Um motor de busca para séries temporais baseado em Teoria da Informação para Cidades Inteligentes

Jordan A. Santos, Danilo Fernandes, Andre L. L. Aquino

[1]Orion Lab., Instituto de Computação
Universidade Federal de Alagoas
Maceió – AL – Brasil

{dfc,alla}@orion.ufal.br, jas2@ic.ufal.br

Abstract. *With the increasing urban digitalization, data lakes have become essential for storing and processing large volumes of data in smart cities, but their complex governance can turn them into data swamps. In this context, ubiquitous computing has emerged as a solution, enabling the continuous and decentralized processing of this data in real time, facilitating the analysis and integration of dispersed information in urban environments. This study proposes an efficient system for automatically identifying correlated time series by combining descriptors based on Information Theory and a vector database. This approach enables effective comparison of time series of different lengths while reducing computational costs. Experimental results show that ordinal pattern histograms outperform conventional statistical descriptors, proving the method's effectiveness for similarity search in big data environments.*

Resumo. *Com a crescente digitalização urbana, data lakes são essenciais para armazenar e processar grandes volumes de dados em cidades inteligentes, mas sua governança complexa pode levá-los a se tornarem data swamps. Nesse cenário, a computação ubíqua surge como uma solução, permitindo o processamento contínuo e descentralizado desses dados em tempo real, facilitando a análise e a integração de informações dispersas em ambientes urbanos. Este trabalho propõe um sistema eficiente para identificar automaticamente séries temporais correlacionadas, combinando descritores baseados na Teoria da Informação e um banco de dados vetorial. Essa abordagem permite comparar séries de diferentes tamanhos de forma eficaz, reduzindo custos computacionais. Os resultados experimentais mostram que histogramas de padrões ordinais superam descritores estatísticos convencionais, comprovando a eficácia do método na busca por similaridade em ambientes de big data.*

1. Introdução

Com o avanço da digitalização urbana, cidades inteligentes estão cada vez mais dependentes da coleta e análise de grandes volumes de dados para otimizar infraestrutura, mobilidade, segurança e sustentabilidade (Ramos et al. 2023). Nesse contexto, os *data lakes* surgem como uma solução essencial para armazenar dados heterogêneos provenientes de sensores IoT, câmeras de vigilância, sistemas de transporte e outros dispositivos conectados (Fernandes et al. 2023). Diferente dos *data warehouses*, que exigem um modelo de

dados estruturado, os *data lakes* permitem a ingestão de informações em diferentes formatos, impulsionando análises avançadas e tomadas de decisão em tempo real (Gorelik 2016).

Contudo, essa ampliação dos horizontes na extração de valor a partir dos dados implica no aumento da complexidade de sua governança (Ramos et al. 2023). Consequentemente, tornam-se imprescindíveis mecanismos de descoberta de dados capazes de permear a obscuridade crescente de um repositório de dados sem um modelo robusto (Fernandes et al. 2024). Caso contrário, o *data lake* vem a degenerar em um inutilizável *data swamp* (Sawadogo & Darmont 2020). Dentre as abordagens para a descoberta de dados, está a busca por similaridade entre dados heterogêneos (Hai et al. 2023).

Essa abordagem é amplamente investigada para dados textuais não-estruturados, os quais são transformados em vetores por uma rede neural *autoencoder* (Hai et al. 2023). Tal vetorização dos dados permite a mensuração de similaridade entre os mesmos e, portanto, a busca dos dados mais próximos a um determinado dado de entrada (Wang et al. 2021). A essa finalidade, foram propostos bancos de dados vetoriais capazes de realizar consultas por meio de variadas métricas de similaridade (Pan et al. 2024). Além disso, há dentre estes, bancos de dados distribuídos capazes de gerenciar volumes massivos de dados característicos de *big data* (Pan et al. 2024).

Entretanto, estima-se que, em uma cidade inteligente, a maioria dos dados são séries temporais provenientes de dispositivos IoT (Ramos et al. 2023). Em um único carro, por exemplo, existe uma gama de sensores que medem, em uma alta frequência, os estados dos seus mais diversos componentes incluindo temperatura do motor, nível de combustível, velocidade e localização. Mediante ao trânsito de carros, esses dados se integram e seu volume rapidamente escala a um *big data*. Tais dados são importantíssimos no contexto de transportes inteligentes, pois refletem o estado do tráfego na cidade e sua qualidade de vida (Fernandes et al. 2023).

