

Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Leonardo Morettini e Castella

**Modelos de predição de séries temporais
aplicados à demanda de energia elétrica da
Califórnia**

**Curitiba
2022**

Leonardo Morettini e Castella

Modelos de predição de séries temporais aplicados à demanda de energia elétrica da Califórnia

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Pr. Dr. Luiz Eduardo S. Oliveira

Curitiba
2022

Modelos de predição de séries temporais aplicados à demanda de energia elétrica da Califórnia

Leonardo Morettini e Castella¹
Pr. Dr. Luiz Eduardo S. Oliveira²

Resumo

O trabalho mostra duas técnicas de predição de séries temporais, o tradicional ARIMA e a de deep learning chamada LSTM, aplicadas à demanda por energia elétrica na Califórnia. As bases de dados contêm a demanda residencial por energia elétrica, baixadas do site do departamento de energia dos Estados Unidos e tratada pelo autor. Foram separadas duas bases de dados para realizar as predições, demanda hora por hora e demanda média diária. O método ARIMA se mostrou mais eficiente para a predição de demanda por hora enquanto o modelo LSTM foi mais eficiente para a predição de demanda diária. O trabalho mostra também que o modelo de deep learning é mais eficiente para predições com bases de dados de maior tamanho do que o modelo tradicional ARIMA.

Palavras-chave: ARIMA, RNN, LSTM, Séries temporais, predição, demanda por energia elétrica.

Abstract

The paper shows two time series prediction techniques, the traditional ARIMA and the deep learning one called LSTM, applied to the electricity demand in California. The databases contain residential electricity demand, downloaded from the US Department of Energy website, and handled by the author. Two databases were separated to perform the predictions, hourly demand, and daily average demand. The ARIMA method proved to be more efficient for predicting hourly demand while the LSTM model was more efficient for predicting daily demand. The work also shows that the deep learning model is more efficient for predictions with larger databases than the traditional ARIMA model.

Keywords: ARIMA, RNN, LSTM, time series, prediction, electric energy demand.

1 Introdução

A demanda por energia elétrica para uso residencial é condicionada por uma série de variáveis externas como

a dos preços de outros bens substituíveis, como o gás natural, o clima e o aumento populacional [8]. A Demanda é um valor calculado, que representa a quantidade de carga de eletricidade dentro do sistema elétrico da (Balancing authority (BA)). Uma BA obtém seu valor de demanda tomando a geração líquida total de eletricidade medida dentro de seu sistema elétrico, e, subtraindo o intercâmbio total de eletricidade líquida medida que ocorre entre a BA e suas BAs vizinhas [20].

O objetivo principal deste artigo é fazer uma predição da demanda por energia elétrica no estado da Califórnia para as próximas horas e para os próximos dias. Para cumprir com esse objetivo foram testados dois modelos utilizados na literatura para predição de séries temporais: O modelo estatístico ARIMA (Autoregressive Integrated moving average model) e o modelo de machine learning LSTM (long short term memory). A performance dos modelos é comparada pelo indicador da RSME (root square mean error) para determinar qual é o melhor modelo para predição de cada agrupamento da demanda, horas ou dias.

A predição de séries temporais foi utilizada para prever a demanda por energia elétrica em outros trabalhos, como o trabalho de Torres, Martínez e Troncoso que utilizaram os modelos ARIMA e LSTM para predição de energia elétrica na Espanha [18]. O Modelo ARIMA também foi explorado por Karin, para predição de energia elétrica em prédios sendo os modelos ARIMA e SARIMA para estimar considerando a sazonalidade dos dados. Foram utilizadas as técnicas de teste de estacionariedade, avaliação de ACF (Autocorrelation function) e PACF (Partial Autocorrelation function) para determinar os parâmetros p , q e d . [7]. No artigo de Jian Qi e Yu DU foi utilizado o modelo LSTM para a predição do consumo de energia com sazonalidade [5].

Rini Nur Hasanah comparou as previsões utilizando VAR (Vector Autoregressive Model) com Recurrent Neural Networks com a presença de camadas LSTM (long short term memory). Os resultados são comparados utilizando os valores de Root mean square errors (RMSE) e a Mean Absolute Error (MAE) [14]. Siami-Namini e Neda Tavakoli mostram uma comparação em séries de dados temporais financeiros entre ARIMA e LSTM em que o modelo de deep learning tem uma redução entre 84% e 85% dos erros obtidos em ARIMA e, portanto, obtendo melhores resultados preditivos [17].

¹Leonardo Morettini e Castella, morettinileonardo@gmail.com.

²Pr. Dr. Luiz Eduardo S. Oliveira - Departamento de Informática/UFPR.

O Artigo de Agrawal e Muchahary utilizou LSTM para predição de consumo de energia elétrica em New England obtendo o seguinte resultado: O modelo proposto encontrou alta precisão com a média absoluta percentual (MAPE) dos erros com um intervalo de confiança de 2.25% [13].

Porém no trabalho de Chai and Draxler é defendido o porquê deve-se usar RMSE ao invés do MAPE para comparar resultados de modelos [2]. O paper de Bouktif e Serhani se aprofunda nos inputs dos hyperparameters para o modelo LSTM aplicado no consumo de energia elétrica. Contém diversos outros artigos como referência que obtiveram resultados com maior precisão utilizando o modelo LSTM [1].

