



EMPREGO DE REDES NEURAIAS LSTM NA PREDIÇÃO DE SÉRIES TEMPORAIS APLICADAS AO CONSUMO DE ENERGIA ELÉTRICA

A. P. Andrade Filho ^{1,2}

¹UFU – Universidade Federal de Uberlândia - Engenharia Elétrica

²UFG – Universidade Federal de Goiás - Mestrado em Engenharia Elétrica e de Computação

Resumo - Com a estiagem em nível recorde no Brasil em 2021, os custos relacionados ao consumo de energia elétrica aumentaram de maneira significativa. Diante desta realidade, o objetivo deste artigo é realizar uma sucinta explanação sobre as redes neurais LSTM (Long Short-Term Memory) e sua aplicação na predição de séries temporais de consumo de energia elétrica. A metodologia empregada baseou-se em um estudo de caso, de uma grande empresa do setor de eletrodomésticos do estado de Minas Gerais. Utilizou-se uma série temporal com 96 amostras de consumo, coletadas em intervalos de 15 minutos, em um período de 24 horas. Os resultados alcançados foram obtidos através de simulação no software Matlab®, onde verificou-se, que o emprego da rede LSTM é capaz de prever com razoável margem de erro, os estados futuros da série temporal após treinadas e testadas com parâmetros adequados. Desta forma, conclui-se que o emprego das redes neurais LSTM são uma alternativa viável para o problema proposto, e podem contribuir para a predição de séries temporais de consumo de energia elétrica.

Palavras-Chave – Predição de Consumo de energia elétrica, Redes neurais LSTM.

USE OF NEURAL NETWORKS LSTM IN THE PREDICTION OF TIME SERIES APPLIED TO ELECTRIC ENERGY CONSUMPTION

Abstract - With the drought at a record level in Brazil in 2021, costs related to electricity consumption increased significantly. Given this reality, the aim of this article is to provide a brief explanation about the application of LSTM (Long Short-Term Memory) neural networks in the prediction of time series of electrical energy consumption. The methodology used was based on a case study of a large company in the home appliance sector in the state of Minas Gerais. We used a time series with 96 consumption samples, collected at 15-minute intervals, over a 24-hour period. The achieved results were obtained through simulation in the Matlab® software, where it was verified

that the use of the LSTM network is able to predict, with a reasonable margin of error, the future states of the time series after being trained and tested with adequate parameters. Thus, it is concluded that the use of LSTM neural networks are a viable alternative for the proposed problem. The present time series was obtained by measuring the parameters of electrical energy consumption of this unit, where it was possible to record in kWh the consumption at 15-minute intervals over a day, totaling 96 sampled in 24 hours.

Keywords – Prediction of Electric Power Consumption, LSTM Neural Networks.

NOMENCLATURA

<i>RNN</i>	Redes Neurais Recorrentes.
<i>RNA</i>	Redes Neurais Artificiais.
<i>LSTM</i>	Rede de memória de longo prazo.
<i>RMSE</i>	Erro médio quadrático.
<i>Tanh</i>	Tangente Hiperbólica.
<i>GPU</i>	Unidade de Processamento Gráfico.
<i>CPU</i>	Unidade Central de Processamento.

I. INTRODUÇÃO

Com a estiagem em nível recorde no Brasil em 2021, até a data da elaboração deste artigo, os custos relacionados a geração e consumo de energia elétrica, têm aumentado de maneira significativa.

A fim de se apresentar uma contribuição para o uso eficiente da energia elétrica, buscando-se a otimização de recursos, este trabalho justifica-se em oferecer uma análise de estudo de caso, em uma grande empresa do setor de eletrodomésticos do estado de Minas Gerais. Utiliza-se como metodologia, as redes neurais artificiais LSTM para a previsão de série temporal de consumo de energia elétrica.

Buscou-se verificar o desempenho, em especial, das redes LSTM, devido a característica peculiar da base de dados, que apesar de curta, retrata o perfil de consumo real diário da empresa supracitada, uma vez que a série temporal é composta de 96 amostras coletadas de maneira automática em intervalos de 15 minutos ao longo 24 horas.

A. Referencial Teórico

Nos últimos anos, diversos estudos relacionados as séries temporais, vem sendo realizados, principalmente em áreas como mercado financeiro, serviços de internet e mercados consumidores como o de energia elétrica.

