutura interna. Esses métodos oferecem uma camada flexível de interpretação que pode ser usada em diferentes modelos. Por exemplo, o LIME pode ser aplicado a qualquer classificador,

fornecendo explicações locais por meio do ajuste de modelos simples e interpretáveis a dados perturbados, o que o torna útil em vários tipos de modelos. Da mesma forma, o SHAP oferece uma medida unificada da importância dos recursos que pode ser aplicada a qualquer modelo, ajudando a entender como os diferentes recursos influenciam as previsões de um determinado modelo. Além disso, os gráficos de dependência parcial (Partial Dependence Plots, PDP) e de expectativa condicional individual (Individual Conditional Expectation, ICE) visualizam o efeito de um ou mais recursos nas previsões do modelo, fornecendo informações sobre a relação geral entre recursos e resultados, aplicáveis a qualquer tipo de modelo. (Burkart; Huber, 2021)

2.3.4.2 Interpretabilidade específica do modelo

Os métodos específicos do modelo são adaptados às características e ao funcionamento interno de determinados modelos, aproveitando a estrutura do modelo para fornecer explicações mais detalhadas e precisas. Por exemplo, o Grad-CAM e o Grad-CAM++ foram projetados especificamente para redes neurais convolucionais para visualizar quais partes de uma imagem contribuem mais para a classificação, destacando as regiões na imagem de entrada que são mais relevantes para fazer uma previsão. A LRP (Layer-Wise Relevance Propagation), usada principalmente para redes neurais profundas, distribui a relevância de uma previsão pelas camadas da rede, ajudando a rastrear como cada camada contribui para a decisão final e oferecendo uma visão detalhada do processo de tomada de decisão. As redes deconvolucionais também são usadas com redes neurais convolucionais para visualizar quais recursos ativam camadas específicas, fornecendo insights sobre o funcionamento interno do modelo e como ele processa os dados de entrada. (Burkart; Huber, 2021)

2.3.5 Tipos de modelos explicados

2.3.5.1 Explicação de modelos Black-Box

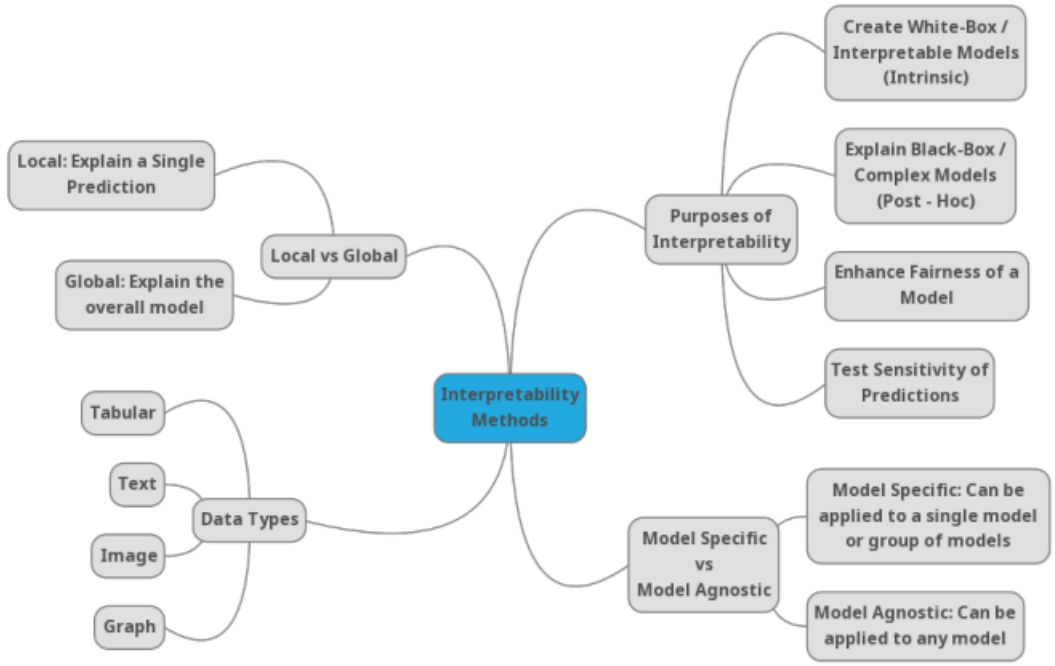
As explicações de caixa preta concentram-se em elucidar as previsões de modelos complexos que não fornecem inerentemente percepções sobre seu processo de tomada de decisão. Esses métodos são essenciais para a compreensão de modelos cujo funcionamento interno não é facilmente interpretável, como redes neurais profundas ou modelos de conjunto. Por exemplo, o LIME gera explicações para qualquer modelo de caixa preta aproximando os limites de decisão local, o que possibilita entender como modelos complexos fazem previsões específicas. O SHAP fornece a importância dos recursos para qualquer modelo, ajudando a explicar a saída dos modelos de caixa preta e oferecendo uma abordagem unificada para entender as previsões de modelos complexos. Além disso, o RISE (Randomized Input Sampling for Explanation) cria mapas de saliência para modelos de caixa preta para destacar a importância de cada recurso, fornecendo uma representação visual de quais

partes dos dados de entrada são mais relevantes para as previsões do modelo. (Burkart; Huber, 2021)