2 Materiais e Métodos

O presente trabalho apresenta duas propostas de predição de séries temporais, o modelo ARIMA e o modelo LSTM. Neste capítulo será apresentado o contexto do conjunto de dados, a limpeza e preparo dos dados com uma breve análise exploratória. Em seguida os modelos são explicados com mais detalhes, como as equações que compõem os modelos e quais os parâmetros que posteriormente foram determinantes para realizar as predições aqui propostas.

2.1 O conjunto de dados

Diferente de outros estudos, a pesquisa utilizou apenas a variável da demanda residencial em Megawatts (Mgw) por hora e por dia, representando a soma da demanda em Mgw de todas as BAs da região da Califórnia, Estados Unidos da América, entre 2015 e 2021. Isso porque, os modelos empregados utilizam apenas uma única variável contra o tempo para fazer a predição.

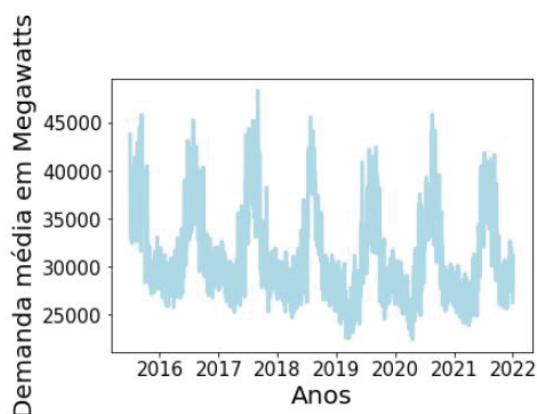


Figura 1: Demanda média residencial diária em Megawatts no estado da Califórnia. Fonte: O Autor (2022).

É possível identificar sazonalidade em que os picos se encontram no meio do ano (meses mais quentes no hemisfério norte).

2.2 Limpeza e preparo dos dados

Os dados foram extraídos diretamente do site do governo americano (US Energy Information Administration) em que é fornecida quantidade total consumida por hora e estado. Para realizar a pesquisa separamos os dados em duas partes: A primeira é o agrupamento por hora da mesma maneira que veio do site e a segunda parte é o agrupamento de demanda média de energia por dia. Os dados são do dia 15 de junho de 2015 até 31 de dezembro de 2021.

Primeiro foi realizada uma análise exploratória dos dados, com a análise de gráficos foi possível observar que havia alguns outliers que precisaram ser removidos das duas bases.

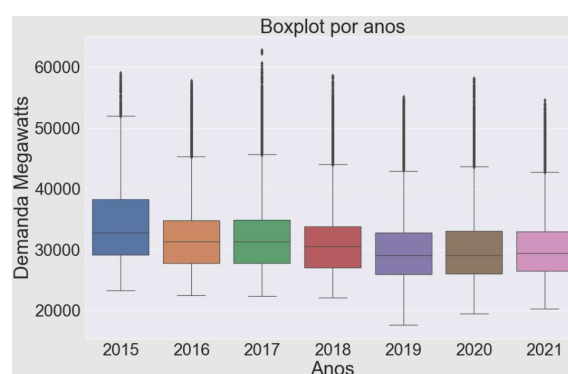


Figura 2: Demandas média, mínimo e máximos ao longo dos anos. Fonte: O Autor (2022).

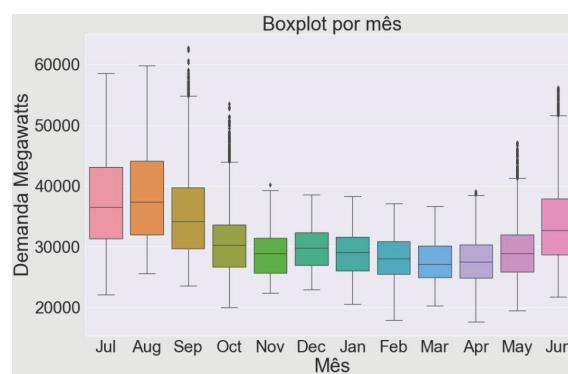


Figura 3: Demandas médias, mínimas e máximas ao longo dos meses. Fonte: O Autor (2022).

A remoção dos outliers foi feita com base no desvio padrão da distribuição dos dados. Dados que estavam a 3 desvios padrão de distância para mais ou para menos foram removidos.

Outro método utilizado foi o dos percentis, dados que estavam abaixo do primeiro quartil (25%) ou acima do terceiro quartil (75%) foram removidos [23].

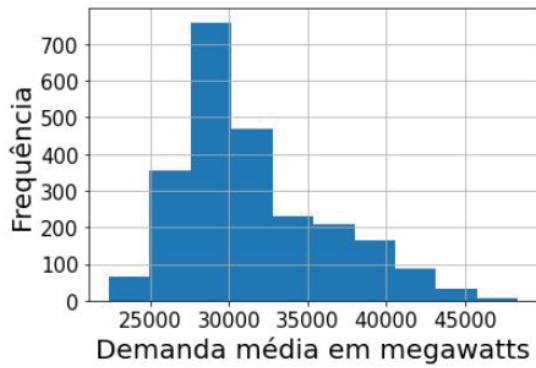


Figura 4: Demandas médias, mínimas e máximas ao longo dos meses. Fonte: O Autor (2022).

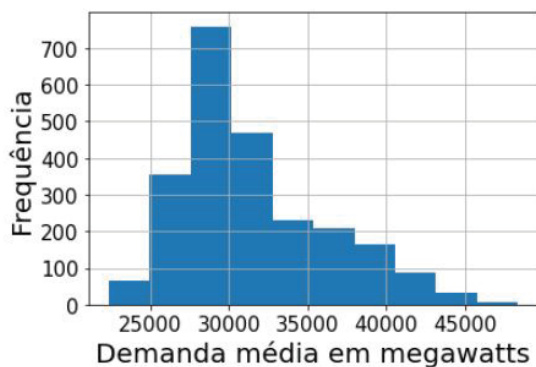


Figura 5: Distribuição de densidade após a remoção de Outliers: Demanda média diária. Fonte: O Autor (2022).

