Classificação de dados quentes e frios em bancos de dados em memória utilizando técnicas de Aprendizado de Máquina

Eric Ferreira Landim Pinto*

José Wellington Franco da Silva†

RESUMO

Com o avanço tecnológico e a crescente demanda por processamento de volumes massivos de dados, o uso de banco de dados tem se expandido significativamente. Diante desse cenário, sistemas eficientes baseados em memória principal, como o *In-Memory DataBase* (IMDB), emergiram como soluções proeminentes. No entanto, o crescente risco de *data overflow* destaca a importância de estratégias de gerenciamento de dados, como o uso de *Online Transaction Processing* (OLTP), para movimentar dados entre diferentes sistemas de armazenamento. Este estudo tem como principal objetivo desenvolver uma abordagem focada em técnicas de Aprendizado de Máquina (AM) para identificar e classificar dados "quentes"e "frios"em bancos de dados baseados em memória. Ao priorizar a gestão eficiente desses dados, busca-se mitigar o problema do *data overflow*. Além disso, o estudo sugere possíveis estratégias para a utilização dos dados identificados, como o armazenamento de dados "quentes"na memória RAM e a alocação de dados “frios” em subsistemas secundários, oferecendo uma base para futuros estudos explorarem essas abordagens de gerenciamento de dados.

Palavras-chave: Banco de dados em memória. IMDB. In-Memory. Dados quentes e frios. OLTP. Aprendizado de Máquina. AM.

ABSTRACT

Abstract With technological advancement and the growing demand for processing massive volumes of data, the use the database has expanded significantly. Given this scenario, efficient systems based in main memory, such as In-Memory DataBase (IMDB), have emerged as prominent solutions. However, the increasing risk of data overflow highlights the importance of data management strategies. data, such as using Online Transaction Processing (OLTP), to move data between different storage systems. This study’s main objective is to develop an approach focused on Machine Learning (ML) techniques to identify and classify "hot" data and "cold" in memory-based databases. When prioritizing the efficient management of this data, we seek mitigate the data overflow problem. Furthermore, the study suggests possible strategies for using of identified data, such as storing "hot" data in RAM and allocating "cold" data on secondary subsystems, providing a basis for future studies to explore these data management approaches.

Keywords: In-memory database. IMDB. In-Memory. Hot and cold data. OLTP. Machine Learning. AM.

* Aluno do Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Ceará - Campus - E-mail: ericlandim@hotmail.com

† Professor Doutor do Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Ceará - Campus - E-mail: wellington@crateus.ufc.br

1 INTRODUÇÃO

A eficiência de um banco de dados é fortemente dependente da velocidade e da capacidade de armazenamento da memória, especialmente ao atender a uma variedade de aplicações que exigem o processamento de grandes volumes de dados em tempo real (SANTOS, 2013). Nesse contexto, de acordo com (MARTINS[1]; POLETTO, 2016), os sistemas de banco de dados baseados em memória principal, conhecidos como IMDB (In-Memory Data Base), emergem como uma solução de destaque. Os IMDB são sistemas de gerenciamento que dependem da memória principal para o armazenamento de dados do computador. Estes sistemas utilizam a memória principal por ser mais rápida e de acesso aleatório. O processador do computador está em constante comunicação com os dados da memória principal. Assim, o banco de dados que utiliza o armazenamento em memória levará menos tempo para executar as instruções da CPU, proporcionando respostas mais rápidas e eficientes para as demandas de processamento de dados em tempo real.

A tecnologia IMDB desempenha um papel fundamental na escalabilidade e no aprimoramento do desempenho de bancos de dados. Seu foco está na otimização dos algoritmos de movimentação de dados, enfrentando desafios como o *data overflow*, que ocorre quando o volume de dados excede a capacidade da memória principal (SANTOS *et al.*, 2021).

A gestão eficaz de dados frios, dados oriundos e quase não utilizados, além de ocupar espaço na memória, se mostra uma abordagem econômica e prática para lidar com o problema do *data overflow*, especialmente diante do crescente uso de aplicações de *Big Data* e do aumento contínuo da quantidade de dados a serem gerenciados (EMMANUEL; STANIER, 2016).

Para a avaliação e comparação dos algoritmos de identificação de dados, este estudo adota um modelo de Aprendizado de Máquina *Long Short Term Memory* (LSTM). O LSTM é uma variação especializada de Rede Neural Recorrente (RNN), onde consegue lidar com problemas de processamento de sequências e reter informações importantes por longos períodos (SANTOS, 2019). A escolha do LSTM neste contexto visa explorar sua habilidade em compreender os padrões complexos presentes nos conjuntos de dados extraídos de bancos de dados em memória, proporcionando uma abordagem robusta para identificar e gerenciar os dados quentes e frios para proporcionar ideias em trabalhos futuros na utilização desses dados classificados.

O presente trabalho está estruturado da seguinte maneira: inicialmente, apresenta-se a introdução, seguida pela fundamentação teórica que abrange os principais temas deste estudo, detalhada no Capítulo 2. Posteriormente, no Capítulo 3, são abordados de maneira sucinta os trabalhos que serviram como base para a pesquisa. Em seguida, no Capítulo 4, são apresentados a metodologia e a abordagem adotadas. O Capítulo 5 engloba a avaliação experimental e os resultados obtidos. O desfecho do trabalho ocorre no Capítulo 6, que compreende a conclusão e as perspectivas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Banco de Dados em Memória

A otimização do desempenho e da capacidade de armazenamento de dados impulsionou o desenvolvimento de sistemas como o IMDB. Esses sistemas, como indicado por (SANTOS, 2013), operam com a memória principal para garantir tempos de resposta significativamente mais rápidos em comparação com os sistemas de armazenamento em disco convencionais. Além disso, conforme o autor citado, a comunicação direta entre o processador e a memória principal reduz consideravelmente as atividades de leitura e escrita, melhorando a eficiência geral do

processamento de dados.

No trabalho de (MARTINS[1]; POLETTO, 2016), é enfatizado que o IMDB se destaca por sua capacidade de otimizar algoritmos de movimentação de dados e resolver problemas de desempenho, utilizando a memória para carregar e acessar dados de forma mais eficiente. O IMDB é escalável tanto horizontal quanto verticalmente. Na escalabilidade horizontal, a adição de novos nós permite atender a um aumento na demanda de capacidade de memória, enquanto na escalabilidade vertical, o foco está em identificar e otimizar pontos de lentidão nas transações sem a necessidade de adicionar *hardware* adicional.

A implementação de escalabilidade horizontal é mais comum em aplicativos como redes sociais *(Facebook, Twitter, Instagram)*, onde as transações são mais independentes e exigem um grande número de operações exclusivas. Por outro lado, a escalabilidade vertical é direcionada para identificar e otimizar os gargalos de desempenho específicos de cada aplicativo, sem a necessidade de expansão do hardware (MARTINS[1]; POLETTO, 2016).

2.2 Arquitetura do IMDB

O IMDB se destaca pela sua simplicidade e velocidade, alocando dados diretamente na memória principal, conforme (SANTOS, 2013). Embora siga as propriedades Atomicidade, Consistência, Isolamento, Durabilidade (ACID), a durabilidade em memória volátil é uma preocupação, uma vez que a tecnologia de memória principal suporta os três primeiros ACID, mas não a durabilidade.

A consistência, crucial na arquitetura de um banco de dados, depende da conformidade com regras, incluindo chaves primárias, estrangeiras e restrições de campo, conforme mencionado por (SANTOS, 2013).

Devido ao gargalo da E/S em aplicações complexas, o IMDB se destaca como uma solução ao armazenar todos os dados e executar operações de E/S inteiramente na memória principal, oferecendo ganhos significativos de desempenho em comparação com os bancos de dados convencionais.

2.3 O que são dados quentes e frios?

Com o notável aumento na capacidade de armazenamento da memória principal, as OLTP têm a capacidade de armazenar dados integralmente na memória, como ressaltado por (SANTOS *et al.*, 2021). No entanto, o rápido crescimento exponencial dos dados em muitas aplicações, incluindo aquelas associadas ao *Big Data* (EMMANUEL; STANIER, 2016), apresenta desafios críticos, como o transbordamento de dados *(data overflow)*, que ocorre quando o volume de dados ultrapassa a capacidade da memória principal.

Nesse contexto, o OLTP desempenha um papel essencial na gestão eficiente dessas situações, identificando e tratando os dados menos acessados, muitas vezes denominados "dados frios", que são subsequentemente alocados em armazenamento secundário, conforme mencionado por (SANTOS *et al.*, 2021). Geralmente, as cargas de trabalho OLTP exibem padrões de acesso nos quais alguns registros são considerados “quentes”, sendo acessados com frequência, enquanto muitos outros são classificados como "frios", sendo raramente acessados ou quase nunca.

