



**Universidade Federal de Uberlândia  
Faculdade de Matemática**

**Bacharelado em Estatística**

**MODELAGEM DO CONSUMO DE  
ENERGIA ELÉTRICA NA REGIÃO  
SUDESTE COM SÉRIES TEMPORAIS E  
REDES NEURAIS ARTIFICIAIS**

**Mateus de Freitas Silva**

**Uberlândia-MG**

**2022**



Mateus de Freitas Silva

**MODELAGEM DO CONSUMO DE  
ENERGIA ELÉTRICA NA REGIÃO  
SUDESTE COM SÉRIES TEMPORAIS E  
REDES NEURAIS ARTIFICIAIS**

Trabalho de conclusão de curso apresentado à Co-  
ordenação do Curso de Bacharelado em Estatística  
como requisito parcial para obtenção do grau de  
Bacharel em Estatística.

Orientadora: Prof<sup>ª</sup> Dra. Nádia Giaretta Biase

**Uberlândia-MG  
2022**





**Universidade Federal de Uberlândia  
Faculdade de Matemática**

**Coordenação do Curso de Bacharelado em Estatística**

A banca examinadora, conforme abaixo assinado, certifica a adequação deste trabalho de conclusão de curso para obtenção do grau de Bacharel em Estatística.

Uberlândia, \_\_\_\_\_ de \_\_\_\_\_ de 2022.

**BANCA EXAMINADORA**

---

Prof<sup>a</sup> Dra. Nádia Giaretta Biase

---

Prof. Dr. Rodrigo Lambert

---

Prof<sup>a</sup> Dra. Maria Imaculada de Sousa Silva

**Uberlândia-MG  
2022**



# AGRADECIMENTOS

Agradeço aos meus pais, Geila e Agnelo, que desde sempre me motivaram a ingressar na UFU, sendo essa a melhor maneira de obter uma educação de qualidade que transformasse minha vida. Além disso, fizeram o que era possível para me proporcionar tranquilidade nos estudos e me incentivar em tudo o que me propus a fazer.

À minha noiva, Rafaela, que desde que nos conhecemos sempre me apoiou e me encorajou nas minhas decisões. Por estar sempre comigo nas conquistas, tristezas e preocupações, me incentivando a fazer o meu melhor e acreditando em mim.

Aos meus amigos, irmãos que a vida me deu, que sempre estiveram e ainda estão presentes em todos os momentos. Por me acolherem quando precisei de ajuda, e por celebrarem minhas conquistas como se fossem deles. Faço questão de levá-los para sempre comigo, todos são muito especiais para mim.

Agradeço também à minha orientadora, Nádia Giaretta Biase, por todos os conhecimentos transmitidos ao longo desses anos. Desde quando foi minha tutora no Programa de Educação Tutorial, depois, quando cursei a disciplina de Séries Temporais e agora, ao longo de todo o TCC.

Por fim, agradeço aos professores Rodrigo Lambert e Maria Imaculada de Sousa Silva, que aceitaram o convite para compor a banca examinadora e dar suas contribuições.



# RESUMO

A questão da energia elétrica, com seu complexo processo de produção, distribuição e consumo, sempre teve muita visibilidade, por questões ambientais, falhas de planejamento, apagões e viabilidade econômica. No Brasil, a região Sudeste é a principal consumidora de energia, e conta com uma matriz diversificada de produção. Por esse motivo, esta pesquisa aborda a série temporal mensal de consumo de energia elétrica na região, com o intuito de encontrar um modelo apto a fazer previsões para essa variável, e absorver o seu comportamento ao longo do tempo, sua tendência e sua sazonalidade. Foram empregadas técnicas de séries temporais, por meio da metodologia de Box-Jenkins, com modelos sazonais autorregressivos, do tipo SARIMA, e modelos de Redes Neurais Recorrentes do tipo LSTM (Long Short Term Memory). O critério utilizado para tomar uma decisão analítica sobre qual modelo é o melhor foi a Raiz do Erro Quadrático Médio, cuja sigla em inglês é RMSE, de modo que o modelo escolhido, em todos os casos, foi aquele com menor valor para o RMSE. Com base nisso, verificou-se que, para previsões a curto prazo, com realimentação de dados, o modelo de Redes Neurais se mostrou bastante eficaz. Por outro lado, se a intenção é desenvolver previsões a longo prazo, projetando o consumo para os 12 meses seguintes, de uma só vez, o modelo SARIMA é o mais adequado, evitando erros muito acentuados em alguns meses.

**Palavras-chave:** SARIMA, Autorregressivo, LSTM, RMSE, Previsão.



# ABSTRACT

The electricity issue, with its complex production, distribution and consumption process, has always had a lot of visibility, for environmental issues, planning failures, blackouts and economic viability. In Brazil, the Southeast region is the main consumer of energy, and has a diversified production matrix. For this reason, this research approaches the monthly time series of electricity consumption in the region, in order to find a model capable of making predictions for this variable, and absorb its behavior over time, its trend and its seasonality. Time series techniques were used using the Box-Jenkins methodology, with autorregressive seasonal models of type SARIMA, and models of Recurrent Neural Networks of type LSTM (Long Short Term Memory). The criterion used to make an analytical decision on which model is best was the Root Mean Squared Error, called by RMSE, so that the chosen model, in all cases, was the one with the lowest value for the RMSE. Based on this, it was found that, for short-term predictions, with data feedback, the Neural Networks model proved to be quite effective. On the other hand, if the intention is to develop long-term forecasts, projecting consumption for the next 12 months, at once, the SARIMA model is the most appropriate, avoiding very sharp errors in certain months.

**Keywords:** SARIMA, Autorregressive, LSTM, RMSE, Forecast.



# SUMÁRIO

<b>Lista de Figuras</b>	<b>I</b>
<b>Lista de Tabelas</b>	<b>III</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Fundamentação Teórica</b>	<b>5</b>
2.1 Séries Temporais . . . . .	5
2.1.1 Estacionariedade . . . . .	5
2.1.2 Processos Estocásticos . . . . .	6
2.1.3 Tendência e Sazonalidade . . . . .	6
2.1.4 O modelo autorregressivo integrado de médias móveis (ARIMA) . . . . .	9
2.1.5 Função de autocorrelação . . . . .	10
2.1.6 Método de Box-Jenkins . . . . .	11
2.1.7 Métodos de comparação de previsão . . . . .	12
2.1.8 Testes de adequação do modelo . . . . .	13
2.2 Redes Neurais Artificiais . . . . .	15
2.2.1 Redes Neurais do tipo ‘Unidades de Memória de Longo Prazo’ . . . . .	16
<b>3 Metodologia</b>	<b>21</b>
<b>4 Resultados</b>	<b>25</b>
4.1 Modelos Autorregressivos . . . . .	25
4.2 Modelos LSTM . . . . .	31
4.3 Comparação entre os modelos SARIMA e LSTM . . . . .	34
<b>5 Conclusões</b>	<b>37</b>
<b>Referências Bibliográficas</b>	<b>39</b>
<b>Apêndice A Modelo SARIMA</b>	<b>41</b>
<b>Apêndice B Modelo LSTM</b>	<b>49</b>



---

# LISTA DE FIGURAS

2.1	Estrutura de uma Rede Neural Artificial (RNA).	15
2.2	Estrutura de uma Rede Neural Recorrente (RNR).	16
2.3	Estrutura de uma Unidade de Memória de Longo Prazo (LTSM).	18
4.1	Consumo de energia elétrica na região Sudeste de janeiro/2002 a dezembro/2021.	25
4.2	Função de Autocorrelação da série temporal (ACF).	26
4.3	Cruzamento das médias com as amplitudes de cada par de observações.	26
4.4	Série original diferenciada em grau 1.	27
4.5	Funções de Autocorrelação e Autocorrelação Parcial para a série diferenciada.	27
4.6	Gráficos de ACF e PACF residuais.	29
4.7	Periodograma acumulado.	30
4.8	Previsões feitas pelo modelo SARIMA(1,1,0)(1,0,1) <sub>[12]</sub> .	30
4.9	Previsões feitas pelo modelo LSTM sem reposição.	32
4.10	Previsões feitas pelo modelo LSTM com reposição.	33
4.11	Série temporal original comparada com as previsões dos modelos ajustados.	35



---

# LISTA DE TABELAS

4.1	Avaliação dos modelos SARIMA propostos . . . . .	28
4.2	Critérios de seleção dos modelos SARIMA com parâmetros significativos . . . . .	28
4.3	Parâmetros estimados para o modelo SARIMA(1,1,0)(1,0,1) <sub>12</sub> . . . . .	29
4.4	Previsões resultantes do modelo SARIMA com seus erros absolutos, em GW/h . . . . .	31
4.5	Valores do RMSE para os 20 modelos LSTM sem reposição ajustados . . . . .	32
4.6	Previsões resultantes do modelo LSTM sem reposição, com seus erros absolutos, em GW/h . . . . .	32
4.7	Valores do RMSE para os 20 modelos LSTM com reposição ajustados . . . . .	33
4.8	Previsões resultantes do modelo LSTM com reposição, com seus erros absolutos, em GW/h . . . . .	34
4.9	Valores do RMSE para os modelos selecionados em cada técnica estudada. . . . .	34



# 1. INTRODUÇÃO

Nas últimas décadas, o setor de energia elétrica sofreu grandes transformações em todo o mundo. No sentido de atender à demanda crescente, foi estabelecida uma estrutura de mercado, em que a chamada indústria de energia elétrica tem como principal característica a livre concorrência na compra e venda deste produto indispensável [2]. Com tantas mudanças, foi necessária uma reestruturação completa na cadeia de produção e distribuição de energia.

No Brasil, mais especificamente no Sudeste, existe uma matriz de produção diversificada, com fontes térmicas, hidrelétricas e nucleares [3]. A infraestrutura dessa região permite que ela disponha da maior e melhor rede de transportes e a mais complexa e interconectada rede logística do país. Por ser a principal região consumidora de energia do país, o Sudeste pode mostrar o caminho para prever o futuro do consumo de energia no Brasil, pois a tendência e a sazão verificadas nessa variável tendem a refletir o que acontece no restante do país. Para atender à demanda, foram desenvolvidas na região alternativas diversificadas de geração que resultaram em uma capacidade instalada significativa.

Segundo artigo do BNDES [3] sobre o potencial de produção de energia da região Sudeste e investimentos feitos pelo banco no setor, a região possui um potencial hidráulico de 43,563 GW/h (gigawatts/hora), dos quais 58% (25,267 GW/h) já se encontram em operação. A capacidade térmica já se encontra em 21 GW/h, segundo o Plano Decenal de Expansão de Energia Elétrica 2023, e mais 2 GW/h correspondentes às usinas nucleares de Angra dos Reis (RJ). Tais informações mostram que a região Sudeste, apesar de não ser a maior região do país em área, contém a maior capacidade de geração de energia hidrelétrica, com grandes reservatórios, especialmente nos estados de Minas Gerais e São Paulo. A característica hidrológica da região, popularmente conhecida como ‘caixa d’água do Brasil’, é estratégica para possibilitar o abastecimento de energia elétrica, o que se deve aos muitos rios que nascem nos estados que a compõem e alimentam outras partes do país.

A quantidade de usinas hidrelétricas espalhadas pela região impressiona, e os rios Paranaíba e Grande são os que mais se destacam pela grande concentração de reservatórios, alguns com imensa capacidade de geração de energia, como Itumbiara (2082 MW/h), Marimbondo (1440 MW/h), e Furnas (1216 MW/h) [6]. Entre as fontes renováveis utilizadas para geração termelétrica, vale destacar a biomassa da cana-de-açúcar, que é produzida em mais de 200 usinas [3]. Dentre as fontes não renováveis, destacam-se o gás natural, a geração termonuclear, lembrada pelas usinas nucleares de Angra dos Reis, e os derivados de petróleo. A geração a gás natural utiliza o gás extraído das bacias de Campos e Santos, no litoral da Região Sudeste, e o gás

natural importado por meio do Gasoduto Brasil-Bolívia ou na forma de gás natural liquefeito (GNL), além das grandes usinas termelétricas (UTE) a gás, todas instaladas no Rio de Janeiro.

Ainda segundo o BNDES [3], quase todo o potencial hidrelétrico na região já está sendo explorado, portanto, resta muito pouco para se aproveitar dessa fonte de energia, além da estrutura atual. As perspectivas sobre investimentos se concentram em projetos de geração termelétrica. Para atender ao crescimento da demanda, há uma forte sinalização de entrada de novas usinas térmicas entre os anos de 2019 e 2023 totalizando 7500 MW/h. Vale lembrar que a concretização dessa expansão depende da disponibilidade de combustível, com a necessidade de explorar novas áreas produtoras de gás na região, e da competitividade dos projetos das termelétricas nos futuros leilões para compra de energia nova.

Outra fonte com grande potencial para a geração termelétrica é a biomassa da cana-de-açúcar, e a Região Sudeste é privilegiada nesse sentido, por concentrar a maior parte da produção de açúcar e etanol do país e ter a maior capacidade instalada em cogeração a partir da biomassa da cana [3]. Somado a isso, o potencial remanescente na região ainda é bastante significativo, com possibilidade de aumento na produção a curto e médio prazo.

Por outro lado, também estão previstos investimentos na transmissão de energia, que por sua vez têm como principais objetivos aumentar a capacidade de interação com as demais regiões e receber energia proveniente dos grandes projetos hidrelétricos na Região Amazônica. As principais concessionárias de distribuição da região estão desenvolvendo projetos de redes elétricas inteligentes, as quais otimizam o consumo, reduzem perdas e aumentam a eficiência na geração de energia [3]. O incentivo à geração distribuída e à conservação do estoque propiciada pelas redes inteligentes são fatores fundamentais para economizar energia, algo tão necessário para uma região historicamente deficitária.

Nesse contexto, a análise de séries temporais e as redes neurais artificiais entram como ferramentas para prever o consumo de energia elétrica dos próximos meses, norteados investimentos para o nosso parque de produção de energia. Essa variável é capaz de produzir dados históricos ao longo de um período determinado, em que as observações são dependentes entre si, caracterizando uma série temporal e permitindo o emprego das técnicas utilizadas ao longo desta pesquisa.

Os dados utilizados para o presente estudo foram coletados da base do Instituto de Pesquisa Econômica Aplicada (IPEA) [4], e representam a série temporal mensal do consumo de energia elétrica na região Sudeste, medido em gigawatts/hora (GW/h). A série começa em Janeiro/2002 e termina em Dezembro/2021, totalizando 240 observações.

A intenção é reservar os últimos meses da série para que sirvam de comparativo entre o valor real e o previsto, de modo a atestar a qualidade do ajuste. Serão obtidos modelos autoregressivos, por meio da metodologia de Box-Jenkins, com a rotina a ser construída no software R [19], e modelos originados de Redes Neurais Artificiais (RNA). Medidas de desempenho normalmente usadas no meio de Séries Temporais serão empregadas para embasar a escolha do melhor modelo, como o MAPE e o RMSE, duas métricas de erro que buscam representar a distância entre os valores reais e previstos.

As Redes Neurais Artificiais se apoiam em técnicas de escolha do número de camadas e de neurônios em cada camada, e seus resultados serão avaliados e comparados com o método em Séries Temporais, de modo a concluir, se possível, sobre qual é o melhor método para se modelar esse tipo de informação. Segundo Rodrigues (2021) [7], a ideia principal dessas redes é que elas possam ser alimentadas reversamente com dados relevantes ao modelo. A mecânica consiste em cada neurônio de uma RNA estar conectado aos neurônios das camadas anterior e seguinte. Portanto, este trabalho vai utilizar as Redes Neurais Recorrentes para modelar a série temporal em questão, uma vez que esta é uma variedade de rede neural projetada para reconhecer padrões em dados com características de uma série temporal, ou como foi descrito em [7], “séries numéricas que emanam de sensores, bolsas de valores e agências governamentais”, de modo que o último caso corresponde à abordagem desta pesquisa.

