



UNIVERSIDADE FEDERAL DO MARANHÃO
Programa de Pós-Graduação em Ciência da Computação

Victor Henrique Bezerra de Lemos

***Estimação de Consumo Elétrico Individual Utilizando Temporal
Convolutional Network***

São Luís
2021

Victor Henrique Bezerra de Lemos

Estimação de Consumo Elétrico Individual Utilizando Temporal Convolutional Network

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA, como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Maranhão - UFMA

Centro de Ciências Exatas e Tecnologia - CCET

Programa de Pós-Graduação em Ciência da Computação - PPGCC

Orientador: Prof. Dr. João Dallyson Sousa de Almeida

Coorientador: Prof. Dr. Anselmo Cardoso de Paiva

São Luís - MA

2021

Victor Henrique Bezerra de Lemos

Estimação de Consumo Elétrico Individual Utilizando Temporal Convolutional Network

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA, como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Trabalho aprovado em São Luís - MA, 28 de Abril de 2020:

**Prof. Dr. João Dallyson Sousa de
Almeida**
Orientador

Prof. Dr. Anselmo Cardoso de Paiva
Co-Orientador

Prof. Dr. Geraldo Braz Junior
(Avaliador Interno)

**Prof. Dr. Ginalber Luiz de Oliveira
Serra**
(Avaliador Externo)

São Luís - MA
2021

À minha família e todos que contribuíram com essa jornada.

Agradecimentos

Agradeço ao meus pais, Maria do Socorro Bezerra Lemos e Cláudio Henrique Sá Freire de Lemos, por terem me dado suporte para seguir estudando e realizar este sonho. E ao meu irmão Adriano, por ser o amigo que sempre esteve presente para escutar os meus problemas.

Agradeço ao meu orientador, João Dallyson Sousa de Almeida, que além dos vários ensinamentos foi sempre muito compreensivo com as minhas dificuldades e ajudou no que fosse possível.

Sou muito grato ao meu coorientador, Anselmo Cardoso de Paiva, a quem tenho extrema admiração e a qual me orgulho de ter sido orientando.

Ao professor Geraldo Braz Junior, pelos anos de trabalho juntos e por todas as orientações.

Ao NCA, laboratório que me deu todo suporte necessário para desenvolver esta pesquisa.

À todos os meus amigos do NCA e PET, vocês fizeram cada dia difícil na universidade um pouco mais fácil.

À minha namorada, Thairine Costa, pelo apoio nos vários momentos de insegurança e aflições que tive durante a pandemia e o mestrado, além de ser minha revisora preferida.

Ao grande amigo Robert Douglas (Roberto) pelas várias vezes que me ajudou.

À companhia elétrica que disponibilizou os dados utilizados nesta pesquisa.

Por fim, agradeço a todas as pessoas que passaram durante essa jornada e ajudaram a moldar a pessoa que sou hoje.

*"A journeyman no longer,
but a Master now."
(Lifestealer, Defence of the Ancients 2)*

Resumo

Todos os anos, as companhias no setor de distribuição de energia elétrica sofrem com perdas devido a problemas na aquisição dos dados de consumo para faturamento. Esses problemas variam desde erros humanos até fraudes realizadas pelos clientes. Com isso, a estimação de consumo mensal de energia é um problema de grande interesse no contexto das concessionárias de energia elétrica, como forma de mitigar problemas de leitura. Uma previsão que minimiza o erro exerce, fator importante na identificação de inconsistências no processo de faturamento mensal. Para isto, companhias elétricas tem investido na utilização da predição para definir limites, entre superior e inferior, onde uma leitura é considerada normal. Assim, neste contexto, este trabalho apresenta um método para predição de consumo elétrico mensal individual. Foi desenvolvido um método baseado em uma rede *Temporal Convolutional Network* (TCN) combinada a aplicação de Otimização dos Hiperparâmetros da arquitetura da rede. Também foi elaborado um fluxo de pré-processamento capaz de amenizar problemas que podem ser encontrados nas séries históricas de consumo elétrico, além de auxiliar a gerar melhor representação da mesma para o modelo preditivo. A abordagem proposta obteve SMAPE total de 16,86%, mostrando-se superior em seis das oito classes de consumidores, que correspondem a 98,23% do dataset, quando comparada a outros métodos encontrados na literatura, como: *Autoregressive Integrated Moving Average* (ARIMA), *Simple Exponential Smoothing* (SES), *HOLT*, *Stochastic Gradient Descent* (SGD), *Long short-term memory* (LSTM) e a própria rede TCN. No geral, se mostrando uma metodologia proposta capaz de performar dentro do esperado, nos mais diferentes cenários.

Palavras-chave: Consumo Elétrico, Séries temporais, Predição, Temporal Convolutional Network.

Abstract

Every year, companies in the electricity distribution sector suffer losses due to problems in acquiring consumption data for billing. These problems range from human error to customer fraud. Thus, estimating monthly energy consumption is a problem of great interest in the context of electricity distribution companies, as a way of mitigating reading problems. A forecast that minimizes error plays an important role in identifying inconsistencies in the monthly billing process. For this, electric companies have invested in the use of prediction to define limits, between upper and lower, where a reading is considered normal. Thus, in this context, this work presents a method for predicting individual monthly electrical consumption. A method based on a *Temporal Convolutional Network* (TCN) network was developed, combined with the application of an Optimization of the Hyperparameters of the proposed architecture. A pre-processing workflow was also created, able to alleviate some of the problems that can be found in the clients' historical series, in addition to helping to generate a better representation of the time series for the predictive model. The proposed approach proposed SMAPE total 16.86 %, being superior in six of the eight consumer classes, which correspond to 98.23 % of the dataset, when compared to other methods found in the literature, such as: *Autoregressive Integrated Moving Average* (ARIMA), *Simple Exponential Smoothing* (SES), *HOLT*, *Stochastic Gradient Descent* (SGD), *Long short term memory* (LSTM) and the TCN network itself. In general, showing a proposed methodology capable of performing as expected, in the most different scenarios.

Keywords: Energy Consumption, Forecasting, Time Series, Temporal Convolutional Network.