Devido à escassez de informação semântica, a integração de séries temporais torna-se um desafio (Yu et al. 2022). No entanto, ao considerar uma cidade inteligente como um sistema de funcionamento orgânico, diferentes séries temporais podem descrever o mesmo fenômeno sob distintas perspectivas. Assim, a análise conjunta dessas séries proporciona uma compreensão mais abrangente e precisa do fenômeno em questão. Por exemplo, um incêndio em um prédio pode ser percebido pelos sensores de temperatura, umidade, luminosidade e fumaça. Em um *data lake* para uma cidade inteligente, um dos desafios seria a identificação dessas séries temporais correlacionadas. Segundo a arquitetura proposta por Fernandes et al. (2023), tal funcionalidade seria desempenhada por um motor de busca atrelado ao sistema de metadados.

Este trabalho enfrenta esse desafio ao propor um mecanismo eficiente para a busca de séries temporais correlacionadas. O sistema é composto por um extrator de descritores e um banco de dados vetorial. Os descritores consistem na caracterização através de Teoria da Informação, a qual permite a extração de um histograma de padrões ordinais que resumem a série temporal (Bandt & Pompe 2002). Isso viabiliza a comparação de séries de diferentes tamanhos, ao mesmo tempo que reduz os custos de armazenamento e processamento durante as buscas. Armazenados os descritores, o banco de dados vetorial possibilita a busca eficiente de séries com padrões similares em um ambiente de *big data*.

Desse modo, pode-se definir a pergunta de pesquisa deste trabalho como "***Dado um conjunto de séries temporais em um data lake, como identificar automaticamente um subconjunto similar a uma série temporal de entrada, considerando a ubiquidade dos dispositivos e a geração contínua de dados em ambientes urbanos conectados?***" Essa pergunta foi respondida favoravelmente mediante a proposição de uma arquitetura e uma avaliação experimental. Nesta, é comparada a efetividade dos descritores da Teoria da Informação em relação aos descritores estatísticos convencionais utilizados para relacionar séries temporais. Logo, a hipótese a ser verificada experimentalmente é se ***"Histogramas de padrões ordinais possibilitam a busca de séries temporais similares em bancos de dados vetoriais de forma mais efetiva que descritores estatísticos clássicos"***. Os resultados experimentais ofereceram evidências que corroboraram para aceitar essa hipótese em alguns cenários. Ao compararmos diretamente os histogramas de padrões ordinais com os descritores estatísticos clássicos, obtemos indícios que a nossa solução se sobrepõe ao considerar a proximidade direta entre as séries.

O restante do trabalho está organizado em cinco partes. Seção 2 descreve alguns trabalhos relacionados. Seção 3 detalha a metodologia e os resultados são apresentados na seção 4. Por fim, a seção 5 propõe alguns trabalhos futuros.

2. Trabalhos Relacionados

Saeedan & Eldawy (2022) propuseram o formato Spatial Parquet, que usa histogramas para particionar e indexar dados geoespaciais em colunas, acelerando consultas espaciais e reduzindo a leitura de arquivos completos. Assim, o uso de histogramas na organização de dados em colunas reduz significativamente a carga computacional ao minimizar a quantidade de dados que precisa ser escaneada.

Tang et al. (2025) desenvolveram o QueryArtisan, que emprega histogramas para estimar distribuições de dados e ajustar dinamicamente planos de consulta, otimizando a execução em ambientes distribuídos. Dessa forma, a ferramenta possibilita que consultas complexas, especialmente aquelas que envolvem grandes volumes de dados distribuídos, sejam processadas de forma mais rápida e eficiente, melhorando a experiência do usuário e reduzindo custos computacionais.

Grzegorowski et al. (2021) demonstraram que histogramas podem guiar a alocação dinâmica de clusters conforme padrões de uso, evitando desperdício de recursos e melhorando a escalabilidade. Como resultado, o processamento de grandes volumes de dados torna-se mais eficiente, evitando o desperdício de recursos computacionais e melhorando a escalabilidade do sistema.

Weng et al. (2021) apresentaram um método de data hiding baseado em histogramas para selecionar subconjuntos representativos em clustering, reduzindo o tempo de treinamento e mantendo a qualidade dos modelos. Isso reduz o tempo necessário para o treinamento de modelos de aprendizado de máquina e melhora a qualidade dos resultados, tornando os histogramas uma ferramenta essencial para a análise de dados em grande escala.

