Aplicação do Modelo ARIMA no Vertica para Previsão da Velocidade do Vento

Gabriel Ciriaco Fornitano[1], Flávio Belizário Mota[2], Vanessa Cristina Oliveira de Souza[1], Arcilan Assireu[3], Melise Maria Veiga de Paula[1]

[1]Instituto de Matemática e Computação – Universidade Federal de Itajubá (UNIFEI)
Caixa Postal 50-37500-903 – Itajubá – MG – Brasil

[2]Descubra Soluções em Decisões Estratégicas

[3]Instituto de Recursos Naturais – Universidade Federal de Itajubá (UNIFEI)

{d2021009763, melise, vanessasouza, arcilan}@unifei.edu.br

flavio.belizario.mota@gmail.com

Abstract. *This study presents an exploratory analysis of applying the ARIMA model directly within the Vertica database for wind speed forecasting. A dataset from the EOSOLAR project was used, containing vertical wind profile measurements from the coastal region of Maranhão, Brazil. The evaluation considered the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) metrics across 105 trained models. The study investigated whether the in-database ARIMA approach in Vertica could provide efficient modeling for wind speed forecasting. The results showed that models with low complexity achieved good predictive performance.*

Resumo. *Este trabalho apresenta um estudo exploratório sobre a aplicação do modelo ARIMA diretamente no banco de dados Vertica para previsão da velocidade do vento. Utilizou-se um conjunto de dados do projeto EOSOLAR, com medições de perfis verticais de vento na região costeira do Maranhão. A avaliação considerou as métricas RMSE (Root Mean Squared Error) e MAE (Mean Absolute Error) sobre 105 modelos treinados. O estudo investigou se a abordagem in-database do ARIMA no Vertica poderia oferecer modelagem eficiente para a previsão da velocidade do vento. Os resultados mostraram que modelos com baixa complexidade alcançaram bom desempenho preditivo.*

1. Introdução

Na última década, tem-se observado avanços no armazenamento, tratamento e análise de dados com o surgimento do conceito da execução de algoritmos estatísticos e de aprendizado de máquina dentro do banco de dados (*In-database machine learning*). Este conceito envolve a execução de algoritmos de aprendizado de máquina diretamente no banco de dados [Cielen et al. 2021]. Essa abordagem oferece diversas vantagens, incluindo simplicidade de infraestrutura, segurança, velocidade, escalabilidade, concorrência, acessibilidade, governança e prontidão para produção [Epstein and Roberts 2022], sendo reforçada por estudos recentes que propõem sistemas que integram o armazenamento dos dados com sua análise e modelagem preditiva, como o *SAVIME* (*Simulation Analysis and Visualization In-Memory Environment*) [Lustosa et al. 2020].

Tradicionalmente, o desenvolvimento de modelos preditivos e de aprendizado de máquina envolve a extração e seleção dos dados, seguida pelo pré-processamento e pelo treinamento em ferramentas especializadas [Raschka et al. 2020]. No entanto, essa abordagem, quando integrada ao banco de dados, pode minimizar o esforço relacionado à extração, possibilitando que todas as etapacutadas diretamente no ambiente do próprio banco, reduzindo a movimentação de dados e o tempo de processamento necessário [Cielen et al. 2021].

Um exemplo dessa abordagem é o Vertica, que implementa algoritmos estatísticos e de aprendizado de máquina nativamente [Epstein and Roberts 2022]. Trata-se de um SGBD relacional distribuído e massivamente paralelo [Fard et al. 2020], cujo subsistema Vertica-ML inclui o modelo ARIMA, amplamente utilizado para previsão de séries temporais, combinando componentes de autoregressão, integração e média móvel.

A aplicação do ARIMA em contextos climáticos, especialmente na previsão do tempo e da velocidade do vento, tem mostrado bons resultados na literatura, apesar de limitações em cenários com alta sazonalidade ou longos horizontes de previsão [Elsaraiti and Merabet 2021, Salman and Kanigoro 2021, Liu et al. 2021].

Este trabalho realiza um estudo exploratório do ARIMA implementado no Vertica para previsão da velocidade do vento. Utilizam-se dados reais do projeto EOSOLAR, que disponibiliza medições verticais de vento na região costeira do Maranhão. A análise considera as etapas de preparação, execução e avaliação dos modelos, sendo utilizadas as métricas RMSE (Root Mean Square Error) e MAE (Mean Absolute Error) para avaliar o desempenho preditivo.