2.3.5.2 Modelos de caixa branca

Os modelos de caixa branca são interpretáveis por design, oferecendo percepções diretas sobre suas previsões sem exigir métodos adicionais de interpretabilidade. Por exemplo, as árvores de decisão fornecem uma explicação clara para cada previsão, com o caminho da raiz para um nó folha servindo como uma narrativa direta do processo de tomada de decisão com base em divisões de recursos. Da mesma forma, os modelos de regressão linear e logística usam coeficientes para indicar a direção e a força da relação entre cada recurso e o resultado, fornecendo uma visão direta e interpretável de como os recursos afetam as previsões. (Burkart; Huber, 2021)

Figura 2 – Representação da classificação dos métodos de interpretabilidade de modelos



Fonte: Linardatos, 2020

2.3.6 DecodeChess

Apesar de não encontrar na literatura artigos em inglês ou português que unem xadrez computacional à inteligência artificial explicável, foi encontrada uma plataforma online que partilha dos mesmos objetivos desse trabalho. DecodeChess é uma ferramenta online que utiliza inteligência artificial para analisar partidas de xadrez, analisando ameaças, planos de ataque, funcionalidade das peças, motivos táticos e estratégicos das posições. ()

DecodeChess parece uma boa ferramenta, entretanto as explicações fornecidas pela plataforma são feitas acerca de detalhes lógicos da posição, e este presente trabalho visa explicar também questões mais abstratas do tabuleiro, e uma visão mais completa da posição.

2.3.7 Redes neurais

As redes neurais são modelos inspirados na forma como o cérebro humano processa informações, consistindo em camadas interconectadas de neurônios artificiais. Elas transformam dados de entrada em saídas úteis por meio de diferentes níveis de representação: as camadas iniciais detectam características simples, como bordas em imagens, enquanto as finais capturam representações mais abstratas e complexas. Isso é feito sem a necessidade de intervenção humana para extrair características, aproveitando o aprendizado hierárquico.

Entre os tipos de camadas presentes em redes neurais, destacam-se as convolucionais, que utilizam filtros para extrair características específicas da entrada. Esse processo reduz significativamente o número de parâmetros da rede, tornando o treinamento mais eficiente. Complementando as camadas convolucionais, temos as camadas de pooling, que reduzem a dimensionalidade dos mapas de características ao concentrar as informações mais relevantes, o que melhora a capacidade de generalização da rede e reduz o risco de overfitting.

As camadas de função de ativação introduzem a não-linearidade no modelo, permitindo o aprendizado de padrões complexos. Dentre as funções de ativação mais comuns, estão o ReLU, que converte entradas negativas em zero, reduzindo o custo computacional, e o Sigmoid e Tanh, que limitam os valores de saída a intervalos específicos. Essas camadas ajudam a rede a decidir quais neurônios serão ativados, com base na entrada, gerando saídas mais robustas.

Na etapa final da rede, as camadas totalmente conectadas são responsáveis por conectar cada neurônio com todos os da camada anterior, criando abstrações de alto nível e produzindo as previsões ou classificações finais. Para medir o desempenho do modelo, camadas de perda, como a de entropia cruzada ou de erro quadrático médio, calculam a discrepância entre as previsões da rede e os valores reais, fornecendo a base para o ajuste dos pesos da rede durante o treinamento.

Além dessas, camadas de regularização, como dropout e drop-weights, são empregadas para reduzir o overfitting. Essas técnicas eliminam aleatoriamente neurônios ou conexões durante o treinamento, forçando a rede a aprender características mais independentes e melhorando sua generalização. A normalização por lote é outra ferramenta importante, responsável por reduzir a variação nas ativações das camadas durante o treinamento, acelerando a convergência, diminuindo problemas relacionados a gradientes desaparecendo e reduzindo a dependência de hiperparâmetros.

Esse conjunto de camadas forma a base estrutural das redes neurais modernas, que são ajustadas para cada aplicação específica, maximizando sua eficiência e capacidade de resolver problemas complexos de aprendizado de máquina. [CITAR]

3 METODOLOGIA

Como apresentado anteriormente, o objetivo desta pesquisa foi desenvolver um artefato que utiliza a XAI para melhorar a compreensão dos lances realizados por engines de xadrez. Essa proposta busca enfrentar o problema central da pesquisa, que é a falta de explicabilidade dessas engines, promovendo maior transparência e acessibilidade às suas decisões.