Wang et al. (2021) criaram o Milvus, um gerenciador de dados vetoriais que suporta vetores de alta dimensão com consultas rápidas e escaláveis, usando CPUs e GPUs via SDKs e APIs RESTful. O sistema oferece suporte a interfaces amigáveis (SDKs

e APIs RESTful), aproveita plataformas heterogêneas com CPUs e GPUs modernas e permite processamento avançado de consultas além da simples busca por similaridade vetorial.

Yu et al. (2022) propuseram um framework semântico de metadados para data lakes industriais de WoT, combinando descritores clássicos de séries temporais e extração automática de entidades para gerenciar dados heterogêneos. O framework proposto aborda séries temporais com descritores clássicos. A validação experimental em um conjunto de dados do mundo real demonstra a eficácia do método, destacando seu potencial para aprimorar a análise inteligente e o gerenciamento adaptativo em ambientes industriais dinâmicos.

Enquanto os trabalhos relacionados exploram aplicações específicas de histogramas e gestão de dados, o método aqui proposto diferencia-se ao combinar descritores derivados da Teoria da Informação e um banco de dados vetorial para identificar séries temporais correlacionadas. Nossa abordagem usa histogramas de padrões ordinais para comparar séries de diferentes tamanhos com baixo custo computacional.

3. Metodologia

O motor de busca proposto neste trabalho compreende duas fases: ingestão e consulta. A primeira descreve o comportamento do sistema quando uma nova série deve ser armazenada no *data lake* (Figura 1). Quando é acionado um novo *pipeline* de ingestão de dados, sua execução deve copiar a série de sua fonte e transferi-la para o sistema de arquivos distribuídos do *data lake*. Seguidamente a série deve ser encaminhada para o motor de busca junto com sua referência no sistema de arquivos. No motor de busca, essa será processada pelo extrator de descritores, resultando em um vetor de dados o qual será armazenado junto com a referência no banco de dados vetorial.

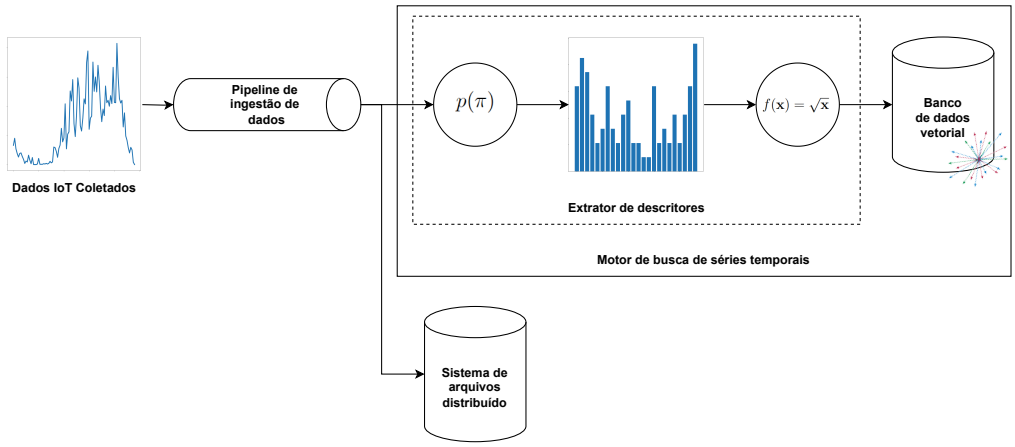

Figura 1. Ingestão de série temporal no *data lake*

Na segunda fase (Figura 2), uma série é enviada para *data lake* por meio de uma API como insumo para a descoberta de dados. Essa série é encaminhada para o motor de busca, onde é processada pelo extrator de descritores e com estes é realizada uma busca por similaridade no banco de dados vetorial. Obtidos os descritores mais próximos, suas respectivas referências são empregadas para recuperar as séries temporais que representam. Por fim, estas são enviadas como retorno da API.