Uma série temporal é um conjunto de observações registradas sequencialmente no decorrer do tempo, podendo ser contínuas ou discretas. As séries temporais, em geral, têm quatro principais componentes, sendo: tendência, ciclo, sazonalidade, irregularidade ou fatores aleatórios [1].

Os problemas, aos quais as RNA's são aplicáveis, consistem basicamente em situações em que existam dados, experimentais ou não, que são apresentados à rede em uma etapa denominada de "treinamento", onde os pesos sinápticos são ajustados com intuito de que alguma tarefa seja realizada. Além da capacidade de aprendizado a partir de exemplos. As redes possuem a capacidade de generalização, que consiste na habilidade desta em apresentar soluções para dados distintos, dados de "validação", diferentes dos dados apresentados na etapa de treinamento. Para [2] as principais características das RNA's são: (i) o aprendizado a partir de exemplos; (ii) adaptabilidade; (iii) capacidade de generalização; (iv) tolerância a falhas; e (v) rápida implementação.

Em outro estudo relacionado as séries temporais, [3] propôs um modelo baseado em misturas gaussianas, na qual é utilizado o algoritmo EM (Expectation Maximization) para se estimar os parâmetros de ajuste das k gaussianas utilizadas no modelo de mistura, esse último método realiza o cálculo da verossimilhança (log Likelihood) para conseguir estimar os parâmetros de média, variância e peso das misturas com precisão.

II. REDES NEURAIS ARTIFICIAIS LSTM

Estudos da neurociência cognitiva revelam que o processo de entendimento daquilo que se lê, está diretamente relacionado com a memória. Para ler com entendimento é preciso inter-relacionar informações novas com informações preexistentes no cérebro do leitor, e para tanto a memória de longo prazo é crucial [4,5].

Notoriamente, os seres humanos não começam seus pensamentos do zero a cada segundo, ao ler o indivíduo entende cada palavra com base em sua compreensão das palavras anteriormente lidas e registradas em sua memória de curto prazo. Seus pensamentos têm persistência, não se começa a pensar do zero novamente, a cada nova palavra lida.

As redes neurais tradicionais não conseguem desempenhar tal comportamento, e isso certamente é um problema. Não está claro como uma rede neural tradicional poderia usar seu raciocínio sobre eventos anteriores para informar os posteriores, por outro lado, as redes neurais recorrentes resolvem esse problema, são redes com loops, permitindo que as informações persistam [6].

Na Fig. 1 um pedaço da rede neural A, olha para alguma entrada x_t e produz um valor h_t , um loop permite que as informações sejam passadas de uma etapa da rede para a próxima. Uma rede neural recorrente pode ser considerada como várias cópias da mesma rede, conforme Fig. 1, cada uma passando uma mensagem a um sucessor [6].

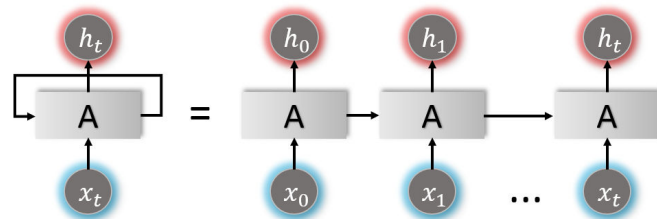


Fig. 1 – RNN desenrolada, adaptado de [6].

Essa natureza em cadeia revela que as redes neurais recorrentes estão intimamente relacionadas a sequências e listas. Elas são a arquitetura natural da rede neural a ser usada para esses dados. Um dos apelos das RNN's é a ideia de que elas podem ser capazes de conectar informações anteriores à tarefa presente, como a previsão de séries temporais.

Em muitas situações é necessário somente avaliar as informações recentes para realizar a tarefa atual. Considera-se um modelo de linguagem tentando prever a próxima palavra na frase "A chuva cai do...". Nota-se que não seria necessário nenhum contexto adicional, pois torna-se evidente que a próxima palavra será *céu*. Nesses casos, onde a lacuna entre as informações relevantes e o local necessário é pequena, os RNN's podem aprender a usar as informações anteriores.