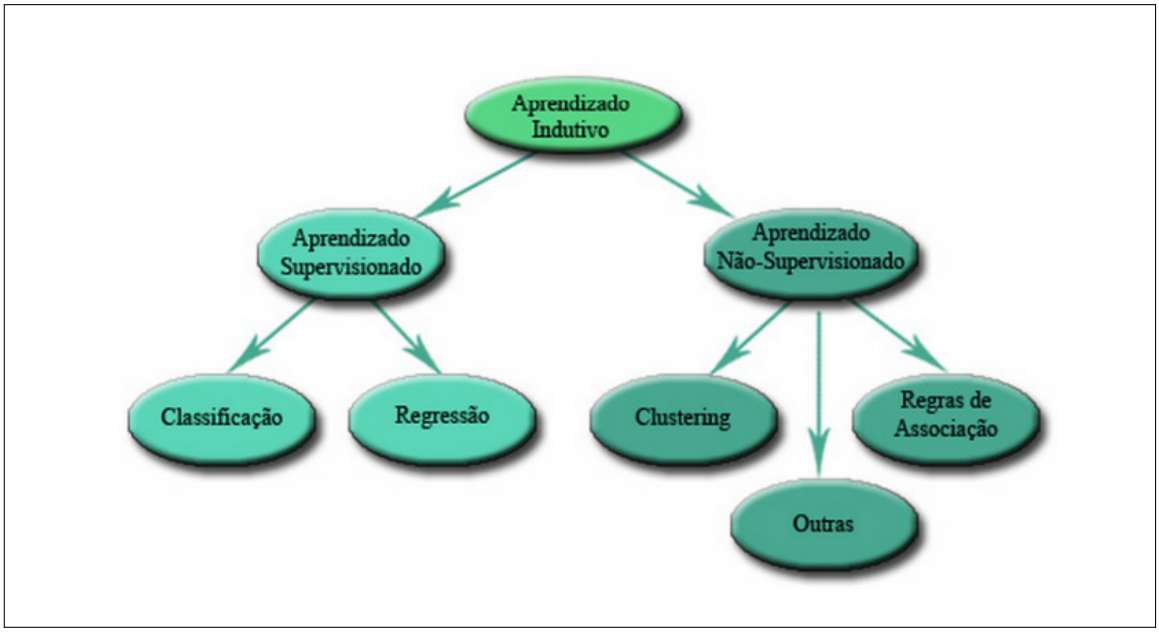
Recentes pesquisas, como apontado por (SANTOS *et al.*, 2021), têm se concentrado na abordagem do problema do transbordamento de dados, com o objetivo de separar de forma eficiente os dados “quentes” e “frios”. Muitos desses estudos destacam o uso das técnicas *Least Recently Used* (LRU) e *Least Frequently Used* (LFU) para distinguir entre dados quentes e frios.

A ênfase atual está na gestão dos dados frios, utilizando estratégias como despejo em disco, compressão ou mesmo exclusão, liberando assim espaço valioso na memória para dados mais ativos como os dados quentes.

2.4 Aprendizado de Máquina

No campo da Inteligência Artificial, o Aprendizado de Máquina (AM) é uma subárea que visa desenvolver modelos capazes de "aprender" com base na experiência, conforme enfatizado por (SCHMITT, 2013). Essa forma de aprendizado se concentra em algoritmos dedutivos, os quais se baseiam em estatísticas e são capazes de extrair regras e padrões a partir de conjuntos extensos de dados. Uma das principais técnicas empregadas no Aprendizado de Máquina é a indução, por meio da qual é possível alcançar conclusões gerais a partir de exemplos específicos. A indução é crucial no AM, representando o conceito de aprendizado através de inferências indutivas baseadas em exemplos. Entretanto, conforme ressalta (REZENDE, 2003), as hipóteses formuladas por meio desse processo podem ou não refletir a verdade com precisão, dependendo da análise resultante da indução (SCHMITT, 2013).

Figura 1 – Hierarquia do aprendizado indutivo



Fonte: (REZENDE, 2003)

A hierarquia do aprendizado indutivo, dividida em aprendizado supervisionado e não supervisionado, envolve a extração de conhecimento a partir de exemplos rotulados e não rotulados. O aprendizado supervisionado utiliza exemplos com entradas e saídas desejadas, enquanto o não supervisionado agrupa ou representa entradas com base em medidas de qualidade, revelando padrões nos dados analisados(SCHMITT, 2013).

A Inteligência Artificial (IA) é uma área de rápido crescimento que está cada vez mais presente no dia a dia da sociedade, impulsionando o foco das linguagens de programação no estudo do aprendizado de máquina e suas subáreas. Diversas aplicações práticas da IA já existem, abrangendo desde soluções de reconhecimento espacial até carros autônomos capazes de operar sem intervenção humana, e até mesmo simples sistemas de reconhecimento facial em dispositivos móveis. No campo do aprendizado de máquina, vários algoritmos são empregados para treinar e capacitar computadores a desenvolverem uma variedade de aplicações de inteli-

gência artificial, incluindo os métodos de aprendizado supervisionado, não supervisionado e por reforço (SAMPAIO, 2022).

Dentre esses métodos, destaca-se o Aprendizado Supervisionado (AS), que está por trás das aplicações mencionadas anteriormente utilizando o campo da IA. Conforme explicado por (SAMPAIO, 2022), o processo de aprendizado supervisionado envolve o uso de conjuntos de dados rotulados. Por exemplo, no contexto do reconhecimento de objetos, para que um computador seja capaz de identificar objetos em imagens, é necessário ter à disposição um conjunto de dados contendo imagens (exemplos) identificadas com os tipos de objetos correspondentes (rótulos). O aprendizado supervisionado incorpora várias abordagens que aprimoram a análise de dados entre diferentes modelos, incluindo as Redes Neurais Profundas (RNP). Esses modelos são construídos com base na hierarquia de camadas de Redes Neurais Artificiais (RNA), cujo objetivo é extrair atributos em vários níveis de detalhamento sobre o objeto modelado.

(SAMPAIO, 2022) destaca que as RNP requerem uma quantidade substancial de dados para sua geração adequada, embora sua aplicação permaneça limitada ao domínio específico. A escassez de dados rotulados pode restringir a construção de modelos robustos, e a Transferência de Conhecimento (TC) surge como uma solução viável. Essa técnica permite o uso de um modelo desenvolvido em um conjunto de dados abrangente (domínio de origem) para aprimorar o aprendizado de um modelo em um conjunto de dados reduzido (domínio alvo). Em (FERREIRA *et al.*, 2020), os autores ressaltam que o aprendizado de máquina tem apresentado resultados promissores em diversos setores, especialmente em problemas com grandes volumes de dados.

2.4.1 Long Short-Term Memory (LSTM)

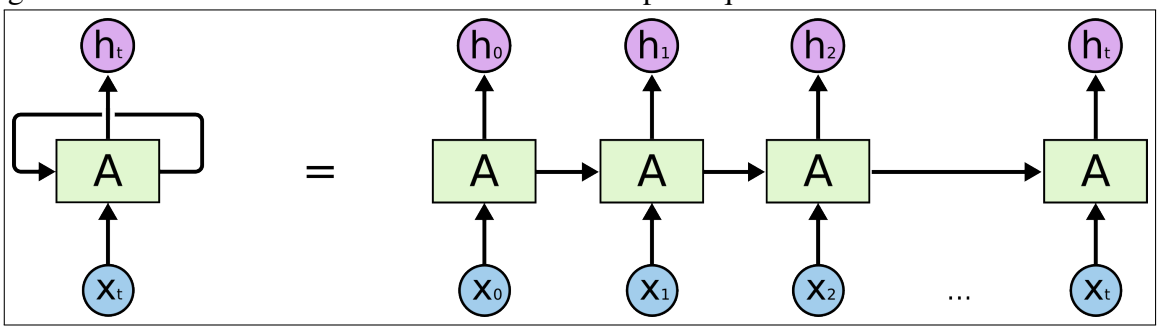
De acordo com (MOGHAR; HAMICHE, 2020), *Long Short-Term Memory* (LSTM) é um tipo de Rede Neural Recorrente (RNN) capaz de capturar dados de estágios anteriores e utilizá-los para previsões futuras. Em geral, uma Rede Neural Artificial consiste em três camadas: a camada de entrada, as camadas ocultas e a camada de saída. Em uma RNA que contém apenas uma camada oculta, o número de nós na camada de entrada depende da dimensão dos dados. Os nós da camada de entrada se conectam à camada oculta por meio de links conhecidos como "sinapses". Cada par de nós (da camada de entrada para a camada oculta) possui um coeficiente chamado peso, que desempenha um papel crucial na detecção de sinais. Durante o processo de aprendizagem, os pesos são ajustados continuamente, levando a uma otimização dos pesos ideais para cada sinapse na RNA.