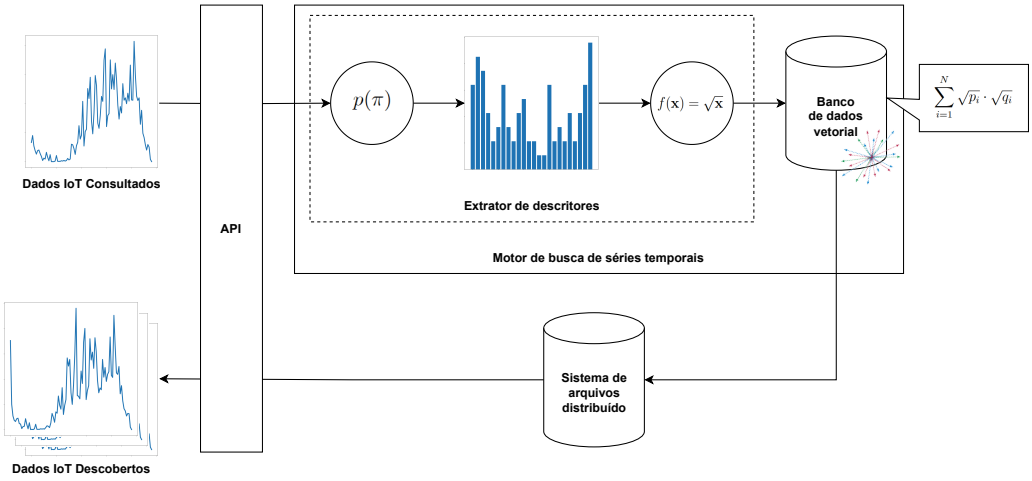

Figura 2. Consulta de séries temporais no *data lake*

O extrator de descritores fundamenta-se no método não paramétrico proposto por Bandt & Pompe (2002) para a análise de séries temporais. Neste, considera-se a série temporal $\{x_t\}_{t=1,\ldots,n}$ composta por n elementos e também vetores d-dimensionais $(d > 1)$ definidos por

$$(s) \to (x_{s-(d-1)}, x_{s-(d-2)}, \ldots, x_{s-1}, x_s),$$

onde $s = d, d+1, \ldots, n$. Para cada um dos $(n-d+1)$ vetores, computa-se a permutação $\pi = (r_0, r_1, \ldots, r_{d-1})$ de $(0, 1, \ldots, d-1)$ definida por $x_{s-r_{d-1}} \leq x_{s-r_{d-2}} \leq \cdots \leq x_{s-r_1} \leq x_{s-r_0}$. As $d!$ possíveis permutações de π são os estados acessíveis de um sistema e, para cada estado, a probabilidade de seu padrão ordinal é

$$p(\pi) = \frac{\#\{s | s \leq n-d+1; (s) \text{ possui o tipo } \pi\}}{n-d+1}, \tag{1}$$

onde o simbolo $\#$ indica o número de ocorrências da permutação π.

Dada as distribuições de probabilidade de padrões ordinais de duas séries temporais, a similaridade entre essas pode ser computada através do coeficiente de Bhattacharyya (Bhattacharyya 1943). A versão discreta desse coeficiente é

$$BC(p, q) = \sum_{x \in \mathcal{X}} \sqrt{p(x)} \cdot \sqrt{q(x)}. \tag{2}$$

onde p e q são funções de distribuição de probabilidade e $\mathcal{X}$ é conjunto de estados possíveis. Estebelecida uma ordem entre os elementos de $\mathcal{X}$, a Eq. 2 pode ser reescrita da seguinte forma conforme

$$BC(p, q) = \sum_{i=1}^{M} \sqrt{p_i} \cdot \sqrt{q_i}. \tag{3}$$

Contudo, bancos de dados vetoriais em geral disponibilizam a similaridade do cosseno para avaliar a proximidade entre os vetores. A fim de calcular o coeficiente de Bhattacharyya por meio da similaridade do cosseno, armazena-se no banco de dados o

vetor p^* para todo histograma p, onde $p^* = \sqrt{p} = (\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_M})$. Com efeito, para os vetores $p^* = \sqrt{p}$ e $q^* = \sqrt{q}$, temos que

$$\cos(p^*, q^*) = \frac{p^* \cdot q^*}{||p^*|| \cdot ||q^*||} = p^* \cdot q^* = \sum_{i=1}^{M} \sqrt{p_i} \cdot \sqrt{q_i} = BC(p, q), \tag{4}$$

visto que

$$||p^*|| = \sum_{i=1}^{M} \sqrt{p_i}^2 = \sum_{i=1}^{M} p_i = 1. \tag{5}$$

Analogamente, temos que $||q^*|| = 1$. Tais transformações sobre os dados, realizadas no motor de busca, são ilustradas nas Figuras 1 e 2.