Lista de ilustrações

Figura 1 – Exemplo de série temporal e alguns de seus componentes.	18
Figura 2 – Rede Neural Recorrente e seus passos de execução.	20
Figura 3 – Célula LSTM e suas componentes.	21
Figura 4 – Célula GRU e suas componentes.	22
Figura 5 – Empilhamento de camadas de convoluções causais.	23
Figura 6 – Empilhamento de camadas convolucionais casuais dilatadas.	24
Figura 7 – Bloco residual e suas respectivas camadas.	25
Figura 8 – Metodologia proposta para predição.	32
Figura 9 – Fluxo de execução do pré-processamento.	34
Figura 10 – Arquitetura base utilizada para a otimização dos hiperparâmetros e o treinamento da rede.	36
Figura 11 – Progresso do número de novos clientes deixam de ser processados pelos sucessivos cortes.	42
Figura 12 – Histórico de consumo de um cliente, um caso onde o último mês é estimado com sucesso por todos os métodos comparados.	43
Figura 13 – Histórico de consumo de um cliente, um caso onde o último mês gera um alto erro na predição por todos os métodos comparados.	44
Figura 14 – Configurações de arquitetura testadas no experimento.	45
Figura 15 – Histórico de consumo de um cliente, um caso onde o último mês gera um alto erro na predição por todos os métodos comparados.	47
Figura 16 – Histórico de consumo da instalação A após interpolação linear utilizada para preencher os valores nulos.	48
Figura 17 – Histórico de consumo da instalação B que possui um valor que foge abruptamente do padrão de consumo.	49
Figura 18 – Histórico de consumo da instalação B após a remoção do registro de consumo considerado anomalia.	49
Figura 19 – Histórico de consumo da instalação C antes da normalização da série.	50
Figura 20 – Histórico de consumo da instalação C após a normalização da série.	50

Lista de tabelas

Tabela 1 – Distribuição quantitativa, percentual, a média e o desvio padrão de clientes por cada classe.	34
Tabela 2 – Quantidade de instalações na base final após a etapa de pré-processamento para filtrar apenas séries com 30 meses de consumo registrado, média de consumo registrado por classe e o desvio padrão.	35
Tabela 3 – Espaço de busca utilizado para otimização de hiperparâmetros.	37
Tabela 4 – Resultados de predição para 6 meses de consumo.	40
Tabela 5 – Resultados de predição para 12 meses de consumo.	41
Tabela 6 – Resultados de predição para 18 meses de consumo.	41
Tabela 7 – Resultados de predição para 24 meses de consumo.	41
Tabela 8 – Resultados de predição para 30 meses de consumo.	42
Tabela 9 – Comparativo entre o resultado das predições por classe para cada uma das configurações testadas. Em verde, destaca-se o melhor resultado obtido.	46
Tabela 10 – Relação dos valores de hiperparâmetros mais escolhidos pelo otimização e suas respectivas quantidades.	51
Tabela 11 – Resultado da métrica MAE para as predições para cada uma das classes aplicando a metodologia proposta.	52
Tabela 12 – Resultado da métrica MAPE para as predições para cada uma das classes aplicando o método proposto.	53
Tabela 13 – Resultado da métrica SMAPE para as predições para cada uma das classes aplicando a metodologia proposta.	53

Lista de abreviaturas e siglas

ANN - *Artificial Neural Network*

ARIMA - *Autoregressive Rntegrated Moving Average*

CAPES - *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*

EMD - *Empirical Mode Decomposition*

FCN - *Fully Convolutional Network*

GRU - *Gated Recurrent Unit*

LSTM - *Long Short-Term Memory*

MAE - *Mean Absoule Error*

MAPE - *Mean Absoule Percentage Error*

MSE - *Mean Squared Error*

PSO - *Particle Swarm Optimization*

RNN - *Recurrent Neural Network*

RNR - *Redes Neurais Recorrentes*

SES - *Seasonal Exponential Smoothing*

SGSC - *Smart Grid Smart City*

SMAPE - *Symmetric Mean Absoule Percentage Error*

SMBO - *Sequential Model Based Optimisation*

SGD - *Stochastic Gradient Descent*

TCN - *Multi-Layer Perceptron*

TPE - *Tree-structured Parzen Estimator*

UFMA - *Universidade Federal do Maranhão*

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	Objetivos Gerais	15
1.1.2	Objetivos específicos	16
1.2	Contribuições	16
1.3	Organização da Dissertação	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Séries Temporais	18
2.1.1	Predição de Séries Temporais	19
2.2	Redes Neurais Recorrentes	19
2.3	Temporal Convolutional Network (TCN)	23
2.4	Otimização de hiperparâmetros	25
2.4.1	Tree-structured Parzen Estimator	26
2.5	Métricas de Estimação	27
3	TRABALHOS RELACIONADOS	29
4	METODOLOGIA	32
4.1	Aquisição de dados	32
4.2	Pré-processamento	34
4.3	Otimização de Hiperparâmetros	35
4.4	Treinamento e Predição	36
5	EXPERIMENTOS E DISCUSSÃO	39
5.1	Experimento 1: Análise de quantidade de meses	39
5.1.1	Resultados	40
5.2	Experimento 2: Estudo do impacto das camadas densas	44
5.2.1	Resultados	45
5.3	Experimento 3: Aplicando o Método Proposto	46
5.3.1	Resultados	46
5.3.1.1	Pré-processamento	47
5.3.1.2	Otimização de Hiperparâmetros	49
5.3.1.3	Predição	51
6	CONCLUSÃO	54
6.1	Trabalhos Futuros	55

REFERÊNCIAS	56
--------------------------	-----------

1 INTRODUÇÃO

Segundo Pesquisa Nacional por Amostra de Domicílios Contínua de 2019 do Instituto Brasileiro de Geografia e Estatística (IBGE), 99,8% da população brasileira possui acesso à energia elétrica ¹. Isso está em cargo de 105 agentes atuando no mercado de distribuição, sendo eles 52 Concessionárias, 52 Permissionárias e 1 Designada, segundo dados da Agência Nacional de Energia Elétrica (ANEEL)². Esse é um mercado que movimentou cerca de 50,19 bilhões de reais no ano de 2018 ³. Estes valores são fortemente impactados negativamente por erros de leituras e fraudes. Os erros de leitura podem levar à uma fatura com valor inferior ao consumo do cliente ou uma fatura com valor superior ao consumo real que tende à acarretar em processos jurídicos. Em ambos os casos, há custos negativos para a companhia de distribuição de energia. Já segundo levantamento da ANEEL, fraudes de leitura geraram um prejuízo estimado de 4,8 bilhões de reais, em 2018⁴.