O tema deste trabalho foi definido aliando o interesse pela área da inteligência artificial à identificação dos problemas de explicabilidade que surgem nesse campo. A explicabilidade tem se tornado uma questão central na inteligência artificial, especialmente à medida que os algoritmos se tornam mais complexos e suas decisões mais difíceis de interpretar. A transparência e a compreensão dos processos de decisão das máquinas são essenciais para a aceitação e confiança nas tecnologias de IA, tanto em aplicações práticas quanto em contextos acadêmicos.

O xadrez e as engines também desempenharam um papel fundamental na escolha deste tema, por ser um excelente objeto de estudo, pois, não é simples ao ponto de ser trivial e nem complexo ao ponto de não poder ser compreendido CITAR SHANON. As partidas de xadrez geradas por engines oferecem um vasto conjunto de dados para análise, onde cada movimento pode ser explorado.

Entender como e por que uma engine decide fazer um lance, considerando milhares ou até milhões de possibilidades, é um desafio que pode proporcionar conhecimentos valiosos para a explicabilidade em IA de maneira geral.

Assim, ao aliar o estudo as engines de xadrez com o desafio técnico da explicabilidade em inteligência artificial, a temática foi delimitada. Através deste, espera-se não apenas contribuir para a compreensão das decisões das engines de xadrez, mas também avançar no campo da explicabilidade em IA, proporcionando conhecimentos que possam ser aplicados em outros domínios.

Para realizar este trabalho, foram conduzidas revisoões sistemáticas sobre engines de xadrez, técnicas de explicabilidade em inteligência artificial e redes neurais. Essas revisões foram essenciais para fundamentar teoricamente a pesquisa, identificar lacunas no conhecimento existente e definir as direções metodológicas mais adequadas para o desenvolvimento do estudo.

A revisão sistemática sobre engines de xadrez teve como objetivo compreender a evolução e as capacidades dessas ferramentas, desde os primeiros motores baseados em regras simples até os mais modernos, que utilizam redes neurais e aprendizado profundo. Foram revisados artigos acadêmicos e livros que detalham o desenvolvimento e a aplicação

de engines de xadrez. Essa revisão permitiu identificar as principais características dos motores mais avançados, como o Stockfish e o AlphaZero, e compreender os algoritmos e técnicas subjacentes que permitem a esses motores alcançar um bom desempenho.

O objetivo da revisão sobre técnicas de explicabilidade foi mapear as metodologias existentes que buscam tornar os processos de decisão de modelos de IA mais transparentes e compreensíveis para os humanos. Foram revisados artigos científicos que tratam de diversas técnicas que visam explicar modelos complexos de machine learning e deep learning.

A revisão das técnicas de explicabilidade permitiu identificar quais métodos seriam mais apropriados para aplicar no contexto das engines de xadrez. Critérios como a capacidade de fornecer explicações detalhadas, a facilidade de interpretação e a aplicabilidade a modelos baseados em redes neurais foram considerados. Esta análise aprofundada das técnicas disponíveis orientou a seleção das ferramentas que seriam testadas na fase experimental do trabalho.

A combinação dessas duas revisões sistemáticas forneceu uma base teórica robusta e direcionou as escolhas metodológicas do estudo. Com base nessas revisões, foi possível desenhar um plano de pesquisa que integra o uso de técnicas avançadas de IA com métodos de explicabilidade, visando oferecer uma compreensão aprofundada das decisões das engines de xadrez.

A pesquisa sobre redes neurais também foi um estágio importante para fundamentar a forma de desenvolvimento da rede neural que funcionou como o modelo que imita as avaliações estáticas do Stockfish, para ser utilizado como objeto de explicação das técncias de XAI

No contexto deste trabalho, foi adotada a abordagem do Design Science Research (DSR) como um metodo conduzir a pesquisa. O DSR é uma metodologia de pesquisa amplamente utilizada nas áreas de ciência da computação e sistemas de informação, focada na criação e avaliação de artefatos projetados para resolver problemas identificados. Sua aplicabilidade se estende ao desenvolvimento de novos modelos, métodos, sistemas e frameworks, fornecendo um caminho estruturado para a inovação e a melhoria contínua.

O processo DSR é estruturado em seis etapas principais. Inicialmente, identificamos a necessidade de explicar as decisões tomadas pelas engines de xadrez, para facilitar a compreensão da comunidade enxadrista acerca do funcionamento das engines, afim de contribuir com o estudo sobre o jogo. Este problema é relevante tanto para a comunidade enxadrística quanto para o campo da inteligência artificial, onde a explicabilidade é um tema de crescente importância. Com base no problema identificado, estabelecemos objetivos claros para a nossa pesquisa, sendo o principal desenvolver um artefato que permita extrair explicações detalhadas e interpretáveis das decisões das engines de xadrez, destacando aspectos como o papel das peças, o domínio das casas e as intenções estratégicas.