2.3 Modelos empregados

Utilizando a linguagem Python em um Jupyter notebook, baixamos as bibliotecas Statsmodels para fazer os testes de hipótese e aplicar o modelo ARIMA e a biblioteca TensorFlow Keras para o modelo de Recurrent Neural Network LSTM.

2.3.1 Modelo ARIMA

O Modelo ARIMA para predição de series temporais, foi apresentado primeiramente por Box-Jenkins. [16]. ARIMA é a Integração dos modelos AR (Autoregressive model) com MA (Moving Average), e seus parâmetros são definidos de acordo com o resultado de cada modelo (AR, I, MA) ou também (p, d, q) [6].

Os modelos AR e MA podem ser expressos nas seguintes equações [6]:

$$y(t) = c + a_1y_{t-1} + \dots + a_p y_{t-p} + e_t \quad (1)$$

$$y(t) = \mu + u_t + m_1u_{t-1} + \dots + m_q u_{t-q} \quad (2)$$

As variáveis portanto, a_1, \dots, a_p , e m_1, \dots, m_q são os parâmetros das porções autorregressivas e de média móvel respectivamente; a constante na expressão é denotada por c e p e q representam a ordem dos respectivos AR e

MA. O ruído branco (White noise) é denotado como e_t . Na equação (2) $u_t, u_{t-1}, \dots, u_{t-q}$ representa termos de ruído branco (erro). Expectativa de y_t é representado por μ . Integrando os dois modelos representado por (1) e (2) usando os mesmos dados do treinamento definido, o ARIMA (p, d, q) é representado pela equação:

$$y(t) = c + a_1y_{t-1} + \dots + a_p y_{t-p} + u_t + m_1u_{t-1} + \dots + m_q u_{t-q} \quad (3)$$

Aqui p representa o valor Autoregressivo (AR) e q representa os termos da Média Móvel (MA). p refere-se ao número de observação de defasagem dos dados históricos que são usados para prever os valores futuros, também chamados de ordem de defasagem. q refere-se ao tamanho do espaço da média móvel. Para este modelo, o principal requisito é tornar os dados da série temporal estatisticamente estacionários em termos de média, variância e correlação. Caso os dados não sejam Estacionários, é preciso determinar um grau de diferenciação representada por “ d ” resultando então no modelo ARIMA (p, d, q) [6].

2.3.2 Modelo LSTM

A arquitetura de LSTM (Long short term memory) é utilizada também para predição de séries temporais [10]. Essa arquitetura é enquadrada em redes recorrentes, cuja principal característica é a capacidade de modelar dependências temporais dos dados. Uma rede LSTM pode ser estruturada de diferentes maneiras dependendo do número de resultados a serem obtidos. Pode ser estruturado com uma entrada e uma saída (um para um), muitas entradas e uma saída (muitos para um), uma entrada e muitas saídas. [18]

A Modelo parte de uma Recurrent Neural Network (RNN) [10]cujo funcionamento parte de duas redes neurais recorrentes simples conforme ilustradas em Figura 4. A ideia é que as conexões recorrentes permitam uma memória de entradas anteriores para persistir no estado interno da rede e, assim, influenciar a saída da rede. A cada passo do tempo a rede neural processa o vetor de entrada x_t , atualiza sua camada oculta por meio de funções de ativação h_t , e o usa para fazer uma previsão de sua saída y_t . Cada nó na camada oculta recebe as entradas da camada anterior e as saídas da camada atual do último passo de tempo. O valor retido na unidade de atraso (Delay Unit) é realimentado para as unidades ocultas (Hidden Units) como entradas [10].

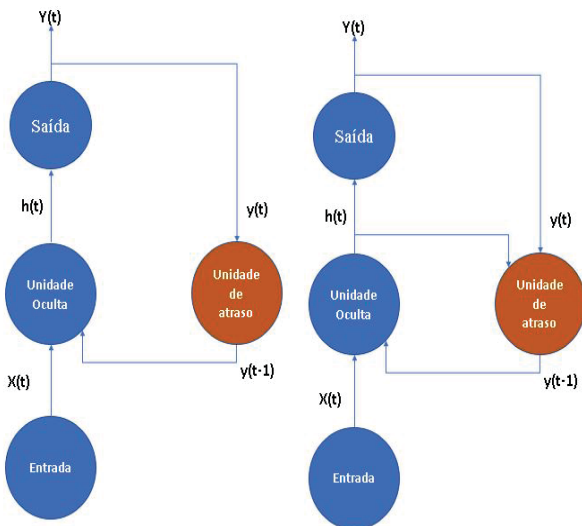


Figura 6: Duas estruturas de redes neurais recorrentes simples. Fonte: Adaptado de LEWIS (2016).