2. Trabalhos Relacionados

O modelo ARIMA representa uma abordagem intuitivamente razoável para a representação de séries temporais [Box et al. 2015]. Ele se mostra especialmente eficaz quando aplicado a séries que apresentam comportamento não estacionário, mas que ainda mantêm homogeneidade estrutural ao longo do tempo. Sua formulação resulta da integração de três componentes estatísticos fundamentais: a autorregressão (AR), a integração (I) — que se refere à diferenciação — e a média móvel (MA) [Hyndman and Athanasopoulos 2021].

A aplicação do ARIMA na previsão da velocidade do vento tem se mostrado eficiente para horizontes de curto prazo [Grigonytė and Butkevičiūtė 2016, Elsaraiti and Merabet 2021, Salman and Kanigoro 2021, Liu et al. 2021]. Em [Grigonytė and Butkevičiūtė 2016], o modelo ARIMA(3,1,1) obteve bons resultados para previsões de até 8 horas no Mar Báltico, com avaliação via RMSE, MAE e MAPE. Observou-se aumento nos erros com o horizonte de previsão e melhor desempenho nos períodos de inverno, outono e verão.

[Elsaraiti and Merabet 2021] compararam o ARIMA com redes neurais profundas, como o modelo LSTM, na previsão da velocidade do vento. Embora o ARIMA tenha apresentado resultados satisfatórios, o LSTM demonstrou desempenho superior, principalmente em previsões de longo prazo e em contextos com padrões sazonais mais complexos, destacando a simplicidade e baixo custo do ARIMA frente à maior capacidade dos modelos neurais.

No contexto de bancos de dados, diversos trabalhos discutem o uso de aprendizado de máquina diretamente em SGBDs por meio de extensões ou bibliotecas. Exemplos incluem PostgreSQL (MadLib), Oracle (OML), Redis (RedisML) e BigQuery (BigQuery ML) [Epstein and Roberts 2022], destacando-se o Vertica, um SGBD relacional, distribuído e orientado a colunas, baseado no protótipo C-Store [Lamb et al. 2012], cuja arquitetura favorece consultas OLAP em grandes volumes de dados [Fard et al. 2020].

No estudo de Fard et al. [Fard et al. 2020], foi realizada uma avaliação do subsistema de aprendizado de máquina distribuído do Vertica. Os autores o compararam ao Spark-MLlib — biblioteca de aprendizado de máquina do Apache Spark — utilizando modelos como *Random Forest*, *K-means* e regressão linear. A comparação considerou tanto o tempo total de treinamento quanto a qualidade preditiva. Os resultados demonstraram que o Vertica-ML apresentou tempos de execução competitivos ou superiores ao Spark, reforçando as vantagens da abordagem *in-database* em aplicações de Big Data.

No Vertica, o comando `ARIMA(...)` é utilizado para criar e treinar modelos ARIMA diretamente sobre séries temporais armazenadas no banco de dados, facilitando a definição do modelo e o uso de dados com intervalos regulares sem a necessidade de exportação para ambientes externos [Vertica 2025].

Após o treinamento do modelo, as previsões podem ser realizadas utilizando a função `PREDICT_ARIMA`. Esta função aplica o modelo ARIMA treinado para prever valores futuros ou realizar previsões no próprio conjunto de dados de treinamento [Vertica 2025].

3. Fluxo de Execução

Inicialmente, os dados foram extraídos de um arquivo no formato MATLAB. O arquivo original contém um cubo de dados multidimensional com diversas variáveis atmosféricas. Para os fins deste trabalho, foram selecionadas apenas as variáveis relacionadas à velocidade do vento e os respectivos registros de tempo, que representam a marca temporal de cada medição.

O projeto EOSOLAR [1] tem como objetivo investigar os recursos eólicos do Nordeste do estado do Maranhão, com foco em sua variabilidade temporal. O projeto utiliza dados coletados por tecnologias de sensoriamento remoto, como o LIDAR (*Light Detection and Ranging*) e o SODAR (*Sound Detection and Ranging*), sobre velocidade e direção do vento. As medições foram realizadas em alturas que variam de 40 a 260 metros, com incrementos de 10 metros.

Foi implementado um algoritmo em Python, utilizando as bibliotecas `NumPy`, `Pandas` e `SciPy`, para acessar e manipular os dados armazenados originalmente em um arquivo no formato `.mat`. Essa etapa inicial de pré-processamento foi necessária devido ao formato específico do arquivo de origem, uma vez que esse não podia ser carregado diretamente com o Vertica.