4. Análise Experimental

A avaliação experimental deste trabalho foi conduzida utilizado uma amostra de um conjunto de dados disponibilizados pelo NetMob 2023 (Martínez-Durive et al. 2023). Este consistiu em um desafio para o desenvolvimento de métodos de análises de dados para consumo de dados de telefonia móvel na França. Neste trabalho, considerou-se a amostra que descreve o consumo de *uplink* do aplicativo *WhatsApp* no dia 16 de março de 2019, na cidade de Lyon, França.

Este subconjunto de dados descreve o consumo por meio do particionamento da cidade em regiões retangulares. Cada região é descrita por um série temporal representando o consumo a cada 5 minutos. Deve-se salientar que o consumo foi descaracterizado multiplicando todo o conjunto do desafio por um número aleatório. Ou seja, os dados foram anonimizados, não sendo possível determinar a medida de transmissão. (Martínez-Durive et al. 2023). Um exemplo de tais séries de consumo de dados é mostrado na Figura 3 para a região 1214. Nesse subconjunto foi realizada uma amostragem aleatória de 1000 regiões, onde 700 foram destinadas a inserção no *data lake* e 300 para a realização de consultas.

Figura 3. Série temporal da região 1214

Para a instanciação da arquitetura, escolheu-se o Milvus como banco de dados vetorial (Wang et al. 2021). A decisão justifica-se por sua robustez em cenários de *big data* e sua ampla gama de recursos em face dos sistemas concorrentes (Pan et al. 2024). Além

disso, o Milvus é um sistema distribuído *open source*. Para extrair os padrões ordinais das séries temporais, foi utilizado a biblioteca *ordpy*. Esta biblioteca foi criada para o Python por Pessa & Ribeiro (2021) com o intuito de aplicar esta e outras manipulações em séries temporais, com base nos conceitos de entropia de permutação de (Bandt & Pompe 2002). Como parâmetros da análise de permutação, utilizou-se uma janela de deslizamento $d_x = 3$ e um atraso $\tau_x = 1$. Estes parâmetros foram definidos seguindo a recomendação em Pessa & Ribeiro (2021), sendo a calibração objetivo de estudo adicional.

A fim de verificar a hipótese inicial deste trabalho, que é ***"Histogramas de padrões ordinais possibilitam a busca de séries temporais similares em bancos de dados vetorias de forma mais efetiva que descritores estatísticos clássicos"***, comparou-se nossa abordagem de vetorização com a de Yu et al. (2022). Estes consideraram a maior variedade de descritores que incluem média, mediana, valores máximo e mínimo, volume, histograma, frequência de amostragem e periodicidade da série temporal. Para determinar o número de intervalos do histograma, utilizou-se a regra da raiz quadrada. Como cada série possui $n = 96$ elementos, então temos $k = \lceil \sqrt{n} \rceil = \lceil \sqrt{96} \rceil = 10$ intervalos. Como métrica nesse espaço de descritores, os mesmos autores indicaram a distância euclidiana. Desse modo, a abordagem concorrente possui um espaço descricional de 17 dimensões, enquanto nossa proposta considera $3! = 6$ dimensões.

No Milvus, foi criada uma coleção separada para cada método. Cada coleção foi populada, segundo seu respectivo método, com os vetores que representam as séries temporais armazenadas no *data lake*. As mesmas consultas foram realizadas em ambas as coleções, considerando cada qual suas respectivas métricas. Para cada consulta, foram retornados os 10 vetores mais próximos e, a partir destes, foram obtidas suas respectivas séries. Assim, para cada série de entrada, obteve-se dois conjuntos de séries similares, um para cada conjunto de descritores.

A fim de avaliar a qualidade das consultas, computou-se a similaridade entre as séries de entrada e aquelas resultantes de suas respectivas consultas. Como o *dataset* descreve o consumo de dados por região, temos que regiões próximas possuem padrões de consumo parecidos, ou seja, séries similares (Martínez-Durive et al. 2023). Contudo, estas possuem escalas distintas, visto que regiões mais centralizadas possuem maior volume de dados consumidos do que a sua vizinhança (Martínez-Durive et al. 2023). Para contornar essa diferença de escala, durante a avaliação, as séries foram normalizadas por seu volume total. Dada a série temporal $\{x_t\}_{t=1,\ldots,n}$, sua versão normalizada é $\{y_t\}_{t=1,\ldots,n}$ onde y_t é