O método descrito até aqui, ou seja, o uso de modelos autorregressivos e posterior comparação dos resultados com aqueles obtidos em um modelo de RNA, já se mostrou bem-sucedido em trabalhos desenvolvidos de maneira similar, como em Campos (2008) [2], que aplicou modelos ARIMA e RNA em dois conjuntos de dados referentes ao consumo de energia elétrica em outras localidades. O estudo não detalhou a construção computacional dos modelos, mas apontou que, apesar de o modelo ARIMA ter um bom ajuste, quando os dados são aplicados a uma Rede Neural Artificial, o ganho de precisão nas previsões é considerável e muito visível graficamente.

Nesse contexto, o presente estudo tem como objetivo modelar e estimar o consumo de energia elétrica da região Sudeste do Brasil, por meio de modelos de séries temporais e redes neurais artificiais, que representem os dados adequadamente. Dentre os modelos ajustados, pretende-se avaliar qual deles apresenta o melhor resultado de previsão.



## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1 SÉRIES TEMPORAIS

Uma série temporal é definida na estatística como um conjunto de observações de uma determinada variável ao longo do tempo, cuja representação é  $Z(t)$  [5]. Quando as observações são obtidas continuamente, isto é, a todo instante ao longo do tempo, diz-se que a série temporal é contínua. Por outro lado, uma série temporal discreta é aquela em que as observações são tomadas pontualmente, em intervalos fixos de tempo, normalmente iguais,  $\Delta t$ . O que faz da análise de séries temporais um método especial com relação a outras técnicas estatísticas é que as observações devem ser dependentes umas das outras, portanto é essencial que a ordem cronológica seja respeitada na apresentação e na análise dos dados [1].

#### 2.1.1 ESTACIONARIEDADE

Uma das suposições mais frequentes que se faz a respeito de uma série temporal é se esta é *estacionária*. Isso acontece quando ela se desenvolve ao longo do tempo  $t$  ao redor de uma média constante  $c$  [5] e a covariância entre dois pontos depende da distância entre eles, independente do tempo  $t$  [1], refletindo alguma forma de equilíbrio estável. Porém, na vida real, a maior parte das séries temporais apresenta alguma forma de não-estacionariedade, em geral *tendências*. O caso mais simples de tendência é aquele em que a série varia ao redor de uma reta, a qual deve ter algum grau de inclinação.

Uma série temporal pode ser estacionária durante um longo período, ou apenas em períodos específicos, podendo mudar seu nível e/ou inclinação. Caso a série não seja estacionária, provavelmente será necessário utilizar uma *série diferenciada* para a construção do modelo preditivo, ou seja, uma série formada pela diferença entre as observações, uma vez que a ordem dos dados não pode sofrer alterações. O objetivo é que a série temporal seja manipulada até que ela se torne estacionária [1]. Isso se deve ao fato de que a maioria dos procedimentos de análise de séries temporais supõe que estas sejam estacionárias [5]. A série diferenciada pode possuir variados graus de diferenciação, sendo a primeira diferença de  $Z(t)$  dada por

$$\Delta Z(t) = Z(t) - Z(t - 1), \quad (2.1)$$

e a segunda diferença é

$$\Delta^2 Z(t) = \Delta[\Delta Z(t)] = \Delta[Z(t) - Z(t-1)], \quad (2.2)$$

ou seja,

$$\Delta^2 Z(t) = Z(t) - 2Z(t-1) + Z(t-2). \quad (2.3)$$

De modo geral, a  $n$ -ésima diferença de  $Z(t)$  é

$$\Delta^n Z(t) = \Delta[\Delta^{n-1} Z(t)]. \quad (2.4)$$

Em situações usuais, tomar uma ou duas diferenças já será suficiente para que a série em questão se torne estacionária.

### 2.1.2 PROCESSOS ESTOCÁSTICOS

Seja  $T$  um conjunto arbitrário. De acordo com Morettin e Tolo (2006) [5], um *processo estocástico* é uma família de variáveis aleatórias  $Z = \{Z(t), t \in T\}$ , definidas em um mesmo espaço de probabilidade  $(\Omega, \mathcal{A}, \mathcal{P})$ , tal que, para cada  $t \in T$ ,  $Z(t)$  é uma variável aleatória. O conjunto  $T$  é tomado geralmente como o conjunto dos números inteiros, ou dos números reais. Como  $Z(t)$  é definida sobre  $\Omega$ , então ela é uma função de dois argumentos,  $Z(t, \omega)$ , com  $t \in T$  e  $\omega \in \Omega$ , de modo que, para cada  $\omega \in \Omega$  fixado, obtém-se uma função de  $t$ . Essa função é definida como uma *realização* do processo estocástico, em outras palavras, é uma *série temporal*.

Em ocasiões nas quais se pretende utilizar modelos para descrever séries temporais, é necessário especificar melhor os processos estocásticos, introduzindo suposições que conduzam a determinadas classes. São elas:

- Processos estacionários ou não-estacionários, de acordo com a dependência ou não, relativamente à origem dos tempos;
- Processos normais ou não-normais, dependendo das distribuições de probabilidades que caracterizam os processos;
- Processos Markovianos ou não-Markovianos, de acordo com a independência de um valor do processo em relação a seus valores precedentes.

O foco deste estudo está somente na classe de processos *estacionários*. Intuitivamente, um processo  $Z$  é dito estacionário caso ele se desenvolva ao longo do tempo de modo que a escolha de uma origem dos tempos não é importante [5]. Isso significa que as características de  $Z(t+\tau)$ , para todo  $\tau$ , são as mesmas de  $Z(t)$ .

### 2.1.3 TENDÊNCIA E SAZONALIDADE

Considerando as observações  $Z_t, t = 1, \dots, N$ , que compõem uma série temporal, pode-se dizer, conforme Morettin e Tolo (2006) [5], que o modelo de decomposição de uma série consiste em escrever  $Z_t$  como soma de três componentes não-observáveis,

$$Z_t = T_t + S_t + \epsilon_t, \quad (2.5)$$

em que  $T_t$  e  $S_t$  representam, respectivamente, a tendência e a sazonalidade, enquanto  $\epsilon_t$  é uma componente aleatória, com média zero e variância  $\sigma_\epsilon^2$ . Portanto, segue-se que  $Z_t$  será em geral uma série não-estacionária [5].

As componentes  $T_t$  e  $S_t$  costumam ter forte relação, e a influência de  $T_t$  sobre  $S_t$  pode ser muito forte, tanto pelo fato de os métodos de estimação da sazonalidade serem muito afetados se a tendência não for levada em consideração, quanto pela especificação de  $S_t$  depender da especificação de  $T_t$  [5].

## TENDÊNCIA

Conforme define Neto (2009) [9], o conceito de tendência pode ser pensado como uma mudança de longo prazo no nível médio da série. A dificuldade seria perceber o que é o “longo prazo”, o que costuma depender principalmente da variável em estudo. Ainda segundo Morettin e Toloí (2006) [5], existem alguns métodos para estimar a tendência, primeiramente considerando a inexistência de sazonalidade:

- Ajuste de uma função do tempo, como um polinômio, uma exponencial ou qualquer outra função suave de  $t$ ;
- Suavização dos valores da série em torno de um determinado ponto, para estimar a tendência naquele ponto;
- Suavização dos valores da série por meio de sucessivos ajustes de retas de mínimos quadrados ponderados.

De acordo com Andrade (2020) [8], para atestar a tendência de uma série temporal, a análise gráfica pode ser um bom caminho, ou ponto inicial. Essa interpretação mais subjetiva pode ser complementada com testes de hipóteses, que podem ser feitos depois da estimação de  $T_t$  ou, de preferência, antes. Na presente pesquisa, será utilizado o teste de Cox-Stuart, citado por Morettin e Toloí (2006) [5].

O teste de Cox-Stuart também é conhecido na literatura como “teste do sinal”, e consiste em agrupar as observações em pares  $(Z_1, Z_{1+c}), (Z_2, Z_{2+c}), \dots, (Z_{N-c}, Z_N)$ , onde  $c = \frac{N}{2}$ , caso  $N$  seja par, ou  $c = \frac{N+1}{2}$ , caso  $N$  seja ímpar. Sendo assim, para cada um desses pares, associa-se o sinal  $+$  se  $Z_i < Z_{i+c}$ , e o sinal  $-$  caso contrário, eliminando os empates [5]. Desse modo, considerando como  $n$  o número de pares nos quais  $Z_i \neq Z_{i+c}$ , o intuito do teste do sinal é testar as hipóteses  $H_0 : P(Z_i < Z_{i+c}) = P(Z_i > Z_{i+c}), \forall i$ : não existe tendência; *versus*  $H_1 : P(Z_i < Z_{i+c}) \neq P(Z_i > Z_{i+c}), \forall i$ : existe tendência.

A estatística do teste pode ser chamada de  $T_2$ , e é igual ao número de pares com sinal positivo. Em situações nas quais  $n \leq 20$ , a regra de decisão é baseada na distribuição Binomial, caso contrário, pode ser usada a distribuição Normal [5]. Considerando  $T_{tab}$  como o valor

tabelado para a distribuição em questão, e que está associado à significância  $\alpha$  aplicada no teste, deve-se concluir que, para  $T_2 > n - T_{tab}$ , a hipótese nula será rejeitada [7].

## SAZONALIDADE

Uma série temporal  $Z_t$  possui um termo referente à componente sazonal  $S_t$ , como podemos ver no modelo (2.5). Essa componente normalmente reflete um comportamento nos dados que ocorre de maneira periódica. Alguns exemplos são o regime de chuvas em uma determinada localidade, cuja curva se repete anualmente; a movimentação em um supermercado, que possui sazonalidade em vários níveis (semanal, mensal e anual); e o consumo de energia elétrica, o qual é objeto de estudo deste trabalho. Sendo assim, um procedimento de ajuste sazonal consiste em:

- Estimar a componente  $S_t$ , por meio de estimativas  $\hat{S}_t$ ;
- Calcular  $Z_t^{SA} = Z_t - \hat{S}_t$  para modelos aditivos, ou  $Z_t^{SA} = \frac{Z_t}{\hat{S}_t}$  no caso de modelos multiplicativos.

Segundo descrevem Morettin e Toloi (2006) [5], existem dois tipos principais de sazonalidade, que podem ser identificados em uma série temporal: a determinística e a estocástica. A sazonalidade determinística ocorre se o padrão sazonal for regular e estável, e tal comportamento pode ser previsto a partir de dados anteriores. Justamente por este fator, os métodos de regressão são ótimos para esse tipo de série. Por outro lado, quando a componente sazonal varia com o tempo, trata-se de uma sazonalidade estocástica, sendo que o método adequado para estimação de  $\hat{S}_t$  é o de médias móveis [8].

Para as sazonalidades determinísticas, existem alguns testes de hipóteses que podem ser utilizados para verificar essa condição. Entre os testes paramétricos, o mais conhecido e utilizado é o teste de Fisher, que é rotineiro a uma análise de variância [2]. O modelo testado por ele é

$$Y_{ij} = S_j + \epsilon_{ij}, i = 1, \dots, n_j, j = 1, \dots, k, \quad (2.6)$$

e supõe  $\epsilon_{ij} \sim N(0, \sigma^2)$ , independentes. Sob a hipótese nula  $H_0 : S_1 = \dots = S_k$ , a estatística

$$T_3 = \frac{N - k}{k - 1} \frac{\sum_{j=1}^k n_j (\bar{Y}_{.j} - \bar{Y})^2}{\sum_{j=1}^k n_j \sum_{i=1}^{n_j} (Y_{ij} - \bar{Y}_{.j})^2} \quad (2.7)$$

tem distribuição  $F(k - 1, N - k)$ . Uma premissa decisiva desse teste é a suposição acerca dos resíduos; estes precisam ter sua normalidade atestada. Tudo isso evidencia a necessidade de se ter bastante cautela ao utilizar tais testes, devido à possibilidade de as suposições necessárias não se concretizarem a priori.

### 2.1.4 O MODELO AUTORREGRESSIVO INTEGRADO DE MÉDIAS MÓVEIS (ARIMA)

Um modelo autorregressivo é definido de maneira que os valores da série no tempo  $t$  dependam dos valores passados. Mais especificamente, o modelo autorregressivo de ordem  $p$ , com notação  $AR(p)$ , é dado por

$$Z_t = \sum_{j=1}^p \phi_j Z_{t-j} + \epsilon_t, \quad (2.8)$$

onde a série  $\epsilon_t$  representa o chamado *ruído branco*. Isso acontece quando as variáveis aleatórias  $\epsilon_t$  não estão correlacionadas, ao mesmo tempo em que possuem média igual a zero e uma variância constante [1].

Considerando um processo linear  $Z_t$ , dizemos que este processo é de médias móveis de ordem  $q$ , denotado por  $MA(q)$ , se este satisfaz a forma

$$Z_t = \mu + \sum_{j=0}^q \theta_j \epsilon_{t-j}, \theta_0 \equiv 1, \quad (2.9)$$

onde  $\mu$  é a média do processo, denotada também por  $E(Z_t)$  [5] e  $\epsilon_t$  é ruído branco com média zero. A generalização dos modelos  $AR(p)$  e  $MA(q)$  dá origem ao chamado *modelo autorregressivo de médias móveis*, ou  $ARMA(p, q)$ , que é definido como

$$Z_t = c + \Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \dots + \Phi_p Z_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}. \quad (2.10)$$

Para os modelos  $ARMA$ , as funções de autocorrelação (ACF) e autocorrelação parcial (PACF) decaem até o infinito em vez de apresentarem um “corte” em alguma defasagem particular [1]. Portanto, para um processo do tipo  $ARMA(p, q)$ , a ACF decairá, seja diretamente ou oscilando ao redor de zero, a partir da defasagem  $q$ . Por sua vez, a PACF decairá, seja diretamente ou oscilando ao redor de zero, a partir da defasagem  $p$ .

Sendo  $d$  um número inteiro não negativo, então uma série temporal  $Z_t$  é considerada um processo  $ARIMA(p, d, q)$  ou um processo integrado de ordem  $d$ , utilizado para séries não estacionárias, se

$$\phi(B)(1 - B)^d Z_t = \theta(B)\epsilon_t, \quad (2.11)$$

em que  $\epsilon_t$  é ruído branco com média zero, e  $\phi(B)$  e  $\theta(B)$  são, respectivamente, polinômios de ordem  $p$  e  $q$ ; sendo que  $B$  é o operador de defasagem,  $\phi(B)$  é o operador autorregressivo,  $\theta(B)$  é o operador de médias móveis e a potência  $d$  é a ordem de diferenciação [1]. O detalhamento de  $\phi(B)$  e  $\theta(B)$  é dado a seguir:

- $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$
- $\theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q$ .

De acordo com Morettin e Toloi (2006) [5], o modelo (2.11) supõe que a  $d$ -ésima diferença da série  $Z_t$  pode ser representada por um modelo  $ARMA(p, q)$ , estacionário e invertível. Um

fato interessante é que, para a maioria dos processos, usa-se  $d = 1$  ou  $d = 2$ , que correspondem aos casos mais comuns da vida real.