Uma representação simplificada de uma rede neural recorrente pode ser vista na figura 2 mostrada à esquerda. O elemento é tal qual uma rede comum, onde observado no tempo *t* um elemento de entrada *xt* é fornecido à rede *A* e uma saída *ht* é fornecida pela rede. No entanto, pode ser observado um laço que permite que a informação de um instante de tempo seja fornecida à rede no próximo instante (SANTOS, 2019).

Porém, uma rede neural recorrente pode ser vista como uma conexão de várias redes neurais usuais, como na Figura 2, à direita. Isto acarreta que grande parte do estudo de outros tipos de redes neurais pode ser reaproveitado no estudo deste tipo de rede. Sabendo disso, redes neurais recorrentes necessitam de um contexto prévio para analisar entradas, sendo ideal quando o contexto está sequencialmente próximo. No entanto, em situações onde o contexto necessário está distante na sequência, resultando em dependências de longo prazo, redes neurais recorrentes simples podem não ser adequadas para lidar com esse desafio.

(SANTOS, 2019) apresenta uma representação simplificada de uma rede neural recorrente, destacando sua capacidade de transmitir informações ao longo do tempo por meio

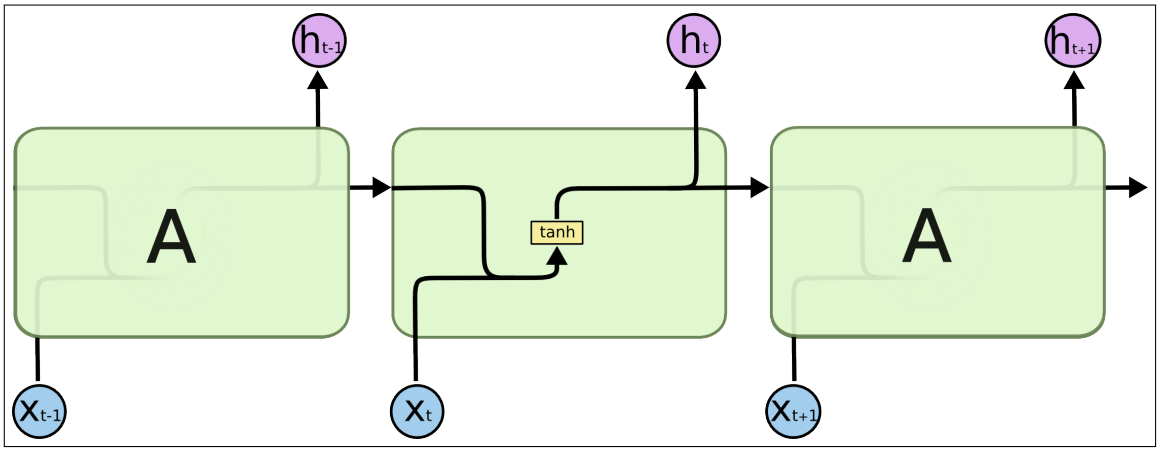
Figura 2 – Rede neural recorrente e rede neural simples equivalente



Fonte: (SANTOS, 2019)

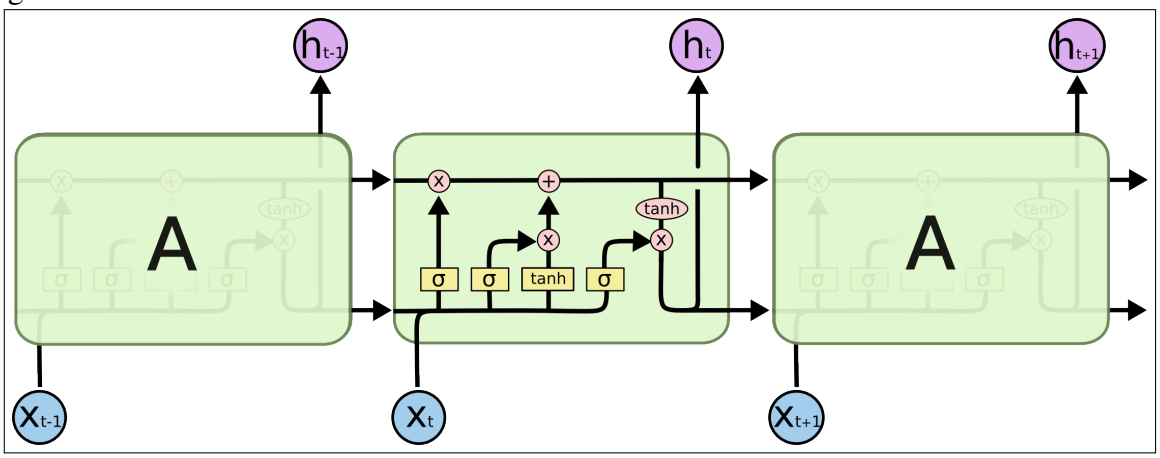
de um laço de retroalimentação. Já as redes LSTM, ao incluírem o estado da célula para transmitir informações, superam os desafio das dependências de longo prazo. Isso possibilita que as informações trafeguem quase intactas pela rede, uma vantagem sobre as redes recorrentes tradicionais que transmitem apenas a saída. A figura 3 e 4 mostra uma comparação entre duas arquiteturas em sua forma expandida.

Figura 3 – RNN simples



Fonte: (SANTOS, 2019)

Figura 4 – Rede LSTM



Fonte: (SANTOS, 2019)

Uma célula LSTM retém uma entrada por um período arbitrário de tempo, o que

impede que o gradiente se anule durante o treinamento com retro-propagação em redes LSTM. Ao contrário das redes recorrentes comuns, em que o gradiente pode se anular em caso de espaçamentos significativos na sequência entre eventos importantes, as redes LSTM podem ser treinadas usando os mesmos otimizadores e funções de perda que outros tipos de redes.

É importante ressaltar que uma rede LSTM apresenta quatro camadas na célula, enquanto uma rede recorrente simples possui apenas uma camada. Conforme (SANTOS, 2019), temos 3 camadas, que atuam como portas que regulam a passagem de informação. Cada porta é composta por uma função de ativação seguida por uma operação de multiplicação, e os pesos em cada porta são ajustados durante o treinamento em conjunto com a operação de ativação padrão. Em síntese, temos:

- Porta de entrada: controla a quantidade pela qual uma nova entrada influencia o valor da célula;
- Porta de esquecimento: controla a quantidade pela qual um valor persiste no estado da célula;
- Porta de saída: controla a quantidade pela qual o valor da célula contribui para a saída.

3 TRABALHOS RELACIONADOS

Nessa seção, será abordado alguns trabalhos correlatos que tratam de algoritmos para a identificação de dados quentes e frios em bancos de dados em memória. A escolha dos trabalhos a serem discutidos foi baseada em sua relevância para o escopo deste estudo e pelos resultados apresentados, buscando compreender as diferentes estrategias e técnicas propostas na literatura. Será destacado a pesquisa de (SANTOS *et al.*, 2021), que apresenta os novos algoritmos *2QCold* e *ARCold*, baseados nos algoritmos clássicos de cache *2Q* e *Adaptive Replacement Cache (ARC)*. Além disso, será discutido o estudo de (LEVANDOSKI *et al.*, 2013) sobre a identificação eficiente de dados quentes e frios em bancos de dados de memória principal, que propõe quatro algoritmos de classificação com base nos registros de acesso. Por fim, será analisado a técnica de política de substituição de *cache* de (CHOI; PARK, 2022), intitulada *"Learning Future Reference Patterns for Efficient Cache Replacement Decisions"*, que visa aumentar a taxa de acertos de cache.

3.1 Algoritmos para Identificação de Dados Frios em Bancos de Dados em Memória (SANTOS *et al.*, 2021)

O estudo aborda a criação e implementação de dois algoritmos, *2QCold* e *ARCold*, derivados de técnicas clássicas de *cache* 2Q e o *ARC*, com o propósito de identificar dados frios. A pesquisa envolve uma avaliação experimental detalhada, realizada em dois cenários, *online* e *offline*, utilizando o banco de dados *Seal-DB* (MORAES *et al.*, 2017), otimizado para memória principal. Os resultados incluem a análise do tempo de resposta e da taxa de acerto, comparando os algoritmos propostos com técnicas tradicionais, como o LRU, que de acordo com (STEINMACHER, 2004), é uma estratégia simples de substituição que não requer parametrização e "descarta"dados menos usados recentemente na memória. A avaliação se baseia em informações coletadas em consultas executadas no banco de dados e em arquivos de log, abordando diferentes cenários de execução. Além disso, o estudo também explora o desempenho

dos algoritmos em relação ao algoritmo de *Belady* para a avaliação da taxa de acerto.