O processo de leitura das instalações de consumo é realizado mensalmente pelas companhias de distribuição de energia, valor esse que é registrado em quilowatt-hora (kWh). Na maioria das vezes, o processo de leitura é realizado manualmente por leituristas, o que pode resultar em erros de leitura e gerar inconsistências (QUINTANILHA et al., 2017). Sendo assim, companhias elétricas fornecem meios para que os leituristas, em campo, sejam capazes de identificar leituras que possuem valores considerados fora do normal e até possíveis fraudes, como a adulteração de medidores.

Identificar esses erros é uma tarefa complexa e que tende a trazer resultados pouco consistentes. Isso se dá, visto que, variações na leitura são suscetíveis à diferentes cenários, como: época do ano, clima, localização, economia, entre outros. Contudo, no contexto atual, companhias costumam utilizar a média de consumo de meses anteriores como abordagem de identificação de inconsistências na leitura. Essa estratégia apesar de simples e eficaz em alguns casos, pode resultar em algumas falhas, principalmente por

¹ INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Educa: jovens. **Domicílios brasileiros**. [S.l.]: IBGE, 2019. Disponível em: <https://educa.ibge.gov.br/jovens/conheca-o-brasil/populacao/21130-domicilios-brasileiros.html>. Acesso em: 22 mar. 2021.

² AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Serviço Público de Distribuição de Energia Elétrica**. Brasília: ANEEL, 2021. Disponível em: <https://www.aneel.gov.br/distribuicao2>. Acesso em: 22 mar. 2021.

³ CANAL ENERGIA. **Mercado de energia movimentou R\$ 50,19 bilhões em 2018, diz CCEE**. São Paulo, 2019. Disponível em: <https://canalenergia.com.br/noticias/53090624/mercado-de-energia-movimentou-r-5019-bilhoes-em-2018-diz-ccce>. Acesso em: 22 mar. 2021.

⁴ ASSOCIAÇÃO NACIONAL DOS CONSUMIDORES DE ENERGIA. **‘Gatos’ fazem Brasil perder energia suficiente para abastecer SC por um ano**. São Paulo: ANACE, 2018. Disponível em: <http://www.anacebrasil.org.br/noticias/gatos-fazem-brasil-perder-energia-suficiente-para-abastecer-sc-por-um-ano/>. Acesso em: 23 mar. 2021.

desconsiderar características de tendência e sazonalidade, fundamentais para minimizar erros de predição.

A partir da estimação de consumo, companhias elétricas utilizam estratégias para definir um intervalo de tolerância na qual o valor real de consumo lido em campo pelo leiturista deve estar inserido. A definição de limites para a predição que tem como objetivo obter intervalo máximo e mínimo onde o valor de consumo registrado pelo leiturista é considerado de acordo com o histórico de consumo do cliente. Os registros de consumo que ultrapassam o intervalo esperado são encaminhados para serem validados por analistas da companhia, que verificam se há necessidade de realização de uma nova leitura, uma vez que, pode ter acontecido um erro humano durante a leitura e registro do consumo. Desta forma, fica evidente que falhas na definição do intervalo esperado podem diminuir consideravelmente a taxa de identificação de inconsistências de leitura, além de gerar diversos custos operacionais. Assim, a predição de consumo mensal se torna ainda mais importante e uma estratégia que seja capaz de minimizar os erros de estimação pode reduzir consideravelmente os gastos das companhias elétricas, aumentar a retenção de arrecadação e evitar reclamações dos clientes por via administrativa ou judicial, favorecendo o bom relacionamento com os mesmos.

Neste contexto, este trabalho apresenta uma metodologia para predição mensal e individual de consumo elétrico, com a finalidade de auxiliar o processo de identificação de falhas e erros do registro de leitura. A metodologia proposta utiliza o histórico mensal de clientes com diferentes perfis de consumo. Para isto, utiliza-se uma abordagem com *Temporal Convolutional Network* (TCN) para realizar a predição do consumo esperado para cada cliente. Além disso, foi utilizada a otimização de hiperparâmetros para garantir que o modelo gerado para cada cliente possua a combinação de parâmetros que possibilita maximizar o seu desempenho preditivo. Neste trabalho, também são apresentados experimentos utilizados para definir a estrutura base da rede e o comprimento das séries temporais utilizadas.

1.1 Objetivos

Nesta Seção, são apresentados os objetivos gerais e específicos a serem obtidos com o desenvolvimento do presente trabalho.

1.1.1 Objetivos Gerais

O objetivo principal deste trabalho é desenvolver um método computacional capaz de prever o consumo mensal esperado de energia elétrica de instalações individuais, utilizando uma metodologia baseada em uma rede TCN, no contexto de companhias do ramo

elétrico, visando minimizar a métrica SMAPE para melhorar o processo de identificação de inconsistências de leitura.

1.1.2 Objetivos específicos

Para alcançar o objetivo geral pretendido é necessário desenvolver os seguintes objetivos específicos:

- Adquirir e construir uma base de dados de histórico de consumo elétrico mensal individual;
- Definir uma arquitetura de rede TCN otimizada aplicada à séries de consumo elétrico;
- Desenvolver um metodologia para aplicar uma rede TCN para predição de consumo mensal de energia elétrica;
- Realizar experimentos para avaliar o desempenho do método proposto;
- Avaliar o desempenho da TCN no problema abordado após a otimização de parâmetros;
- Comparar o desempenho do método proposto com outras técnicas de predição em séries temporais;
- Analisar as vantagens e limitações do método proposto.

1.2 Contribuições

Destacam-se como principais contribuições do estudo realizado:

- Utilização de uma rede TCN para estimação de consumo elétrico individual;
- Comparação da TCN com outros métodos de predição em séries temporais considerados na literatura como superiores nesta tarefa;
- Avaliação do desempenho da otimização de hiperparâmetros no contexto de séries temporais de consumo elétrico individual.

1.3 Organização da Dissertação

O trabalho está organizado como segue: O [Capítulo 2](#) aborda os principais conceitos sobre séries temporais e redes neurais, relevantes para a compreensão deste trabalho. Em seguida, o [Capítulo 3](#) apresenta alguns dos principais trabalhos relacionados ao tema

abordado. Nele, cada trabalho é descrito com foco em suas principais características e contribuições. No [Capítulo 4](#) aborda-se a metodologia proposta para prever o consumo mensal de energia elétrica. Também é apresentado o design e desenvolvimento dos experimentos realizados. Já no [Capítulo 5](#) são apresentados os experimentos realizados, bem como os resultados obtidos, seguido de uma discussão sobre os mesmos. Por último, o [Capítulo 6](#) descreve as conclusões obtidas deste trabalho, destacando as contribuições geradas por esta pesquisa e os trabalhos futuros que podem ser explorados.