Supondo agora uma série temporal sazonal não estacionária  $X_t$  observada  $s$  períodos por ano, de maneira que  $s = 4$  para observações trimestrais e  $s = 12$  para observações mensais, tem-se que uma das formas de remover a sazonalidade presente na série e transformá-la em uma série estacionária  $Z_t$ , para que um modelo ARIMA possa ser empregado, é realizar uma diferenciação sazonal, nos termos da diferenciação já vista [1]. Assim,

$$X_t - X_{t-s} = (1 - B^s)X_t = Z_t. \quad (2.12)$$

Contudo, em alguns casos faz-se necessário adicionar uma modelagem de  $Z_t$  determinada por seu padrão sazonal, o qual é aleatório entre os ciclos  $s$ . Se a sazonalidade de  $Z_t$  já tiver sido filtrada, esta pode ser representada por um modelo ARIMA( $p, d, q$ ). Assim, chega-se à classe de modelos *sazonais multiplicativos* ARIMA( $p, d, q$ )x( $P, D, Q$ ) ou SARIMA,

$$\phi(B)\Phi(B^s)(1 - B)^d(1 - B^s)^D Z_t = \theta(B)\Theta(B^s)\epsilon_t, \quad (2.13)$$

em que  $\phi(B)$  e  $\theta(B)$  são os mesmos operadores relatados em (2.11),  $\Phi(B^s)$  é o operador autorregressivo sazonal de ordem  $P$ ,  $\Theta(B^s)$  é o operador de médias móveis sazonal de ordem  $Q$  e  $\epsilon_t$  é ruído branco com média zero [1]. Sendo assim, os operadores sazonais podem ser dados por

- $\Phi(B^s) = 1 - \Phi_1 B^s - \dots - \Phi_P B^{sP}$
- $\Theta(B^s) = 1 - \Theta_1 B^s - \dots - \Theta_Q B^{sQ}$ .

Nessa classe de modelos, são permitidas tanto a diferenciação regular quanto a diferenciação sazonal. Percebe-se que a série diferenciada pode ser representada utilizando tanto componentes autorregressivos de médias móveis regulares quanto sazonais. Normalmente, tem-se que  $D \leq 1$  e  $P, Q \leq 2$ . O processo representado pela equação (2.13) é dito *causal* se, e somente se, as raízes dos polinômios autorregressivos se encontrarem fora do círculo unitário. Esses modelos teóricos podem ser aplicados em várias metodologias de previsão. No caso desta pesquisa, será utilizada a metodologia de Box-Jenkins, um método que emprega a classe de modelos ARIMA em sua concepção, e que, portanto, será necessário e bastante útil para o objetivo principal do estudo.

### 2.1.5 FUNÇÃO DE AUTOCORRELAÇÃO

Cada par de observações  $(X_{t+h}, X_t)$  consecutivas em uma série temporal  $Z$  estacionária possui uma *função de autocorrelação*, que será denotada aqui por ACF, e é definida como:

$$\rho_x(h) = \frac{\gamma_x(h)}{\gamma_x(0)} = \text{cor}(X_{t+h}, X_t). \quad (2.14)$$

A função de autocorrelação fornece uma medida útil do grau de dependência entre os valores de uma série temporal [1]. Pode ser calculada para qualquer conjunto de dados e não fica

restrita a observações de séries temporais estacionárias. Para dados contendo tendência, a ACF exibirá um decaimento lento à medida que o tempo  $t$  aumenta. No caso de dados com um componente periódico determinístico, como a sazonalidade, a ACF exibirá um comportamento similar ao período. Isso mostra a importância de se calcular e analisar o gráfico da função de autocorrelação, a fim de identificar ou confirmar a existência dessas características no conjunto de dados.

Sob o ponto de vista dos modelos autorregressivos, conforme Morettin e Toloí (2006) [5], os processos  $AR(p)$ ,  $MA(q)$  e  $ARMA(p, q)$  apresentam ACF com características especiais, de modo que:

- Um processo  $AR(p)$  tem ACF que decai de acordo com exponenciais e/ou senoides amortecidas, e que é infinita em extensão;
- Um processo  $MA(q)$  tem ACF finita, apresentando um corte após o “lag”  $q$ ;
- Um processo  $ARMA(p, q)$  tem ACF infinita em extensão, que decai de acordo com exponenciais e/ou senoides amortecidas após o “lag”  $q - p$ .

Estas observações devem ser lembradas e consideradas durante o processo de identificação do modelo, de modo a tentar escolher um ou mais modelos que cumpram as características descritas acima e que, portanto, descrevem o processo estocástico [5]. Um outro instrumento para facilitar essa identificação é a *função de autocorrelação parcial*, ou PACF, como será chamada neste trabalho.

Considerando  $\phi_{kj}$  como o  $j$ -ésimo coeficiente de um modelo  $AR(k)$ , de tal modo que  $\phi_{kk}$  seja seu último componente, sabe-se que

$$\phi_{kk} = \frac{|P_k^*|}{|P_k|} \quad (2.15)$$

é a PACF, onde  $P_k$  é a matriz de autocorrelações e  $P_k^*$  é a matriz  $P_k$ , com a última coluna trocada pelo vetor de autocorrelações [5]. Assim como ocorre com as ACF, no caso das PACF existem algumas propriedades definidas:

- Um processo  $AR(p)$  tem  $PACF \neq 0$  para  $k \leq p$  e  $PACF = 0$  para  $k > p$ ;
- Um processo  $MA(q)$  ou  $ARMA(p, q)$  tem PACF que se comporta similarmente à ACF de um processo  $AR(p)$ .

### 2.1.6 MÉTODO DE BOX-JENKINS

Considerando uma série temporal não sazonal e não estacionária  $Z_t$ , que possa ser representada por um modelo de classe  $ARIMA(p, d, q)$ , o objetivo da metodologia de Box-Jenkins é encontrar um modelo linear de classe  $ARIMA$  que possa ter gerado a série em questão, bem como ser utilizado para fornecer valores de previsão para o futuro desta [1]. Caso apresente sazonalidade, a série temporal pode ser representada por um modelo  $SARIMA(p, d, q) \times (P, D, Q)$ ,

conforme a equação (2.13). A estratégia de modelagem começa a ser montada com a identificação do melhor modelo. Esta etapa consiste em selecionar valores para  $p, d, q$  e  $P, D, Q$  (no caso de modelos sazonais), envolvendo subjetividade e julgamento pessoal, portanto, é considerada a mais difícil e delicada, e não há consenso sobre qual a melhor estratégia a ser seguida.

Dentre as várias estratégias existentes, duas se destacam: a análise das funções de autocorrelação e autocorrelação parcial amostrais, e o uso de um critério de seleção de modelos. Para utilizar a primeira estratégia é preciso reconhecer modelos ARIMA por meio das características da ACF e da PACF [1]. Para modelos sazonais, o comportamento delas deve ser analisado também de acordo com a defasagem sazonal. Além de identificar os valores de  $p$  e  $q$ , o grau de diferenciação da série também precisa ser conhecido. Para um modelo não sazonal, um comportamento suave consistente nas autocorrelações amostrais em defasagens altas indica necessidade de diferenciação, ou seja, indica que a série é não estacionária.

A segunda estratégia para identificar os parâmetros é implementar *funções penalizadoras*. As mais utilizadas são o AIC (Akaike Information Criterion) e o BIC (Bayesian Information Criterion). Esses critérios incorporam um termo penalizador, que aumenta à medida que o número de parâmetros ( $p, q$ ) do modelo aumenta [5], enquanto que a variância residual diminui. Dessa forma, serão escolhidos os modelos com o menor número de parâmetros possível. As equações para esses critérios são

$$AIC = -2\log\hat{L} + 2(p + q),$$

$$BIC = -2\log\hat{L} + (p + q)\log T,$$

onde  $\hat{L}$  representa a verossimilhança maximizada e  $T$  é o número de observações [1]. Tais funções devem ser minimizadas para que se identifique os parâmetros ideais de um modelo ARMA( $p, q$ ).

O critério AIC apresenta alguns problemas, na visão proposta por Morettin e Tolo (2006) [5], como a superestimação assintótica da ordem verdadeira do modelo, tendendo a escolher modelos superparametrizados. Já o BIC é um critério conhecido por fornecer estimadores consistentes para os valores verdadeiros de  $p$  e  $q$  à medida que  $T$  tende ao infinito. Por outro lado, o AIC é assintoticamente mais eficiente para modelos puramente autorregressivos, mas que raramente são encontrados. Na prática, a seleção de modelos é feita calculando o valor do critério para todos os modelos ARIMA ou SARIMA associados aos valores de  $p, d, q$  e  $P, D, Q$  [1]. Assim, seleciona-se o modelo que apresentar o menor valor para o critério, e modelos alternativos cuja diferença para o valor mínimo do AIC ou do BIC seja inferior a 2.

### 2.1.7 MÉTODOS DE COMPARAÇÃO DE PREVISÃO

Um dos métodos de escolha do melhor mecanismo de previsão é a comparação dos valores previstos  $\hat{Z}_t$  com os valores observados da série  $Z_t$ , o que aponta a acurácia, ou a capacidade preditiva do modelo aplicado. Os dois métodos que serão utilizados no presente estudo para medição da acurácia levam em consideração os resíduos em seus cálculos [1]. Esses métodos

são a Raiz Quadrática do Erro Médio (RMSE) e o Erro Percentual Médio Absoluto (MAPE). Basicamente, os erros são definidos como a diferença do valor previsto para o valor observado correspondente ( $Z_t - \hat{Z}_t$ ).

O Erro Quadrático Médio de Previsão é a média do quadrado dos valores de cada resíduo. Segundo Bertolo (2013) [10], o EQMP destaca os grandes erros, comparados aos erros de menor magnitude. Por outro lado, o Erro Percentual Médio Absoluto avalia a magnitude do erro com relação à série histórica, e é a mais popular das formas apresentadas para medir a acurácia da previsão. Entretanto, o uso desta fórmula torna-se inviável caso a série temporal contenha valores iguais a zero. Por se tratar de uma medida de erro relativo, ele não depende da escala, e com isso, permite comparar a acurácia da previsão entre séries temporais de proporções diferentes. Para melhor entendimento, o cálculo dos dois métodos aparece a seguir:

$$EQMP = \frac{\sum_{t=1}^n e_t^2}{n}, \quad (2.16)$$

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{e_t}{Z_t} \right|}{n}. \quad (2.17)$$

Não há consenso entre os estatísticos sobre qual método é preferível. Assim, se os erros elevados de previsão forem inaceitáveis, então o uso do EQMP faz-se necessário. E o MAPE é utilizado para comparar a precisão de duas séries temporais diferentes [1]. Dessa forma, tanto o EQMP quanto o MAPE serão utilizados como critério de acurácia para a comparação dos modelos de previsão que serão aplicados na metodologia da pesquisa.

Vale ressaltar que, durante a geração dos resultados, no software R, utiliza-se a Raiz Quadrada do Erro Médio (RMSE), que pode ser definida simplesmente como a raiz quadrada da equação 2.16, e que tem a vantagem de apresentar valores na mesma escala da variável estudada [5], podendo ser interpretada como uma medida do desvio médio entre o valor observado e o previsto.

### 2.1.8 TESTES DE ADEQUAÇÃO DO MODELO

Após a estimação do modelo, é preciso ainda verificar sua adequabilidade sobre os dados. Para isso, podem ser feitos alguns testes de diagnóstico sobre o modelo. Normalmente, estes testes se baseiam nas autocorrelações estimadas dos resíduos [5].

#### TESTE DE AUTOCORRELAÇÃO RESIDUAL

Depois de estimar  $\phi$  e  $\theta$ , o modelo passa a ter estimativas para os resíduos, representadas por  $\hat{\epsilon}_t$ . Caso o modelo seja adequado, essas estimativas deverão estar próximas dos  $\epsilon_t$  e, portanto, deverão ser aproximadamente não-correlacionadas [5]. Denotando por  $\hat{r}_k$  as autocorrelações desses resíduos estimados, espera-se que

$$\hat{r}_k = \frac{\sum_{t=k+1}^n \hat{\epsilon}_t \hat{\epsilon}_{t-k}}{\sum_{t=1}^n \hat{\epsilon}_t^2} \sim \mathcal{N}\left(0, \frac{1}{n}\right). \quad (2.18)$$

O objetivo é comprovar o comportamento de ruído branco por parte dos resíduos, e isso pode ser feito comparando  $\hat{r}_k$  com os limites  $\pm \frac{2}{\sqrt{n}}$ , de modo que as autocorrelações se mantenham dentro destes limites. Vale lembrar que uma sequência de resíduos  $\{\epsilon_t\}$  é denominada *ruído branco* caso a série temporal seja não-correlacionada, e cada valor tenha média zero e variância constante [5].

### TESTE DE BOX-PIERCE

Esse teste não é capaz de detectar quebras específicas no comportamento de ruído branco, mas pode indicar se os resíduos estão muito altos. Sua estatística é dada por  $Q(K)$ , de modo que, caso o modelo seja apropriado, tem-se:

$$Q(K) = n(n+2) \sum_{j=1}^K \frac{\hat{r}_j^2}{(n-j)} \sim \chi_{K-p-q}^2. \quad (2.19)$$

A hipótese nula considera a existência do ruído branco, e será rejeitada se a estatística  $Q(K)$  ultrapassar os limites de significância da distribuição Qui-Quadrado [5].

### TESTE DO PERIODOGRAMA ACUMULADO

Supondo  $\epsilon_t$  como sendo observações de um processo estocástico, e definindo  $p_\epsilon(f)$  como um espectro do processo, tem-se que  $p_\epsilon(f)$  pode ser estimado por

$$I_\epsilon(f_i) = \frac{1}{2\pi n} \left[ \left( \sum_{t=1}^n \epsilon_t \cos\left(\frac{2\pi i}{n} t\right) \right)^2 + \left( \sum_{t=1}^n \epsilon_t \sin\left(\frac{2\pi i}{n} t\right) \right)^2 \right], \quad (2.20)$$

com  $0 < f_i < \frac{1}{2}$ , onde tal estimador é chamado *periodograma*. Segundo Morettin e Toloi (2006) [5], é fácil provar que se  $\epsilon_t$  for ruído branco, então seu espectro é igual a  $2\sigma_\epsilon^2$  no intervalo  $\left[0, \frac{1}{2}\right]$ .

Outra maneira bastante utilizada para tirar essa conclusão é o método gráfico, por meio da plotagem da estimativa do espectro acumulado ( $C(f_j)$ ) *versus* os valores de  $f$ , onde

$$C(f_j) = \frac{\sum_{i=1}^j I_\epsilon(f_i)}{n\hat{\sigma}_\epsilon^2}, \quad (2.21)$$

de modo que o gráfico de  $C(f_j)$  *versus*  $f_j$  deve ter um aspecto parecido ao de uma reta  $y = x$ , para que os erros sejam considerados ruído branco.

## 2.2 REDES NEURAIS ARTIFICIAIS

Redes Neurais Artificiais (RNA) são modelos computacionais cuja função é simular uma estrutura neural natural, como as que ocorrem em seres vivos inteligentes, os quais adquirem conhecimento através de experiências. São constituídas por unidades de processamento, que por sua vez estão conectadas por canais de comunicação associados a pesos sinápticos. A inteligência dessas estruturas se deve à interação entre as unidades de entrada e seus respectivos pesos. Existem diversos modelos que podem ser usados como regra de treinamento, e é através destes modelos que a rede neural aprende e aprimora seu desempenho [11].

As Redes Neurais podem ser usadas para resolver uma grande variedade de problemas, uma vez que são capazes de calcular qualquer função computável [12]. Elas vêm sendo parte importante da Ciência de Dados nas últimas décadas, e têm colaborado para aumentar a acurácia dos modelos de previsão, especialmente nos campos científico e comercial. Basicamente, uma RNA possui uma primeira camada com vários pontos de entrada, algumas camadas ocultas com seus respectivos neurônios, e uma camada de saída, por onde aparecem os resultados, como se pode ver na Figura 2.1.