Este processo sistemático e iterativo, guiado pelo DSR, nos permite desenvolver uma solução eficaz para o problema de explicabilidade das engines de xadrez, e avaliar os resultados obtidos, garantindo que as conclusões sejam robustas e aplicáveis a contextos reais de uso. Assim, o DSR se mostra uma abordagem essencial para a condução deste trabalho, proporcionando uma base sólida para a inovação e o avanço no entendimento das decisões das engines de xadrez.

3.1 Preparação do ambiente

A preparação do ambiente para o desenvolvimento do artefato foi um passo importante para garantir a consistência, reprodutibilidade e eficiência durante todo o processo de análise das engines de xadrez. O ambiente foi configurado de forma dockerizada, utilizando Python como linguagem de programação principal, MySQL como gerenciador do banco de dados e Alembic para gerenciar as migrações do banco de dados. Esta abordagem permitirá um desenvolvimento mais organizado, modular e fácil de manter, além de facilitar a colaboração entre diferentes membros da equipe.

Para iniciar a preparação do ambiente, foi utilizado o Docker, uma plataforma de contêineres que possibilita criar, gerenciar e executar aplicações em ambientes isolados e padronizados. O Docker simplifica a gestão das dependências e configurações do sistema, garantindo que o ambiente de desenvolvimento seja idêntico ao ambiente de produção, minimizando problemas de incompatibilidade.

Após a instalação do Docker, foi criado um arquivo Dockerfile que definiu a imagem do Docker para o projeto. Este arquivo especifica a base da imagem (neste caso, uma imagem oficial do Python), bem como todas as dependências necessárias, como bibliotecas e pacotes Python que foram utilizados no desenvolvimento. O conteúdo do Dockerfile incluiu comandos para instalar essas dependências utilizando o gerenciador de pacotes 'pip', garantindo que todas as versões dos pacotes sejam consistentes em todos os ambientes onde a imagem for executada.

Em seguida, foi criado um arquivo docker-compose.yml para definir e gerenciar serviços Docker multi-contêiner. Permitindo a utilização de um container do MySql em conjunto com o container do Python. A utilização do Docker Compose simplificiy o processo de iniciar, parar e gerenciar os contêineres, garantindo que todos os serviços estejam corretamente interligados e configurados.

Após a configuração dos containers, o Alembic foi configurado, para criar, aplicar e gerenciar versões de esquema do banco de dados de forma controlada e eficiente. Para configurar o Alembic, foi necessário instalar o pacote através do pip e inicializar o diretório de migrações. O arquivo de configuração do Alembic (alembic.ini) e o script de ambiente (env.py) foram ajustados para se conectarem ao banco de dados MySQL definido no Docker

Compose. O env.py contém a lógica para obter a URL de conexão do banco de dados a partir das variáveis de ambiente, garantindo que as migrações possam ser executadas em diferentes ambientes sem modificações adicionais no código.

Para o desenvolvimento do modelo de avaliação estática de posições de xadrez o framework PyTorch foi escolhido como a base para a construção e treinamento da rede neural, devido à sua flexibilidade, facilidade de uso e suporte a treinamento acelerado por GPU.

Para aproveitar ao máximo o desempenho durante o treinamento, foi instalada a versão de PyTorch com suporte a CUDA, permitindo que os cálculos computacionalmente intensivos fossem delegados à GPU disponível no sistema. Além disso, a biblioteca NumPy foi utilizada para manipulação eficiente de matrizes e arrays numéricos, enquanto o Pandas foi empregado para o processamento e tratamento inicial dos dados provenientes do Lichess.

A biblioteca h5py também foi instalada, permitindo a manipulação de grandes volumes de dados em formato HDF5, o que facilitou o armazenamento eficiente de tensores de entrada e rótulos durante o treinamento do modelo. Outras dependências incluem a biblioteca python-chess, que foi essencial para a manipulação de tabuleiros de xadrez e conversão de posições no formato FEN para representações matriciais, e o tqdm, usado para gerar barras de progresso que facilitam o monitoramento do treinamento.

Por fim, ferramentas como o PyTorch Lightning foram utilizadas para facilitar o gerenciamento do fluxo de treinamento, embora o desenvolvimento tenha priorizado a implementação direta no PyTorch para maior controle sobre os detalhes da arquitetura.

Com todas as ferramentas dispostas e conteinerizadas, será feita a modelagem do banco de dados para guardar as partidas devidamente classificadas.

3.2 Teste de cenários relevantes para aplicação da explicabilidade

O xadrez é um jogo dinâmico, e a avaliação de uma posição muitas vezes vai além de uma análise puramente posicional. Em determinadas situações, como aquelas envolvendo sequências táticas inevitáveis que levam a ganhos materiais significativos, o desenrolar dessas jogadas táticas têm prioridade sobre a árvore de avaliações estáticas da posição, tornando a análise do conjunto de avaliações estáticas irrelevante nesse contexto.