O estudo de desempenho dos algoritmos *2QCold*, *ARCold* e LRU em um cenário *online* revelou uma ligeira redução no tempo de resposta, com *2QCold* e *ARCold* apresentando 1% e 5%, respectivamente, em comparação com o LRU. No cenário *offline*, o *2QCold* registrou uma redução significativa de 17% a 28% em relação ao *Forward*, enquanto o *ARCold* mostrou uma redução entre 17% e 23%.

A análise das taxas de acerto dos algoritmos *2QCold*, *ARCold*, LRU e *Belady*, tanto no cenário online quanto no offline, indicou aumentos significativos. No cenário online, *2QCold* e *ARCold* demonstraram um aumento de cerca de 0,6% e 27,25%, respectivamente, em comparação com o LRU. No cenário offline, os algoritmos *2QCold* e *ARCold* alcançaram aumentos de cerca de 1% e 27%, respectivamente, em relação ao *Forward*. Além disso, observou-se que ambos os algoritmos foram capazes de reduzir o tempo de resposta do sistema em 5% e 28%, respectivamente, e melhoraram a taxa de acerto na identificação de dados frios em 27% em comparação com as pesquisas relacionadas.

3.2 *Identifying Hot and Cold Data in Main-Memory Databases* (LEVANDOSKI *et al.*, 2013)

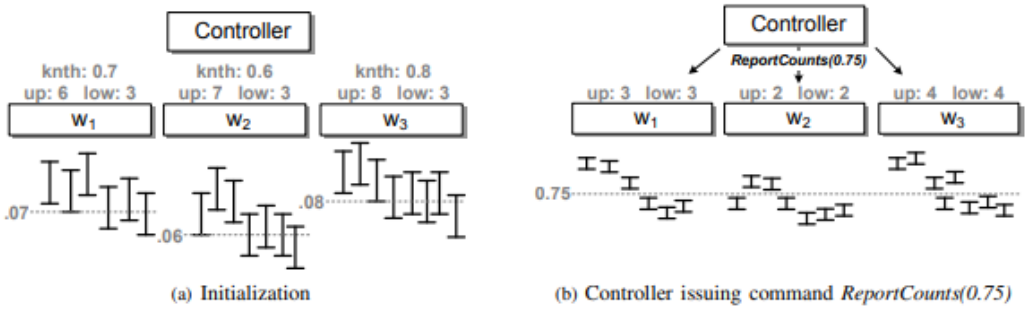
O trabalho de (LEVANDOSKI *et al.*, 2013) mostra que as cargas de trabalho OLTP consistem em acessos a registros com diferentes temperaturas, classificados como “quentes”, “mornos” e “frios”. O desempenho do sistema depende fortemente dos registros quentes, enquanto os registros frios podem ser movidos para armazenamento secundário. Seu trabalho se concentra em gerenciar efetivamente esses registros frios, identificando-os com precisão para a migração apropriada, a fim de otimizar o sistema OLTP em memória principal.

O estudo apresenta o *Hekaton*, um mecanismo OLTP otimizado para memória da *Microsoft*, juntamente com a Sibéria, um *framework* para gerenciamento de dados frios. A Sibéria possui quatro componentes fundamentais, incluindo classificação eficiente de dados frios, armazenamento adequado, mecanismos de acesso e migração eficientes, e redução do acesso desnecessário ao armazenamento frio, todos projetados para otimizar o desempenho do sistema e garantir uma transição transparente dos dados frios para o armazenamento secundário.

O estudo avaliou experimentalmente a taxa de acerto da *Exponential Smoothing* (ES) em comparação com duas técnicas de substituição de cache bem conhecidas,LRU-2 e *ARC*. Eles utilizaram 1 bilhão de registros para classificar os registros em ordem de acordo com sua frequência de acesso estimada, identificando os principais registros como “quentes”. A análise comparou o ES com um classificador perfeito fictício e revelou que o ES é consistente e preciso, mantendo uma perda na taxa de acerto abaixo de 1% para todos os tamanhos de dados quentes. Enquanto isso, o algoritmo LRU-2 também apresentou uma precisão razoável para tamanhos menores.

Como mostrado na Figura 5, a suavização exponencial ES foi comparada com duas técnicas de substituição de cache: LRU-2 e *ARC*. A perda na taxa de acerto foi avaliada em relação a um classificador perfeito, demonstrando que o ES é o mais consistente e preciso, mantendo uma perda na taxa de acerto abaixo de 1% para todos os tamanhos de dados quentes. O algoritmo LRU-2 se mostrou razoavelmente preciso para tamanhos menores, mas exibiu uma

Figura 5 – Comparação de Taxa de Acerto do ES com *ARC* e LRU-2



(a) Initialization (b) Controller issuing command *ReportCounts(0.75)*

Fig. 5. Backward parallel classification example

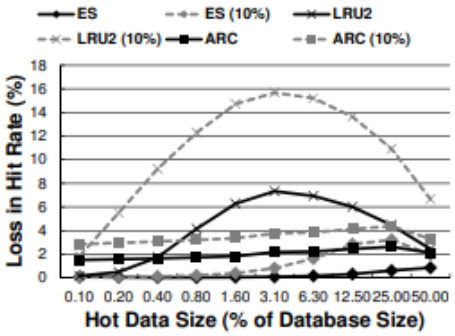


Fig. 6. Loss in hit rate (Zipf)

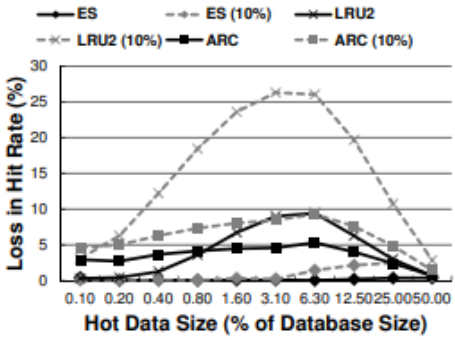


Fig. 7. Loss in hit rate (TPC)

Fonte: (LEVANDOSKI *et al.*, 2013)

perda de taxa de acerto tão alta quanto 7,8% para tamanhos maiores de *hot set*. Por outro lado, o ARC apresentou uma perda consistente de 2% na taxa de acerto, superando o LRU-2 na maioria dos tamanhos de dados quentes. Os experimentos relataram o desempenho dos algoritmos de classificação, considerando o tempo necessário para executar a classificação em um *log* de 1 bilhão de acessos, utilizando oito *threads* para os algoritmos paralelos. Os algoritmos foram testados com conjuntos de dados *Zipf* e *TPC-E*.

3.3 Learning Future Reference Patterns for Efficient Cache Replacement Decisions (CHOI; PARK, 2022)

O trabalho atual propõe uma técnica de política de substituição de cache com o objetivo de aumentar a taxa de acertos de cache, otimizando assim o gerenciamento e o desempenho do cache. Conforme descrito por (CHOI; PARK, 2022), a proposta visa empregar um método de aprendizado de máquina para antecipar os blocos que serão solicitados no futuro, visando evitar decisões inadequadas. A operação principal do método ocorre quando uma falha de cache ocorre, momento em que o modelo de aprendizado de máquina prevê uma sequência futura de referência de bloco com base na referência de bloco da sequência de entrada. O bloco previsto é adicionado ao *buffer* de previsão e removido do buffer sem acesso, se presente. Após preencher o *buffer* de previsão, a política de substituição convencional pode ser substituída por uma complexidade de tempo O(1), substituindo o bloco com um *buffer* sem acesso. Esse método aprimora o algoritmo menos recentemente usado LRU em 77%, o algoritmo menos frequentemente utilizado LFU em 65%, e o cache de substituição adaptável ARC em 77%. Além disso, fortalece a política heurística, garantindo desempenho consistente para cargas de trabalho compatíveis com LRU e LFU.

Nos experimentos mostrados, um total de 10 *datasets* foram usados, *web07* e *web12* foram usados para comparar a eficiência da política de substituição. Além disso, *cscope*, *vislumbre*, *postgres* e *sprites* são os *datasets* que são extraídos das várias aplicações. Como também, *2-pools*, *multi1, multi2*, e *multi3* são os datasets sintéticos que são obtidos executando aplicativos concorrentemente. A Figura 6 a seguir fornece uma descrição detalhada dos *datasets* que são usados nos experimentos.

Figura 6 – Descrição do conjunto de dados.