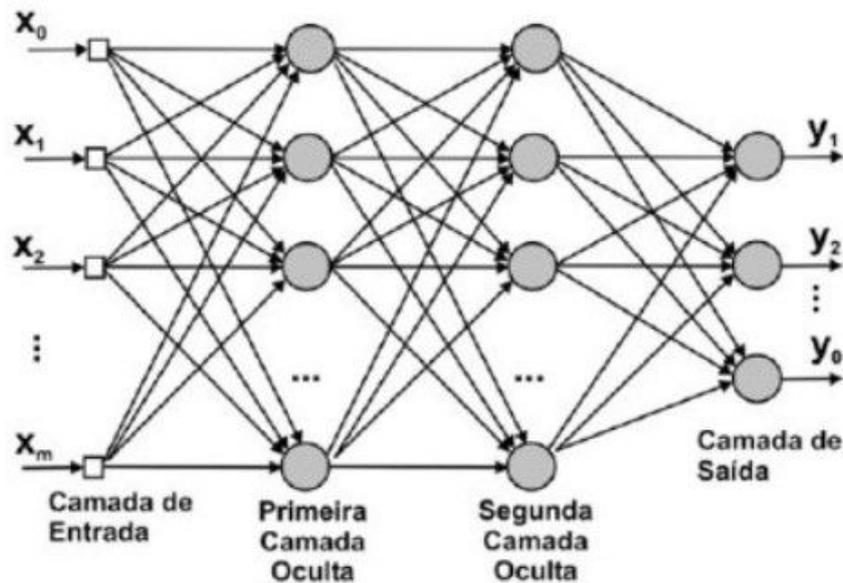


Figura 2.1: Estrutura de uma Rede Neural Artificial (RNA).

Uma Rede Neural Recorrente (RNR) é um tipo de RNA que será aplicada no presente estudo e é projetada para reconhecer padrões em sequências de dados, como texto, genomas, caligrafia, linguagem natural ou dados provenientes de séries temporais [7]. A ideia principal é que uma RNR possa ser alimentada reversamente com dados relevantes ao modelo (neste caso, uma série histórica de consumo de energia elétrica), no intuito de realizar previsões [13]. É possível definir, por exemplo, que o modelo receberá como entrada dados de 12 meses consecutivos de consumo de energia elétrica e retornará o consumo no mês seguinte. Sem treinar o modelo, o resultado será irrelevante, porém, quando o mesmo é treinado por várias vezes, juntamente com o resultado do mês seguinte, espera-se que ele seja calibrado para prever o consumo para

uma determinada entrada.

Esse tipo de funcionamento é baseado no fato de que cada neurônio de uma RNR está conectado a todos os neurônios das camadas imediatamente anterior e seguinte. Em cada neurônio é atribuído um peso para as entradas recebidas, de modo que esses pesos representam o treinamento recebido e portanto constituem a memória da rede. Em outras palavras, cada neurônio reconhece determinado tipo de situação e para essa situação de entrada, existe um determinado neurônio que está mais ativado, portanto, para um problema complexo, são utilizadas grandes quantidades de neurônios e camadas [13]. O papel de cada neurônio é a execução de funções matemáticas e não deve ser associado a uma parcela intuitiva de um reconhecimento feito pelo cérebro humano. A definição de uma RNR pode ser formulada como:

$$h^t = \sigma(W_h X + W_r y^{t-1}) \quad (2.22)$$

$$y = \sigma(W_y h^t), \quad (2.23)$$

onde  $h$  é o chamado *estado oculto*,  $W$  são os pesos,  $t$  representa o tempo,  $r$  é o estado da célula,  $X$  são os dados de entrada,  $y$  os dados de saída e  $\sigma$  é a função sigmoide.

Conforme Haykin (2007) [11], o processo de realimentação proporciona grande influência na rede em termos de treinamento e desempenho. A Figura 2.2 apresenta um exemplo de RNR, com as camadas de entrada e saída, e as camadas ocultas, entre as quais acontece o processo de recorrência.

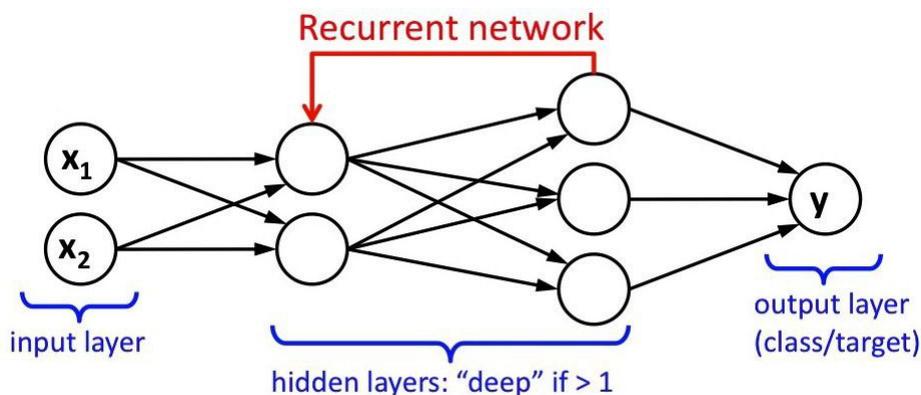


Figura 2.2: Estrutura de uma Rede Neural Recorrente (RNR).

### 2.2.1 REDES NEURAIAS DO TIPO ‘UNIDADES DE MEMÓRIA DE LONGO PRAZO’

Unidades de Memória de Longo Prazo, ou simplesmente LSTM, são um tipo especial de Rede Neural Recorrente capaz de aprender comportamentos a longo prazo [7]. As redes LSTM têm a capacidade de remover ou adicionar informações ao estado da célula, cuidadosamente regulado por estruturas chamadas portões, que nada mais são que uma forma opcional de permitir ou não a passagem de informações. Esses portões são compostos por uma camada de

rede neural sigmóide e uma operação de multiplicação pontual [14]. Existem três tipos [15], sendo eles:

- **Portão de esquecimento:** As informações que não são mais úteis no estado da célula são removidas aqui. As entradas  $x_t$  (entrada no momento específico) e  $h_{t-1}$  (saída de célula anterior) são alimentadas ao portão e multiplicadas por matrizes de peso, seguidas pela adição de um viés. O resultante é passado por uma função de ativação que fornece uma saída binária. Se para um determinado estado de célula a saída for 0, a informação é esquecida e, para a saída 1, a informação é retida para uso futuro.
- **Portão de entrada:** Este portão é responsável pela adição de informações úteis ao estado da célula. Primeiro, a informação é regulada segundo a função sigmóide, que filtra os valores a serem lembrados de forma similar ao portão de esquecimento, usando as entradas  $h_{t-1}$  e  $x_t$ . A partir daí, um vetor é criado usando a função tangente hiperbólica ( $\tanh$ ), que dá saídas entre -1 a 1, e que contém todos os valores possíveis de  $h_{t-1}$  e  $x_t$ . Os valores do vetor e os valores regulados são multiplicados para obter as informações úteis.
- **Portão de saída:** A função de extrair informações úteis do estado da célula atual, para serem apresentadas como uma saída, é do portão de saída. Primeiramente, um vetor é gerado aplicando a função  $\tanh$  na célula. Então, a informação é regulada usando a função sigmóide, que filtra os valores a serem lembrados usando as entradas  $h_{t-1}$  e  $x_t$ . Os valores do vetor e os valores regulados são multiplicados para serem enviados como uma saída e entrada para a próxima célula.

Conforme Dametto (2018) [16], os nós de memória de uma rede neural artificial da categoria LSTM são denominados células, uma vez que esses tipos de estrutura possuem propriedades mais complexas que as RNR tradicionais pelo fato de aprenderem dependências a longo prazo, bem como identificar as informações que devem ser “esquecidas” a partir de um certo ponto. Esquemáticamente, na Figura 2.3 é possível ver a estrutura de uma Rede Neural Artificial LSTM e seus itens constituintes.

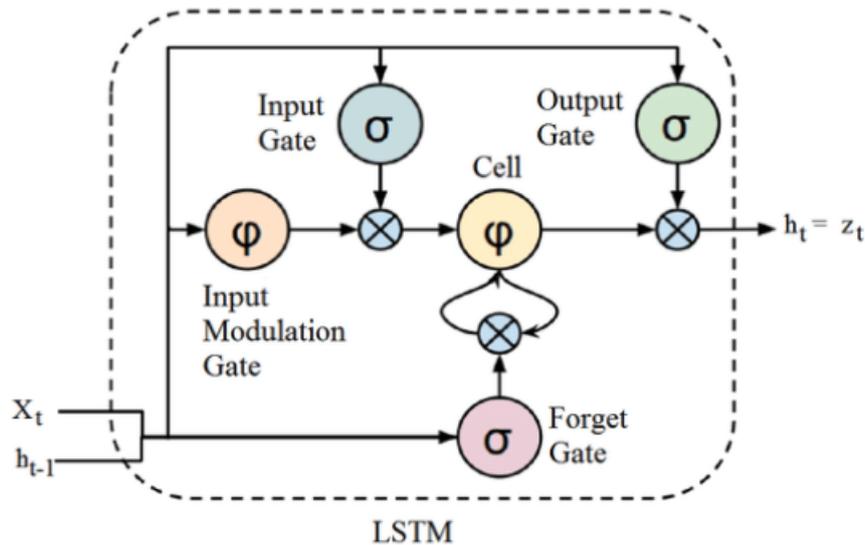


Figura 2.3: Estrutura de uma Unidade de Memória de Longo Prazo (LSTM).

De acordo com Rodrigues (2021) [7], a simulação de uma rede LSTM depende de duas condições: ter dados estacionários, devendo eliminar a tendência ao tomar as diferenças sucessivas, e a função de ativação das unidades da rede deve estar em uma escala de -1 a 1. Além disso, deve estabelecer alguns parâmetros:

- **Amostras de treino:** o número de entradas utilizadas para treinar o modelo.
- **Amostras de teste:** o número de entradas utilizadas para testar o modelo.
- **Épocas:** o conjunto de treino é alimentado à rede tantas vezes quanto o número de épocas.
- **Tamanho batch:** o *batch* é o conjunto de entradas que a rede neural irá processar antes de ajustar os pesos do modelo, e tem grande impacto na velocidade do treinamento.
- **Número de Neurônios:** representa o número de unidades LSTM.
- **Unidades anteriores:** quantas unidades consecutivas compõem uma única entrada. Pode ser entendido como o tamanho da janela que se desloca à medida que os dados são lidos.
- **Horizonte de predição:** até quantas unidades após a entrada serão feitas previsões.
- **Função de Perda:** a principal função de qualquer rede neural é minimizar perdas, uma vez que ela aprende por meio dos próprios erros. Essa é a função utilizada para determinar quão ruim está a resposta da rede durante o treino. Um exemplo seria o Erro Médio Quadrático, já citado na seção de Séries Temporais.
- **Otimizador:** um método numérico usado na otimização, para encontrar um mínimo da função (entrada, peso)  $\rightarrow$  (saída). Normalmente usa-se um esquema iterativo onde em cada

passo se toma a direção negativa do gradiente calculado durante a alimentação reversa da rede.

- **Função de ativação:** A função pela qual passa os valores de entrada de um neurônio submetidos ao peso, cujo resultado representa o estado de ativação do neurônio.



### 3. METODOLOGIA

O conjunto de dados a ser estudado na presente pesquisa é a série temporal mensal do consumo de energia elétrica na região Sudeste, medido em Gigawatts/hora (GW/h). Essa série está disponibilizada no site do Instituto de Pesquisa Econômica Aplicada (IPEA) [4], e para esta análise será considerada a sequência de observações que começa no mês de janeiro de 2002 e termina em dezembro de 2021, totalizando 240 meses, ou 20 anos. Para verificar a adequação dos modelos, as 12 últimas observações foram retidas, de modo que sejam fornecidas previsões para os meses de janeiro até dezembro de 2021, bem como para os meses seguintes. A intenção é comparar, ao fim do processo, os valores realizados com os que foram previstos.

#### MODELAGEM ARIMA

Os dados foram primeiramente importados para o software R, em que foi possível efetuar os cálculos e ajustar modelos de séries temporais. O R é um software livre, com recursos computacionais específicos para tratar de séries temporais univariadas, especialmente o pacote *forecast* [20], que tem suma importância nas etapas de identificação e estimação da modelagem ARIMA. Inicialmente, uma análise descritiva da série temporal foi efetuada para verificação do comportamento geral dos dados, por meio do cálculo de medidas de precisão e dispersão, como a média e o desvio-padrão, por exemplo.

As abordagens para verificação de tendência e sazonalidade nos dados foram bastante analíticas. Primeiramente, foi aplicado o teste de Cox-Stuart, o qual está disponível no pacote *randtests* [21], do R. Caso  $H_0$  seja rejeitada, deve-se concluir que existe tendência, e nesse caso a série original será substituída pela série diferenciada. Este passo deve ser repetido quantas vezes for necessário, com incremento no grau de diferenciação a cada repetição; mas espera-se que este grau de diferenciação não seja maior que 2, chegando a um conjunto de dados sem tendência.

Um detalhe que precisa ser considerado é que, a cada passo, pode-se fazer uma interpretação gráfica, com o intuito de verificar visualmente se a tendência foi retirada com sucesso. Como já foi dito, as funções de autocorrelação e autocorrelação parcial podem ser plotadas, e, se estas convergirem para zero a partir de uma determinada defasagem, significa que não existe mais tendência nos dados.

A partir daí, a série estará autorizada a prosseguir para a verificação da sazonalidade. O teste de Fisher foi o responsável, nesta pesquisa, por avaliar a presença da componente sazonal. Seu comando está contido no pacote *GeneCycle* [22], e sua hipótese nula afirma que não existe

sazonalidade. Portanto, caso esta seja rejeitada, deve-se concluir que a componente sazonal é significativa e, portanto, será necessário ajustar um modelo do tipo SARIMA para os dados.

O próximo passo é colocar em prática a metodologia de Box-Jenkins, começando pelo ajuste dos modelos ARIMA. O pacote *forecast* é muito bem estruturado para esse tipo de procedimento. É o momento de testar diferentes modelos ARIMA, com diferentes combinações de parâmetros, a fim de compará-los e encontrar aquele que se ajusta melhor aos dados. As medidas de adequabilidade podem ser calculadas para cada modelo, e as mais recomendadas são exatamente os critérios penalizadores, de Akaike e de Bayes. Normalmente, o que se espera é encontrar um modelo que contenha os menores valores tanto para o AIC quanto para o BIC, em detrimento dos seus concorrentes. Esse modelo deverá ser reconhecido como o melhor de todos, e prosseguir para a análise dos seus resíduos e posteriormente para o procedimento de predição. O comando *auto.arima*, proveniente do pacote *MLmetrics* [23], pode ser utilizado para ajudar nessa identificação.

A análise dos resíduos deve ser implementada para atestar a normalidade destes. Serão aplicados os testes de autocorrelação residual, o teste do periodograma acumulado e, como último passo antes de iniciar o processo de predição, o teste de Box-Pierce, que serve para verificar se os resíduos são independentes e compõem um ruído branco [17].

Para realizar a predição, o pacote *forecast* dispõe de um comando de previsão em que é inserido o modelo ARIMA e a partir de sua execução são gerados os valores preditivos. O resultado pode ser plotado e exibido, em comparação com os valores realizados.