Entretanto em algumas situações a análise do contexto se torna ainda mais relevante, ao se considerar tentar prever, por exemplo, a última palavra no texto "Eu cresci no Brasil ... eu falo português". Informações recentes sugerem que a próxima palavra é provavelmente o nome de um idioma, mas se, deseja-se restringir qual idioma, precisamos do contexto de *ter nascido no Brasil*, já informado no início da frase. É perfeitamente possível que a lacuna entre as informações relevantes e o ponto onde elas são necessárias se torne muito grande. Infelizmente, à medida que essa lacuna aumenta, as RNN's tornam-se incapazes de aprender a conectar as informações. Estudos práticos demonstram que as RNN's não parecem ser capazes de aprendê-los [6].

Já as redes LSTM's não apresentam tal limitação, redes de memória de longo prazo - geralmente chamadas apenas de LSTM's - são um tipo especial de RNN, capazes de aprender dependências de longo prazo [7].

As LSTM's são projetadas explicitamente para evitar o problema de dependência de longo prazo. Todas as redes neurais recorrentes têm a forma de uma cadeia de módulos repetidos de rede neural. Em RNN's padrões, esse módulo de repetição terá uma estrutura muito simples, com uma única camada de função de ativação Tanh. As funções de ativação são extremamente importantes nas redes neurais artificiais, atuando basicamente no processo de decisão sobre a ativação ou não de um neurônio específico, tomando a decisão final se a informação que o neurônio está recebendo é relevante para a informação fornecida ou deve ser ignorada [8,9]. A Fig. 2 ilustra a função de ativação escolhida, Tanh, bem similar a função sigmoide, simétrica em relação a origem, variando entre -1 e 1, ela é contínua e diferenciável em todos os pontos, não é linear, permitindo-se realizar o *backpropagation* facilmente nos erros.

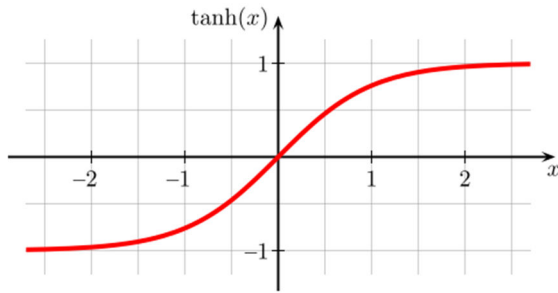


Fig. 2 – Função de ativação tanh (Tangente Hiperbólica).

As LSTM's também possuem tal estrutura em cadeia, entretanto, o módulo de repetição utiliza um mecanismo diferente, ao invés de ter uma única camada de rede neural, existem quatro, interagindo de uma maneira peculiar, conforme ilustrado na Fig. 3 [6].

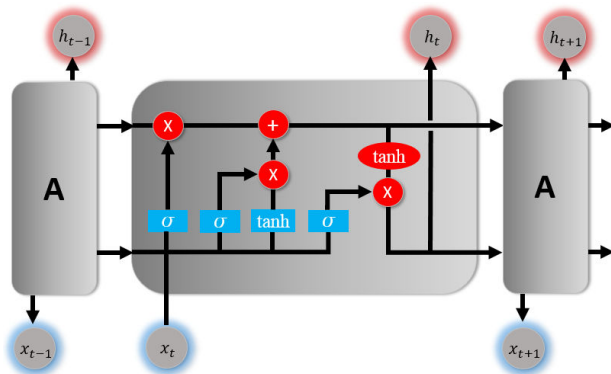


Fig. 3 – Módulo de repetição de uma LSTM, adaptado de [6].

Na Fig. 3 cada linha carrega um vetor inteiro, da saída de um nó às entradas de outros. Os círculos vermelhos representam operações pontuais, como adição, multiplicação e tangente hiperbólica de vetor, enquanto as caixas azuis são camadas de rede neural aprendidas. A fusão de linhas denota concatenação, enquanto uma bifurcação de linha denota seu conteúdo sendo copiado e as cópias indo para locais diferentes. A chave para as LSTM's é o estado da célula, a linha horizontal que atravessa a parte superior do diagrama, conforme Fig. 4. O estado da célula é como uma correia transportadora, ele percorre toda a cadeia, com apenas algumas interações lineares menores.