O Modelo LSTM é semelhante ao RNN, porém com a diferença de que nas Unidades ocultas ficam bloqueios de memória (Memory Blocks) [10]. Qualquer modelo RNN funciona em uma estrutura em cadeia, porém no LSTM cada módulo tem 4 camadas diferentes funcionando de maneira bem específica:

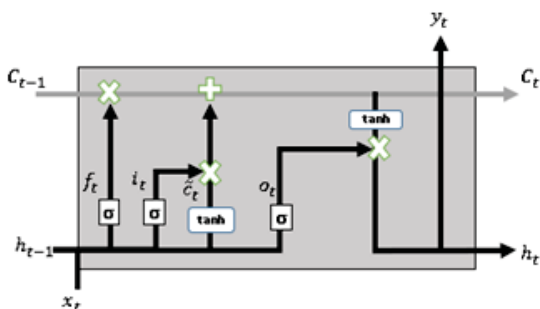


Figura 7: Célula LSTM (long short term memory) de uma rede neural recorrente. Fonte: Adaptada de TORRES (2022).

As camadas são constituídas de acordo com as equações abaixo [18]:

$$\tilde{c}_t = \tanh(W_c \cdot [a_{t-1}, x_t] b_c) \tag{4}$$

$$i_t = \sigma(w_u \cdot [a_{t-1}, x_t] + b_u) \tag{5}$$

$$f_t = \sigma(w_f \cdot [a_{t-1}, x_t] + b_f) \tag{6}$$

$$O_t = \sigma(W_o \cdot [a_{t-1}, x_t] + b_o) \tag{7}$$

$$C_t = i_t * \tilde{c}_t + f_t * c_{t-1} \tag{8}$$

$$a_t = o_t * \tanh c_t \tag{9}$$

Cada célula LSTM recebe a informação modelada da célula passada (C_{t-1} e h_{t-1}) e os dados do instante (x_t). A função Sigmoid e a função tanh são as camadas da rede neural [21] cuja resposta varia de 0 a 1 sendo um

valor próximo de 0 significa que a informação vai ser esquecida e próximo de 1 vai ser retida. f_t é a camada forget gate, que serve para “esquecer” o que aprendeu vindo do input anterior; i_t é a camada de input e o O_t é o portao de saída. i_t decide qual informação O_t vai usar para atualizar a memória de C_t . Portanto C_t é atualizado em função de f_t e i_t . Finalmente O_t decide qual informação vai sair pelo portao final e ir para a próxima unidade escondida.

2.4 Aplicação do modelo ARIMA

Para as previsões ARIMA, houve a necessidade de cortes de dados em um tempo menor, uma vez que o computador utilizado para a análise dos dados não dispunha de memória suficiente para operar com o tamanho original da base de dados. Além disso, antes de rodar o modelo ARIMA, foi necessário fazer alguns testes antes, como o teste de estacionariedade e os testes de Partial Correlation. Isso porque, os testes fornecem as informações necessárias para saber se precisamos transformar os dados em logaritmo e quais serão os parâmetros de autorregressão e média móvel para inserir no modelo. Dados estacionários se comportam de uma maneira específica, cuja distribuição de probabilidade não muda no decorrer do tempo e sempre tem um comportamento “normal”. Para executar seu teste, é realizada a análise de gráficos e observação da existência ou não de sazonalidade na demanda por energia elétrica. Nos meses de inverno a demanda aumenta e nos meses de mais quente ela diminui. Para tratar os dados de maneira que se possa ter um comportamento estacionário, foi realizada a transformação logarítmica dos dados diários e obtida então uma distribuição normal e dados estacionários:

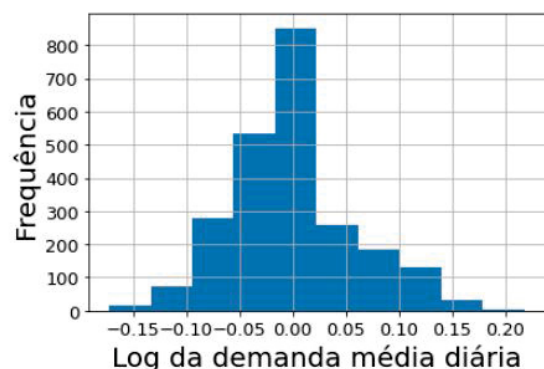


Figura 8: Histograma do log da demanda média diária. Fonte: O Autor (2022). Fonte: O Autor (2022).

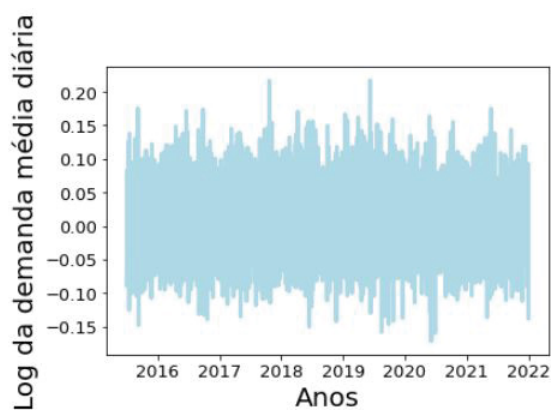


Figura 9: Demanda diária transformada em logaritmo.
Fonte: O Autor (2022).

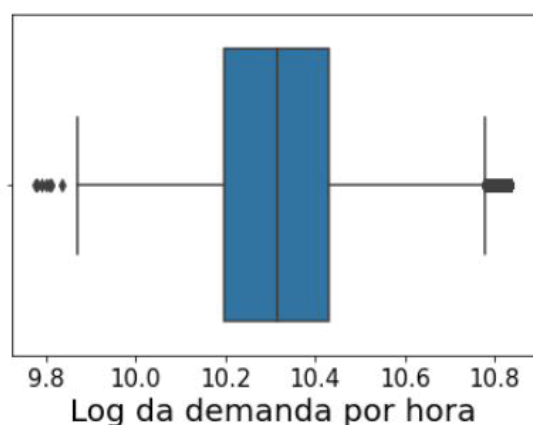


Figura 10: Demanda por hora transformada em logaritmo sem outliers. Fonte: O Autor (2022).