## MODELAGEM EM REDES NEURAIS

A modelagem e posterior previsão utilizando Redes Neurais do tipo LSTM serão feitas utilizando a linguagem de programação Python [18]. A escolha deste recurso se deve ao fato de ser um software livre e gratuito, cujo uso em análises de dados e aplicação de técnicas estatísticas vem crescendo amplamente, e pela grande quantidade de publicações que fazem uso da ferramenta. O ambiente (IDE) a ser utilizado para processar e executar os códigos Python será o Jupyter Notebook, que funciona de modo que todo o processo é executado em nuvem. Os pacotes utilizados para as análises serão o *Keras*, uma biblioteca em redes neurais, além das bibliotecas *Pandas* para estruturação dos dados, *Matplotlib* para a construção de gráficos, *scikit-learn* para regressão linear e normalização dos dados e *NumPy* para trabalhar com vetores e funções matemáticas.

O primeiro passo será obter a série diferenciada dos dados, com o mesmo grau obtido durante a análise de Séries Temporais feita no R, para eliminar a tendência e tornar os dados estacionários. Depois disso, será feita uma transformação escalar nos dados, de maneira que os mesmos se concentrem entre -1 e 1, por meio do comando *MinMaxScaler*. Em seguida, será construído e ajustado um modelo de Rede Neural do tipo LSTM aos dados de treinamento. A rede vai utilizar uma função de perda de Erro Quadrático Médio e o algoritmo de otimização ADAM, que tem o papel de melhorar funções estocásticas. A partir dos modelos ajustados, é possível gerar as previsões para os últimos 12 meses (janeiro/2021 a dezembro/2021), como já

foi mencionado. Para as análises, serão adotadas duas técnicas diferentes para previsão, antes de ser feita a comparação definitiva, com o modelo ARIMA obtido no R. Essas duas técnicas são definidas da seguinte forma:

- **LSTM sem reposição:** Neste tipo de modelo, a previsão será feita de uma só vez para os 12 meses de 2021, a partir de uma entrada  $t$ . Dentro do código Python, é possível fixar o número de sequências de saída em 12 e o número de testes em 1, sendo que o ponto  $t$  deve ser o último mês anterior ao período separado para previsão (no caso, dezembro/2020). Esse formato se assemelha muito ao modelo ARIMA, pois a previsão é gerada uma única vez, para todos os 12 meses, sem qualquer tipo de iteração.
- **LSTM com reposição:** Aqui a previsão terá uma espécie de “recorrência mensal”, com uma saída para cada entrada, no período estabelecido de 12 meses. Para isso, serão feitos 12 testes, e em cada teste o número de sequências de saída será 1, e, da mesma forma que no modelo anterior, o ponto  $t$  será o mês de dezembro de 2020. Este formato já não se assemelha aos modelos ARIMA, pois este último não trabalha com reposição dos dados.

Depois de feitas as previsões, pode-se reverter as transformações de escala e diferença realizadas nos dados anteriormente e obter o valor da Raiz Quadrática do Erro Médio (RMSE), para avaliar a assertividade das previsões. Quanto às especificações técnicas do modelo, é preciso definir a quantidade de neurônios e de épocas da qual o mesmo irá dispor. Na literatura, até o momento, não foi encontrada uma regra clara, ou um cálculo que norteie essa decisão, que será tomada no decorrer do trabalho. A tendência é que sejam escolhidos alguns modelos, com quantidades diferentes pré-estabelecidas de neurônios e épocas, para proceder com a comparação entre eles.

Ao final de todo o processo, com os modelos ajustados, parte-se para a comparação dos três modelos construídos (ARIMA, LSTM sem reposição e LSTM com reposição), o cálculo da Raiz Quadrática do Erro Médio (RMSE) de cada um deles, e então a escolha sobre qual é o melhor, buscando o menor valor do RMSE.



## 4. RESULTADOS

Na Figura 4.1 é ilustrado o comportamento da série temporal ao longo do tempo, o qual apresenta fortes indícios de tendência e alguns indícios de sazonalidade, de modo a concluir que a série não é estacionária. Esse comportamento sazonal parece ter sofrido uma quebra durante o ano de 2020, mais precisamente entre os meses de Abril e Julho. Uma possível explicação para isso é o início da pandemia de COVID-19. Vale lembrar que o consumo de energia elétrica está sendo medido em GW/h.

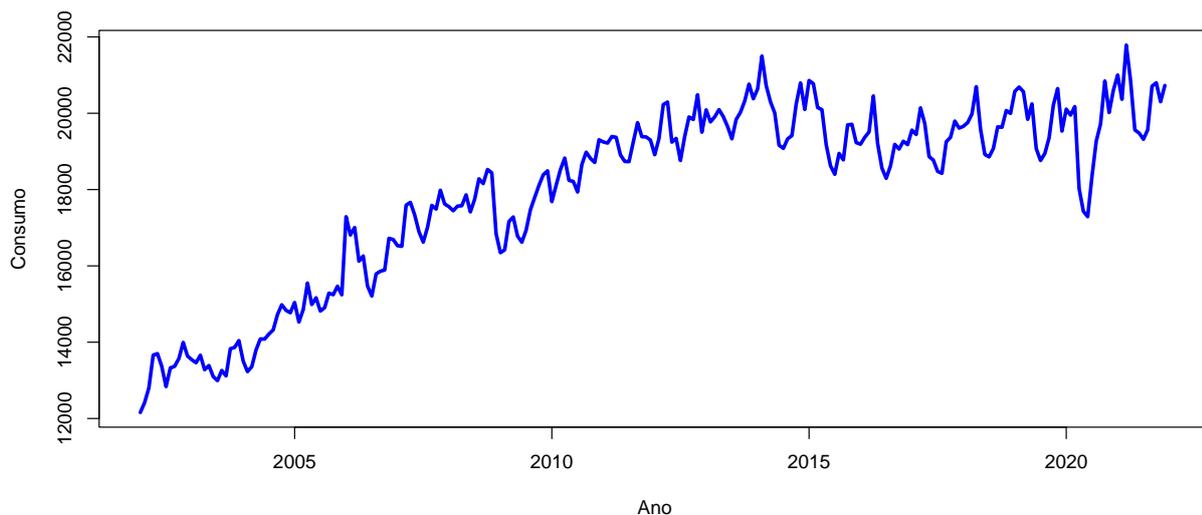


Figura 4.1: Consumo de energia elétrica na região Sudeste de janeiro/2002 a dezembro/2021.

### 4.1 MODELOS AUTORREGRESSIVOS

A metodologia utilizada no R foi a mesma proposta por Box-Jenkins [5], a qual se inicia com a análise gráfica para verificação da estacionariedade. A Função de Autocorrelação (ACF) está ilustrada na Figura 4.2 e permite orientar essa conclusão. Caso o gráfico apresente uma convergência para zero num curto espaço, pode-se concluir que a série é estacionária. Este claramente não parece ser o caso da série estudada.

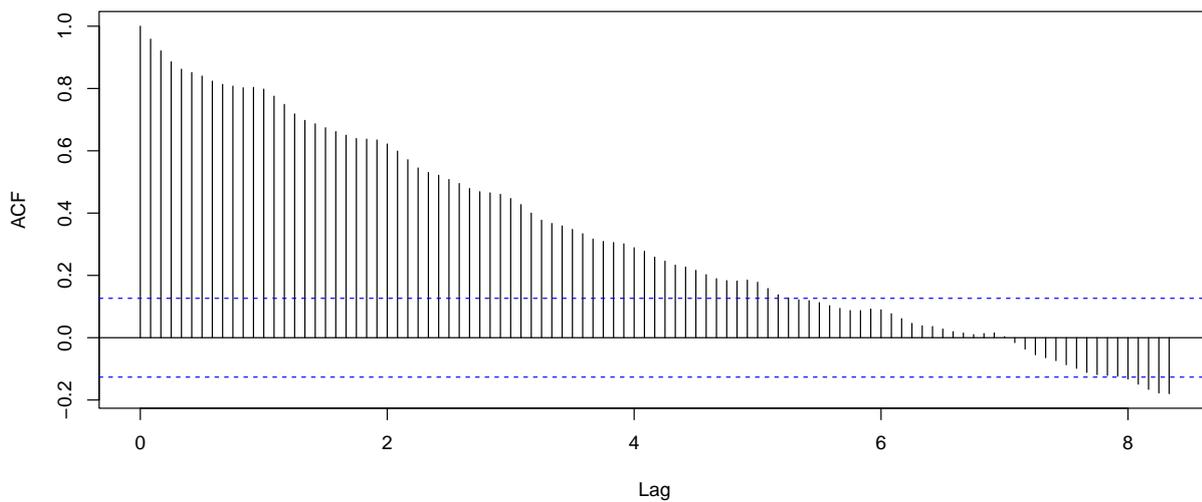


Figura 4.2: Função de Autocorrelação da série temporal (ACF).

Outro recurso utilizado como diagnóstico é o gráfico da média em função da amplitude. As duas medidas são calculadas dividindo a série temporal em pares de observações, de modo que no gráfico apareçam 120 médias e amplitudes. É possível identificar, na Figura 4.3, que não é necessário implementar transformações nos dados, uma vez que não há indícios de que a amplitude dependa da média.

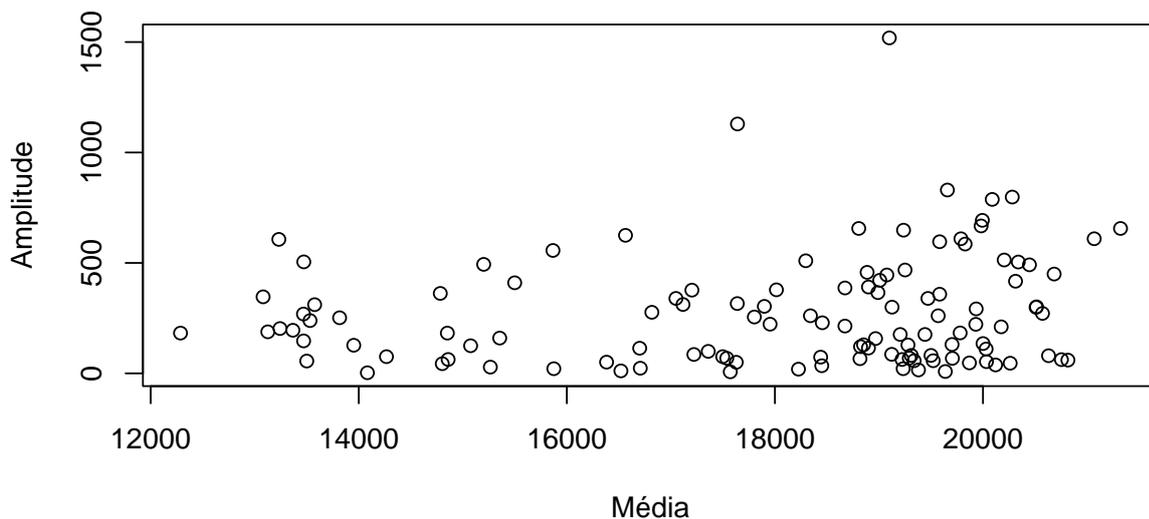


Figura 4.3: Cruzamento das médias com as amplitudes de cada par de observações.

A próxima etapa foi a execução do teste de Cox-Stuart para tendência, com o intuito de concluir analiticamente sobre a existência dessa componente na série temporal em estudo. Vale lembrar que a hipótese nula deste teste supõe que não existe tendência. O resultado obtido foi

um p-valor muito menor que 5%, ou qualquer outra significância que pudesse ser proposta, de modo que a hipótese  $H_0$  foi rejeitada, e concluindo que é preciso seguir para a geração da série diferenciada, repetindo então os passos já realizados.

Foi aplicada na série uma diferenciação em grau 1, tendo suas observações substituídas pelas diferenças entre os dados originais adjacentes. O objetivo aqui é eliminar a componente de tendência, fazendo com que o processo se torne estacionário. A Figura 4.4 contém a série diferenciada, e neste caso, fica nítida a mudança de comportamento, com a média constante e menor variação. Da mesma forma, as funções de autocorrelação (ACF) e autocorrelação parcial (PACF) indicam, devido à convergência dos “lags”, que a tendência foi removida, como se pode ver na Figura 4.5. Ainda com base nesses resultados, percebe-se que as defasagens significativas na ACF seguem um padrão, reforçando a presença de uma componente sazonal, que deverá ser considerada pelo modelo.

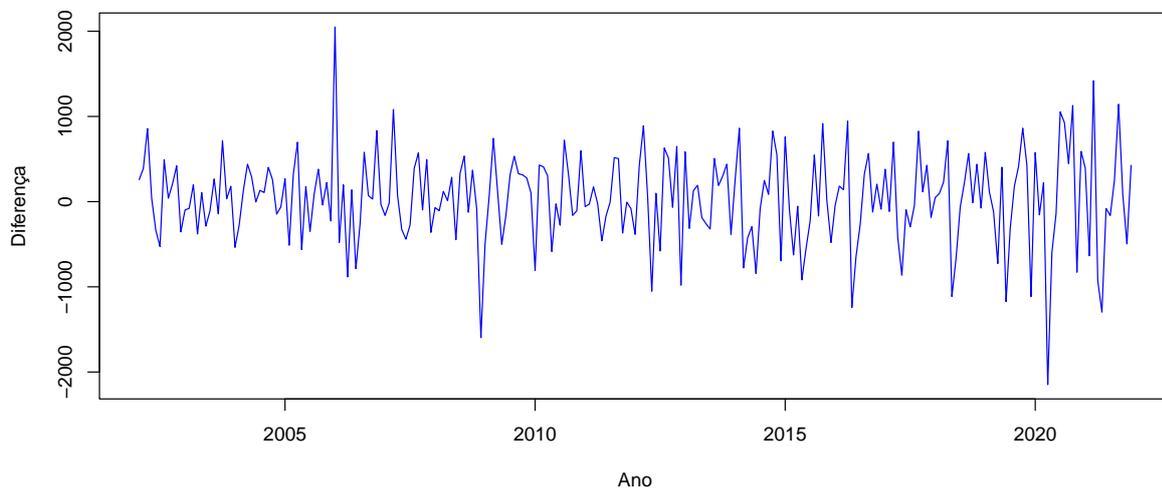


Figura 4.4: Série original diferenciada em grau 1.

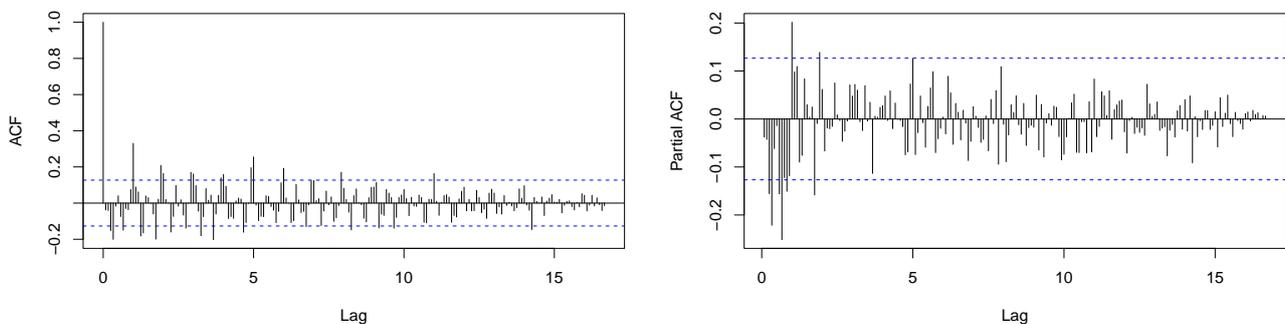


Figura 4.5: Funções de Autocorrelação e Autocorrelação Parcial para a série diferenciada.