Para testar em quais momentos deve ser empregado o uso de XAI nas avaliações estáticas, foi realizado um pequeno estudo comparativo que visa comparar o comportamento do Stockfish com dois tipos de função de avaliação, a função de avaliação clássica (Hand-Crafted Evaluation - HCE) e a função de avaliação com redes neurais (NNUE).

O objetivo dessa abordagem foi foi avaliar em quais cenários o stockfish com o NNUE se sobressairá ou não sobre o stockfish com HCE, evidenciando cenários que exigem

mais de uma boa função de avaliação. A análise se deu em relação ao desempenho do Stockfish com NNUE em comparação com o Stockfish com a função de avaliação HCE, visando identificar momentos em que a função de avaliação estática robusta é determinante para o resultado. Essa análise é fundamental para a aplicação de técnicas de XAI, uma vez que tais técnicas foram utilizadas para extrair explicações das avaliações estáticas da engine.

Para realizar o estudo foram definidos dois cenários distintos de posição no xadrez: o primeiro cenário é a posição inicial do jogo, e o segundo é uma posição de final de peões empatada.

Inicialmente, foram criadas duas equipes de engines Stockfish: uma composta por 25 versões com NNUE ativado e outra por 25 versões sem NNUE, utilizando apenas a função de avaliação clássica. A escolha de 25 engines em cada equipe visou incluir a variável "profundidade de busca"no experimento. Assim, cada engine tinha sua profundidade de pesquisa limitada entre 1 e 25, permitindo a análise do impacto dessa variável no desempenho das engines.

Para cada cenário, cada Stockfish com NNUE enfrentou cada um dos Stockfish sem NNUE em duas partidas: uma com a engine NNUE jogando de brancas e outra de negras, garantindo equilíbrio nas condições de jogo. Assim, uma engine com NNUE e profundidade 1 enfrentou todas as engines clássicas, desde profundidade 1 até profundidade 25, repetindo-se esse padrão até as engines NNUE de profundidade 25.

Os duelos entre as engines permitiram observar momentos em que o Stockfish com redes neurais superava o Stockfish clássico, evidenciando a importância de uma função de avaliação aprimorada e, por extensão, a relevância de técnicas XAI nesses contextos.

Inversamente, em situações onde a função de avaliação se mostrou irrelevante (como finais onde a profundidade de pesquisa garantiu a precisão dos resultados), ficou claro que o uso de XAI para explicação das avaliações estáticas seria desnecessário, pois a árvore de pesquisa já desempenha papel decisivo.

A seleção de uma posição final empatada teve o intuito de avaliar a redução da importância da função de avaliação em profundidades maiores, pois, ao alcançar o final do jogo, as engines conseguem antecipar o resultado independentemente da função de avaliação. Assim, para essas posições, conclui-se que a aplicação de XAI não é relevante.

Os resultados obtidos das partidas que começam da posição inicial, mostraram que o Stockfish com função de avaliação NNUE superou o Stockfish clássico na maioria das partidas. Observou-se que, em partidas com profundidade menores que 15, a profundidade de pesquisa era um fator preponderante, a vantagem de 2 unidades de profundidade era um fator determinante para garantir a vitória de qualquer engine na partida, independentemente do tipo de função de avaliação. Porém, a partir de profundidades superiores a 20,

a diferença de profundidade deixou de ser um fator determinante, e as engines com NNUE passaram a vencer a maioria dos confrontos.

[FIGURA AQUI]

No cenário de final empatado, a posição é inevitavelmente empatada em 20 lances, caso nenhum dos dois jogadores cometa um erro, apesar da ligeira vantagem material de 2 peões a mais para as negras. Devido à diferença material, apesar de ser um lance empatado, jogar de brancas nessa posição é mais desafiador.

O experimento mostrou que quando o Stockfish com NNUE jogou de brancas, desempenhou a função de "sobrevivência"melhor que o Stockfish com HCE.

[FIGURA AQUI]

3.3 Seleção de técnica de explicabilidade

Após identificar as técnicas de explicabildiade encontradas na literatura durante a revisão sistemática, foi iniciado o processo de seleção, considerando a natureza específica do problema em análise: Explicabilidade das avaliações estáticas realizadas por um modelo de neural network treinado com base no Stockfish.

O objetivo foi selecionar um conjunto representativo de abordagens para explicar a importância dos elementos do tabuleiro na avaliação, priorizando técnicas que fossem adequadas ao escopo do estudo.

Inicialmente foi preciso definir dois importantes critérios de exclusão para selecionar as técnicas que foram aplicadas. O primeiro critério é referente ao escopo de explicação, técnicas como PDP, ICE e ALE foram descartadas, pois seu foco está na explicação de modelos globais, enquanto o problema demanda explicações locais específicas para posições de xadrez.

O segundo critério de exclusão é referente ao objeto de análise, abordagens como Guided BackPropagation, Redes Deconvolucionais e CAM foram desconsideradas, pois o interesse da pesquisa está em explicar diretamente a relevância das entradas do modelo, e não a ativação de camadas intermediárias.