Devido à extensão da base de dados separados por horas, foi preferível realizar um recorte da última semana do ano de 2021 para a predição retirando-se assim toda a sazonalidade e variância acima do normal. Para testar a estacionariedade utilizamos do teste de ADF (Augmented Dicker Fuller) incluído na biblioteca "Statsmodels" do python. O teste consiste em avaliar duas hipóteses:

Hipótese Nula (H_0): Se não for rejeitada, sugere que a série temporal tem uma raiz unitária, o que significa que não é estacionária. Tem alguma estrutura dependente do tempo.

Hipótese Alternativa (H_1): A hipótese nula é rejeitada; sugere que a série temporal não tem uma raiz unitária, o que significa que é estacionária. Não possui estrutura dependente do tempo.

Interpreta-se esse resultado a partir do valor p do teste. Um valor de p abaixo de um limite estipulado (1%, 5% ou 10%) sugere a rejeição a hipótese nula (estacionária), do contrário, um valor de p acima do limite sugere a não rejeição da hipótese nula (não estacionária).

p-value > 0,05: Falha ao rejeitar a hipótese nula (H_0), os dados têm raiz unitária e não são estacionários.

p-value <= 0,05: Rejeita-se a hipótese nula (H_0), os dados não possuem raiz unitária e são estacionários.

Os valores críticos são maiores que o valor do teste ADF, o que significa que podemos rejeitar a hipótese nula e considerar a transformação logarítmica como eficiente para a transformação dos dados em estacionários. Os dados da demanda por hora não foram transformados para logaritmo e ainda assim apresentam estacionariedade.

Tabela 1: Teste de Estacionariedade Augmented Dickey-Fuller.

Resultados	Dados por hora	Dados diários
Estatística teste ADF	-2.36	-9.30
Valor p	0.15	0.00
Valores Críticos		
1%	-3.47	-3.43
5%	-2.88	-2.86
10%	-2.58	-2.57

Para definir os parâmetros dos testes ARIMA existem os testes de autocorrelação PACS (Partial Autocorrelation statistics). O gráfico indica os valores fora da faixa de confiança em azul claro a ordem do modelo AR, em que a partir da terceira observação, existe uma correlação negativa, ou seja apenas duas horas anteriores são necessárias para explicar a demanda da hora seguinte.

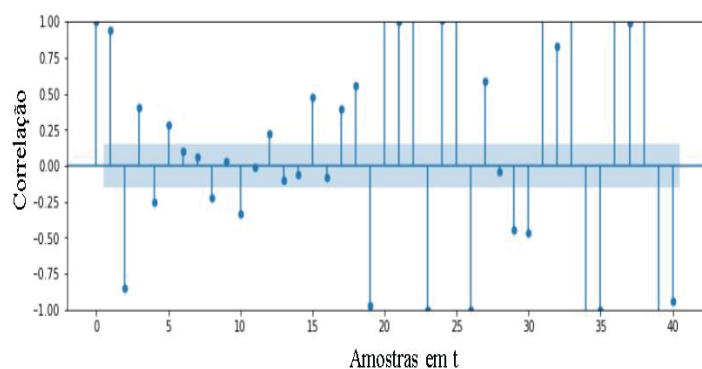


Figura 11: Gráfico Auto-correlação parcial (Partial Autocorrelation) para dados diários. Fonte: O Autor (2022). Para determinar o valor da função MA a partir da análise do Correlograma observamos as faixas que estão mais longe da faixa de confiança em azul claro. O eixo x indica o valor em t e o eixo y o grau de correlação.

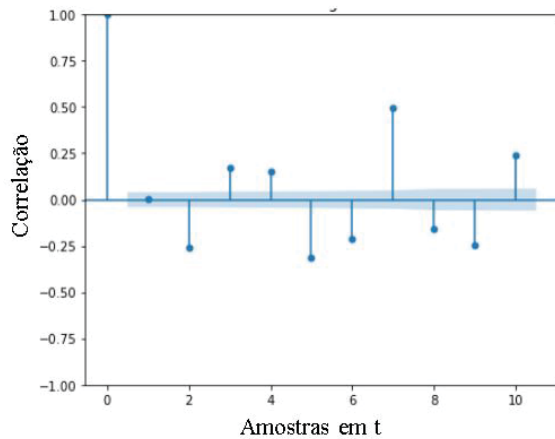


Figura 12: Correlograma da demanda diária. Fonte: O Autor (2022). O eixo x indica o valor em t e o eixo y o grau de correlação. O gráfico mostra que apenas a primeira hora é suficiente para explicar a próxima dentro do modelo MA. Com isso determinamos os parâmetros (p,d e q) da análise diária e por hora como (2,0,1).