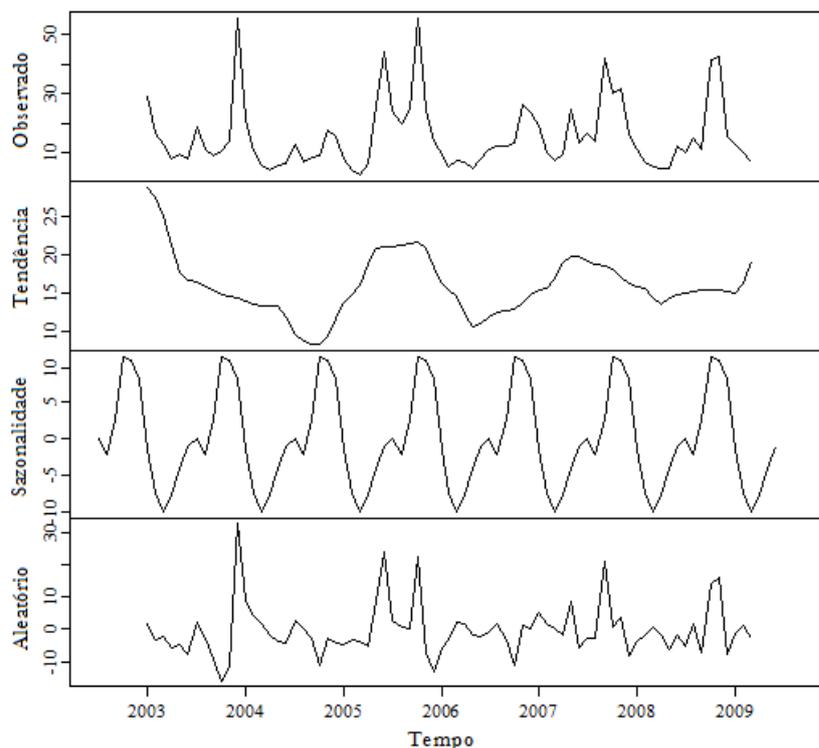
2 FUNDAMENTAÇÃO TEÓRICA

Esta seção discorre sobre os conceitos relevantes para melhor compreensão do tema abordado, sendo eles: Séries Temporais, Redes Neurais Recorrentes (RNR), *Temporal Convolutional Network*, Otimização de Hiperparâmetros, *Tree-structured Parzen Estimator* e Métricas de Estimação.

2.1 Séries Temporais

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo (EHLERS, 2007). Em uma grande gama de aplicações estatísticas, os dados disponíveis consistem de séries temporais, ou seja, uma série de observações, $x_j(n)$; $j = 1, \dots, p$; $n = 1, \dots, N$, feitas através do tempo (WEI, 2006). Dentre essas aplicações, tem-se áreas como biologia, meteorologia, economia, engenharia, entre outros (DAVIS, 1941; ENRIGHT, 1965; GONELLA, 1972; PALIT; POPOVIC, 2006; BAR-JOSEPH; GITTER; SIMON, 2012; DUCHON; HALE, 2012; GRANGER; NEWBOLD, 2014; SALGOTRA; GANDOMI; GANDOMI, 2020). Na Figura 1 pode-se observar um exemplo de série temporal bem como seus componentes.

Figura 1 – Exemplo de série temporal e alguns de seus componentes.



Fonte: Bayer, Castro e Bayer (2012)

A partir das séries temporais é possível extrair informações importantes que podem ser utilizadas para entender melhor o comportamento da série. Ainda na Figura 1 são apresentadas as componentes de tendência, sazonalidade e movimentos aleatórios. A tendência apresenta a inclinação de uma série para aumentar ou diminuir ao longo do tempo. Já a sazonalidade, mostra as forças rítmicas que operam de forma regular e cíclica em até um ano. Por fim, os movimentos aleatórios ou irregulares são simplesmente variações aleatórias na série.

2.1.1 Predição de Séries Temporais

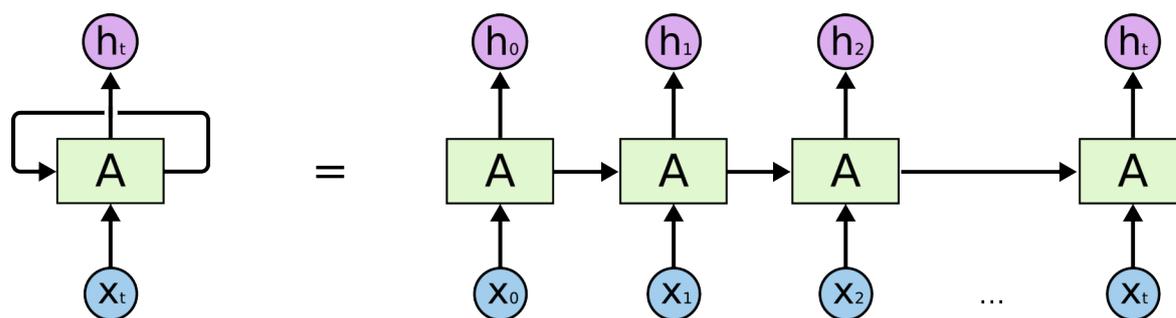
As séries temporais são amplamente utilizadas para tarefas de predição de valores futuros (WEBBY; O'CONNOR, 1996; GOOIJER; HYNDMAN, 2006). Por isso, há um grande interesse nas técnicas capazes de realizar este processamento com êxito. O problema de previsão de série temporal consiste na previsão de valores futuros com base nos valores anteriores e atuais da série temporal (SORJAMAA et al., 2007). Savi (2006) afirma que a predição em séries temporais consiste em ajustar um modelo aos dados de uma série.

Existem várias opções de métodos para predição de séries temporais, desde os métodos mais simples até as abordagens de Redes Neurais Recorrentes. Na literatura, há modelos de previsão com destaque, como o *Simple Exponential Smoothing* (SES) (JR, 1985), *Multilayer Perceptron* (MLP) (RUMMELHART et al., 1986), *Support Vector Regression* (SVR) (BOSER; GUYON; VAPNIK, 1992), *Logistic Regression* (WRIGHT, 1995), Redes Neurais Recorrentes (MEDSKER; JAIN, 2001) e *AutoRegressive Integrated Moving Average* (ARIMA) (BOX et al., 2015).