Dataset Name	Summary	Reference Count	Coverage BlockNo. (%)	Coverage Delta (%)
glimpse	Text information retrieval utility	6015	69.78%	100%
postgres	Joins queries among four relations in a relational database system	10448	76.67%	100%
cscope	An interactive C source program examination tool	6781	73.74%	100%
sprite	The Sprite network file system	133996	81.59%	64.77%
2-pools	Multi-user database	100000	58.45%	10.18%
multi1	cscope, cpp	15858	75.43%	70.84%
multi2	cscope, cpp, postgres	26311	62.83%	56.08%
multi3	cpp, gnuplot, glimpse, postgres	30241	56.36%	51.19%
web07	Product detail page HTTP requests	76118	51.86%	32.78%
web12	Product detail page HTTP requests	95607	66.59%	29.76%

Fonte: (CHOI; PARK, 2022)

Alguns dos vestígios mencionados no trabalhos são classificados como quatro tipos de padrões de acesso ao cache de arquivo:

- Padrão Sequencial: Todos os blocos são solicitados um após outro e nunca são acessados novamente;
- Padrão de *loop*: Todos os blocos são solicitados repetidamente após um intervalo regular (período);
- Padrões agrupados temporalmente: Os blocos que são solicitados mais recentemente são mais prováveis a ser solicitados em breve;
- Padrões Probabilístico: Cada bloco possui uma probabilidade de referência estacionária, e todos os blocos são solicitados independentemente de acordo com as probabilidades associadas.

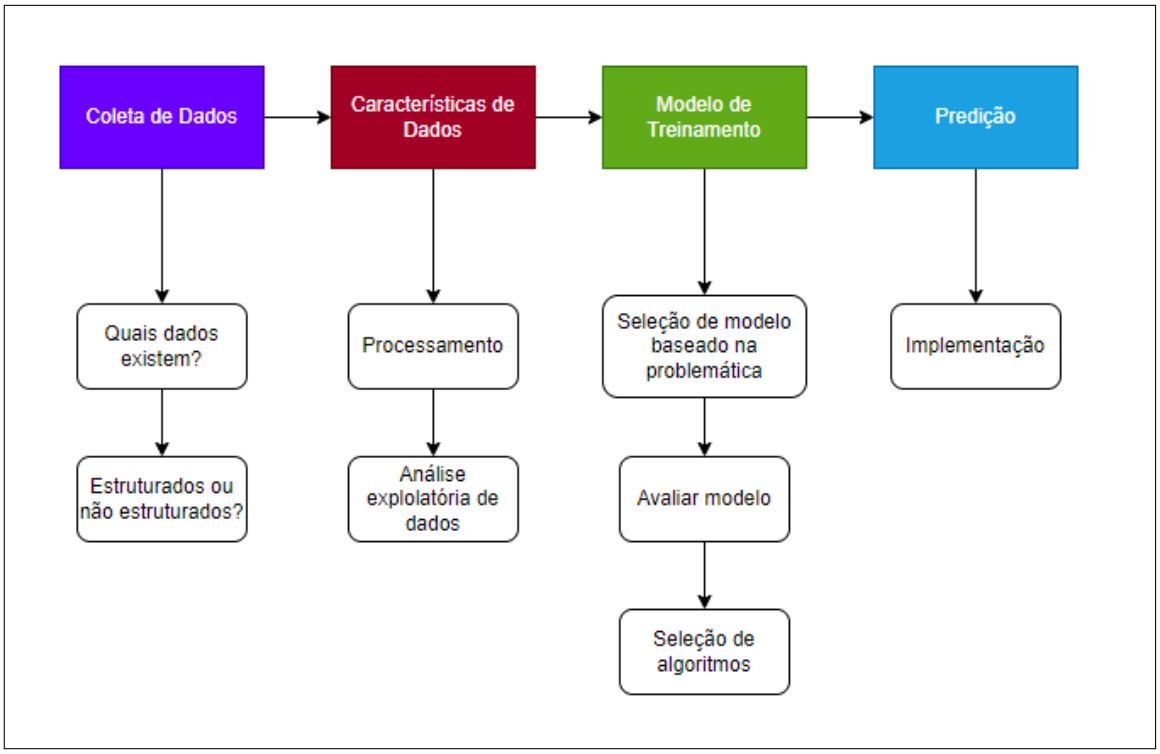
O estudo de (CHOI; PARK, 2022) introduz o modelo de aprendizado *Seq2Seq* para antecipar sequências futuras de blocos em um contexto de cache. O modelo utiliza um *buffer* de previsão e um método eficiente de *O(1)* para determinar alvos a serem descartados. Ao integrar o modelo à heurística existente da política de substituição de cache, o método supera LRU em 77%, LFU em 65%, e ARC em 77%. A combinação com políticas heurísticas, como LRU, permite adaptação a cargas de trabalho compatíveis. Em resumo, a proposta é uma política heurística de substituição de cache que incorpora informações passadas e futuras. (CHOI; PARK, 2022) sugere que essa abordagem, ao empregar aprendizado de máquina, tem o potencial de aprimorar a eficiência em diversos padrões de carga de trabalho.

4 METODOLOGIA E ABORDAGEM

Este trabalho foca na análise e validação de grandes conjuntos de dados em bancos otimizados para memória principal, buscando extrair *insights* significativos para compreensão dos padrões e tendências subjacentes.

Ao enfrentar os desafios inerentes ao processamento de grandes volumes de dados em tempo real, propusemos uma solução baseada em técnicas de aprendizado de máquina e análise estatística. Com a implementação de algoritmos e estratégias de validação meticulosamente planejadas, este estudo busca oferecer uma visão abrangente do comportamento dos sistemas em estudo. Priorizamos a compreensão e análise aprofundada dos dados extraídos, garantindo assim a confiabilidade e relevância das interpretações e previsões. Com esta abordagem inovadora e meticulosa, visamos preencher uma lacuna significativa na pesquisa atual, fornecendo uma base sólida para estudos e avanços futuros no campo dos sistemas baseados em memória principal.

Figura 7 – Pipeline Aprendizado de Máquina



Fonte: Do Autor

A metodologia adotada abrangeu a coleta e caracterização dos dados, seguidas pela análise exploratória. Após o processamento, implementou-se um modelo de treinamento, com escolha e avaliação baseadas na problemática. A fase de predição encerrou o processo, analisando as previsões em relação aos dados. Essa abordagem proporcionou uma compreensão aprofundada da identificação de dados quentes e frios com o conjunto de dados de um banco otimizado para memória.

4.1 Coletor de dados e estatísticas

A base de dados utilizada neste estudo foi coletada por meio de múltiplas requisições ao *Seal-DB*, um sistema de banco de dados educacional projetado para apoiar o ensino de conceitos relacionados a bancos de dados (MORAES *et al.*, 2017). Para uma descrição mais aprofundada, o *Seal-DB* fornece um ambiente prático para aprendizado, permitindo a interação dos usuários com consultas *SQL* e operações de manipulação de dados. O conjunto de dados gerado a partir dessas interações serviu como uma representação realista e dinâmica

de requisições sequenciais a um banco de dados em memória, contribuindo para a validação e teste do modelo LSTM proposto neste trabalho. Vale destacar que o conjunto de dados baseia-se no *benchmark Transaction Processing Performance Council* (TPC-C), que simula um ambiente de processamento de transações *online* OLTP, envolvendo consultas complexas e operações de manipulações de dados.

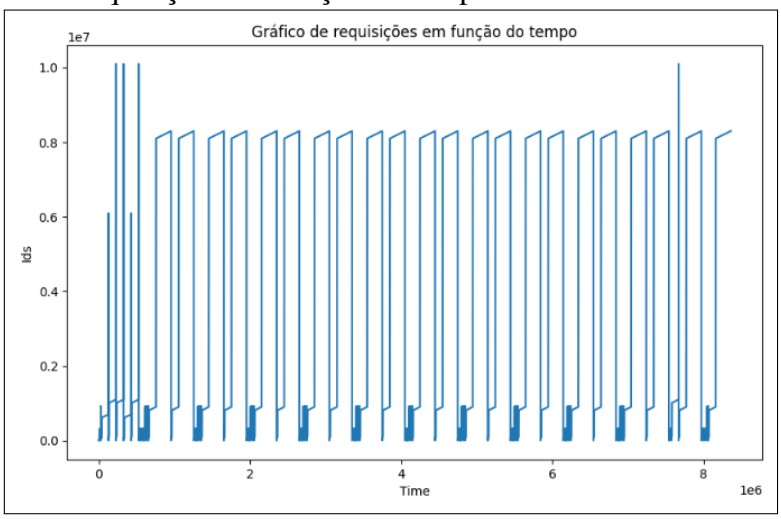
A Tabela 1 apresenta o conjunto de dados principal, que abrange uma ampla gama de requisições coletadas ao longo de um determinado período de tempo. As colunas incluem informações sobre o tempo das requisições, os *IDs* associados a cada requisição e os tipos de requisições. As primeiras e últimas linhas do conjunto de dados são apresentadas abaixo:

Time	**ID**	**Tipo de Requisição**
0	31	R
1	32	R
2	33	R
3	34	R
4	35	R
...	...	...
8361115	8299583	R
8361116	8299584	R
8361117	8299585	R
8361118	8299586	R
8361119	8299587	R

Tabela 1 – Conjunto de Dados de Requisições

Além disso, foram criadas visualizações gráficas para representar o padrão de movimento dos *IDs* em relação ao tempo. A Figura 8 ilustra o gráfico das requisições em função do tempo. A análise inicial dessas visualizações é crucial para compreender a distribuição e o comportamento geral dos dados.