Após o processo de exclusão, foram selecionadas técnicas que poderiam oferecer diferentes abordagens para explicar a importância das entradas do modelo. A seleção final incluiu cinco técnicas, escolhidas por sua diversidade metodológica: SmoothGrad, Saliency Map, LRP, DeepLIFT e LIME.

Os métodos SmoothGrad e Saliency Map utilizam gradientes diretos para gerar explicações, permitindo uma visualização das entradas mais sensíveis às alterações no output.

Layer-wise Relevance Propagation (LRP) e DeepLIFT propagam relevâncias ou diferenças ao longo das camadas, fornecendo uma perspectiva mais estruturada de como cada entrada contribui para o resultado final. LIME: ajusta modelos explicativos simples em torno de perturbações do input, gerando interpretações localmente compreensíveis e independentes do modelo principal.

Essa seleção foi realizada com o intuito de cobrir diferentes abordagens metodológicas e testar como cada uma poderia contribuir para explicar a avaliação de posições de xadrez. Dessa forma, o conjunto de técnicas selecionado permite analisar o problema sob múltiplas perspectivas, maximizando a compreensão sobre a relevância dos elementos do tabuleiro na avaliação.

3.4 Modelo aproximado

Devido a incompatibilidade do stockfish com as técnicas de explicabilidade, houve a necessidade da criação de um modelo aproximado, que imitasse as avaliações do stockfish e fosse utilizado como para a aplicação das técnicas de explicablidiade, pois, algumas das técnicas aplicadas são intrínsecas e precisam dos tensores de entrada da rede neural para realizar a explicação.

Inicialmente, foi necessário construir um dataset robusto, utilizando como fonte o site Lichess [POR O LINK AQUI], uma plataforma amplamente reconhecida por sua base de dados extensa e diversificada de partidas. Optou-se por selecionar partidas de jogadores de elite, uma vez que estas tendem a conter posições mais ricas em termos táticos e posicionais, o que contribuiria para o aprendizado do modelo. O processo de coleta resultou em um arquivo CSV com 2.400.000 registros. Cada linha do arquivo correspondia a uma posição de xadrez representada em formato FEN (Forsyth-Edwards Notation), acompanhada da avaliação estática gerada pelo motor Stockfish 17.

Essa avaliação foi extraída diretamente da análise do motor, que atribui um valor numérico às posições com base em critérios como controle de espaço, segurança do rei, material, entre outros, com a utilização do NNUE.

Durante o pré-processamento, uma atenção especial foi dada à qualidade dos dados. Primeiramente, todas as posições duplicadas foram removidas, garantindo que o modelo não fosse treinado de maneira redundante. Adicionalmente, posições em que o rei estava em cheque foram eliminadas, uma vez que o Stockfish prioriza cálculos dinâmicos nessas situações e não retorna uma avaliação puramente estática. Assim, o dataset resultante era composto apenas por posições em que a avaliação estática do Stockfish era confiável, assegurando que o modelo aprendesse com dados consistentes e representativos.

Para viabilizar o uso dessas informações em um modelo de aprendizado profundo, foi necessário transformar as posições em formato FEN em uma representação matricial

tridimensional (13x8x8). Esta representação foi projetada para ser compatível com redes neurais convolucionais, que são particularmente eficazes para processar dados estruturados como imagens e tabuleiros. Cada matriz tridimensional possuía 13 planos: os primeiros 12 planos representavam a presença de peças no tabuleiro, com distinção entre peças brancas e pretas (seis tipos de peças para cada cor), enquanto o 13º plano indicava os movimentos legais disponíveis na posição.

Devido à grande quantidade de dados e à limitação de memória RAM disponível no ambiente de trabalho, o dataset completo foi dividido em 24 partes, cada uma contendo 100.000 registros. Cada segmento foi processado individualmente, com as matrizes de entrada (X) e as avaliações do Stockfish (Y) sendo convertidas em tensores PyTorch.

Esses tensores foram então armazenados em arquivos separados para posterior uso. Após o processamento de todas as partes, os dados foram consolidados em um único arquivo no formato HDF5. Este formato foi escolhido por sua eficiência no armazenamento e recuperação de grandes volumes de dados, permitindo acesso rápido durante o treinamento.

O modelo desenvolvido para avaliar posições estáticas consistiu em uma rede neural convolucional projetada especificamente para capturar características espaciais do tabuleiro de xadrez. Ele era composto por duas camadas convolucionais seguidas por camadas totalmente conectadas.

As camadas convolucionais utilizavam filtros para identificar padrões relevantes nas representações matriciais, como alinhamento de peças e controle de casas críticas. As camadas totalmente conectadas, por sua vez, processam essas informações para gerar uma saída numérica única, representando a avaliação da posição. Funções de ativação ReLU foram empregadas após cada camada para introduzir não-linearidades e permitir que o modelo aprendesse relações complexas. Adicionalmente, técnicas de inicialização de pesos, como He e Xavier, foram utilizadas para garantir uma convergência mais eficiente do treinamento.