2.2 Redes Neurais Recorrentes

Rede Neural Recorrente (do inglês, *Recurrent Neural Network* - RNN) são uma variação de redes neurais que trabalham com o conceito de realimentação. Esse tipo de rede utiliza a informação do tempo atual bem como as percepções de padrões de tempos anteriores. Assim, as RNNs possuem duas fontes de entrada, o presente e o passado recente. A RNN combinam essas informações para gerar o conhecimento necessário. Lang, Waibel e Hinton (1990) iniciaram com a ideia de compartilhar parâmetros entre diferentes instantes de tempo. As RNNs foram desenvolvidas visando o aprendizado de padrões sequenciais ou séries temporais. Isso se dá por que as Redes Neurais Artificiais (RNA) não oferecem recursos capazes de armazenar e utilizar informações passadas, pontos essenciais para realizar esse tipo de tarefa. As RNNs podem ser aplicadas em uma ampla variedade de cenários, como classificação de sentimento, descrição de imagens, tradução de textos e predição de séries temporais (ZHANG; JIANG; TONG, 2016; WANG et al., 2016; SINGH et al., 2017; HAN et al., 2019).

Figura 2 – Rede Neural Recorrente e seus passos de execução.



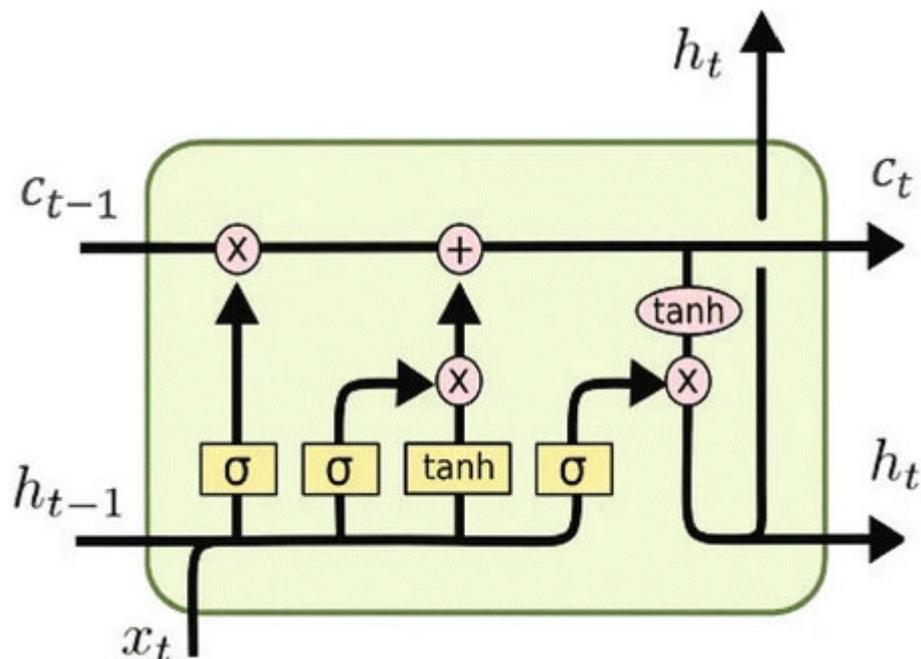
Fonte: Olah (2015)

Existem duas abordagens principais para RNNs, a rede recorrente simples de Elman (ELMAN, 1990) e as redes parcialmente recorrentes de Jordan (JORDAN, 1997). Na abordagem de Elman o processo de realimentação se dá através de uma camada conhecida como oculta. Já Jordan utiliza a própria camada de saída da rede como forma de realimentar. Na Figura 2 observa-se uma representação gráfica de uma RNN, bem como o desenrolar de suas etapas de execução. Nela percebe-se o processo de retroalimentação evidenciado seguindo o padrão proposto por Elman, onde em cada tempo t o neurônio recebe duas entradas, sendo uma delas vinda da camada escondida.

Em uma RNN, a informação sequencial é mantida em um estado oculto, sendo ele capaz de preservar essa informação por várias camadas da rede através do tempo. Esse tipo de problema sequencial é conhecido como dependência de longo prazo ou *long-term dependency*, isso por que o evento no tempo atual é uma função de um ou mais tempos anteriores. Porém, as RNNs apresentam grande dificuldade para treinamento devido aos problemas de *vanishing gradient* e *exploding gradient*, além da dificuldade de aprender padrões de longa duração (LI et al., 2018a). No problema de *vanishing gradient*, o gradiente tende a reduzir à medida volta-se nas camadas ocultas, o que resulta nos neurônios nas camadas anteriores terem um aprendizado consideravelmente mais lento. Já o contrário ocorre no problema de *exploding gradient*, onde o gradiente fica muito maior nas camadas anteriores. Visando resolver esses problemas, principalmente quando trabalhados com séries mais longas, foram criadas a *Long-Short Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) e *Gated Recurrent Unit* (GRU) (CHO et al., 2014). Essas variantes das RNNs utilizam do conceito de *gates* ou portões para persistir informações relevantes por mais tempo. Os *gates* são capazes de aprender ao longo do tempo quais são as informações mais importantes que devem ser mantidas ou descartadas. Ambos são semelhantes na sua essência, porém existem algumas diferenças pontuais. O LSTM possui três *gates*, *input gate*, *output gate* e *forget gate*, e o GRU implementa apenas dois, *reset* e *update gate*. Na LSTM ainda existe o *cell state*, recurso responsável por armazenar informações importantes ao longo do processamento da sequência. Os *gates* na LSTM atuam sobre o *cell state* realizando operações para atualizar o seu valor, dada a

necessidade. Por outro lado, a GRU utiliza a camada escondida para realização da mesma tarefa.

Figura 3 – Célula LSTM e suas componentes.



Fonte: Olah (2015)

Na Figura 3 têm-se as partes que formam uma célula LSTM. Para o tempo t , tem-se: x_t - dado da sequência no tempo t ; C_t - cell state; h_t - saída da célula; i_t - input gate; f_t - forget gate; e o_t - output gate.