Figura 8 – Gráfico de requisições em função do tempo.

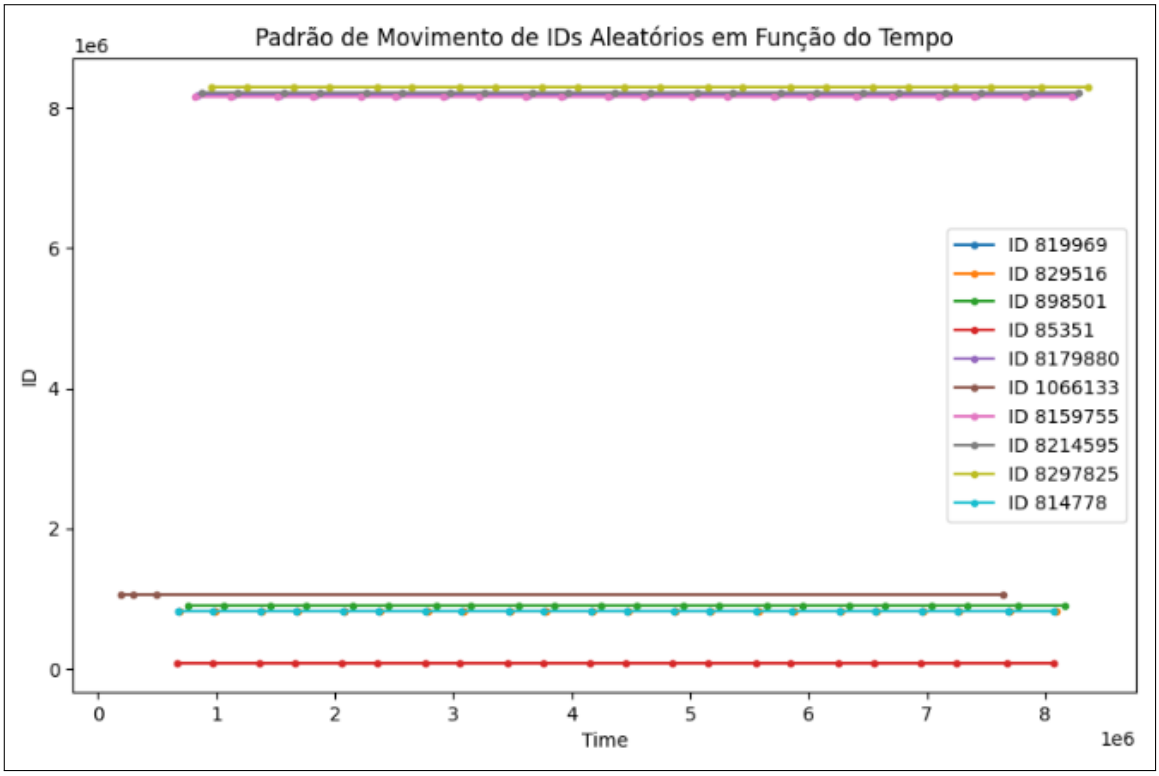


Fonte: Do Autor

A análise do padrão de movimento dos *IDs* ao longo do tempo revelou *insights* sobre o comportamento dos sistemas subjacentes. Por meio da representação gráfica da figura 8 nos

pontos de acesso dos IDs no eixo do tempo, observamos uma variedade de padrões que refletem a dinâmica das interações nos sistemas em estudo. Esses pontos de acesso, representados por pequenos pontos multicoloridos no gráfico, destacam os momentos específicos em que cada ID realizou uma requisição. A observação desses pontos ao longo do tempo oferece uma compreensão mais profunda da frequência e regularidade das atividades dos *IDs* no sistema.

Figura 9 – Padrão de movimento de *IDS* aleatórios em função do tempo.



Fonte: Do Autor

Além do mais, a identificação e compreensão desses padrões de acesso são fundamentais para o desenvolvimento de estratégias de previsão eficazes. Como visto na Figura 9, o padrão de acesso dos *IDs* irão variar, alguns são acessados no começo do tempo de acesso, visualizado pelo eixo y de time e ser acessados só posteriormente no fim, outros ja variam com o padrão de acesso, e assim por diante. A compreensão do comportamento passado dos *IDs* permite a implementação de técnicas preditivas que podem projetar o próximo ponto de tempo para cada *ID* com base nos padrões históricos identificados. Esta análise serve como um ponto de partida crucial para o próximo capítulo, onde discutiremos a implementação de técnicas de previsão baseadas em aprendizado de máquina para antecipar os tempos de acesso futuros dos *IDs* no sistema.

5 AVALIAÇÃO EXPERIMENTAL

Nesta seção, apresentamos os experimentos que foram realizados e os resultados obtidos. Configurações da máquina de testes segue sendo um *Aspire* A515-54G, com um processador Intel(R) Core(TM) i5 de 10º geração, 8GB de memória RAM.

Configuração/Resultado	**Detalhes**
`Método Utilizado`	LSTM com Bidirectional e GPU
`Separador de Conjunto de Treino`	70% do Dataset (df)
`Separador de Conjunto de Validação`	30% do Dataset (df_validation)
`Número de Épocas`	200
`Otimizador`	Adam
`Outras Configurações do Modelo`	Camada LSTM: 50 unidades, Ativação: ReLU
	Camada Dropout: Taxa de 0.2
	Camada Dense 1: Unidades 1, Regularização L1: 0.01
	Camada Dense 2: Unidades 1, Regularização L2: 0.001

Tabela 2 – Configurações do Algoritmo

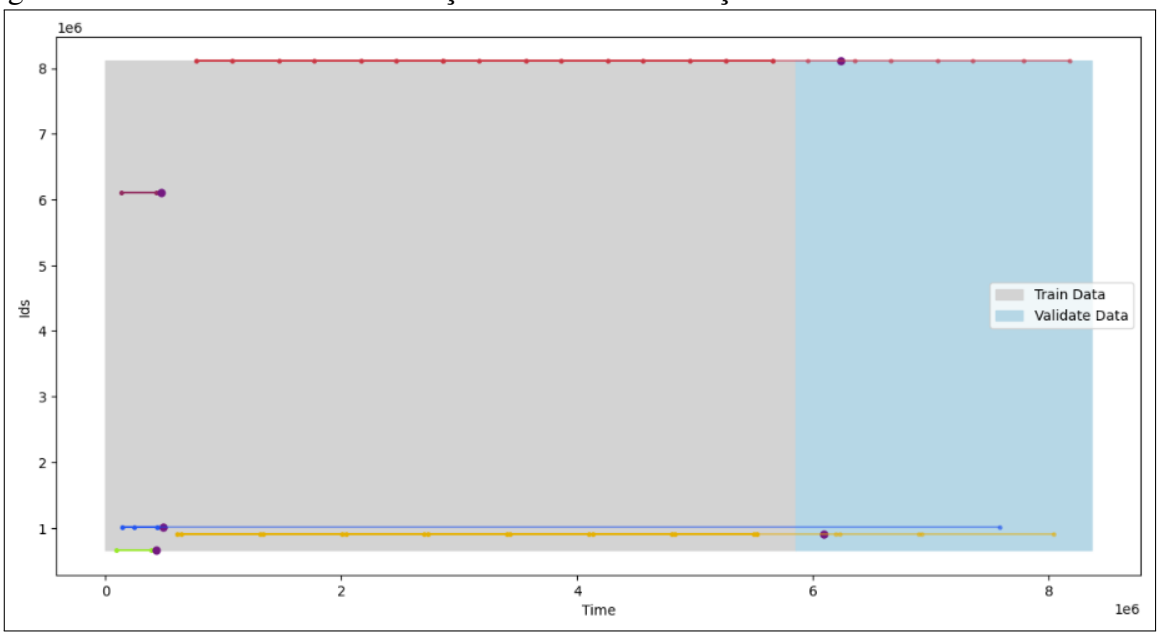
A Tabela **??** apresenta as configurações utilizadas pelo modelo LSTM aplicado no algoritmo. Ao incorporar a Graphics Processing Unit (GPU) na criação do modelo LSTM, buscamos otimizar o desempenho computacional durante o treinamento. A utilização do ambiente paralelo proporcionado pela GPU acelera significativamente o processamento das operações, permitindo uma análise mais eficiente das sequências temporais. A arquitetura do modelo é definida com uma camada bidirecional de LSTM, que é conhecida por capturar padrões temporais complexos. Além disso, é adicionada uma camada de *dropout* para mitigar o risco de *overfitting*. O uso de regularizadores nos neurônios densos contribui para a estabilidade do modelo. A compilação é feita com o otimizador *Adam* e o Erro Médio Absoluto (MAE), otimizando a rede para prever com precisão os padrões de tempo nas sequências. Esse aprimoramento visa aproveitar ao máximo os recursos de hardware disponíveis, melhorando a eficiência do treinamento e, consequentemente, a capacidade do modelo de generalizar padrões temporais em dados não vistos.