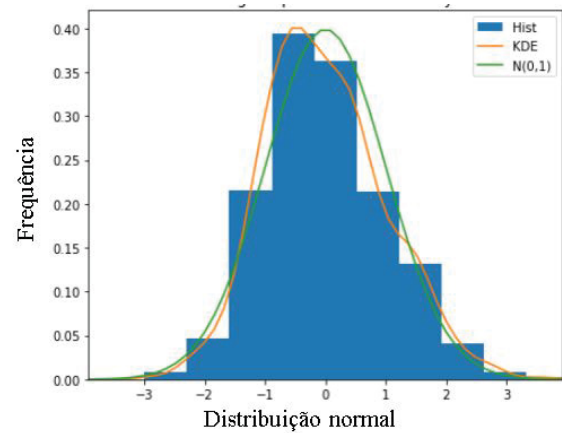


Figura 13: Distribuição dos dados log da demanda diária. Fonte: O Autor (2022). O eixo x representa os valores em log e o eixo y a frequência que eles aparecem. Os dados diários com transformação logarítmica obtiveram erros muito baixos e uma distribuição próxima de uma normal. O estimador KDE (Kernel density estimator) que traz a função da densidade de distribuição dos dados de forma suavizada, enquanto as barras azuis mostram a forma estática. A função $N(0,1)$ mostra a função de distribuição normal.

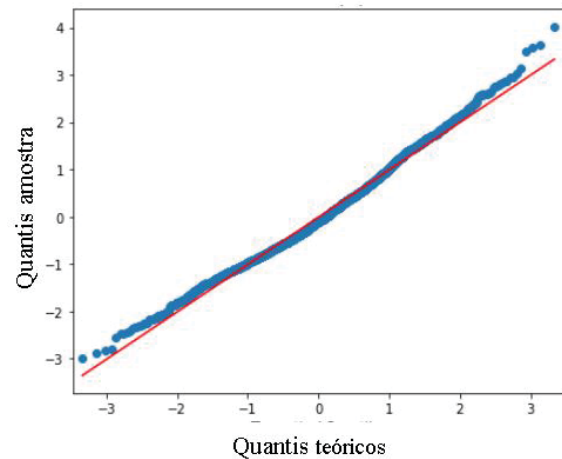


Figura 14: Gráfico normal quantil-quantil demanda diária. Fonte: O Autor (2022). O eixo x são os quantis teóricos e o eixo Y os quantis da amostra. O gráfico "Normal Q-Q" mostra se o quanto próximo a distribuição de frequência dos dados está para uma distribuição de probabilidade normal.

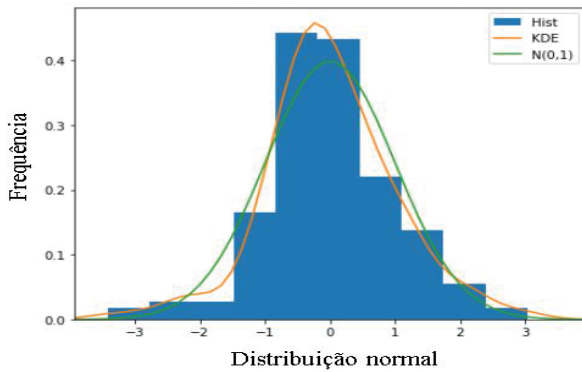


Figura 15: Distribuição dos dados demanda por hora. Fonte: O Autor (2022). O eixo x representa os valores em log e o eixo y a frequência que eles aparecem. Os dados diários com transformação logarítmica obtiveram erros muito baixos e uma distribuição próxima de uma normal. O estimador KDE (Kernel density estimator) que traz a função da densidade de distribuição dos dados de forma suavizada, enquanto as barras azuis mostram a forma estática. A função $N(0,1)$ mostra a função de distribuição normal. Os dados da demanda por eletricidade hora por hora da última semana de 2021 também tiveram comportamento esperado para o teste ARIMA.

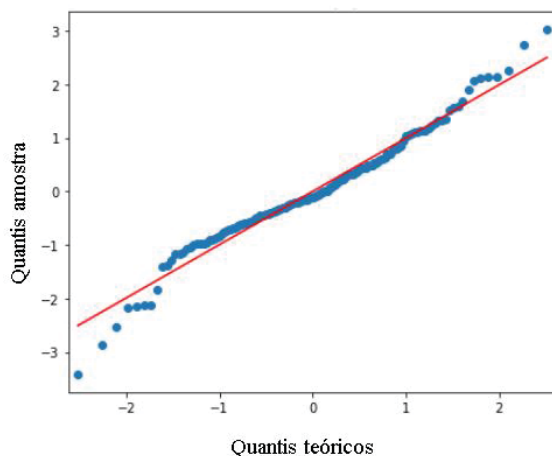


Figura 16: Gráfico normal quanti-quantil demanda por hora. Fonte: O Autor (2022). O eixo x são os quantis teóricos e o eixo Y os quantis da amostra. Os dados da demanda por hora ainda que tenham alguns pontos que fogem da reta da distribuição, apresentam em sua maioria um comportamento esperado de estacionariedade.

2.5 Aplicação do Modelo LSTM

Para o modelo LSTM utilizamos a biblioteca Keras e importamos um modelo já pronto e separamos as bases de treino com 70% do total dos dados e teste com os restantes 30% do treino. O modelo treinou com 310 epochs, learning rate 0.01, 1 unidade em: input layer, hidden size, number of layers e number of classes.

A base de horas e diária foram treinadas integralmente sem a necessidade de fazer os recortes e sem transformar em logaritmo. Em seguida foram feitas as transforma-

ções logarítmicas para ambas as bases e rodado o modelo com os mesmos parâmetros. Primeiramente a base foi transformada em uma matriz de 1 coluna para 56215 linhas para a base de horas e 2367 para a base diária.

A base foi dividida em duas bases: dataX e dataY, como os recursos independentes (excluindo a variável objetivo) e as variáveis dependentes (variáveis objetivo). O modelo usou a base trainX como recursos e trainY como dados observados. Para realizar o teste utilizou-se a testX como os recursos e testY para validar os resultados preditos.