O treinamento foi realizado utilizando o algoritmo de otimização Adam, devido à sua capacidade de ajustar dinamicamente as taxas de aprendizado para cada parâmetro. A função de perda escolhida foi o erro quadrático médio (MSE), ideal para o problema de regressão em questão, já que a saída do modelo era um valor contínuo. O treinamento ocorreu em um ambiente equipado com suporte a GPU, acelerando significativamente o processamento. Os dados foram alimentados no modelo em batches de tamanho 64, utilizando um DataLoader configurado para embaralhar os registros a cada época, garantindo que o modelo não se adaptasse a padrões específicos da sequência de treinamento.

O treinamento foi realizado ao longo de 50 épocas, com o monitoramento constante da função de perda para avaliar a evolução do aprendizado do modelo, terminando em 0.1612 de perda. Além disso, técnicas de regularização, como o gradient clipping, foram

aplicadas para evitar explosões de gradientes e melhorar a estabilidade do processo. Ao final de cada época, métricas de desempenho foram registradas para análise futura.

Após o término do treinamento, o modelo foi salvo no formato PyTorch, permitindo sua reutilização em outras aplicações e análises. O arquivo HDF5 foi encerrado, liberando os recursos do sistema.

Para demonstrar a capacidade do modelo de aproximação desenvolvido em replicar as avaliações estáticas do Stockfish 17, foi realizada uma análise de desempenho em um conjunto de teste composto por 10 mil posições de xadrez, distintas daquelas utilizadas no treinamento. Os resultados obtidos foram comparados com avaliações realizadas entre diferentes versões do Stockfish, a fim de fornecer um referencial para análise dos erros e da correlação.

Na comparação entre o modelo desenvolvido e o Stockfish 17, os resultados demonstraram um Erro Médio Absoluto (MAE) de 0.6320, indicando uma diferença média relativamente pequena entre as avaliações do modelo e do motor. O Erro Médio Quadrático (MSE), que amplifica discrepâncias maiores, foi de 0.7721, enquanto a Raiz do Erro Quadrático Médio (RMSE), uma métrica mais interpretável por retornar os erros na mesma escala das avaliações, apresentou um valor de 0.8787. O Coeficiente de Correlação (R^2), que avalia a capacidade do modelo em explicar a variância nos dados, foi de 0.8676, evidenciando uma forte correlação entre o modelo e o Stockfish 17.

Como parâmetro de comparação, foi avaliada também a similaridade entre diferentes versões do Stockfish. Entre o Stockfish 17 e o Stockfish 15, o MAE foi de 0.4117, enquanto o MSE alcançou 1.1069 e o RMSE, 1.0521. O R^2, neste caso, foi de 0.8101, indicando uma boa correlação, mas inferior à encontrada entre o modelo desenvolvido e o Stockfish 17. Já na comparação entre o Stockfish 17 e o Stockfish 16, os resultados apresentaram maior divergência, com MAE de 0.5566, MSE de 5.7834, RMSE de 2.4049 e R^2 de apenas 0.1436, refletindo uma baixa correlação entre as avaliações dessas versões.

Esses resultados demonstram que o modelo de aproximação proposto alcança um nível de similaridade com o Stockfish 17 que se compara ou até supera a consistência observada entre versões distintas do próprio motor de xadrez. A baixa variabilidade nos erros e a alta correlação reforçam a eficácia do modelo em reproduzir avaliações do Stockfish 17 de forma confiável, destacando sua adequação como substituto para análises baseadas em técnicas de interpretabilidade, que demandam maior transparência em relação ao processo de avaliação de posições estáticas.

3.5 Aplicação da técnica de explicabilidade

Para aprofundar a análise da capacidade explicativa do modelo de aproximação, foram aplicadas técnicas de explicabilidade em duas posições específicas.

[IMAGEM AQUI] A posição da figura é conhecida na literatura enxadrística, foi uma partida realizada entre a engine Leela Chess Zero, e o stockfish, o recorte dessa posição se dá pelo próximo lance ser um movimento que sacrifica material em troca de um ganho posicional, um tema fundamental para o escopo do trabalho, que foca na análise posicional mais do que tática. Além disso, a escolha dessa posição foi motivada por sua análise detalhada em um vídeo do YouTube [LINK??], o que permite uma comparação qualitativa entre os resultados das explicações geradas pelo modelo e as análises humanas.

[IMAGEM AQUI] A segunda posição é uma posição "absurda", com uma configuração dificilmente alcançada em partidas reais, mas que dispõe de uma configuração onde a vantagem material não é o suficiente para superar a vantagem posicional.

Para o SmoothGrad, gerou-se ruído gaussiano em torno da entrada original, criando variações levemente modificadas da matriz inicial. Em seguida, cada variação foi processada pelo modelo, acumulando-se os gradientes calculados em cada iteração. O cálculo médio dos gradientes foi realizado para destacar quais áreas da entrada tiveram maior impacto nas decisões do modelo.