$$y_t = \frac{x_t}{\sum_{t=1}^{n} x_t}. \tag{6}$$

Desse modo, a similaridade para avaliação da qualidade foi calculada por meio da distância euclidiana entre as séries temporais normalizadas. Logo, reflete, entre duas regiões, a diferença geométrica percentual do consumo de dados em relação ao volume total do referido dia. Essa interpretação possibilita compreender a diferença qualitativa entre os padrões de consumo quantificada pela métrica de avaliação. A Figura 4 dispõe os *boxplots* das distâncias euclidianas observadas entre 10 séries de entrada e suas respectivas saídas de consulta segundo cada abordagem. Nessas consultas, observou-se uma superioridade da abordagem proposta sobre o concorrente da literatura.

A Figura 5 apresenta uma série inserida como entrada em uma consulta no motor

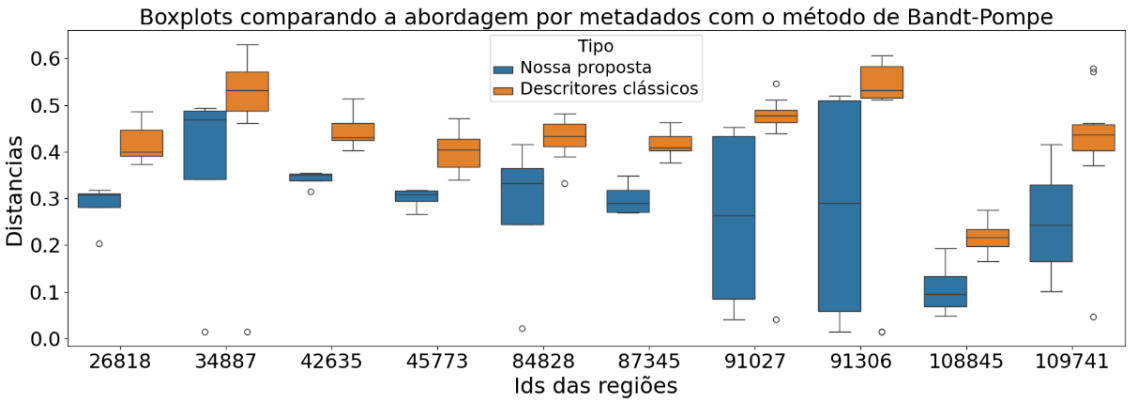

Figura 4. Boxplots das distâncias euclidianas

de busca e duas séries similares obtidas segundo a nossa abordagem e de Yu et al. (2022). Nesse exemplo, a série obtida através da primeira foi mais próxima à série de entrada do que aquela obtida pela estratégia concorrente. Deve ser salientado que esse *baseline* do cálculo direto das distâncias entre as séries é realizável nesse contexto experimental em decorrência do conjunto de dados possuir apenas séries de igual tamanho. Isso não ocorre necessariamente em cenários reais de tal modo que as buscas possam ocorrer baseadas nessas distâncias diretas. Além disso, tal abordagem demandaria $16\times$ mais espaço em disco e tempo em computações do que a abordagem de histogramas de padrões ordinais com $d_x = 3$. Essa diferença de consumo de recursos cresce linearmente com o tamanho das séries armazenadas.

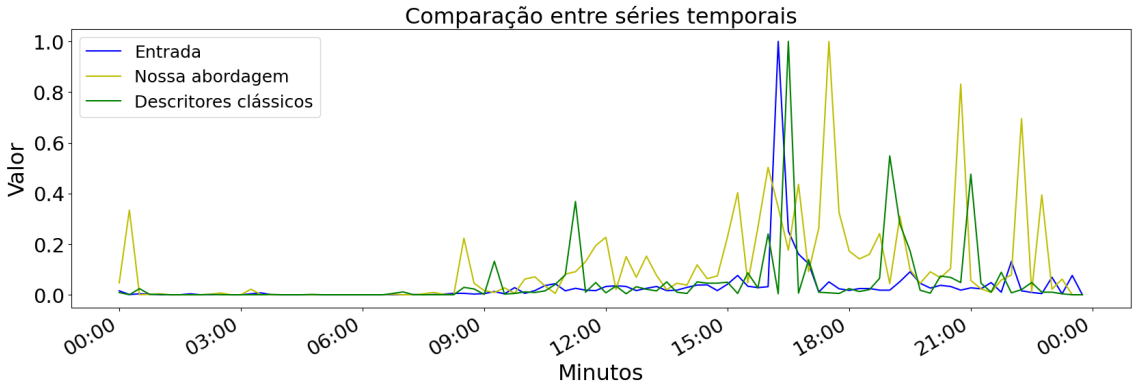

Figura 5. Comparação entre uma série usada como entrada em uma consulta e séries similares identificadas pela abordagem proposta e descritores clássicos