Aplicando novamente o teste de Cox-Stuart, finalmente foi obtido o resultado esperado. Com  $p\text{-valor} = 0,4635$ , a hipótese nula não foi rejeitada, e pode-se dizer, analiticamente, que a diferenciação da série temporal eliminou a tendência. Com esta componente já removida dos dados, foi aplicado o teste de Fisher para sazonalidade, o qual indicou a presença da componente sazonal nas observações, por meio de um  $p\text{-valor}$  significativo ( $2 \times 10^{-7}$ ). Esse resultado evidencia a necessidade de se ajustar modelos autorregressivos integrados de médias móveis sazonais (SARIMA).

Finalizadas as definições iniciais do modelo, foi iniciada a busca pelo modelo ideal, por meio da estimação de vários modelos, com diversas combinações de parâmetros. O comando *auto.arima* gerou um modelo automático considerado como ideal, mas que neste estudo foi comparado a outros dez modelos, gerados manualmente pelo comando *arima*. Todos eles, com suas respectivas combinações de parâmetros, estão representados na Tabela 4.1.

Tabela 4.1: Avaliação dos modelos SARIMA propostos

Modelo	Parâmetros	P-valores
Modelo 0 (Automático)	SARIMA(1,1,1)(1,0,0) <sub>[12]</sub>	Significativos
Modelo 1	SARIMA(1,1,1)(1,0,1) <sub>[12]</sub>	Significativos
Modelo 2	SARIMA(1,1,1)(2,0,1) <sub>[12]</sub>	Não significativos
Modelo 3	SARIMA(1,1,0)(1,0,0) <sub>[12]</sub>	Não significativos
Modelo 4	SARIMA(1,1,0)(1,0,1) <sub>[12]</sub>	Significativos
Modelo 5	SARIMA(1,1,0)(1,0,2) <sub>[12]</sub>	Não significativos
Modelo 6	SARIMA(2,1,1)(1,0,0) <sub>[12]</sub>	Não significativos
Modelo 7	SARIMA(0,1,0)(1,0,1) <sub>[12]</sub>	Significativos
Modelo 8	SARIMA(2,1,0)(1,0,1) <sub>[12]</sub>	Não significativos
Modelo 9	SARIMA(2,1,1)(1,0,1) <sub>[12]</sub>	Não significativos
Modelo 10	SARIMA(3,1,0)(1,0,1) <sub>[12]</sub>	Não significativos

A Tabela 4.1 traz a informação sobre a significância dos parâmetros, uma vez que este foi o primeiro critério utilizado para definir os modelos adequados e selecionar os melhores. Dessa forma, atenderam a essa exigência os modelos 0, 1, 4 e 7. Para estes, foram obtidos os valores dos critérios AIC e BIC, e dos métodos residuais MAPE e RMSE. Os resultados estão considerados na Tabela 4.2.

Tabela 4.2: Critérios de seleção dos modelos SARIMA com parâmetros significativos

Modelo	MAPE	RMSE	AIC	BIC
Modelo 0 - SARIMA(1,1,1)(1,0,0) <sub>[12]</sub>	0,02157311	616,6564	3445,65	3462,78
Modelo 1 - SARIMA(1,1,1)(1,0,1) <sub>[12]</sub>	0,0197604	568,0248	3437,44	3454,564
Modelo 4 - SARIMA(1,1,0)(1,0,1) <sub>[12]</sub>	0,01946521	468,1798	3447,52	3461,221
Modelo 7 - SARIMA(0,1,0)(1,0,1) <sub>[12]</sub>	0,02228577	532,8438	3451,1	3461,376

O modelo 4 possui a vantagem de ter os menores valores para o MAPE e para o RMSE, enquanto o modelo 1 dispõe dos menores valores para os critérios de penalização de Akaike (AIC) e de Bayes (BIC). Com o intuito de escolher um desses modelos como o melhor e mais preciso, o presente estudo optou por dar preferência ao modelo com menores erros de previsão. Portanto, optou-se pelo modelo 4, uma vez que este também apresenta uma quantidade menor de parâmetros, cujas estimativas estão detalhadas na Tabela 4.3.

Tabela 4.3: Parâmetros estimados para o modelo SARIMA(1,1,0)(1,0,1)<sub>12</sub>

Modelo	Parâmetro	Estimativa	Desvio-padrão
SARIMA(1,1,0)(1,0,1) <sub>12</sub>	$\phi_1$	-0,161	0,0675
	$\Phi_1$	0,9662	0,0292
	$\Theta_1$	-0,8327	0,077

Para cancelar a adequabilidade do modelo, foi aplicado o teste de autocorrelação residual, que avalia as funções de autocorrelação (ACF) e autocorrelação parcial (PACF) dos resíduos. A análise gráfica aparece na Figura 4.6, e é possível observar que existem no máximo 2 “lags” significativos, indicando que os resíduos não são correlacionados.

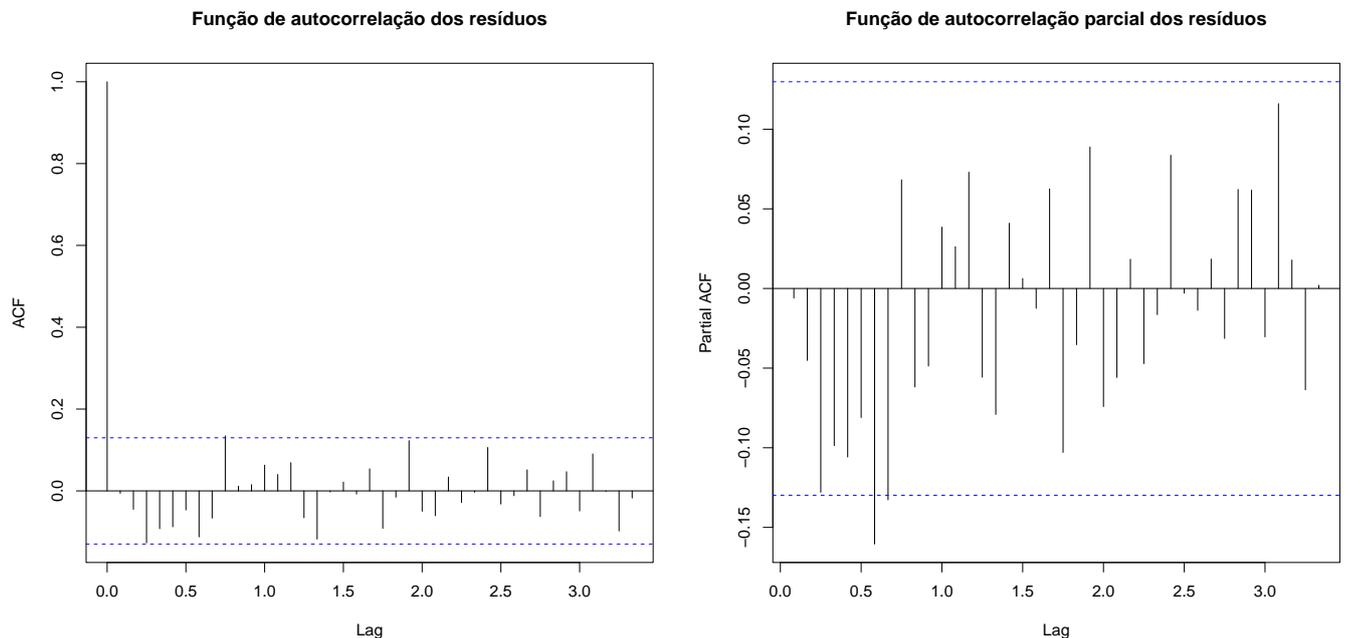


Figura 4.6: Gráficos de ACF e PACF residuais.

O teste de Box-Pierce foi executado logo em seguida, sob a hipótese nula de que os resíduos fossem ruído branco. O p-valor obtido foi de 0,9273, bastante alto e que atestou o comportamento de ruído branco por parte dos resíduos. Essa conclusão foi reforçada com a análise gráfica do periodograma acumulado, exibido na Figura 4.7.

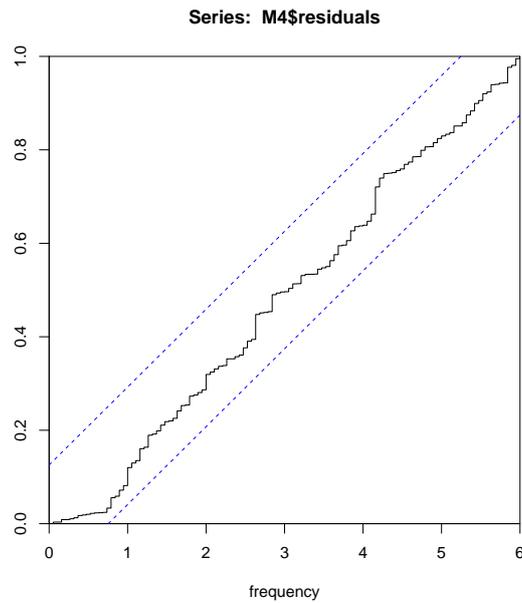


Figura 4.7: Periodograma acumulado.

Finalmente o modelo proposto está validado, e é interessante avaliar graficamente suas previsões, que foram realizadas para o período de Janeiro/2021 a Dezembro/2021, e comparadas com os valores reais na Figura 4.8.

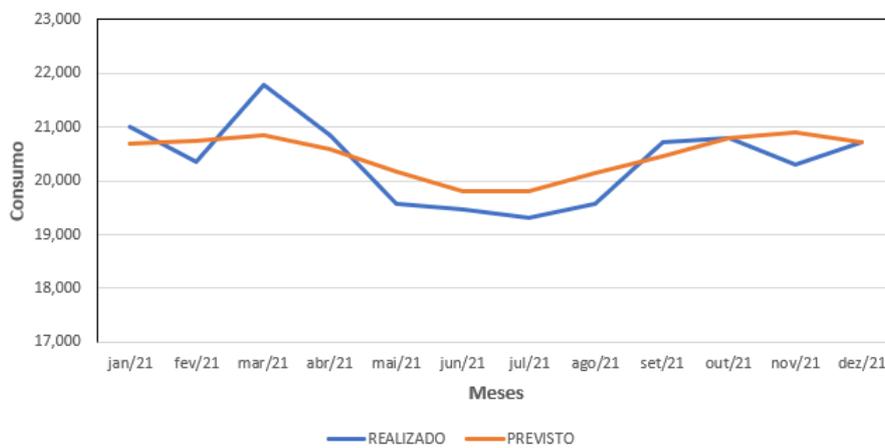


Figura 4.8: Previsões feitas pelo modelo  $SARIMA(1,1,0)(1,0,1)_{[12]}$ .

De modo geral, as previsões se mantiveram próximas dos valores reais, ora um pouco abaixo ora um pouco acima, exceto pelo mês de Março/2021. Por outro lado, o valor observado parece ser anormal para o mês em questão, do ponto de vista da sazonalidade histórica envolvida na variável, o que acaba por dificultar a previsão. O modelo mostrou também uma boa capacidade de se manter próximo aos valores reais, mesmo depois de muitos meses. Os valores de previsão e os erros absolutos estão representados na Tabela 4.4.

Tabela 4.4: Previsões resultantes do modelo SARIMA com seus erros absolutos, em GW/h

Mês	Realizado	Previsto	Erro absoluto
Jan/2021	21,002	20,684	0,318
Fev/2021	20,366	20,743	0,377
Mar/2021	21,786	20,859	0,927
Abr/2021	20,858	20,588	0,270
Mai/2021	19,562	20,166	0,604
Jun/2021	19,481	19,798	0,377
Jul/2021	19,318	19,803	0,485
Ago/2021	19,567	20,133	0,466
Set/2021	20,710	20,449	0,261
Out/2021	20,798	20,798	0
Nov/2021	20,302	20,906	0,604
Dez/2021	20,728	20,710	0,017

## 4.2 MODELOS LSTM

O procedimento de modelagem em Redes Neurais do tipo LSTM ocorreu todo em Python, e foram ajustados ao todo 20 modelos, com diferentes combinações de número de neurônios e de épocas. O primeiro passo foi obter a série diferenciada, para retirar a tendência dos dados, e torná-los estacionários. Usando o comando *MinMaxScaler*, do pacote *sklearn*, os dados foram padronizados para se situarem entre -1 e 1.

Logo depois, os modelos LSTM começaram a ser ajustados aos dados, utilizando o algoritmo de otimização ADAM e, como métrica de perda, a Raiz Quadrada do Erro Médio (RMSE). Foram aplicadas as combinações com 15, 50, 100, 250 e 400 neurônios; e com 200, 500, 1000 e 2000 épocas; obtendo-se o valor do RMSE para todos esses modelos, de modo que esse fosse o critério para a escolha do melhor modelo.

Os modelos de rede foram ajustados com lotes de tamanho 1, com o objetivo de prever o consumo de energia elétrica mensal na região Sudeste entre os meses de Janeiro/2021 a Dezembro/2021. Todo o procedimento que envolve o modelo foi feito usando os dados normalizados, de modo que, somente após geradas as previsões, foi executado um processo de inversão; primeiro das transformações escalares, seguido da inversão das diferenças, antes que fosse calculado o RMSE.

Para os modelos sem reposição, a informação sobre essa medida de perda está contida na Tabela 4.5, e mostra que o modelo escolhido foi aquele com 15 neurônios e 200 épocas.

Tabela 4.5: Valores do RMSE para os 20 modelos LSTM sem reposição ajustados

Neurônios	200 Épocas	500 Épocas	1000 Épocas	2000 Épocas
15	798,309	1018,118	1167,403	3211,928
50	877,824	983,435	1301,831	2543,430
100	892,469	993,581	1325,135	3701,134
250	867,339	1013,066	1438,387	2381,813
400	853,810	1026,142	1529,167	3547,728

Com a escolha do modelo mais adequado, foi possível obter os valores previstos e compará-los com os valores reais. A comparação gráfica entre os dois grupos aparece na Figura 4.9, enquanto os valores de previsão obtidos estão na Tabela 4.6, juntamente com os erros absolutos de previsão.

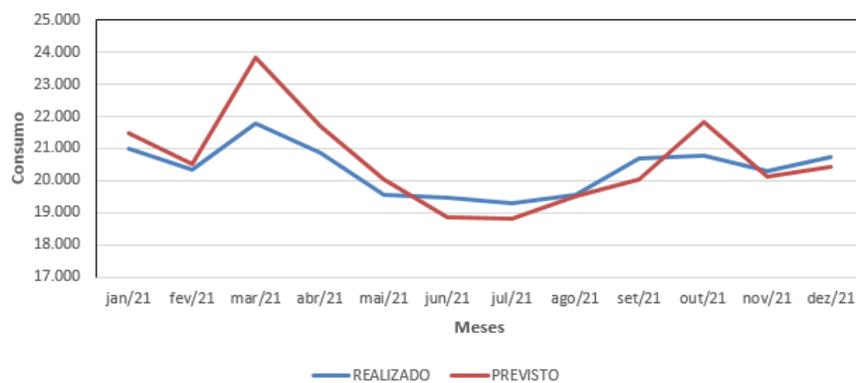


Figura 4.9: Previsões feitas pelo modelo LSTM sem reposição.