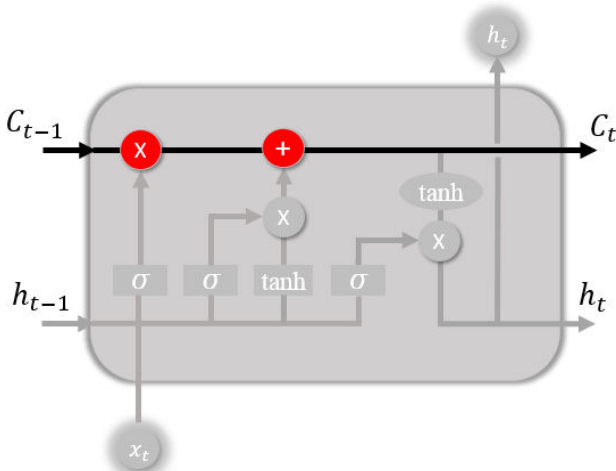


Fig. 4 – Estado da célula, adaptado de [6].

As LSTM's têm a capacidade de remover ou adicionar informações ao estado da célula, cuidadosamente regulado por estruturas chamadas de *portões*. Os *portões* são uma forma opcional de permitir a passagem de informações. Eles são compostos de uma camada de rede neural sigmoide e uma operação de multiplicação pontual. A camada sigmoide emite números entre zero e um, descrevendo quanto de cada componente deve ser deixado passar. Um valor de zero significa "não deixe nada passar", enquanto um valor de um significa "deixe tudo passar!". Uma rede LSTM tem três dessas portas, para proteger e controlar o estado da célula [6].

III. METODOLOGIA

Para prever os valores das etapas de tempo futuras da série apresentada na Fig. 5, treina-se uma rede LSTM de regressão sequência a sequência, em que as respostas são as sequências de treinamento com valores alterados por uma etapa de tempo.

A cada etapa de tempo da sequência de entrada, a rede LSTM aprende a prever o valor da próxima etapa de tempo. Para prever os valores de várias etapas de tempo no futuro, utiliza-se a função *predictAndUpdateState* do Matlab® para prever as etapas de tempo uma de cada vez e atualizar o estado da rede em cada previsão [10].

A saída é uma matriz de células, onde cada elemento é uma única etapa de tempo. Os dados são então remodelamos em um vetor de linha, para finalmente serem plotados. As etapas do processo podem ser observadas como se segue:

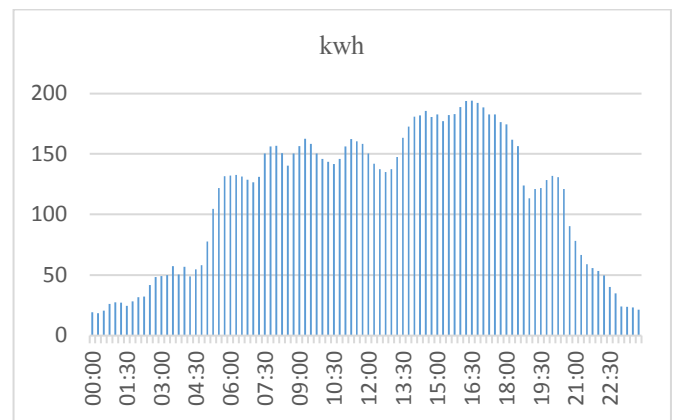


Fig. 5 – Série temporal escolhida.

A. Etapa Inicial

Nesta etapa a série ilustrada na Fig. 5, no formato Microsoft Excel é carregada ao Software Matlab®. Cria-se uma matriz de dados para armazenar as informações. Particiona-se os dados para treinamento e teste, respectivamente em 95% e 5%, e realiza-se o teste com o particionamento escolhido.

B. Padronização dos Dados

A fim de se evitar divergências no treinamento, efetua-se a padronização dos dados da série temporal, calculando e armazenando a Média e o Desvio Padrão, e aplicando-os a função *dataTrainStandardized* Matlab®.

C. Preparação dos Preditores e Respostas

Para prever os valores das etapas de tempo futuras, especifica-se as respostas como sequências de treinamento com valores alterados por uma etapa de tempo. Ou seja, a cada etapa de tempo da sequência de entrada, a rede LSTM aprende a prever o valor da próxima etapa de tempo. Os preditores são as sequências de treinamento sem o intervalo de tempo final.