Em relação aos testes, é importante mencionar que foram conduzidos com uma seleção específica de *IDs*, escolhidos aleatoriamente devido às limitações de configuração do ambiente de teste. O conjunto de dados utilizado não foi extenso, mas representativo o suficiente para as avaliações realizadas. Considerando outros otimizadores, os resultados obtidos foram comparáveis ao Adam, porém muita das vezes obtendo resultados semelhantes, dado que o conjunto de dados apresenta uma predominância de requisições sequenciais. Dessa forma, devido à popularidade do Adam como otimizador eficaz para redes LSTM, foi mantido como a escolha principal, alinhando-se com a natureza sequencial do conjunto de dados em questão.

5.1 Resultados

A Figura 10 apresenta os resultados da predição de acessos utilizando a arquitetura LSTM. Cada linha multi-colorida corresponde a um *ID* específico, exibindo suas frequências

Figura 10 – Resultados da Classificação de *IDs*: Visualização com 5 *IDs*



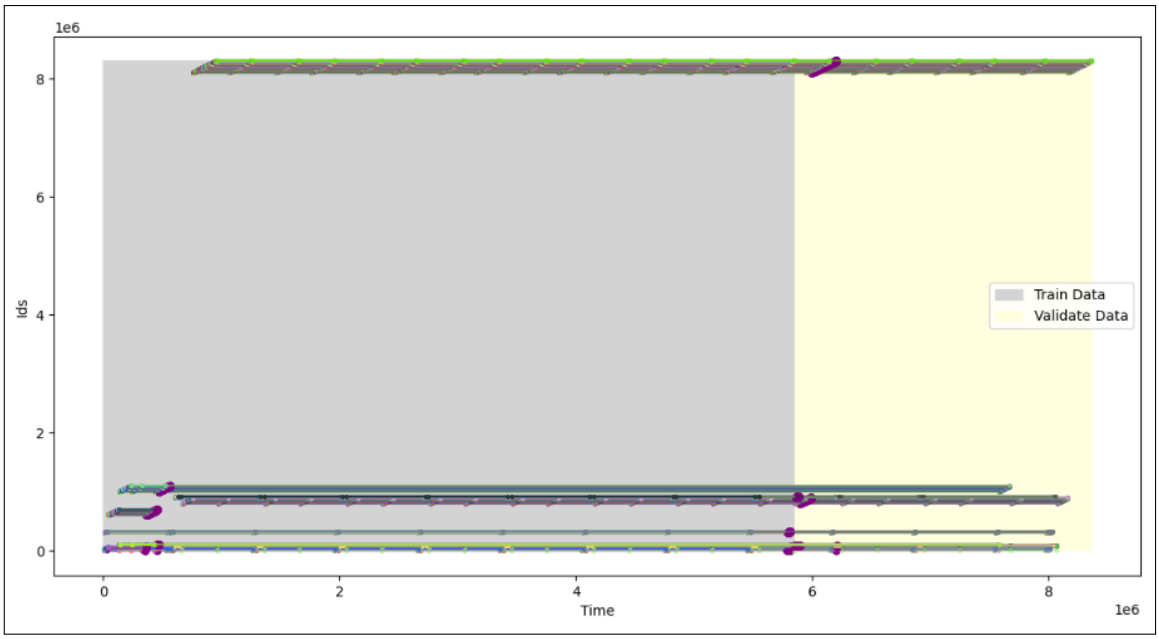
Fonte: Do Autor

temporais ao longo do tempo. Os pontos violetas representam as predições feitas pelo modelo LSTM para cada ID. Nesta amostra, o modelo identificou 2 *IDs* como quentes e 3 como frios. Vale notar que os *IDs* classificados como quentes exibem acessos futuros. No entanto, um desses *IDs*, apesar de prever um futuro acesso, teve apenas alguns no início do corte de treino, sendo considerado frio. Similarmente, os outros *IDs* classificados como frios possuem predições no conjunto de treino, indicando que provavelmente não serão acessados no futuro

Essas predições têm um papel crucial na classificação de *IDs* como quentes ou frios, pela forma que o modelo foi treinado, deixando *IDs* com possíveis aparições futuras em um corte do *dataset* de validação. *IDs* frios são caracterizados pela previsão do modelo, indicando que não serão frequentemente acessados em poucos intervalos de tempo. Isso é evidenciado pelas predições na amostra em questão desses *IDs* frios, presentes no corte cinza do gráfico que representa o conjunto de dados de treino. No entanto, é importante ressaltar que alguns *IDs* frios podem ter futuras aparições, embora esses eventos estejam espaçados por longos períodos de tempo. Por outro lado, *IDs* quentes são identificados pelas previsões no corte azul do gráfico, representado pelo corte de validação sugerindo que serão acessados em períodos mais próximos no futuro, de acordo com o modelo.

5.2 Teste com conjuntos de *IDs* aleátorios

Figura 11 – Resultados da Classificação de IDs: Visualização com 1000 *IDs*



Fonte: Do Autor

***IDs* Quentes**	***IDs* Frios**
586	414

Tabela 3 – Classificação de *IDs*

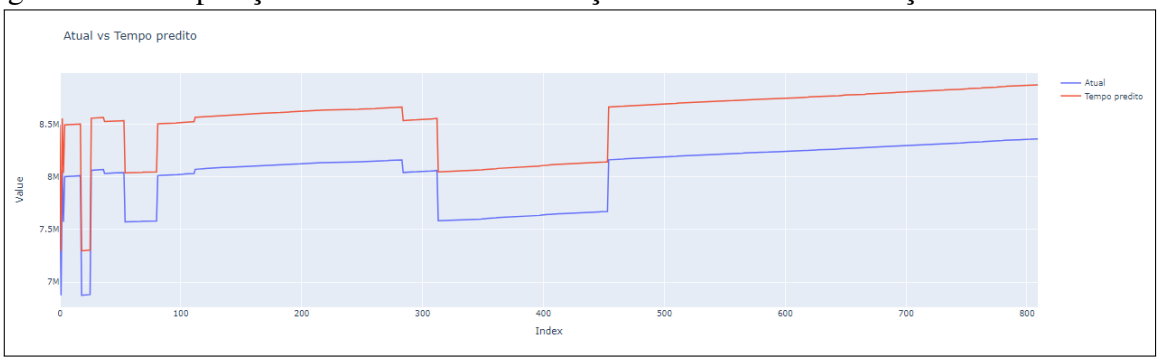
Para avaliar o desempenho do modelo em uma escala mais ampla, foram selecionados aleatoriamente conjuntos de cortes de *IDs* no *dataset*, cada um composto por **1000** *IDs*. A figura 11 resultante apresenta uma visão abrangente, revelando a capacidade do modelo de distinguir entre *IDs* "quentes" e "frios". Em uma das sessões de testes, é observado na Tabela 3 de **1000** *IDs*, que aproximadamente **414** foram classificados como "frios", enquanto **586** foram classificados como "quentes" com base nas predições nos dados de treino e validação do corte do *dataset* e no uso do modelo. Vale ressaltar que, devido à sobreposição de pontos no gráfico, a visualização detalhada pode ser desafiadora, contudo, a separação das predições é notável.

O desafio visual decorre da densidade de dados e da natureza sobreposta das predições, mas a capacidade do modelo em realizar a classificação é clara. É importante notar que, dada a limitação computacional do ambiente de teste atual, a análise de milhões de *IDs* não foi possível devido às restrições de memória.

O gráfico apresentado na Figura 12 compara os valores reais dos tempos de acesso (representados pela linha azul) com as previsões do modelo (representadas pela linha laranja) para os *1000 IDs* aleatórios no conjunto de validação. Notavelmente, os padrões de comportamento das previsões do modelo mostram uma notável semelhança com os valores reais de validação, isso também se da por conta que o *dataset* utilizado é composto por um formato de milhares de requisições sequenciais.