Tabela 4.6: Previsões resultantes do modelo LSTM sem reposição, com seus erros absolutos, em GW/h

Mês	Realizado	Previsto	Erro absoluto
Jan/2021	21,002	21,490	0,488
Fev/2021	20,366	20,522	0,156
Mar/2021	21,786	23,828	2,042
Abr/2021	20,858	21,716	0,858
Mai/2021	19,562	20,037	0,475
Jun/2021	19,481	18,866	0,615
Jul/2021	19,318	18,823	0,495
Ago/2021	19,567	19,520	0,047
Set/2021	20,710	20,022	0,688
Out/2021	20,798	21,845	1,047
Nov/2021	20,302	20,124	0,178
Dez/2021	20,728	20,443	0,285

É possível perceber que, embora o modelo tenha feito uma boa detecção da sazonalidade, cometeu também alguns “exageros”, principalmente nas previsões para Março/2021 e Outubro/2021. Essa superestimação parece ter ocorrido em todos os modelos LSTM sem reposição, afetando os valores da Raiz do Erro Quadrático Médio. Já foi mencionado que essa medida costuma valorizar muito os grandes erros, portanto um desvio maior do que o normal para um ou dois meses é capaz de prejudicar o modelo, do ponto de vista do critério de perda que foi utilizado.

Para a técnica com reposição, foram ajustados mais 20 modelos, contemplando as mesmas combinações de neurônios e épocas. Os valores calculados do RMSE estão na Tabela 4.7, e, como se pode ver, o menor valor para a função de perda foi obtido no modelo com 250 neurônios e 200 épocas.

Tabela 4.7: Valores do RMSE para os 20 modelos LSTM com reposição ajustados

Neurônios	200 Épocas	500 Épocas	1000 Épocas	2000 Épocas
15	357,300	403,397	646,828	544,408
50	319,435	620,712	739,845	1020,525
100	311,338	451,792	551,838	624,784
250	293,471	501,305	919,406	1347,902
400	304,003	457,544	511,364	626,884

A partir desse modelo, foi executado o bloco de código Python responsável por gerar as previsões para os meses de 2021. O gráfico contido na Figura 4.10 faz a comparação entre os valores ocorridos e os previstos, enquanto a Tabela 4.8 contém os valores numéricos, e os erros absolutos de previsão.

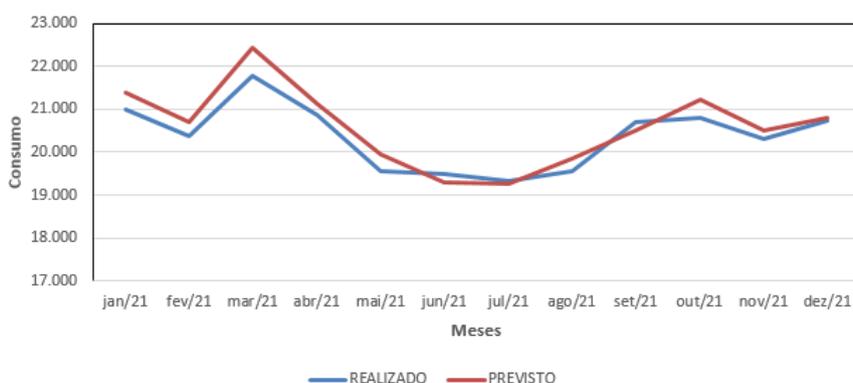


Figura 4.10: Previsões feitas pelo modelo LSTM com reposição.

Tabela 4.8: Previsões resultantes do modelo LSTM com reposição, com seus erros absolutos, em GW/h

<b>Mês</b>	<b>Realizado</b>	<b>Previsto</b>	<b>Erro absoluto</b>
Jan/2021	21,002	21,397	0,395
Fev/2021	20,366	20,686	0,320
Mar/2021	21,786	22,436	0,650
Abr/2021	20,858	21,138	0,280
Mai/2021	19,562	19,963	0,401
Jun/2021	19,481	19,306	0,175
Jul/2021	19,318	19,248	0,070
Ago/2021	19,567	19,841	0,274
Set/2021	20,710	20,498	0,212
Out/2021	20,798	21,223	0,425
Nov/2021	20,302	20,505	0,203
Dez/2021	20,728	20,812	0,084

Analisando o gráfico e os erros absolutos, nota-se que o modelo LSTM com reposição conseguiu seguir com excelência a sazonalidade da série ao longo do ano de 2021, com os erros bastante estáveis, e diminuindo com o passar dos meses. Esse resultado comprova a capacidade dos modelos de Redes Neurais Recorrentes para prever séries temporais a curto prazo.

### 4.3 COMPARAÇÃO ENTRE OS MODELOS SARIMA E LSTM

Partindo do ponto de que os modelos estão ajustados, utilizando as três técnicas escolhidas para este trabalho, estabeleceu-se o RMSE como critério para avaliar qual deles apresentou a melhor performance. Os ajustes mais adequados em cada situação já vinham sendo escolhidos com base nos menores valores para a Raiz do Erro Quadrático Médio, e na Tabela 4.9 está registrado o RMSE para cada um dos três modelos selecionados anteriormente.

Tabela 4.9: Valores do RMSE para os modelos selecionados em cada técnica estudada.

<b>Modelo</b>	<b>RMSE</b>
Modelo 4 - SARIMA(1,1,0)(1,0,1) <sub>[12]</sub>	468,180
LSTM sem reposição - 15 neurônios e 200 épocas	798,309
LSTM com reposição - 250 neurônios e 200 épocas	293,471

Para dar uma dimensão melhor sobre o desempenho dos modelos, foi gerada também uma visão gráfica das estimativas de todos eles, juntamente com os valores reais do consumo de

energia elétrica na região Sudeste, ocorridos ao longo dos meses de 2021, e representados na Figura 4.11.

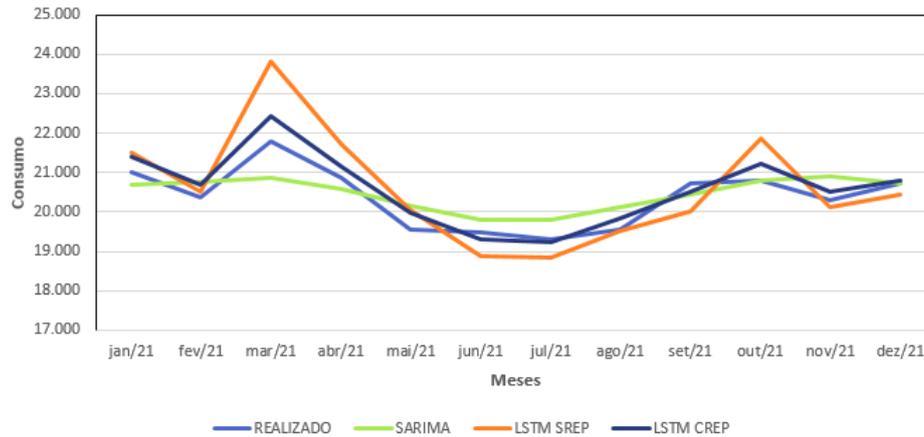


Figura 4.11: Série temporal original comparada com as previsões dos modelos ajustados.

Considerando como critério de avaliação o RMSE, tem-se que o modelo mais preciso é o LSTM com reposição, composto por 250 neurônios e 200 épocas. Ao mesmo tempo, é preciso considerar que modelos LSTM com reposição atuam de maneira distinta dos outros dois tipos. Os modelos SARIMA e LSTM sem reposição calculam as previsões para os próximos 12 meses de uma só vez, sem que o aprendizado se repita. Sendo assim, comparando o modelo SARIMA(1,1,0)(1,0,1) com o modelo LSTM sem reposição com 15 neurônios e 200 épocas, vê-se que a metodologia de Box-Jenkins [5] trouxe um resultado mais próximo da realidade, mesmo que ainda não seja o ideal. Resultados parecidos foram obtidos por Campos (2008) [2], reforçando a vocação das Redes Neurais Artificiais para previsões de curto prazo.



## 5. CONCLUSÕES

O modelo obtido pela técnica de Box-Jenkins, que envolve métodos autorregressivos sazonais de médias móveis, e é do tipo SARIMA(1,1,0)(1,0,1)<sub>12</sub>, se mostrou útil para entender e prever os registros mensais do consumo de energia elétrica na região Sudeste do Brasil. No entanto, a linha de previsão parece suavizar bastante a sazonalidade, de modo a evitar diferenças bruscas de previsão entre um mês e outro.

Entre os modelos que utilizam Redes Neurais Recorrentes, foi escolhido um modelo LSTM sem reposição e outro com reposição. O modelo sem reposição, composto por 250 neurônios e 200 épocas, se adequou bem à sazonalidade, porém elevou muito seus erros em alguns meses do ano, fazendo com que suas previsões, de modo geral, não ficassem tão precisas. Um risco associado a erros como esse é que, supondo que um modelo de previsão seja seguido por empresas concessionárias de energia elétrica, o planejamento de produção e distribuição durante um determinado mês seria comprometido, resultando em excesso de energia estocada, ou a falta dela em alguns lugares.

No caso do modelo LSTM com reposição, com 15 neurônios e 200 épocas, que trabalha com a realimentação das informações, as previsões se tornaram mais certas, como era esperado. O comportamento sazonal foi muito bem interpretado, portanto o modelo se mostrou seguro e fácil de ajustar, sendo plenamente capaz de estimar valores a curto prazo.

Outro ponto a ser considerado é a quantidade de neurônios e épocas utilizadas no modelo. Analisando as medidas de erro referentes aos modelos, no caso o RMSE, foi possível identificar que os mais precisos tendem a ser aqueles com menos épocas. Portanto, em uma análise futura, caso o modelo necessite de modificações, é recomendável reavaliar as combinações de neurônios e épocas, na direção de reduzir a quantidade de épocas empregadas.

Como conclusão, a metodologia de RNR, utilizando modelos LSTM com reposição, se mostrou altamente precisa, especialmente para o curto prazo. Portanto, recomenda-se utilizar essa estrutura para prever o consumo nos primeiros meses de 2022, por exemplo. Por outro lado, se a intenção é implementar uma única previsão de 12 meses, então a comparação deve ser feita entre o modelo LSTM sem reposição e a metodologia de Box-Jenkins, com o uso de modelos do tipo SARIMA. Nesse caso, a última opção foi mais capaz de interpretar a tendência e a sazonalidade da variável em estudo sem correr o risco de se especializar nos dados de entrada e cometer erros bruscos em determinados meses.



# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Escola de Administração Fazendária: *Modelo de Previsão para Arrecadação Tributária*. VI Prêmio Tesouro Nacional, 2001.
- [2] Campos, R. J.: *Previsão de séries temporais com aplicações a séries de consumo de energia elétrica*. Dissertação de Mestrado, Universidade Federal de Minas Gerais, 2008.
- [3] Siffert Filho, N. F. et. al.: *O BNDES e a questão energética e logística na região Sudeste*, 2015. [https://web.bndes.gov.br/bib/jspui/bitstream/1408/4401/1/0%20BNDES%20e%20a%20quest%C3%A3o%20energ%C3%A9tica%20e%20logistica%20na%20Regi%C3%A3o%20Sudeste\\_5\\_P.pdf](https://web.bndes.gov.br/bib/jspui/bitstream/1408/4401/1/0%20BNDES%20e%20a%20quest%C3%A3o%20energ%C3%A9tica%20e%20logistica%20na%20Regi%C3%A3o%20Sudeste_5_P.pdf), acessado em 27/11/2021.
- [4] Instituto de Pesquisa Econômica e Aplicada: *Consumo de energia elétrica na Região Sudeste (SE): quantidade*, 2022. <http://www.ipeadata.gov.br/Default.aspx>, acessado em 13/02/2022.
- [5] Morettin, P. A. e Tolo, C. M. C.: *Análise de séries temporais*. Blucher, 2<sup>a</sup> ed., 2006.
- [6] Eletrobras: *Eletrobras Furnas: números do sistema*, 2022. <https://www.furnas.com.br/geracao/?culture=pt>, acessado em 07/02/2022.
- [7] Rodrigues, J. N.: *Inflação no Brasil: uma aplicação de Séries Temporais e Redes Neurais Recorrentes*. Monografia, Universidade Federal de Uberlândia, 2021.
- [8] Andrade, C. A.: *Análise de séries temporais aplicada ao programa de coleta seletiva no município de Uberlândia (MG)*. Monografia, Universidade Federal de Uberlândia, 2020.
- [9] Neto, A.: *Análise de Séries Temporais - Notas de Aula*, 2009. [https://docs.ufpr.br/~soniaisoldi/ce001/CE017\\_09\\_08.pdf](https://docs.ufpr.br/~soniaisoldi/ce001/CE017_09_08.pdf), acessado em 27/02/2022.
- [10] Bertolo, L. A.: *Métodos básicos de previsão no Excel*, 2013. <http://www.bertolo.pro.br/MetodosQuantitativos/Simulacao/MetodosBasicosDePrevisaoDeSeriesTemporaisNoExcel.pdf>, acessado em 28/02/2022.
- [11] Haykin, S.: *Redes neurais: princípios e prática*. Bookman Editora, 2<sup>a</sup> ed., 2007.
- [12] Lewis, N. D.: *Neural Networks for Time Series Forecasting with R: An Intuitive Step by Step Blueprint for Beginners*. CreateSpace, 2017.

- [13] Vassalli, L. C.: *Aplicação de redes neurais LSTM para a previsão de curto prazo de vazão do Rio Paraíba do Sul*. Monografia, Universidade Federal de Juiz de Fora, 2018.
- [14] Olah, C.: *Understanding LSTM Networks*, 2015. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, acessado em 28/02/2022.
- [15] Data Science Academy: *Deep Learning Book*, 2021. <https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-long-short-term-memory/>, acessado em 28/02/2022.
- [16] Dametto, R. C.: *Estudo da aplicação de redes neurais artificiais para predição de séries temporais financeiras*. Dissertação de Mestrado, FEB, 2018.
- [17] Minitab.com: *Estatística qui-quadrado de Box-Pierce (Ljung-Box) ARIMA*, 2022. <https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/time-series/how-to/arma/interpret-the-results/all-statistics-and-graphs/modified-box-pierce-ljung-box-chi-square-statistics/>, acessado em 01/03/2022.
- [18] Python: *Python Software Foundation*, 2022. <https://www.python.org/>, acessado em 03/03/2022. <https://pt.overleaf.com/project/625f441e0dc2d70a1d157b95>
- [19] R: *A Language and Environment for Statistical Computing*, 2021. <https://www.R-project.org/>, acessado em 15/03/2022.
- [20] Hyndman, R. J. e Khandakar, Y.: *Forecasting functions for time series and linear models*, 2022. <https://pkg.robjhyndman.com/forecast/>, acessado em 16/05/2022.
- [21] Caeiro, F. e Mateus, A.: *randtests: Testing randomness in R*, 2014. <https://CRAN.R-project.org/package=randtests>, acessado em 18/05/2022.
- [22] Ahdesmaki, M. et. al.: *GeneCycle: Identification of Periodically Expressed Genes*, 2021. <https://CRAN.R-project.org/package=GeneCycle>, acessado em 19/05/2022.
- [23] Yachen, Y.: *MLmetrics: Machine Learning Evaluation Metrics*, 2013. <https://CRAN.R-project.org/package=MLmetrics>, acessado em 19/05/2022.