Um conjunto de operações são realizadas como somas e multiplicações, como pode ser visto nas Equações (2.1 à 2.6).

$$i_t = \sigma(X_t * U_i + H_{t-1} * W_i) \quad (2.1)$$

$$f_t = \sigma(X_t * U_f + H_{t-1} * W_f) \quad (2.2)$$

$$O_t = \sigma(X_t * U_o + H_{t-1} * W_o) \quad (2.3)$$

$$\tilde{C}_t = \tanh(X_t * U_c + H_{t-1} * W_c) \quad (2.4)$$

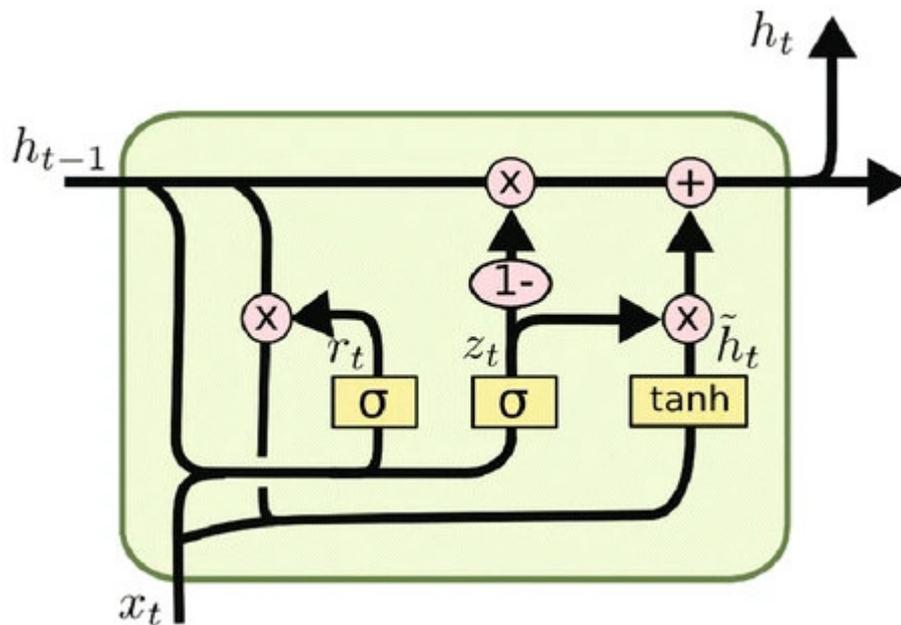
$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \quad (2.5)$$

$$h_t = \tanh(C_t) * O_t \quad (2.6)$$

A saída da *cell state*, \tilde{C}_t , e a saída, h_t , são computados utilizando a função de ativação Tangente Hiperbólica, representada por \tanh . Já os *gates* fazem uso de ativações sigmoide, onde os valores são reduzidos entre 0 e 1. Isso é de grande importância para que informações sejam esquecidas ou mantidas, uma vez que, quando multiplicado por 0, o valor é removido. Já quando o valor é multiplicado por 1, a informação é preservada.

Comparativamente, o diagrama da GRU pode ser visto na Figura 4 e as Equações de 2.7 à 2.10. Na GRU, o *update gate* z_t se assemelha à uma combinação entre o *input gate* e *forget gate* da LSTM. O mesmo é computado em função de x_t e h_{t-1} e, em seguida, aplicando a função de ativação sigmóide.

Figura 4 – Célula GRU e suas componentes.



Fonte: Olah (2015)

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (2.7)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (2.8)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \quad (2.9)$$

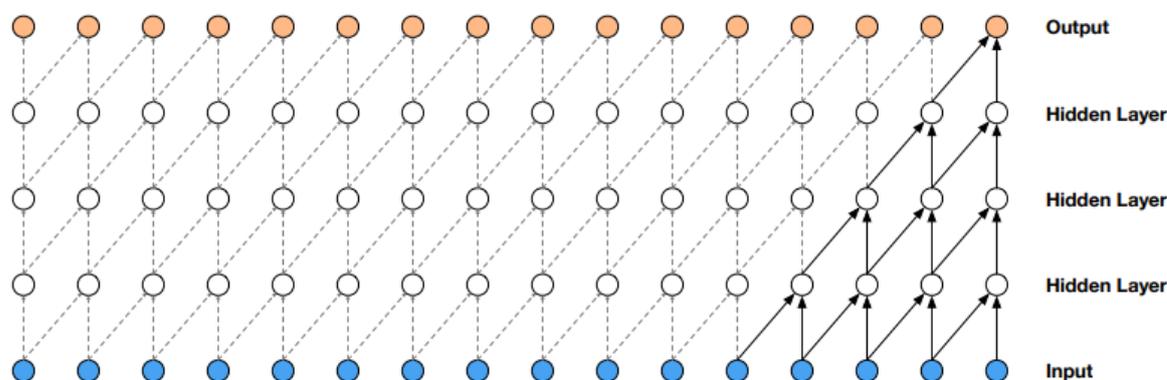
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2.10)$$

2.3 Temporal Convolutional Network (TCN)

Redes Convolucionais Temporais (do inglês, *Temporal Convolutional Network* - TCN) são arquiteturas de redes propostas por [Bai, Kolter e Koltun \(2018\)](#), baseadas em duas características principais: realiza convoluções causais e permite entrada de tamanho variável. Convoluções Causais significam que nenhuma informação é transmitida do futuro para o passado, e assim permite entradas de diferentes tamanhos, mapeando para qualquer sequência de saída, como em uma RNN. A TCN, também pode usar dados do passado mais distante para realizar previsões, semelhante a uma rede LSTM. Desta forma, ela combina simplicidade de aplicação, a capacidade auto-regressiva e memória de longa duração. E como vantagem, se destaca por permitir paralelismo, ser flexível quanto ao tamanho das entradas e ao controle de uso de memória de forma eficaz. Estes atributos contribuíram para a escolha desta rede para compor o método proposto neste estudo.

Nas convoluções causais, as previsões $p(x_{t+1}|x_1, \dots, x_t)$ geradas no tempo t não podem depender de quaisquer valores no futuro $x_{t+1}, x_{t+2}, \dots, x_T$ como representado na Figura 5, utilizam dois conceitos como base. O primeiro conceito, é que as redes geram uma saída de mesmo tamanho da entrada. O segundo, é o fato de que não há informações do futuro compartilhadas com o passado. Para tratar os diferentes tamanhos de saída, é utilizada uma Rede Unidimensional Convolutiva Totalmente Conectada (FCN - 1D Fully Convolutional Network) ([LONG; SHELHAMER; DARRELL, 2015](#)). Para o segundo conceito, são utilizadas convoluções causais, onde cada saída em tempo t é concluída apenas com dados do mesmo tempo ou anterior. Porém, essa estratégia leva a redes extremamente profundas e com o uso de filtros grandes.

