

LSTNETA: UM MODELO DE REDES NEURAS ARTIFICIAIS HÍBRIDAS PARA A PREDIÇÃO DE CONSUMO DE ENERGIA ELÉTRICA¹

 <https://orcid.org/0000-0001-5511-9453>  [Ricardo Augusto Manfredini²](#)

Resumo: Apresentamos um estudo comparativo das previsões de consumo de eletricidade utilizando o método SARIMAX (Seasonal Auto Regressive Moving Average eXogenous variables), o modelo HyFis2 (Hybrid Neural Fuzzy Inference System) e o modelo LSTNetA (Long and Short Time Series Network Adapted), uma rede neural híbrida contendo camadas GRU (Gated Recurrent Unit), CNN (Convolutional Neural Network) e camadas densas, especialmente adaptadas para este estudo de caso. O estudo experimental comparativo desenvolvido mostrou um resultado superior para o modelo LSTNetA com previsões de consumo muito mais próximas do consumo real. O modelo LSTNetA no estudo de caso teve um rmse (root mean squared error) de 198,44, o modelo HyFis2 602,71 e o método SARIMAX 604,58.

Palavras-chave: redes neurais; energia elétrica - consumo; análise de séries temporais.

INTRODUÇÃO

Este trabalho teve como objetivo a predição do consumo de energia elétrica de um prédio comercial com utilização de RNAs (Redes Neurais Artificiais). Foram utilizadas e testadas diversas arquiteturas de RNAs e selecionada uma arquitetura híbrida (Densa, Convolutacional e Recorrente), descrita originalmente por Lai, G. et al. (2018) e adaptada para este estudo de caso.

As seções deste trabalho estão organizadas da seguinte forma: inicialmente é feita uma fundamentação teórica dos conceitos, técnicas e ferramentas utilizadas. Posteriormente são definidos os materiais e métodos que serão utilizados para a validação do modelo proposto. Finalmente, são apresentados os resultados do modelo proposto comparando-o com outros modelos.

2 METODOLOGIA

Este trabalho foi realizado no *Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development* (GECAD³), centro de pesquisa localizado no Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto ISEP/IPP, Porto, Portugal. Da mesma forma que o modelo HyFIS2 (Josi et al.; 2016), o modelo proposto utiliza os dados reais de consumo elétrico de setores do Prédio N do

¹Projeto de Pós-Doc em colaboração com o GECAD do ISEP/IPP.

² Professor do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, *Campus Farroupilha*. Contato: ricardo.manfredini@farroupilha.ifrs.edu.br.

³<http://www.gecad.isep.ipp.pt/GECAD/Pages/Pubs/PublicationsPES.aspx>.

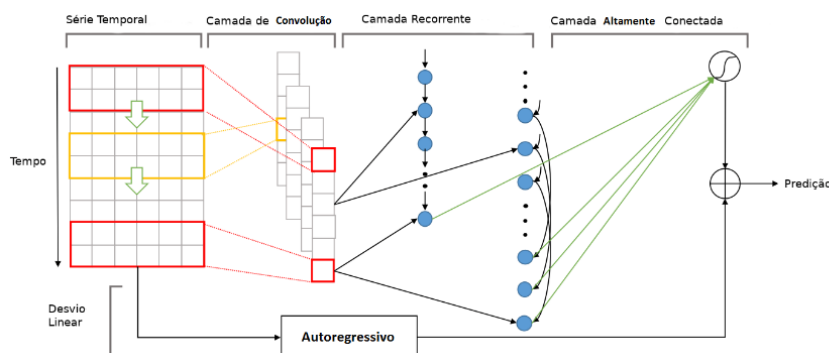
ISEP/IPP onde está localizado o GECAD. O prédio possui cinco medidores de energia que armazenam os dados de consumo de energia elétrica de setores específicos do prédio, com intervalo de tempo de 10 segundos. Essas informações, bem como dados meteorológicos, são armazenados num servidor SQL automaticamente, através de agentes desenvolvidos em Java.

Para validação do modelo descrito a seguir, foram realizados testes utilizando os mesmos dados de consumo aplicados ao modelo SARIMAX e o HyFIS2. O setor dos laboratórios do Prédio N não foi computado, pois possui grande variação de consumo em razão dos experimentos lá conduzidos que geram muitos *outliers* no histórico de consumo. Para os testes dos experimentos, foram realizadas médias horárias dos consumos armazenados a cada dez segundos, devido à necessidade de predição da próxima hora de consumo.