## A. MODELO SARIMA

Rotina implementada no R para o procedimento de Box-Jenkins, resultando em um modelo do tipo SARIMA:

```
library(readxl)
dados <- read_excel("arquivo.xls", col_types = c("date", "numeric"))
attach(dados)
dados <- ts(Consumo,start=2002,frequency=12)

dados_modelo <- read_excel("Dados sem 2021.xls", col_types = c("date", "numeric"))
dados_novo <- ts(dados_modelo$Consumo,start=2002,frequency=12)

ts.plot(dados,col="blue",ylab="Consumo",xlab="Ano",lwd=3)

# Função de autocorrelação
acf(dados, lag.max = 100,main=)
pacf(dados, lag.max = 40)

# Verificando a necessidade de transformação
# Calculando a média
media<-vector()
j<-1
vet<-dados
for(i in 1:240){
  media[i]<-(vet[j]+vet[j+1])/2
  j<-j+2
}

# Calculando o desvio padrão
desvio <- vector()
z<-1
for(i in 1:240){
  desvio[i]<-sd(c(vet[z],vet[z+1]))
```

```

    z<-z+2
}

plot(media,desvio,ylab="Amplitude",xlab="Média")

### TESTE DE COX STUART PARA TENDÊNCIA
require(randtests)
cox.stuart.test(dados)

# Hipótese nula: Não existe tendência
# Hipótese alternativa: Existe tendência

# Série diferenciada

diffdados=diff(dados)
ts.plot(diffdados,col="blue",ylab="Diferença",xlab="Ano",lwd=1)

# Função de autocorrelação
par(mfrow=c(1,2))
acf(diffdados, lag.max = 200,main=)
pacf(diffdados, lag.max = 200,main=)

require(randtests)
cox.stuart.test(diffdados)
# H0 não tem tendência

#####

### TESTE DE FISHER PARA SAZONALIDADE
library(GeneCycle)
fisher.g.test(diffdados)
diffdados
# Hipótese nula: Não existe sazonalidade
# Hipótese alternativa: Existe sazonalidade

# Ajuste dos modelos

library(forecast)
library(MLmetrics)

```

```

modelo<-auto.arima(dados_novo)
modelo
previsao_modelo <- forecast(modelo, 12, level=c(95))
t1_modelo <- 0.6885/0.0694
t2_modelo <- -0.9194/0.0338
t3_modelo <- 0.376/0.0684
dt(t1_modelo, length(dados_novo)) #pvalor ar1
dt(t2_modelo, length(dados_novo)) #pvalor ma1
dt(t3_modelo, length(dados_novo)) #pvalor sar1
MAPE(previsao_modelo$mean,dados[229:240])
RMSE(previsao_modelo$mean,dados[229:240])

M1=arima(dados_novo, order = c(1, 1, 1),seasonal = list(order=c(1,0,1),period=12))
M1
previsao_M1 <- forecast(M1, 12, level=c(95))
t1_M1 <- 0.6528/0.0815
t2_M1 <- -0.8767/0.0477
t3_M1 <- 0.9534/0.0372
t4_M1 <- -0.8/0.0875
dt(t1_M1, length(dados_novo)) #pvalor ar1
dt(t2_M1, length(dados_novo)) #pvalor ma1
dt(t3_M1, length(dados_novo)) #pvalor sar1
dt(t4_M1, length(dados_novo)) #pvalor sma1
MAPE(previsao_M1$mean,dados[229:240])
RMSE(previsao_M1$mean,dados[229:240])
BIC(M1)

M2=arima(dados_novo, order = c(1, 1, 1),seasonal = list(order=c(2,0,1),period=12))
M2
previsao_M2 <- forecast(M2, 12, level=c(95))
t1_M2 <- 0.6568/0.0809
t2_M2 <- -0.8805/0.0466
t3_M2 <- 1.1037/0.1062
t4_M2 <- -0.1308/0.0907
t5_M2 <- -0.8666/0.0856
dt(t1_M2, length(dados_novo)) #pvalor ar1
dt(t2_M2, length(dados_novo)) #pvalor ma1
dt(t3_M2, length(dados_novo)) #pvalor sar1
dt(t4_M2, length(dados_novo)) #pvalor sar2
dt(t5_M2, length(dados_novo)) #pvalor sma1

```

```
MAPE(previsao_M2$mean,dados[229:240])
RMSE(previsao_M2$mean,dados[229:240])
BIC(M2)
```

```
M3=arima(dados_novo, order = c(1, 1, 0),seasonal = list(order=c(1,0,0),period=12))
M3
previsao_M3 <- forecast(M3, 12, level=c(95))
t1_M3 <- -0.1058/0.0685
t2_M3 <- 0.3656/0.0686
dt(t1_M3, length(dados_novo)) #pvalor ar1
dt(t2_M3, length(dados_novo)) #pvalor sar1
MAPE(previsao_M3$mean,dados[229:240])
RMSE(previsao_M3$mean,dados[229:240])
BIC(M3)
```

```
M4=arima(dados_novo, order = c(1, 1, 0),seasonal = list(order=c(1,0,1),period=12))
M4
previsao_M4 <- forecast(M4, 12, level=c(95))
t1_M4 <- -0.161/0.0675
t2_M4 <- 0.9662/0.0292
t3_M4 <- -0.8327/0.077
dt(t1_M4, length(dados_novo)) #pvalor ar1
dt(t2_M4, length(dados_novo)) #pvalor sar1
dt(t3_M4, length(dados_novo)) #pvalor sma1
MAPE(previsao_M4$mean,dados[229:240])
RMSE(previsao_M4$mean,dados[229:240])
BIC(M4)
```

```
M5=arima(dados_novo, order = c(1, 1, 0),seasonal = list(order=c(1,0,2),period=12))
M5
previsao_M5 <- forecast(M5, 12, level=c(95))
t1_M5 <- -0.1557/0.0679
t2_M5 <- 0.9761/0.0269
t3_M5 <- -0.7602/0.0902
t4_M5 <- -0.1065/0.0799
dt(t1_M5, length(dados_novo)) #pvalor ar1
dt(t2_M5, length(dados_novo)) #pvalor sar1
dt(t3_M5, length(dados_novo)) #pvalor sma1
dt(t4_M5, length(dados_novo)) #pvalor sma2
MAPE(previsao_M5$mean,dados[229:240])
```

```
RMSE(previsao_M5$mean,dados[229:240])
```

```
BIC(M5)
```

```
M6=arima(dados_novo, order = c(2, 1, 1),seasonal = list(order=c(1,0,0),period=12))
```

```
M6
```

```
previsao_M6 <- forecast(M6, 12, level=c(95))
```

```
t1_M6 <- 0.6914/0.0821
```

```
t2_M6 <- -0.0509/0.0731
```

```
t3_M6 <- -0.8766/0.0477
```

```
t4_M6 <- 0.3774/0.0707
```

```
dt(t1_M6, length(dados_novo)) #pvalor ar1
```

```
dt(t2_M6, length(dados_novo)) #pvalor ar2
```

```
dt(t3_M6, length(dados_novo)) #pvalor ma1
```

```
dt(t4_M6, length(dados_novo)) #pvalor sar1
```

```
MAPE(previsao_M6$mean,dados[229:240])
```

```
RMSE(previsao_M6$mean,dados[229:240])
```

```
BIC(M6)
```

```
M7=arima(dados_novo, order = c(0, 1, 0),seasonal = list(order=c(1,0,1),period=12))
```

```
M7
```

```
previsao_M7 <- forecast(M7, 12, level=c(95))
```

```
t1_M7 <- 0.9688/0.0339
```

```
t2_M7 <- -0.8559/0.0854
```

```
dt(t1_M7, length(dados_novo)) #pvalor sar1
```

```
dt(t2_M7, length(dados_novo)) #pvalor sma1
```

```
MAPE(previsao_M7$mean,dados[229:240])
```

```
RMSE(previsao_M7$mean,dados[229:240])
```

```
BIC(M7)
```

```
M8=arima(dados_novo, order = c(2, 1, 0),seasonal = list(order=c(1,0,1),period=12))
```

```
M8
```

```
previsao_M8 <- forecast(M8, 12, level=c(95))
```

```
t1_M8 <- -0.1657/0.0685
```

```
t2_M8 <- -0.0291/0.0678
```

```
t3_M8 <- 0.9655/0.0295
```

```
t4_M8 <- -0.8296/0.0778
```

```
dt(t1_M8, length(dados_novo)) #pvalor ar1
```

```
dt(t2_M8, length(dados_novo)) #pvalor ar2
```

```
dt(t3_M8, length(dados_novo)) #pvalor sar1
```

```
dt(t4_M8, length(dados_novo)) #pvalor sma1
```

```
MAPE(previsao_M8$mean,dados[229:240])
RMSE(previsao_M8$mean,dados[229:240])
BIC(M8)
```

```
M9=arima(dados_novo, order = c(2, 1, 1),seasonal = list(order=c(1,0,1),period=12))
M9
previsao_M9 <- forecast(M9, 12, level=c(95))
t1_M9 <- -0.1551/0.0679
t2_M9 <- 1.0876/0.1015
t3_M9 <- -0.1092/0.0883
t4_M9 <- -0.8803/0.0821
t5_M9 <- -0.8/0.0869
dt(t1_M9, length(dados_novo)) #pvalor ar1
dt(t2_M9, length(dados_novo)) #pvalor ar2
dt(t3_M9, length(dados_novo)) #pvalor ma1
dt(t4_M9, length(dados_novo)) #pvalor sar1
dt(t5_M9, length(dados_novo)) #pvalor sma1
MAPE(previsao_M9$mean,dados[229:240])
RMSE(previsao_M9$mean,dados[229:240])
BIC(M9)
```

```
M10=arima(dados_novo, order = c(3, 1, 0),seasonal = list(order=c(1,0,1),period=12))
M10
previsao_M10 <- forecast(M10, 12, level=c(95))
t1_M10 <- -0.1693/0.0684
t2_M10 <- -0.0456/0.068
t3_M10 <- -0.1261/0.0681
t4_M10 <- 0.962/0.0328
t5_M10 <- -0.8269/0.0812
dt(t1_M10, length(dados_novo)) #pvalor ar1
dt(t2_M10, length(dados_novo)) #pvalor ar2
dt(t3_M10, length(dados_novo)) #pvalor ar3
dt(t4_M10, length(dados_novo)) #pvalor sar1
dt(t5_M10, length(dados_novo)) #pvalor sma1
MAPE(previsao_M10$mean,dados[229:240])
RMSE(previsao_M10$mean,dados[229:240])
BIC(M10)
```

```
par(mfrow=c(1,2))
```

```
# Análise dos resíduos
ts.plot(M4$residuals)
hist(M4$residuals)

require(nortest)
# H0 é normal
lillie.test(M4$residuals)

acf(M4$residuals, lag.max = 40, main="Função de autocorrelação dos resíduos")
pacf(M4$residuals, lag.max = 40, main="Função de autocorrelação parcial dos resíduos")
Box.test(M4$residuals, type=c("Box-Pierce"))
cpgram(M4$residuals) # Periodograma

## Predição
require(forecast)
previsao<-forecast(M4, 12, level=c(95))
plot(previsao_M4, lwd=2, col="black",xlab="Ano", ylab=NA)
previsao
```



## B. MODELO LSTM

Rotina implementada no Jupyter Notebook, em linguagem Python, para a obtenção de modelos de Redes Neurais Recorrentes do tipo LSTM:

```
import pandas as pd
import numpy as np
import keras
import sklearn as sk
import matplotlib.pyplot as plt

from keras.models import Sequential, load_model, Model
from keras.layers import Dense, Dropout, LSTM, Input, Embedding
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from keras.models import Model
from keras.layers import Input, Dense, LSTM, Dropout
from datetime import datetime
import math

base = pd.read_excel('arquivo.xls')
base = base.dropna()

plt.figure(figsize = (15,10))
plt.plot(base['Data'],base['Consumo'], label='Consumo')
plt.legend(loc='best')
plt.show()

def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = pd.DataFrame(data)
    cols, names = list(), list()
```

```

for i in range(n_in, 0, -1):
    cols.append(df.shift(i))
    names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
for i in range(0, n_out):
    cols.append(df.shift(-i))
    if i == 0:
        names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
    else:
        names += [('var%d(t+%d)' % (j+1, i)) for j in range(n_vars)]
agg = pd.concat(cols, axis=1)
agg.columns = names
if dropnan:
    agg.dropna(inplace=True)
return agg

```

```

n_lag = 1
n_sec = 12
raw_values = base['Consumo'].values
raw_values = raw_values.reshape(len(raw_values), 1)
supervised = series_to_supervised(raw_values, n_lag, n_sec)

```

```

n_test = 12
supervised_values = supervised.values
train, test = supervised_values[0:-n_test], supervised_values[-n_test:]

```

```

def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return pd.Series(diff)

```

```

def prepare_data(series, n_test, n_lag, n_seq):
    raw_values = series.values
    raw_values = raw_values.reshape(len(raw_values), 1)
    diff_series = difference(raw_values, 1)
    diff_values = diff_series.values
    diff_values = diff_values.reshape(len(diff_values), 1)
    scaler = MinMaxScaler(feature_range=(-1, 1))
    scaled_values = scaler.fit_transform(diff_values)

```

```

scaled_values = scaled_values.reshape(len(scaled_values), 1)
supervised = series_to_supervised(scaled_values, n_lag, n_seq)
supervised_values = supervised.values
train, test = supervised_values[0:-n_test], supervised_values[-n_test:]
return scaler, train, test

```

```
scaler, train, test = prepare_data(base['Consumo'], n_test, n_lag, n_sec)
```

```
# Transformando em uma matriz 3D
```

```

X_train, y_train = train[:, 0:n_lag], train[:, n_lag:]
X_train = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test, y_test = test[:, 0:n_lag], test[:, n_lag:]

```

```
# Montando a Rede
```

```

model = Sequential()
model.add(LSTM(15, batch_input_shape=(1, X_train.shape[1], X_train.shape[2]), stateful=True))
model.add(Dense(y_train.shape[1]))
model.compile(loss='mean_squared_error', optimizer='adam')
# Realizando treinamento
model.fit(X_train, y_train, epochs=200, batch_size=1, verbose=0, shuffle=True) # Com reposição (shuffle=True); Sem reposição (shuffle=False)
model.reset_states()

```

```
def forecast_lstm(model, X):
```

```

    # Transformando X para o formato 3D [samples, timesteps, features]
    X = X.reshape(1, 1, len(X))
    # Realizando predição
    forecast = model.predict(X, batch_size=1)
    # Convertendo para array
    return [x for x in forecast[0, :]]

```

```
y_pred = list()
```

```
for i in range(len(test)):
```

```

    X_test, y_test = test[i, 0:n_lag], test[i, n_lag:]
    forecast = forecast_lstm(model, X_test)
    y_pred.append(forecast)

```

```
def inverse_transform(series, forecasts, scaler, n_test):
```

```
    inverted = list()
```

```

for i in range(len(forecasts)):
    # Criando um array das predições
    forecast = np.array(forecasts[i])
    forecast = forecast.reshape(1, len(forecast))
    # Invertendo escala
    inv_scale = scaler.inverse_transform(forecast)
    inv_scale = inv_scale[0, :]
    # Invertendo diferença
    index = len(series) - n_test + i - 1
    last_ob = series.values[index]
    inv_diff = inverse_difference(last_ob, inv_scale)
    inverted.append(inv_diff)
return inverted

```

```
y_pred = inverse_transform(base['Consumo'], y_pred, scaler, 12)
```

```
# Avaliando os resultados
```

```

for i in range(n_sec):
    actual = base['Consumo'][227:239]
    predicted = [forecast[i] for forecast in y_pred]
    rmse = math.sqrt(mean_squared_error(actual, predicted))
    print('t+%d RMSE: %f' % ((i+1), rmse)) # O RMSE válido é o último.

```

```
predicted
```

```
# Construção do gráfico usando o matplotlib
```

```

meses = ['Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho', 'Julho', 'Agosto', 'Setembro',
'Outubro', 'Novembro', 'Dezembro']
colors=['orange','blue']

```

```

plt.figure(figsize = (15,5)) # Dimensões
plt.gca().set_prop_cycle(color=colors)
plt.plot(meses,predicted,label='Previsto')
plt.plot(meses,actual,label='Realizado')
plt.title("Previsões do modelo LSTM",fontsize=15)
plt.xlabel("Meses",fontsize=13)
plt.ylabel("Consumo",fontsize=13)
plt.legend(loc='best')
plt.show()

```