D. Definindo a Arquitetura da rede LSTM

Cria-se uma rede de regressão LSTM, e especifica-se que a camada LSTM deve ter 200 unidades ocultas, esta quantidade de unidades ocultas foi escolhida levando-se em considerações simulações realizadas. Especifica-se as opções de treinamento, definindo o solucionador como 'adam' e treino por 250 épocas. Para evitar que os gradientes explodam, define-se o limite do gradiente para 1.

Especifica-se a taxa de aprendizado inicial em 0,005 e diminui-se a taxa de aprendizado após 125 épocas, multiplicando-a por um fator de 0,2.

E. Treinamento da LSTM

Treina-se a rede LSTM com as opções de treinamento especificadas usando o comando `trainNetwork` Matlab®.

F. Previsão das Etapas de Tempo Futuro

Para prever os valores de várias etapas de tempo no futuro, utiliza-se a função `predictAndUpdateState` Matlab® para prever as etapas de tempo uma de cada vez e atualizar o estado da rede em cada previsão. Para cada previsão, utiliza-se a previsão anterior como entrada para a função.

Padroniza-se então os dados de teste utilizando-se os mesmos parâmetros dos dados de treinamento.

G. Inicialização do Estado da Rede Neural

Inicializa-se o estado da rede, realizando-se uma previsão nos dados de treinamento utilizando-se o comando `XTrain` Matlab®.

Em seguida, faz-se a primeira previsão utilizando-se a última etapa da resposta de treinamento `YTrain` Matlab®. Implementa-se um loop nas previsões restantes e insere-se a previsão anterior em `predictAndUpdateState` Matlab®.

Para grandes coleções de dados, longas sequências ou grandes redes, as previsões na GPU são geralmente mais rápidas de calcular do que as previsões na CPU. Caso contrário, as previsões na CPU geralmente são mais rápidas de calcular. Para previsões de passo único de tempo, usamos a CPU. Para usar a CPU para previsão, defini-se a opção 'ExecutionEnvironment' de `PredictAndUpdateState` para 'cpu'.

H. Despadronização das Previsões e Cálculo do RMSE (Erro Médio Quadrático)

A despadronização das previsões é implementada utilizando-se os parâmetros calculados anteriormente, conforme apresenta-se em (1).

$$Y_{Pred} = Desvio\ Padr\tilde{a}o * Y_{Pred} + M\acute{e}dia \quad (1)$$

Plota-se o gráfico de progresso do treinamento apresentando-se o erro quadrático médio (RMSE) calculado a partir dos dados padronizados. Calcula-se o RMSE a partir das previsões não padronizadas.

Em seguida, plota-se a série de tempo de treinamento com os valores previstos, desta forma, pode-se comparar os valores previstos com os dados de teste. Na Fig. 6 apresenta-se o fluxograma do processo de previsão da série temporal.

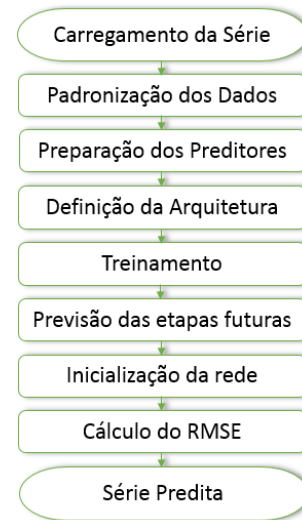


Fig. 6 – Fluxograma do processo de previsão da série temporal.

I. Atualização de Estado da Rede com Valores Observados

Pode-se atualizar o estado da rede com os valores observados em contrapartida aos valores previstos, uma vez que se tem acesso aos valores reais das etapas de tempo entre as previsões. Primeiramente inicializa-se o estado da rede. Para fazer previsões sobre uma nova sequência, redefine-se o estado da rede usando `resetState` Matlab®. A redefinição do estado da rede evita que as previsões anteriores afetem as previsões nos novos dados. Redefine-se o estado da rede e inicializa-se o estado da rede prevendo os dados de treinamento. Para cada previsão, se prevê a próxima etapa de tempo utilizando-se o valor observado da etapa de tempo anterior. Define-se a opção `ExecutionEnvironment` de `PredictAndUpdateState` para 'cpu'. Novamente despadroniza-se as previsões, calcula-se o RMSE e compara-se os valores previstos com os dados de teste. Entretanto, neste momento, as previsões são mais precisas ao atualizar-se o estado da rede com os valores observados em contrapartida aos valores previstos.