2.1 O Modelo Long and Short Time series Network Adapted (LSTNetA)

O modelo desenvolvido para a predição de consumo energético utilizou como base o modelo proposto por Lai et al. (2018), representado na Figura 1, constitui-se de uma RNA híbrida, com três camadas distintas. Inicialmente possui uma camada convolucional para a extração de padrões de curto prazo da série temporal, tem como entrada a série temporal, a saída dessa camada é a entrada da camada recorrente que memoriza informações históricas da série temporal, que, por sua vez, sua saída é a entrada da camada densa altamente conectada. Finalmente a saída da camada altamente conectada é combinada com a saída da regressão linear autorregressiva (ARMA) (Zhang, G. P.; 2003) garantindo que a saída terá a mesma escala da entrada, compondo assim a predição.

Figura 1 - Arquitetura do modelo LSTNetA



Fonte: Adaptado de Lai *et al* (2018).

A Figura 2 sumariza a implementação da rede LSTNetA. A camada de convolução está representada pela classe **Conv2D**, a camada recorrente está representada pelas classes **GRU**, a camada densa está representada pelas classes **Dense**, a autorregressão está representada na classe **PostARTrans**.

É importante salientar que a camada recorrente usa uma das variantes de RNN (*Recurrent Neural Network*) a GRU (*Gated Recurrent Unit*) (Chung et al, 2014), esse modelo de RNA assim como as LSTM (*Long Short-Term Memory*) visa resolver o problema da memória de curto prazo das RNN que, em séries longas, têm dificuldade de transportar os resultados de etapas anteriores para as posteriores.

Figura 2 - Sumário da implementação do LSTNet.

```
This may be caused by multiline strings or comments not indented at the same level as the code.
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 24, 24)]	0	
reshape (Reshape)	(None, 24, 24, 1)	0	input_1[0][0]
conv2d (Conv2D)	(None, 19, 1, 100)	14500	reshape[0][0]
dropout (Dropout)	(None, 19, 1, 100)	0	conv2d[0][0]
reshape_1 (Reshape)	(None, 19, 100)	0	dropout[0][0]
pre_skip_trans (PreSkipTrans)	(None, 1, 100)	0	reshape_1[0][0]
gru (GRU)	[(None, 100), (None, 60600)		reshape_1[0][0]
gru_1 (GRU)	[(None, 5), (None, 5 1605		pre_skip_trans[0][0]
dropout_1 (Dropout)	(None, 100)	0	gru[0][1]
post_skip_trans (PostSkipTrans)	(None, 95)	0	gru_1[0][1] input_1[0][0]
pre_ar_trans (PreARTrans)	(None, 24)	0	input_1[0][0]
concatenate (Concatenate)	(None, 195)	0	dropout_1[0][0] post_skip_trans[0][0]
flatten_1 (Flatten)	(None, 24)	0	pre_ar_trans[0][0]
flatten (Flatten)	(None, 195)	0	concatenate[0][0]
dense_1 (Dense)	(None, 1)	25	flatten_1[0][0]
dense (Dense)	(None, 24)	4704	flatten[0][0]
post_ar_trans (PostARTrans)	(None, 24)	0	dense_1[0][0] input_1[0][0]
add (Add)	(None, 24)	0	dense[0][0] post_ar_trans[0][0]