No caso do LIME, a matriz 13x8x8 foi achatada em um vetor unidimensional para compatibilidade com a ferramenta. Foi utilizada a técnica de perturbações locais, onde múltiplas amostras com modificações pontuais foram geradas e submetidas ao modelo. Uma explicação foi gerada para cada entrada alterada, permitindo a identificação das contribuições individuais de cada elemento do vetor, posteriormente mapeadas de volta para a matriz original.

A técnica de LRP foi aplicada calculando a relevância de cada elemento da entrada com base na propagação reversa dos valores pelo modelo. O input foi configurado para permitir gradientes, e o gradiente de saída foi multiplicado diretamente pela entrada, produzindo um mapa de relevâncias. Esse mapa foi extraído diretamente da operação matemática e projetado de volta para a representação espacial do tabuleiro.

Para o DeepLIFT, foi necessário definir uma baseline de referência, escolhida como uma matriz de zeros representando um tabuleiro vazio. O modelo foi executado comparando a entrada real com a baseline, o que permitiu determinar a contribuição relativa de cada elemento da matriz para a avaliação final. O processo utilizou a biblioteca Captum para calcular as atribuições diretamente.

Por fim, com Saliency Maps, configurou-se o tensor de entrada para calcular gradientes diretos em relação à saída do modelo. Realizou-se o backpropagation da saída do modelo em relação à entrada, utilizando os gradientes absolutos para identificar as regiões mais sensíveis na matriz 13x8x8.

Cada técnica foi implementada de forma independente, mantendo a integridade do modelo e sem realizar ajustes em sua arquitetura ou parâmetros. O foco foi utilizar as ferra-

mentas computacionais disponíveis para gerar explicações que poderiam ser posteriormente analisadas quanto à sua correlação com princípios enxadrísticos.

4 RESULTADOS

[AQUI VOU FALAR SOBRE OS GRÁFICOS PLOTADOS PELAS TÉCNICAS NAS POSIÇÕES (selecionar bem, pq tem muita imagem)]

REFERÊNCIAS

BRANGE, H. Evaluating heuristic and algorithmic improvements for alpha-beta search in a chess engine. 2021. Citado na página 23.

BURKART, N.; HUBER, M. F. A survey on the explainability of supervised machine learning. **Journal of Artificial Intelligence Research**, v. 70, p. 245–317, 2021. Citado nas páginas 26, 27, 28, 34, 35, 36 e 37.

DAVID, O. E.; NETANYAHU, N. S.; WOLF, L. Deepchess: End-to-end deep neural network for automatic learning in chess. In: _____. **Lecture Notes in Computer Science**. Springer International Publishing, 2016. p. 88–96. ISBN 9783319447810. Disponível em: http://dx.doi.org/10.1007/978-3-319-44781-0_11. Citado na página 23.

disservin. **Chess pieces movements**. 2024. Disponível em: https://static.significados.com.br/foto/movimentos-das-pecas-de-xadrez.jpg. Acesso em: 01 JUN. 2024.

disservin. **NNUE**. 2024. Disponível em: https://disservin.github.io/stockfish-docs/nnue-pytorch-wiki/docs/nnue.html. Acesso em: 01 JUN. 2024. Citado nas páginas 25 e 26.

HAQUE, R.; WEI, T. H.; MüLLER, M. On the road to perfection? evaluating leela chess zero against endgame tablebases. In: _____. **Advances in Computer Games**. Springer International Publishing, 2022. p. 142–152. ISBN 9783031114885. Disponível em: http://dx.doi.org/10.1007/978-3-031-11488-5_13. Citado nas páginas 23 e 24.

KLOSOWSKI, K. Chess engine using deep reinforcement learning. 2019. Citado na página 22.

LEMLEY, J. *et al.* Cwu-chess: An adaptive chess program that improves after each game. In: **2018 IEEE Games, Entertainment, Media Conference (GEM)**. [S.l.: s.n.], 2018. p. 1–9. Citado na página 23.

LINARDATOS, P.; PAPASTEFANOPOULOS, V.; KOTSIANTIS, S. Explainable ai: A review of machine learning interpretability methods. **Entropy**, MDPI, v. 23, n. 1, p. 18, 2020. Citado nas páginas 29, 30, 31, 32 e 33.

ØSTENSEN, E. F. **A complete chess engine parallelized using lazy smp**. Dissertação (Mestrado), 2016. Citado na página 13.

RODRÍGUEZ, R. M. **Developing a chess engine**. Tese (Doutorado) — Universitat Politècnica de València, 2022. Citado nas páginas 21, 22, 23 e 24.

VRZINA, S. **Piece By Piece: Building a Strong Chess Engine.** Tese (Doutorado) — Vrije Universiteit Amsterdam, 2023. Citado nas páginas 22 e 24.