3 Resultados e discussões

Neste tópico é discutida a acurácia de cada modelo e é feito a comparação de seus resultados. A métrica utilizada para comparação é a RMSE (root mean square error), utilizada outras vezes na literatura [14] por ser justamente a melhor maneira de se comparar modelos de predição de séries temporais [2]. No final deste tópico mostramos também a performance de cada modelo com cada base de dados plotado no gráfico mostrando os dados contra o que foi predito.

3.1 Medidas de performance

A medida RMSE foi usada para comparação de modelos de séries temporais no trabalho de Peter and Pius, 2019 [12]. Para explicar o modelo digamos que temos uma amostra n de erros E no modelo calculado como $(e_i, i=1,2,\dots, n)$. Assumimos que o erro da amostra não tem viés, temos então a seguinte equação:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (10)$$

Assumindo que os erros têm uma distribuição normal, temos então que o RMSE nos ajuda a ter uma noção geral da distribuição dos erros [2].

3.2 Resultados de predições dos modelos

As análises dos gráficos mostram que os modelos foram bem-sucedidos para fazer a predição da demanda de energia elétrica das próximas horas e dos próximos dias, tanto o modelo ARIMA para períodos mais curtos (um mês ou uma semana) quanto o LSTM para períodos mais longos (1 ano ou mais, mesmo com toda a oscilação sazonal).

O modelo ARIMA foi aplicado primeiro na base de dados diária e obteve uma predição com RMSE de 640, transformando de volta da base logarítmica e 696 para a predição das próximas horas da última semana de dezembro de 2021. Pelas figuras pode-se observar que a predição das horas foi mais eficiente do que as dos dias.

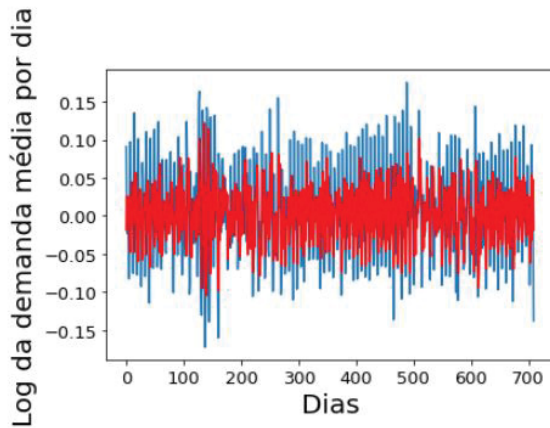


Figura 17: Predição ARIMA em vermelho e log da demanda diária em azul. Valor RMSE 0.064. Fonte: O Autor (2022).

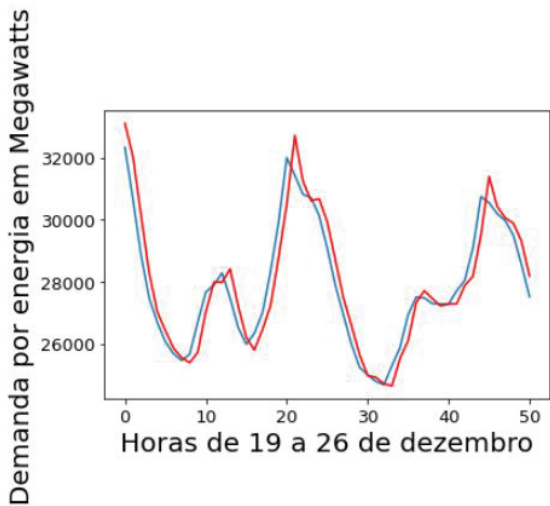


Figura 18: Predição ARIMA em vermelho e demanda por hora do dia 19 ao dia 26 de dezembro em azul. Valor do RMSE 683. Fonte: O Autor (2022).

O modelo LSTM para o log da demanda por horas obteve um RMSE de 0.074:

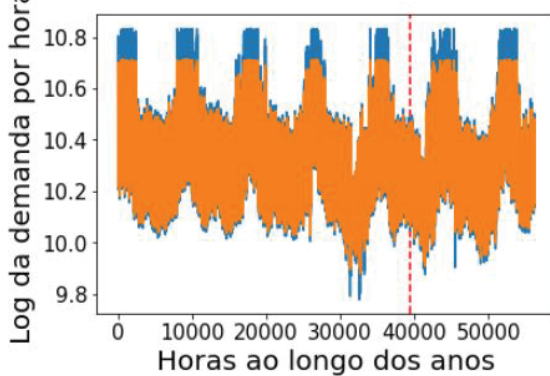


Figura 19: Predição LSTM em laranja e observações do log da demanda por horas em azul. Fonte: O Autor (2022).

O modelo LSTM para a base de dados diária obteve um RMSE de 0.060:

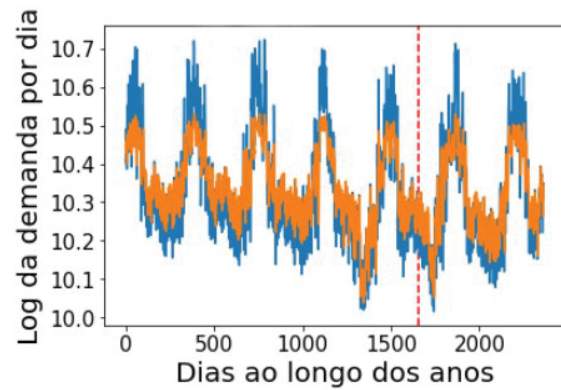


Figura 20: Predição LSTM em laranja e log da demanda diária em azul. Fonte: O Autor (2022).