```

Total params: 81,434
Trainable params: 81,434
Non-trainable params: 0

```

Fonte: Elaborado pelo autor (2022).

Na etapa de *backpropagation*, processo de aprendizagem das RNAs, as RNNs sofrem com o problema da dissipação do gradiente (*The Vanishing Gradient*

Problem) (Schmidhuber, J; 2015). Gradientes são valores usados para atualizar os pesos das redes neurais. O problema da dissipação do gradiente é quando esses propagados durante o treinamento de uma rede, vão sofrendo multiplicações por valores menores que 1 a cada camada da rede atravessada, chegando às camadas iniciais da rede com valores ínfimos. Isso faz com que o ajuste dos pesos, calculados a cada iteração do treinamento da rede, sejam também ínfimos, isso onera o treinamento da rede.

Dessa forma, nas RNNs, as camadas que recebem uma pequena atualização do gradiente param de aprender, com isso, as RNNs podem esquecer o que foi visto em sequências mais longas, tendo assim uma memória de curto prazo.

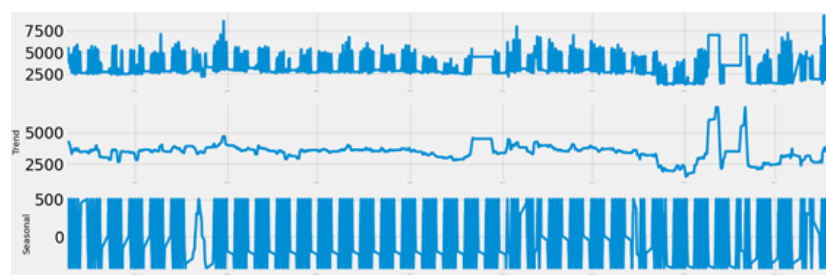
Numa arquitetura típica de uma GRU, basicamente o que a difere de uma RNN padrão são as portas de descarte (*reset gate*) e de atualização (*update gate*), que através da aplicação das funções de ativação *Sigmoid* e *tanh*, é definido se a saída anterior h_{t-1} será considerada ou descartada para o cálculo da nova saída.

O modelo LSTNetA foi desenvolvido na linguagem de programação Python versão 3.7 (Python; 2021) utilizando a biblioteca de aprendizagem de máquina, desenvolvida pelo Google, TensorFlow versão 2.0.

3 RESULTADOS

A Figura 3 representa a série temporal do consumo de energia elétrica utilizada pelo modelo SARIMAX, para treinar e testar o modelo LSTNetA e o HyFIS2. O gráfico superior representa a série histórica do consumo em *watts/hora*, que inicia às zero horas de 08/04/2019 às oito horas de 20/12/2019. O gráfico ao centro mostra a tendência calculada da série e o inferior à sua sazonalidade.

Figura 3 - Série histórica de consumo.



Fonte: Elaborado pelo autor (2022).

3.1 SARIMAX

O método SARIMAX é um método estatístico de análise de séries temporais, possibilitando a predição através de regressões lineares. Dessa forma, não se pode caracterizá-lo como um algoritmo de aprendizagem de máquina. No escopo deste trabalho, foi aplicado para obter dados de predição de um modelo amplamente utilizado, obtendo-se resultados para comparação com o modelo proposto e com o modelo HyFIS2.

Para verificar a acurácia de todos os modelos abordados, neste trabalho, foram utilizados os últimos 120 registros, correspondentes a cinco dias de consumo, para comparação entre o consumo real e o predito. Para cálculo do erro utilizado para a verificação dos resultados deste trabalho, em todos os modelos, foi utilizada a raiz do erro quadrático médio (*Root Mean Square Error* – RMSE). A aplicação desse modelo resultou num RMSE médio de 604,72 que foi considerado como acurácia desse modelo, neste trabalho.

3.2 Modelo HyFIS2