Figura 5 – Empilhamento de camadas de convoluções causais.

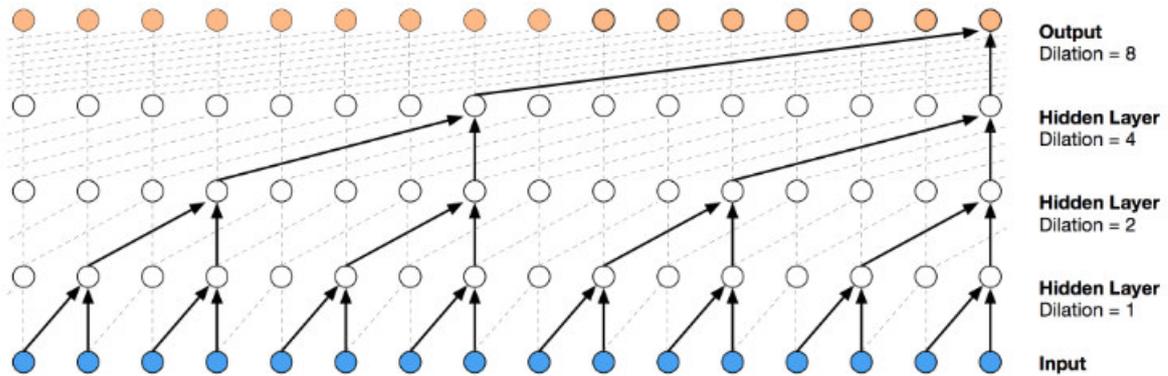


Fonte: [Oord et al. \(2016\)](#)

Outro conceito importante aplicado na TCN são as convoluções dilatadas (Figura 6). Elas partem do trabalho proposto por [Oord et al. \(2016\)](#), aplicando convoluções dilatadas permitindo um largo campo receptivo com a aplicação de poucas camadas, como demonstrado por [Yu e Koltun \(2015\)](#). Essas convoluções podem ser utilizadas para facilitar

a aplicação das convoluções causais em tarefas de sequência, principalmente quando é necessário treinar dados com um histórico mais longo. Isso se dá, visto que, uma convolução causal pode apenas olhar um histórico de tamanho linear na profundidade da rede.

Figura 6 – Empilhamento de camadas convolucionais casuais dilatadas.



Fonte: Oord et al. (2016)

Uma convolução dilatada, dada uma sequência de entrada $x \in \mathbb{R}^n$ e um filtro $f : \{0, \dots, k-1\} \rightarrow \mathbb{R}$, é uma operação F no elemento s em uma sequência definida como na Equação 2.12

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \quad (2.11)$$

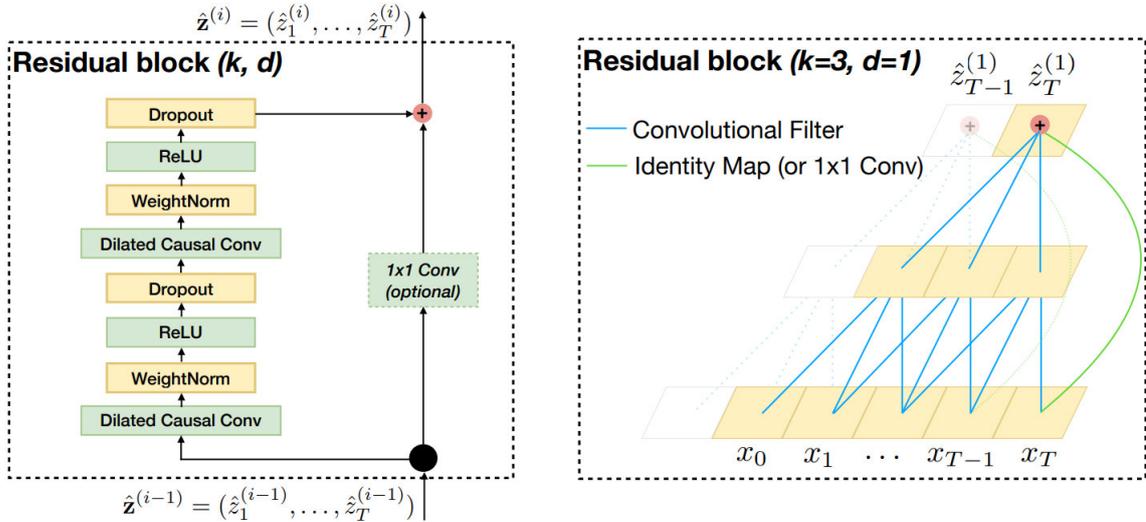
onde d é a taxa de dilatação, k é o tamanho do filtro, e $s - d \cdot i$ é a direção em que a convolução será aplicada. No caso de $d = 1$, a convolução dilatada se reduz a uma convolução regular. O campo receptivo na TCN está diretamente ligado à profundidade da rede, assim como o tamanho do filtro k e o fator de dilatação d , tendo um grande impacto na estabilidade da rede.

Com isso, cada camada consiste em diversos filtros para extração de características. Logo, é aplicado um bloco residual (HE et al., 2016) em substituição às camadas convolucionais. Cada bloco residual contém um ramo que leva um conjunto de transformações F , cujas saídas são adicionadas à entrada x do bloco:

$$o = \text{Activation}(x + F(x)). \quad (2.12)$$

O bloco residual em uma TCN é composto de duas camadas de convoluções casuais dilatadas, ativação ReLU (NAIR; HINTON, 2010), normalização ponderada (SALIMANS; KINGMA, 2016) aplicada aos filtros convolucionais e, por fim, uma camada *spatial dropout* (SRIVASTAVA et al., 2014) aplicada em cada convolução dilatada, como ilustrado na Figura 7.

Figura 7 – Bloco residual e suas respectivas camadas.