Essa congruência sugere que o modelo está eficientemente identificando os padrões

Figura 12 – Comparação Valores Reais vs Predições nos dados de validação



Fonte: Do Autor

temporais distintivos associados aos *IDs* quentes presentes no conjunto de validação. Esses resultados enfatizam a capacidade do modelo em antecipar com precisão os tempos de acesso para *IDs* com comportamentos futuros distintos. Isso destaca a correspondência entre as previsões do modelo e os valores reais, indicando sua eficácia na identificação de padrões temporais associados a *IDs* quentes no conjunto de validação.

6 CONCLUSÃO

Os resultados obtidos por meio da implementação do modelo LSTM para identificação e predição de dados quentes e frios em conjuntos de dados de banco de dados gerados pelo *Seal-DB* configurado em memória são promissores. Os gráficos de predição revelaram uma notável capacidade do modelo em classificar *IDs* quentes e frios nos dados de validação, destacando padrões sequenciais de acessos. A comparação entre as predições e os dados reais na validação evidenciou um desempenho consistente e uma significativa semelhança, especialmente em um contexto de conjuntos de dados compostos por milhares de requisições sequenciais. A robustez do LSTM em destacar *IDs* quentes e frios com base nas predições, antecipando acessos futuros, destaca seu potencial em cenários de gerenciamento de grandes volumes de dados no cenário de banco de dados em memória. Além disso, os dados coletados, especialmente aqueles classificados como frios, apresentam oportunidades para trabalhos futuros. Explorar estratégias como o envio desses dados para uma memória secundária, otimizando assim o uso da memória principal.

Esses resultados reforçam a viabilidade da aplicação do LSTM nesse contexto e fornecem uma base sólida para futuras explorações e otimizações do modelo. O enfoque na análise temporal e a possibilidade de considerar também a dimensão da distância de acesso abrem perspectivas interessantes para aprimorar ainda mais a compreensão e a previsão do comportamento de *IDs* em grandes conjuntos de dados.

6.1 Trabalhos Futuros

À medida que esta pesquisa avança na identificação e predição de dados quentes e frios através da implementação do modelo LSTM em conjuntos de dados de banco de dados em memória, uma série de possíveis direções para trabalhos futuros se destacam, oferecendo

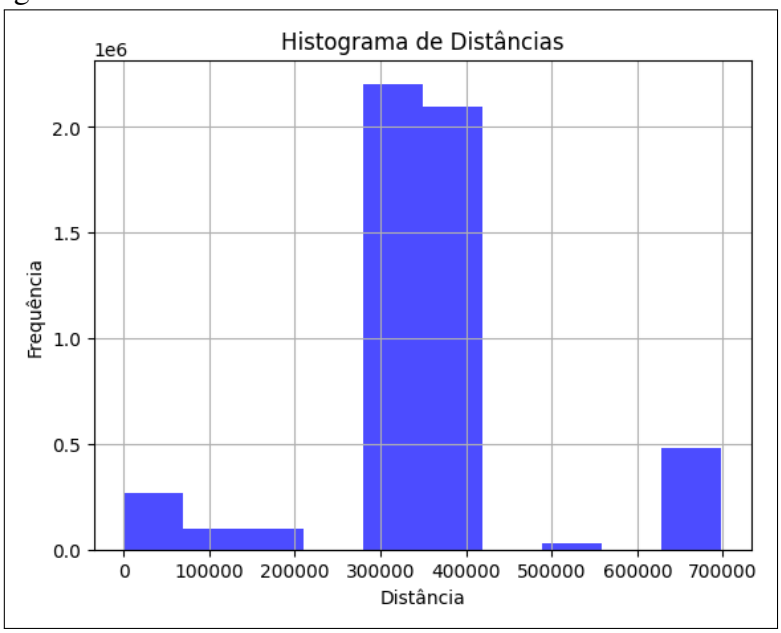
oportunidades empolgantes para expandir e aprimorar essa abordagem.

Dentre as potenciais melhorias, uma consideração crucial reside na otimização contínua do modelo LSTM. Explorar arquiteturas alternativas e ajustar hiper-parâmetros pode aprimorar significativamente a capacidade do modelo de generalização. Além disso, a aplicação de técnicas avançadas de regularização pode mitigar o risco de *overfitting*, contribuindo para predições mais robustas. Investigar a aplicabilidade do modelo em diferentes domínios e conjuntos de dados oferece *insights* importantes sobre sua adaptabilidade. Aprofundar a avaliação do desempenho, incorporando métricas específicas, proporciona uma compreensão mais refinada de como o modelo se comporta em cenários diversos.

Explorar estratégias para implementar o modelo em ambientes distribuídos, bem como investigar integrações com outras tecnologias emergentes, são áreas de pesquisa que podem melhorar a escalabilidade e lidar com desafios específicos, como problemas de privacidade e segurança.

6.2 Explorando Modelos na Distância de Acesso

Figura 13 – Histograma de Distância dos *IDs*



Fonte: Do Autor

Uma perspectiva promissora para pesquisas futuras é a expansão deste estudo para considerar não apenas o tempo, mas também a distância de acesso dos *IDs*. Isso poderia envolver a aplicação de modelos específicos para análise da distância, explorando como variações nesse aspecto afetam as predições. A construção de histogramas da distribuição da distância de acesso também seria uma ferramenta valiosa para compreender padrões não capturados pela análise temporal. Integrar dimensões temporal e de distância pode proporcionar uma visão mais completa do comportamento dos *IDs* em grandes conjuntos de dados.

A título exploratório, o histograma apresentado na figura 13 neste trabalho fornece uma visão inicial da distribuição das distâncias no conjunto de dados atual. As barras do

histograma sugerem um comportamento que, visualmente, assemelha-se a uma distribuição gaussiana. Investigar essa observação de forma mais sistemática e quantitativa pode abrir caminho para o desenvolvimento de modelos mais sofisticados baseados na distância de acesso.

REFERÊNCIAS

CHOI, H.; PARK, S. Learning future reference patterns for efficient cache replacement decisions. **IEEE Access**, IEEE, v. 10, p. 25922–25934, 2022.

EMMANUEL, I.; STANIER, C. Defining big data. In: **Proceedings of the International Conference on Big Data and Advanced Wireless Technologies**. [*S. l.*: *s. n.*], 2016. p. 1–6.

FERREIRA, F.; GRUENDEMANN, F.; ARAUJO, R.; YAMIN, A.; AGOSTINI, L. Avaliação do uso de aprendizagem de máquina na inferência de perfis de infusões intravenosas. **XXXVIII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS-SBrT**, v. 2020, p. 5, 2020.

LEVANDOSKI, J. J.; LARSON, P.-Å.; STOICA, R. Identifying hot and cold data in main-memory databases. In: IEEE. **2013 IEEE 29th International Conference on Data Engineering (ICDE)**. [*S. l.*], 2013. p. 26–37.

MARTINS[1], J. dos S.; POLETTO, A. S. R. de S. Um estudo exploratório acerca de banco de dados in-memory comparado aos bancos de dados convencionais. 2016.

MOGHAR, A.; HAMICHE, M. Stock market prediction using lstm recurrent neural network. **Procedia Computer Science**, Elsevier, v. 170, p. 1168–1173, 2020.

MORAES, G.; FILHO, J. d. A. M.; BRAYNER, A. Seal-db: Uma ferramenta de suporte ao aprendizado de banco de dados. In: **32th Brazilian Symposium on Databases DEMOS AND APPLICATIONS SESSION PROCEEDINGS**. [*S. l.*: *s. n.*], 2017. p. 35–40.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. [*S. l.*]: Editora Manole Ltda, 2003.

SAMPAIO, H. T. L. TransferÊncia de conhecimento com aprendizagem profunda para classificaÇÃo de textos em lÍngua portuguesa. p. 41, 2022.

SANTOS, A. V.; PINHEIRO, V.; MONTEIRO, J. M. Algoritmos para identificação de dados frios em bancos de dados em memória. In: SBC. **Anais do XXXVI Simpósio Brasileiro de Bancos de Dados**. [*S. l.*], 2021. p. 241–252.

SANTOS, R. P. dos. Um estudo exploratÓrio sobre bancos de dados in-memory. p. 33, 2013.

SANTOS, T. N. S. Aprendizado automático utilizando um modelo lstm aplicado como auxiliar no controle de orientação e velocidade de robô móvel. 2019.

SCHMITT, V. F. Uma análise comparativa de técnicas de aprendizagem de máquina para prever a popularidade de postagens no facebook. 2013.

STEINMACHER, I. Técnicas de web caching e prefetching com prioridades. **Trabalho Individual–Instituto de Informática, UFRGS, Porto Alegre**, 2004.