O modelo HyFIS2 (*Hybrid neural Fuzzy Inference System*) utiliza uma abordagem híbrida com a combinação de RNA densa e lógica difusa (*fuzzy logic*). O sistema inclui cinco camadas. Na primeira camada, os nós são as entradas que transmitem os sinais para a próxima camada. Na segunda e na quarta camadas, os nós atuam como funções de pertinência para expressar as variáveis linguísticas difusas de entrada-saída. Nessas camadas, os conjuntos *fuzzy* definidos para as variáveis de entrada-saída são representados como: grande (L), médio (M) e pequeno (S). No entanto, para algumas aplicações, estes podem ser mais específicos e representados como, por exemplo, positivo grande (LP), positivo pequeno (SP), zero (ZE), negativo pequeno (SN) e negativo grande (LN). Na terceira camada, cada nó é um nó de regra e representa uma regra difusa. Os pesos de conexão entre a terceira e a quarta camada representam fatores de certeza das regras associadas, ou seja, cada regra é ativada e controlada pelos valores de peso. Por fim, a quinta camada contém o nó que representa a saída do sistema.

A aplicação desse modelo resultou num RMSE médio de 602,71 que foi considerado como acurácia desse modelo, neste trabalho.

3.3 Aplicação do Modelo LSTNETA

O treinamento da RNA LSTNetA foi realizado conforme descrito anteriormente, utilizando-se os dados do consumo real de energia elétrica do prédio N do ISEP/IPP onde está localizado o GECAD, excetuando-se o setor de laboratórios. A série histórica analisada foi das zero horas de 08/04/2019 às oito horas de 20/12/2019, com medições a cada dez segundos, totalizadas a cada hora, resultando 4186 registros, contendo hora e consumo. O treinamento foi realizado com taxa de aprendizagem de 0.0003, utilizando o método estocástico Adam (Kingma and Ba, 2015) de otimização da descida do gradiente para atualização dos pesos no processo de *backpropagation*. Para os pesos iniciais da RNA, foi utilizado o algoritmo *VarianceScaling* (He, K. et al, 2010) que gera pesos iniciais com valores na mesma escala das entradas. O kernel de convolução utilizado foi uma matriz identidade 6x6 e foi realizado um loop de treinamento com 1000 épocas. Todos esses parâmetros foram obtidos de forma experimental e os de melhor resultado, selecionados. A comparação entre o consumo real e o predito, é demonstrado na Figura 4. A Figura 5 mostra os erros RMSE apurados. A aplicação desse modelo resultou num RMSE médio de 198,44 que foi considerado como acurácia desse modelo, neste trabalho.

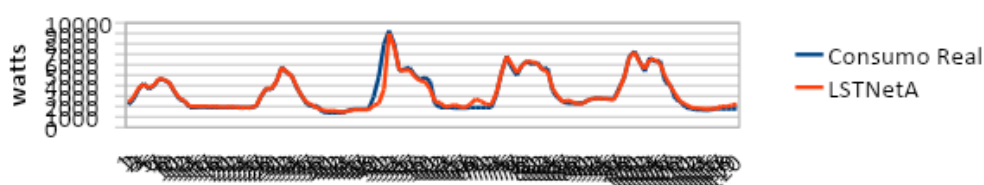
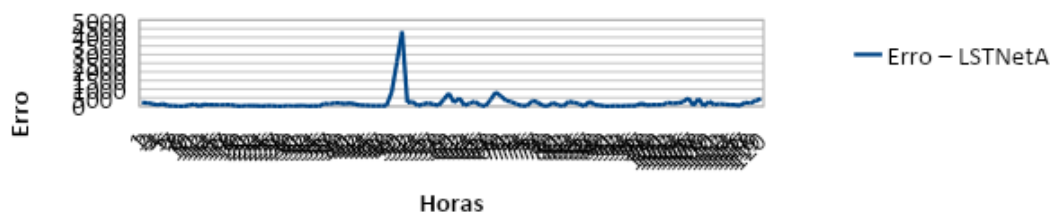


Figura 4 - Comparativo Consumo Real X LSTNetA.

Fonte: Elaborado pelo autor (2022).

Figura 5 - Erros verificados do modelo LSTNetA.



Fonte: Elaborado pelo autor (2022).

4 CONSIDERAÇÕES FINAIS

A Tabela 1 mostra um fragmento dos resultados dos três modelos, a coluna *Data e Hora*, a coluna *Real* mostrando o consumo real de eletricidade em *watts* naquela data e hora, a coluna *LSTNetA* a predição desse modelo na data e hora, a coluna *Erro – LSTNetA* o erro absoluto desse modelo na predição, a coluna *HyFIS2* a predição desse modelo na data e hora, a coluna *Erro – HyFIS2* o erro absoluto desse modelo na predição, finalmente as colunas *SARIMAX* e *Erro – SARIMAX*, representando a predição e o erro absoluto, respectivamente, no modelo SARIMAX.

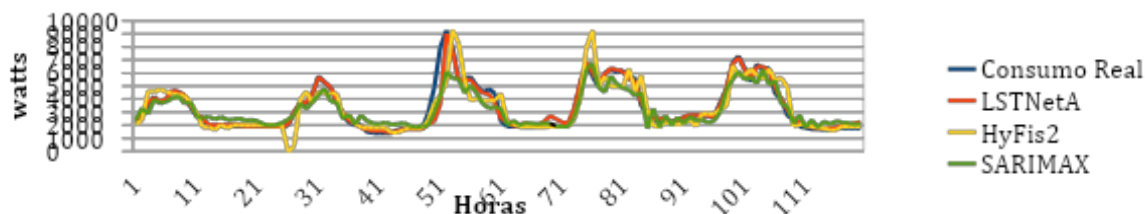