IV. SIMULAÇÃO E RESULTADOS

Nesta seção apresenta-se as simulações computacionais e os resultados alcançados com o emprego da rede LSTM. A Fig. 7 ilustra a série real utilizada, padronizada. As Fig. 8 a 11 ilustram os resultados alcançados.

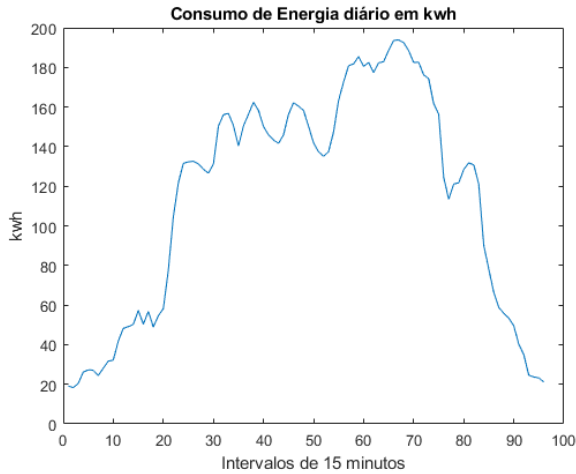


Fig. 7 – Série real.

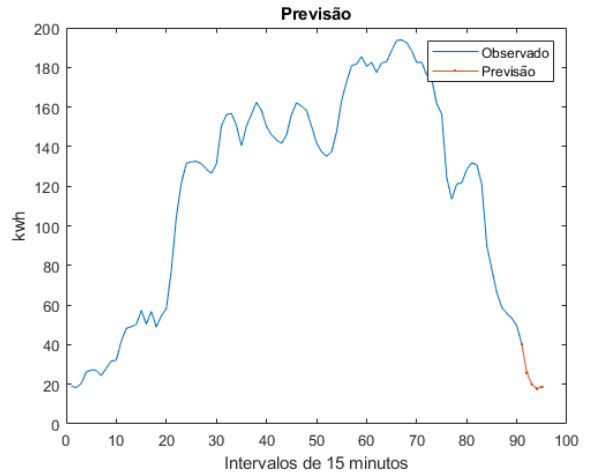


Fig. 10 – Série com valores atualizados

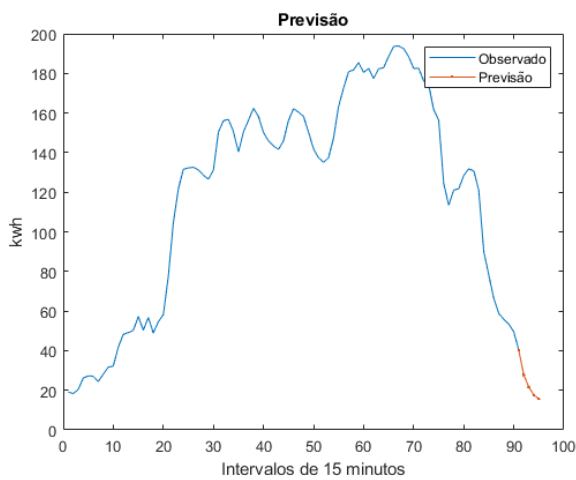


Fig. 8 – Série com valores Previstos

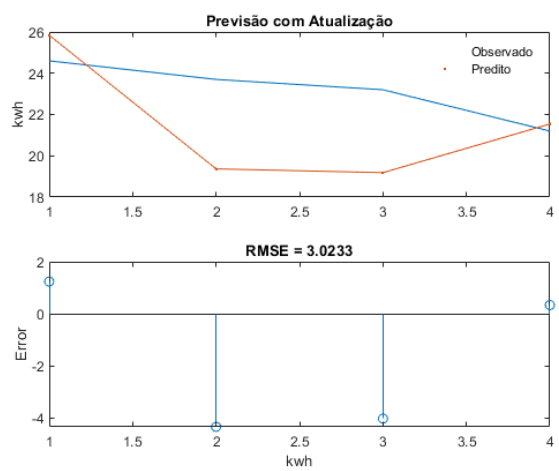


Fig. 11 – Erro para valores atualizados

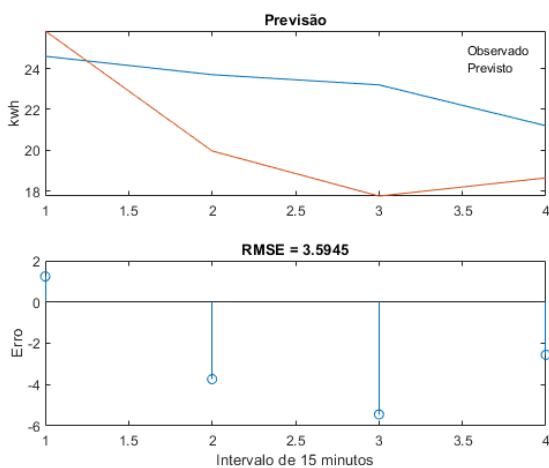


Fig. 9 – Erro para valores Previstos

V. CONCLUSÃO

Verificou-se que o emprego das redes LSTM na predição de séries temporais é viável, uma vez que tais redes conseguem prever com razoável margem de erro os estados futuros, após treinadas e testadas com parâmetros adequados. Durante o processo de simulação e treinamento, experimentou-se outras séries temporais, com o intuito de se avaliar o desempenho da rede frente a perfis de consumo diversos. Observou-se que o modelo empregado apresenta melhor desempenho quando aplicado em séries estacionárias, não obstante, a escolha do percentual de treinamento deve ser avaliada com cuidado, a fim de se garantir relevante e significativa amostra da série temporal, desta forma consegue-se minimizar previsões errôneas. Constata-se que o emprego das redes neurais LSTM podem contribuir para a predição de séries temporais de consumo de energia elétrica de maneira satisfatória. Sugere-se para trabalhos futuros a utilização de um banco de dados maior, com amostras diárias no decorrer de um mês, o que contribuiria para o conceito de generalização, trazendo com isso a possibilidade de avaliar melhor o *Overfitting*.

VI. AGRADECIMENTOS