Embora o tratamento inicial tenha ocorrido fora do banco, isso reflete uma particularidade do experimento, e não uma limitação da abordagem proposta. O foco deste trabalho está em investigar os benefícios da modelagem diretamente dentro do ambiente

[1] `https://eosolar.equatorialenergia.com.br/`

Vertica. Assim, ainda que os dados utilizados não estivessem previamente armazenados em um banco analítico, a análise demonstra que, caso estivessem, o uso do Vertica poderia eliminar etapas de movimentação de dados, proporcionando maior desempenho, segurança e integração no processo de previsão.

A base de dados utilizada contempla 7561 registros de velocidade do vento, coletados em diferentes alturas. As medições abrangem o período de 16 de setembro de 2021 (18:00) a 8 de novembro de 2021 (14:30). Os valores de velocidade do vento variam entre 1,83 m/s e 17,02 m/s, com uma velocidade média de aproximadamente 9,44 m/s. A diversidade vertical das medições permite capturar de forma detalhada o perfil de vento ao longo da camada atmosférica de interesse para aplicações eólicas [Assireu et al. 2024]. Para os experimentos realizados, foi selecionada a série de velocidade do vento medida a 150 metros de altura, por estar alinhada à faixa de operação de turbinas eólicas comerciais [Chastre and Lúcio 2012].

O segundo passo consistiu na preparação do ambiente computacional necessário para realizar os testes com o modelo ARIMA. Para isso, foi utilizado um contêiner Docker executando uma instância do banco de dados Vertica Community Edition (CE). O uso de contêineres foi adotado com o objetivo de minimizar possíveis ruídos e inconsistências entre execuções, buscando maior reprodutibilidade do experimento ao proporcionar um ambiente controlado e padronizado [Boettiger 2015].

3.1. Treinamento do Modelo ARIMA

Neste trabalho, foram utilizados 7500 dos 7561 registros para o treinamento e os 61 mais recentes para validação. Apesar das proporções variarem na literatura, é comum priorizar uma base ampla de treino, desde que se mantenha uma janela recente para avaliação preditiva, respeitando a ordem temporal [Grigonytė and Butkevičiūtė 2016, Elsaraiti and Merabet 2021, Liu et al. 2021].

Com os dados particionados em tabelas no banco, foi iniciado o treinamento do modelo ARIMA para prever a velocidade do vento a 150 metros. A seleção dos parâmetros p, d e q é crucial para o desempenho preditivo, sendo adotada a abordagem de *grid search*, também utilizada em [Salman and Kanigoro 2021], para identificar as melhores combinações segundo RMSE e MAE.

Definiram-se os intervalos $p \in 0, 1, 2, 3, 4, 5$, $d \in 0, 1, 2$ e $q \in 0, 1, 2, 3, 4, 5$, totalizando 105 combinações. Exceções foram feitas aos modelos ARIMA(0,0,0), ARIMA(0,1,0) e ARIMA(0,2,0), não suportados pelo Vertica por exigirem ao menos um parâmetro diferente de zero. A escolha dos intervalos foi inspirada em trabalhos anteriores [Salman and Kanigoro 2021, Elsaraiti and Merabet 2021, Grigonytė and Butkevičiūtė 2016].

Realizou-se também uma segunda rodada de testes para analisar o impacto do aumento de p no tempo de execução e no desempenho. Foram testados modelos do tipo ARIMA$(p, 2, 2)$ com $p \in 8, 16, 32, 64, 128$, mantendo os demais parâmetros fixos para isolar o efeito do termo autorregressivo.

As previsões foram geradas pela função `PREDICT_ARIMA()`, aplicada diretamente à continuação da série de treinamento. O modelo utilizou os próprios valores previstos como entrada nos passos seguintes, até atingir o número de previsões, definido pelo

parâmetro `npredictions`da função.

A avaliação dos modelos foi feita com as métricas RMSE e MAE, comumente adotadas na previsão de variáveis meteorológicas por ARIMA [Grigonytė and Butkevičiūtė 2016, Elsaraiti and Merabet 2021, Salman and Kanigoro 2021, Liu et al. 2021]. As previsões foram comparadas aos valores reais da tabela de teste, composta pelos 61 registros subsequentes aos dados de treinamento. A comparação foi organizada por meio de uma nova tabela que alinha os pares previstos e observados por índice temporal.

4. Resultados

No experimento dedicado à variação de p, com valores de p crescendo de 8 até 128, observou-se um aumento expressivo no tempo de treinamento dos modelos. Esse crescimento seguiu um padrão exponencial, com o tempo de execução atingindo mais de 7000 segundos para o modelo ARIMA$(128, 2, 2)$. Essa tendência está ilustrada na Figura 1.