Tabela 2: Resultados dos modelos de predição.

Resultados RMSE	Dados por hora*	Dados diários
LSTM	0.074	0.060
ARIMA	0.068	0.064

*Modelo ARIMA foi rodado apenas para uma semana

4 Conclusões

Os modelos foram eficientes para realizar as predições, porém a métrica RMSE não se mostrou suficiente para a escolha do melhor modelo, tendo em vista a performance das predições pelos gráficos e a necessidade de se fazer um recorte dos dados para testar ARIMA. No entanto, o modelo ARIMA se mostrou muito eficiente mesmo com dados não estacionários, como foi o caso da predição de demanda diária, em que pelos gráficos conseguimos ver que o modelo obteve valores muito próximos do observado. O modelo ARIMA também se mostrou eficiente para a predição de curto prazo, ou seja, para a predição da demanda das próximas horas dentro de uma semana, além de não precisar fazer uma transformação logarítmica. O modelo LSTM, em compensação, consegue fazer predições para grandes bases de dados sem a necessidade de recorte e com uma acurácia relevante. Para trabalhos futuros seria interessante utilizar de modelos que suportam variáveis externas como preços, oferta e demanda de energias substitutas como gás natural, temperatura e sazonalidade como por exemplo o modelo SARIMAX, ou até mesmo outros modelos de machine learning e comparar o tempo computacional para uma série de dados grande como a de demanda por horas, semanas e/ou dias.

Referências

- [1] Bouktif S, Fiaz A, Ouni A, Serhani MA, *Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting*. Energies. 2020; 13(2):391.
- [2] Chai, T. and Draxler, R. R., *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments*

- against avoiding RMSE in the literature, (Geosci. Model Dev., 7, 1247–1250. 2014).
- [3] Fattah J, Ezzine L, Aman Z, El Moussami H, Lachhab A, *Forecasting of demand using ARIMA model. International Journal of Engineering Business Management*, (January, 2018).
- [4] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, *Time Series Analysis: Forecasting and Control*, (Wiley, 2016), 5^a ed.
- [5] Jian Qi Wang, Yu Du, Jing Wang, *LSTM based long-term energy consumption prediction with periodicity*, (Energy, Vol 197. 2020).
- [6] Kakoli Goswami and Aditya Bihar Kandali. *Electricity Demand Prediction using Data Driven Forecasting Scheme: ARIMA and SARIMA for Real-Time Load Data of Assam*, (2020 International Conference on Computational Performance Evaluation (ComPE) North-Eastern Hill University, Shillong, Meghalaya, India.2020.)
- [7] Karin Kandananond. *Electricity demand forecasting in buildings based on ARIMA and ARX models*, In Proceedings of the 8th International Conference on Informatics, Environment, Energy and Applications (IEEA '19). Association for Computing Machinery, New York, NY, USA, 268–271. 2019.
- [8] Khan PW, Kim Y, Byun Y-C, Lee S-J. *Influencing Factors Evaluation of Machine Learning-Based Energy Consumption Prediction*. Energies. 2021
- [9] Hwang E, Rho S, Baik SW. *Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM*. Applied Sciences. 2019
- [10] N.D. Lewis. *DEEP TIME SERIES FORECASTING With PYTHON An Intuitive Introduction to Deep Learning for Applied Time Series Modeling*. 2016.
- [11] Nelson Fumo and M.A. Rafe Biswas. *Regression analysis for prediction of residential energy consumption. Mechanical Engineering Department, The University of Texas, Tyler, United States, 2015*.
- [12] N. Cruz, L. G. Marín and D. Sáez. *Prediction Intervals With LSTM Networks Trained By Joint Supervision*, International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2019.
- [13] Peter T. Yamak, Li Yujian, and Pius K. Gadosey. *A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting*. In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2019). Association for Computing Machinery, New York, NY, USA, 49–55. 2019.
- [14] Rahul Kumar Agrawal, Frankle Muchahary. *Long Term Load Forecasting with Hourly Predictions based on Long-Short-Term-Memory Networks*. Dept. of Electrical Engineering, Delhi Technological University, New Delhi, India. 2018.
- [15] Rini Nur Hasanah. *Comparison Analysis of Electricity Load Demand Prediction using Recurrent Neural Network (RNN) and Vector Autoregressive Model (VAR)*. Electrical Engineering Department Faculty of Engineering Universitas Brawijaya. Malang, Indonesia. 2020
- [16] Rui Fu, Zuo Zhang and Li Li. *Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction*. 31st Youth Academic Annual Conference of Chinese Association of Automation. Wuhan, China. Nov-2016.
- [17] Sepp Hochreiter and Jurgen Schmidhuber. *Long Short-Term Memory*. MIT. Neural Computation 9, 1735–1780. 1997.
- [18] Sima Siami-Namini, Neda Tavakoli and Akbar Siami Namin. *A Comparison of ARIMA and LSTM in Forecasting Time Series*. 17th IEEE International Conference on Machine Learning and Applications. 2018.
- [19] Torres, J.F., Martínez-Álvarez, F. Troncoso, A. *A deep LSTM network for the Spanish electricity consumption forecasting*. Neural Comput Applic (2022).
- [20] U.S. Energy Information Administration. Washington, DC, 2022. Disponível em: <https://www.eia.gov/electricity/data.php>. Acesso em: 5 jun. 2022.
- [21] COLAH'S BLOG. Understanding LSTM Networks. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 4 jun. 2022. MACHINE LEARNING MAESTRY. Disponível em: <https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/:text=We>