À Escola de Engenharia Elétrica, Mecânica e Computação da Universidade Federal de Goiás, pela oportunidade de desenvolvimento deste trabalho, a CAPES DS pelo apoio financeiro, e aos membros participantes da XIX CEEL (Conferência de Estudos em Engenharia Elétrica) onde este trabalho foi apresentado.

VII. REFERÊNCIAS

- [1] R. Adhikari and R. K Agrawal, 2013 An Introductory Study on Time Series Modeling and Forecasting. LAP Lambert Academic Publishing, Germany, 2013.
- [2] A. P. Braga, Carvalho, A. C., e Ludemir, T.B. Redes Neurais Artificiais: teoria e aplicações. Livraria Técnica Científica, 2000.
- [3] A. P. Andrade Filho, GMM applied in time series detection processes. Stochastic Processes - University of Goiás, June, 2021.
- [4] R. M. Cosenza e L. B. Guerra, Neurociência e educação: como o cérebro aprende. PortoAlegre: Artmed; 2011.
- [5] S. Dehaene, Os neurônios da leitura: como a ciência explica a nossa capacidade de ler. Tradução Leonor Scliar-Cabral. Porto Alegre: Penso; 2012.
- [6] J. Colah, Understanding LSTM Networks, 2015.
- [7] S. Hochreiter and J. Urgan Schmidhuber, Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
- [8] C. A. Assis, Redes Neurais Profundas, 2021 – Notas de aula do professor, Universidade Federal de Goiás, Junho de 2021.
- [9] N. G. Andrew. Neural networks and deep learning. Coursera. <https://www.coursera.org/learn/neuralnetworks-deep-learning>, 325, 2017.
- [10] Wathworks, Time Series Forecasting Using Deep Learning. June de 2021.
<https://la.mathworks.com/help/deeplearning/ug/time-series-forecasting-using-deep-learning.html>

VIII. DADOS BIBLIOGRÁFICOS



Alfredo de Paulo Andrade Filho, nascido em Serro-MG em 18/04/1983. Graduado em Engenharia Elétrica pela Universidade Federal de Uberlândia (UFU), Especialista em Engenharia Elétrica e Engenharia de Controle e Automação. Discente do curso de Mestrado em Engenharia Elétrica e Computação da EMC/UFG.