Entretanto, ao analisar o impacto do aumento de p sobre a qualidade preditiva, observou-se que, apesar do acréscimo significativo no custo computacional, os valores da métrica MAE permaneceram relativamente estáveis. Não se verificou uma melhoria consistente na acurácia das previsões com o aumento de p. Essa análise está representada na Figura 2.

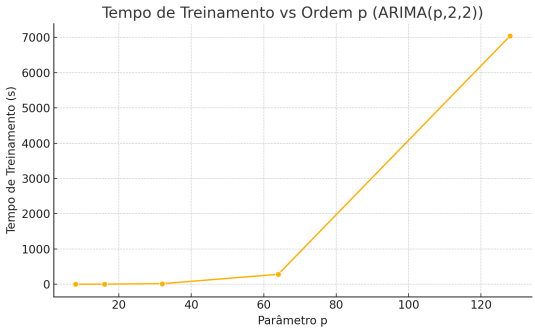

Figure 1. Tempo de treinamento dos modelos ARIMA$(p, 2, 2)$

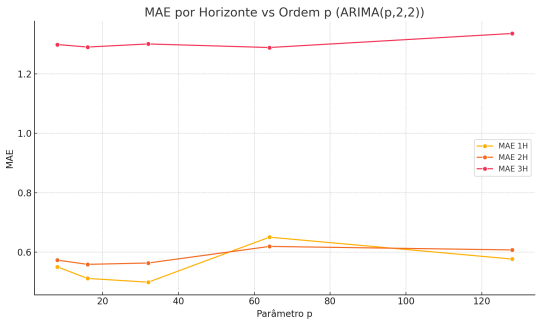

Figure 2. Evolução da métrica MAE com relação à ordem p

Ambos os gráficos confirmam que modelos mais simples, como $(16, 2, 2)$ e $(32, 2, 2)$, alcançam os menores valores de RMSE e MAE no horizonte de 1 hora, sem exigir os altos custos de processamento observados nos modelos com ordens mais elevadas.

Uma busca em grade foi conduzida para identificar as melhores combinações de (p, d, q) com base em 105 configurações testadas. O modelo ARIMA$(4, 2, 0)$ apresentou o menor RMSE no intervalo de uma hora, (0,3559), enquanto o ARIMA$(3, 2, 0)$ obteve o menor MAE (0,311), ambos com tempos de execução inferiores a 500 ms, demonstrando a viabilidade computacional da abordagem.

A Tabela 1 destaca os 5 melhores modelos treinados e seus valores de RMSE que variam de 0,3595 sendo esse o menor RMSE no período de 1 hora, até 1,8268 no previsão de 3 horas. Com relação ao valores de todos os testes, os piores modelos se afastaram

bastante da previsão real em modelos como o ARIMA$(0, 0, 1)$ tendo um RMSE de 4,7943 e MAE de 4,6999 no período de 1 hora.

Table 1. Top 5 modelos com menor RMSE (1h)

Modelo	1H	2H	3H
(4,2,0)	**0,3595**	1,384	3,0568
(3,2,0)	0,3842	1,545	3,303
(5,2,0)	0,3845	1,1567	2,7068
(0,2,1)	0,417	0,7007	1,9096
(1,2,1)	0,4976	0,6647	1,8268

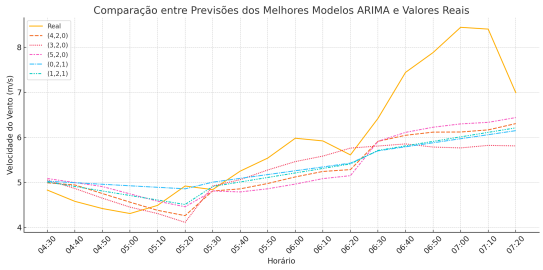

Figure 3. Comparação entre valores reais e previsões dos melhores modelos ARIMA

Por fim, utilizando os cinco modelos ARIMA com menor RMSE identificados na etapa de avaliação, foi realizada uma análise preditiva focada nas três primeiras horas do conjunto de teste, correspondendo aos 18 primeiros registros. Considerando que os melhores desempenhos de RMSE ocorreram no horizonte de uma hora, adotou-se uma estratégia de previsão recursiva com janelas de uma hora: a previsão da segunda hora foi realizada utilizando os dados reais da primeira hora, enquanto a previsão da terceira hora utilizou os valores reais da segunda hora. Essa abordagem permitiu manter sempre um intervalo de previsão de uma hora, minimizando a propagação acumulativa de erros. A Figura 3 ilustra a comparação entre os valores reais observados e as previsões geradas pelos cinco melhores modelos.