Para cada consulta c, computou-se $\hat{\mu}_{d,c}$ e $\hat{\mu}_{p,c}$ como as médias distâncias euclidianas normalizadas para as abordagens baseada em descritores clássicos e padrões ordinais, respectivamente. Seguidamente, foi calculada as diferenças entre os pares de médias de cada consulta c, ou seja, $D_c = \hat{\mu}_{p,c} - \hat{\mu}_{d,c}$. A Figura 6 apresenta o histograma das diferenças D entre as médias das distâncias para as 300 séries de entrada. As diferenças negativas correspondem as consultas na quais a nossa proposta apresentou em média uma qualidade superior, ou seja, retornou dados mais próximos ao requerido.

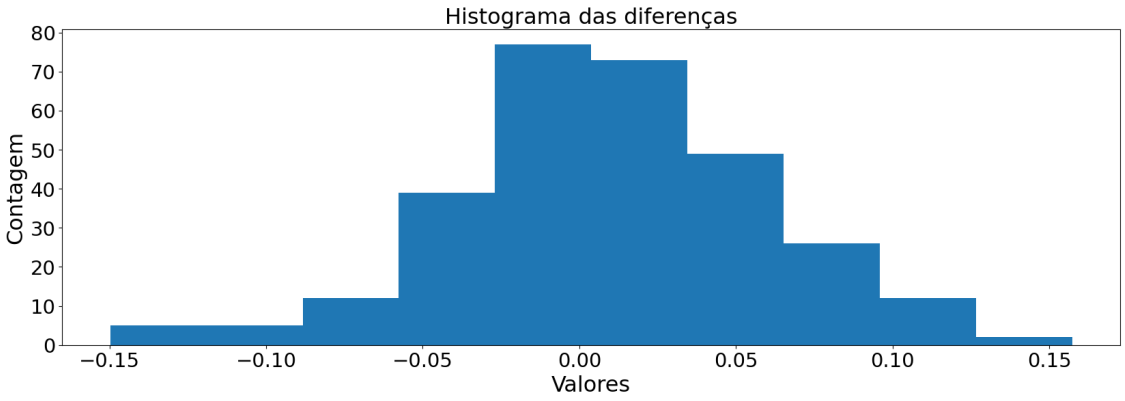

Figura 6. Histograma das diferenças entre as médias das distâncias euclidianas de avaliação

5. Conclusão

Neste trabalho, foi proposto um motor de busca eficiente de séries temporais com base na análise estatística entre histogramas de padrões ordinais com descritores estatísticos clássicos. Os resultados demonstram que os histogramas de padrões ordinais apresentam um desempenho melhor, a partir da distância euclidiana entre séries temporais. Notou-se que em alguns casos, ocorre o oposto. Isto poderá ser analisado em trabalhos futuros, mas no geral, o desempenho da proposta apresentado é bem satisfatório. Em suma, este trabalho promoveu uma ferramenta para descoberta de dados IoT em cidades inteligentes, o que impulsiona a ubiquidade por meio de ciência de dados.

Para trabalhos futuros, pretende-se avaliar os cenários em que nossa proposta mostrou-se pouco satisfatória. Para tal, far-se-á análises das estatísticas das similaridades das séries em grupos de controle, de modo que haja um resultado esperado. Adicionalmente, pretende-se empregar um maior conjunto de dados, considerando outros aplicativos e datas.