Comparando os resultados dos modelos SARIMAX, HyFIS2 e LSTNetA, pode-se observar, como demonstrado na Figura 6, que o método LSTNetA, com os dados utilizados para teste, foi o que apresentou as predições mais próximas do consumo real de energia elétrica, onde a linha vermelha, que representa as predições do modelo LSTNetA, em grande parte do período, sobrepõe a linha azul que representa o consumo real. Isso demonstra uma predição muito próxima do valor real de consumo, com erros baixos.

Tabela 1 - Fragmento de Predições e Erros dos 3 Modelos

Data e Hora	Consumo Real	LSTNetA	Erro – LSTNetA	HyFis2	Erro – HyFis2	SARIMAX	Erro – SARIMAX
19/12/2019 09:00	4759,38	4824,27	64,89	3427,13	1332,25	4721,76	37,619
19/12/2019 10:00	6781,51	6685,28	96,2346	6583,38	198,13	5516,26	1265,2476
19/12/2019 11:00	7279,1	7194,26	84,8373	5798,56	1480,54	6124,2	1154,8976
19/12/2019 12:00	6332,88	6247,08	85,8038	5798,38	534,5	5497,1	835,7849
19/12/2019 13:00	5350,34	5569,95	219,6063	6322,98	972,64	5653,27	302,9276
19/12/2019 14:00	6677,56	6499,5	178,0639	5798,37	879,19	5197,56	1479,9983

Fonte: Elaborado pelo autor (2022).

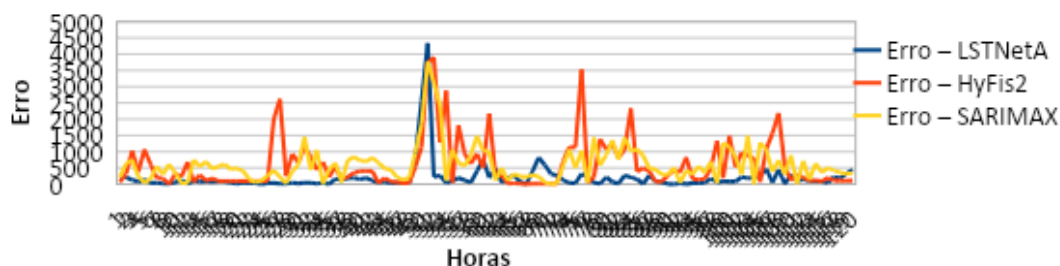
Figura 6. Comparativo Consumo Real X Modelos de Predição.



Fonte: Elaborado pelo autor (2022).

A Figura 7 representa os erros (RSME) dos três modelos, permitindo comparar a assertividade das predições de cada um dos métodos e, ainda, concluir que o método LSTNetA apresentou uma melhor eficácia em suas predições em comparação aos métodos SARIMAX e HyFIS2. Essa afirmação pode ser corroborada com os dados apresentados na Tabela 2, onde o erro total médio do modelo LSTNetA é significativamente menor que os demais modelos.

Figura 7 - Comparativos de erros verificados em todos os modelos.



Fonte: Elaborado pelo autor (2022).

Tabela 2 - RSME dos 3 Modelos Testados

	Erro – LSTNetA	Erro – HyFis2	Erro – SARIMAX
RSME	198,4496	602,7109	604,581

Fonte: Elaborado pelo autor (2022).

REFERÊNCIAS

Chung, J., Gulcehre, C., Cho, K., Bengio, Y. **Empirical evaluation of gated recurrent neural network on sequence modeling**. in NIPS 2014 Workshop on Deep Learning, December 2014.

He, K. at al. **Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification**. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

Lai, G., Chang, W.C., Yang, Y., and Liu, H. **Modeling long- and short-term temporal patterns with deep neural networks**. 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018, 95–104.

Kingma, D. P. and Ba, J. Adam: **A Method for Stochastic Optimization**. Computer Science, Mathematics ICLR 2015. (SSCI). 6-9 Dec. 2016R. Disponível em: <https://arxiv.org/abs/1412.6980.pdf>. Acesso em: mar. 2021.

Python. **Python is a programming language that lets you work quickly and integrate systems more effectively**. Disponível em: <https://www.python.org>. Acessado em: 01/03/2021

SARIMAX. **SARIMAX: Introduction**. Disponível em: https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_sarimax_state.html. Acessado em: 01/03/2021

SCHMIDHUBER, J. **Deep learning in neural networks: An overview**. Neural Networks, volume 61, January 2015, Pages 85-117. Disponível em: <https://arxiv.org/pdf/1404.7828.pdf>. Acesso em: 01 mar. 2021.

TensorFlow. **Uma plataforma completa de código aberto para machine learning**. Disponível em: <https://www.tensorflow.org>. Acesso em: mar. 2021.

Zhang, G. P. **Time series forecasting using a hybrid arima and neural network model**. Neurocomputing, 50:159–175, 2003.