5. Conclusões

A análise realizada demonstrou que o modelo ARIMA, quando ajustado por meio de busca em grade, apresentou desempenho satisfatório na previsão da velocidade do vento. Os modelos ARIMA$(4, 2, 0)$ e ARIMA$(3, 2, 0)$ obtiveram os menores valores de RMSE e MAE, respectivamente, evidenciando boa capacidade preditiva dentro do contexto do estudo. Observou-se que os valores previstos foram capazes de refletir, com certo erro, as tendências na velocidade do vento, embora apresentassem menor acurácia em cenários com mudanças abruptas — um comportamento esperado em modelos ARIMA, dada sua ênfase em padrões lineares e estacionários.

Adicionalmente, os resultados indicaram que o aumento do parâmetro p não resultou necessariamente em melhorias no desempenho. Modelos com valores intermediários de p, como $(3, 2, 0)$ e $(4, 2, 0)$, foram suficientes para capturar a dinâmica da série com custo computacional reduzido. Por outro lado, configurações com p elevados acarretaram aumento significativo no tempo de execução.

Como trabalho futuro, sugere-se a ampliação da análise explorando outros algoritmos de aprendizado de máquina disponíveis nativamente no Vertica, como regressão linear, floresta negra e k médias. Além disso, seria relevante realizar comparações entre diferentes abordagens de modelagem aplicadas à mesma base de dados, incluindo métodos estatísticos e modelos baseados em aprendizado profundo, a fim de avaliar possíveis ganhos de desempenho preditivo na tarefa de previsão da velocidade do vento.

References

Assireu, A. T., Fisch, G., Carvalho, V. S. O., Pimenta, F. M., de Freitas, R. M., Saavedra, O. R., Neto, F. L. A., Júnior, A. R. T., Oliveira, D. Q., Lopes, D. C. P., de Lima, S. L., Marcondes, L. G. P., and Rodrigues, W. K. S. (2024). Sea breeze-driven effects on wind down-ramps: Their implications for wind farms along the north-east coast of brazil. *Energy*, 294:130804.

Boettiger, C. (2015). An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition.

Chastre, C. and Lúcio, V. (2012). Torres pré-fabricadas de betão para suporte de turbinas eólicas. In *Estruturas pré-moldadas no mundo – Aplicações e comportamento estrutural*, pages 91–106. Universidade NOVA de Lisboa.

Cielen, D., Meysman, A. D. B., and Ali, M. (2021). *Data Science: Principles and Practice*. Manning Publications.

Elsaraiti, M. and Merabet, A. (2021). A comparative analysis of the arima and lstm predictive models and their effectiveness for predicting wind speed. *Energies*, 14(20):6782.

Epstein, B. and Roberts, P. (2022). *Accelerate Machine Learning with a Unified Analytics Architecture*. O'Reilly Media, Inc., Sebastopol, CA, USA.

Fard, A., Zhang, B., Katepalli, K., Stonebraker, M., and Rundensteiner, E. A. (2020). Vertica-ml: Distributed machine learning in vertica database. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 755–768. ACM.

Grigonytė, E. and Butkevičiūtė, E. (2016). Short-term wind speed forecasting using arima model. *Energetika*, 62(1–2):17–26.

Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3 edition. Accessed on March 26, 2025.

Lamb, A., Fuller, M., Varadarajan, R., Tran, N., Vandiver, B., Doshi, L., and Bear, C. (2012). The vertica analytic database: C-store 7 years later. *Vertica Systems, An HP Company*.

Liu, X., Lin, Z., and Feng, Z. (2021). Short-term offshore wind speed forecast by seasonal arima-a comparison against gru and lstm. *Energy*, 227:120492.

Lustosa, H., Costa, F., Guimarães, J., and de Oliveira, D. (2020). Savime: An array dbms for simulation analysis and ml models predictions. In *International Conference on Database and Expert Systems Applications*, pages 357–367. Springer.

Raschka, S., Patterson, J., and Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, 11(4):193.

Salman, A. G. and Kanigoro, B. (2021). Visibility forecasting using autoregressive integrated moving average (arima) models. *Procedia Computer Science*, 181:586–593.

Vertica (2025). Arima - vertica 25.1.x documentation. Accessed: March 26, 2025.