Referências

Bandt, C. & Pompe, B. (2002), 'Permutation entropy: A natural complexity measure for time series', *Phys. Rev. Lett.* **88**, 174102.
URL: *https://link.aps.org/doi/10.1103/PhysRevLett.88.174102*

Bhattacharyya, A. (1943), 'On a measure of divergence between two statistical populations defined by their probability distributions', *Bull. Calcutta math. Soc.* **35**, 99–109.
URL: *https://cir.nii.ac.jp/crid/1572261550690788352*

Fernandes, D., L. L. Moura, D., Santos, G., S. Ramos, G., Queiroz, F. & L. L. Aquino, A. (2023), Towards edge-based data lake architecture for intelligent transportation system, *in* 'Proceedings of the Int'l ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks', MSWiM '23, ACM, New York, NY, USA, p. 1–8.
URL: *http://dx.doi.org/10.1145/3616394.3618270*

Fernandes, D., Ramos, G. S., Pinheiro, R. G. & Aquino, A. L. (2024), 'A multi-start simulated annealing strategy for data lake organization problem', *Applied Soft Computing* **160**, 111700.
URL: *http://dx.doi.org/10.1016/j.asoc.2024.111700*

Gorelik, A. (2016), *The Enterprise Big Data Lake*, O'Reilly Media, Sebastopol, CA, USA.

Grzegorowski, M., Zdravevski, E., Janusz, A., Lameski, P., Apanowicz, C. & Ślezak, D. (2021), 'Cost optimization for big data workloads based on dynamic scheduling and cluster-size tuning', *Big Data Research* **25**, 100203.
URL: *https://www.sciencedirect.com/science/article/pii/S2214579621000204*

Hai, R., Koutras, C., Quix, C. & Jarke, M. (2023), 'Data lakes: A survey of functions and systems', *IEEE Transactions on Knowledge and Data Engineering* **35**(12), 12571–12590.
URL: *http://dx.doi.org/10.1109/TKDE.2023.3270101*

Martínez-Durive, O. E., Mishra, S., Ziemlicki, C., Rubrichi, S., Smoreda, Z. & Fiore, M. (2023), 'The netmob23 dataset: A high-resolution multi-region service-level mobile data traffic cartography'.

Pan, J. J., Wang, J. & Li, G. (2024), 'Survey of vector database management systems', *The VLDB Journal* **33**(5), 1591–1615.
URL: *http://dx.doi.org/10.1007/s00778-024-00864-x*

Pessa, A. A. B. & Ribeiro, H. V. (2021), 'ordpy: A python package for data analysis with permutation entropy and ordinal network methods', *Chaos: An Interdisciplinary Journal of Nonlinear Science* **31**(6).
URL: *http://dx.doi.org/10.1063/5.0049901*

Ramos, G. S., Fernandes, D., Coelho, J. A. P. d. M. & Aquino, A. L. L. (2023), *Toward Data Lake Technologies for Intelligent Societies and Cities*, Springer International Publishing, Cham, pp. 3–29.

Saeedan, M. & Eldawy, A. (2022), Spatial parquet: a column file format for geospatial data lakes, *in* 'Proceedings of the 30th International Conference on Advances in Geographic Information Systems', SIGSPATIAL '22, ACM, p. 1–4.
URL: *http://dx.doi.org/10.1145/3557915.3561038*

Sawadogo, P. & Darmont, J. (2020), 'On data lake architectures and metadata management', *Journal of Intelligent Information Systems* **56**(1), 97–120.
URL: *http://dx.doi.org/10.1007/s10844-020-00608-7*

Tang, X., Liu, W., Wu, S., Yao, C., Yuan, G., Ying, S. & Chen, G. (2025), 'Queryartisan: Generating data manipulation codes for ad-hoc analysis in data lakes', *Proc. VLDB Endow.* **18**(2), 108–116.
URL: *https://doi.org/10.14778/3705829.3705832*

Wang, J., Yi, X., Guo, R., Jin, H., Xu, P., Li, S., Wang, X., Guo, X., Li, C., Xu, X., Yu, K., Yuan, Y., Zou, Y., Long, J., Cai, Y., Li, Z., Zhang, Z., Mo, Y., Gu, J., Jiang, R., Wei, Y. & Xie, C. (2021), 'Milvus: A purpose-built vector data management system', *Proceedings of the 2021 International Conference on Management of Data* .
URL: *https://api.semanticscholar.org/CorpusID:235474148*

Weng, S., Tan, W., Ou, B. & Pan, J.-S. (2021), 'Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm', *Information Sciences* **549**, 13–33.
URL: *https://www.sciencedirect.com/science/article/pii/S0020025520310689*

Yu, H., Cai, H., Liu, Z., Xu, B. & Jiang, L. (2022), 'An automated metadata generation method for data lake of industrial wot applications', *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **52**(8), 5235–5248.