Fonte: Bai, Kolter e Koltun (2018)

2.4 Otimização de hiperparâmetros

Em Aprendizado de Máquina, hiperparâmetros são os parâmetros que não são otimizados durante a etapa de treinamento. Corriqueiramente se lança mão de um certo número de hiperparâmetros que se relacionam: ao pré-processamento de dados, à regularização, ao treinamento e à definição estrutural de um modelo, para obter um melhor desempenho de predição (JÚNIOR, 2018). Friedman et al. (2001) se refere ao problema de otimização de hiperparâmetros como *model selection*. Essa definição compreende desde a definição do algoritmo de aprendizado até a escolha das funções de *kernel*. Os hiperparâmetros determinam características estruturais do modelo, como o número de camadas ocultas, função de ativação, entre outros. Além de parâmetros que moldam como o modelo é treinado, como por exemplo, taxa de Aprendizagem, número de épocas de treinamento, etc.

Para garantir que o conjunto ótimo de hiperparâmetros sejam alcançados é necessário definir uma função objetivo e uma estratégia de busca. A função objetivo consiste em buscar o conjunto de hiperparâmetros capaz de minimizar ou maximizar uma determinada métrica, respeitando um sistema linear de restrições. A escolha entre minimizar e maximizar varia conforme o comportamento que é desejado pela função objetivo. Já a estratégia de busca é o método de otimização que é responsável por fazer a seleção de um conjunto de hiperparâmetros. Existem diversas abordagens de algoritmos de otimização, desde as mais simples como, *Grid Search* e *Random Search* (BERGSTRA; BENGIO, 2012), até abordagens baseadas em meta-heurísticas como *Simulated Annealing* (KIRKPATRICK; GELATT; VECCHI, 1983) e estratégias evolutivas como Otimização de Enxame de Partículas (do inglês, *Particle Swarm Optimization* - PSO) (KENNEDY; EBERHART, 1995) e Algoritmo Genético (HOLLAND, 1992).

Ainda há estratégias baseadas no modelo de otimização sequencial (do inglês, *Sequential Model-based Optimization* - SMBO) (HUTTER; HOOS; LEYTON-BROWN, 2011), função dos hiperparâmetros que podem ser ajustados baseados nas tentativas anteriores e assim apontar como alcançar o mínimo local. O SMBO é um formalismo sucinto da otimização Bayesiana (do inglês, *Bayesian optimization*) (FRAZIER, 2018). O SMBO faz uso de um modelo de regressão probabilístico para prever o desempenho de cada configuração de hiperparâmetros. Esse modelo é utilizado por uma função de aquisição para gerar um novo conjunto de parâmetros a serem avaliados, que pode ser definido pelo pseudo-algoritmo 1, onde $f : X \rightarrow \mathbb{R}$ é a função de *fitness* e x^* é o ponto que maximiza a função Surrogate (LANGE; HUNTER; YANG, 2000).

Algorithm 1 SMBO

SMBO(f, M_0, T, S):

- 1: $H = \emptyset$
- 2: **for** $t = 1$ em T **do**
- 3: $x^* = \operatorname{argmin}_x S(x, M_{t-1})$
- 4: Avaliar $f(x^*)$
- 5: $H = H \cup (x^*, f(x^*))$
- 6: Treinar novo modelo M_t para H
- 7: **end for**
- 8: **return** H

end function

A estratégia de busca atua sobre um conjunto de possíveis valores que cada hiperparâmetro pode assumir, denominado de espaço de busca. O tamanho do espaço de busca definido está diretamente relacionado ao tempo de otimização, dado que o número de possibilidades a serem testadas aumentam consideravelmente. Isso tende a ser um problema principalmente em estratégias de busca exaustiva, onde todas as combinações de hiperparâmetros são avaliadas até que uma condição de parada seja atendida. Métodos que se encaixam nesse perfil são o *Grid Search* e *Random Search*. Assim, outros métodos de otimização que melhor convergem para o conjunto ótimo de hiperparâmetros e com um menor esforço computacional acabam sendo preferíveis.

2.4.1 Tree-structured Parzen Estimator

Tree-structured Parzen Estimator (TPE) (BERGSTRA et al., 2011) é um dos principais métodos de otimização. O TPE é um modelo SMBO que faz uso de modelos construídos sequencialmente para aproximar o desempenho de um conjunto de hiperparâmetros alvo baseado no histórico de leituras. Portanto, dado um conjunto de hiperparâmetros M aplicado ao modelo de predição $F(M)$ que gera um erro e , visa-se encontrar o conjunto ótimo de hiperparâmetros que minimize o erro gerado na predição. Logo, TPE se resume a uma abordagem Bayesiana para otimizar o conjunto ótimo de hiperparâmetros.

Assim, o TPE consiste em um conjunto de núcleos estimadores organizados em árvore para preservar a dependência condicional. Essa estratégia gera dois modelos probabilísticos, $l(x)$ e $g(x)$, baseados em processos hierárquicos, onde os mesmos modelam o domínio de variáveis quando a função objetiva está acima ou abaixo de um limiar de desempenho y^* . Um dos benefícios de utilizar o TPE como algoritmo de otimização é o fato de possuir suporte natural à domínios com variáveis condicionais (DEWANCKER; MCCOURT; CLARK, 2015). Esse comportamento é descrito pela Equação 2.13 abaixo,

$$p(y|x) = \frac{p(x|y) * p(y)}{p(x)} \quad (2.13)$$

onde $p(x|y)$ é a probabilidade dos hiperparâmetros dado o escore da função objetiva, determinada pela Equação 2.14:

$$p(x|y) = \begin{cases} l(x) & \text{se } y < y^* \\ g(x) & \text{se } y \geq y^* \end{cases} \quad (2.14)$$

A TPE também utiliza-se de um melhoramento esperado estimado utilizando uma *Expected Improvement* (EI) definido pela Equação 2.15, valor este proporcional à razão $l(x) - g(x)$. Sendo assim, para maximizar o EI é também necessário maximizar a razão. O TPE traça amostras de hiperparâmetros a partir de $l(x)$, avaliando-os em termos de $l(x) - g(x)$ e retornando o conjunto com o maior $l(x) - g(x)$ correspondente ao maior EI. Esses hiperparâmetros são então avaliados sob a função objetiva.

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1 - \gamma) g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)} (1 - \gamma) \right)^{-1} \quad (2.15)$$

2.5 Métricas de Estimação

Esta seção descreve as métricas comumente utilizadas para avaliar as estimativas em séries temporais e que também são utilizadas na validação dos experimentos realizados neste estudo.

As métricas de erro absoluto permitem comparar e avaliar resultados de predições com base no erro absoluto entre o valor predito e o valor real. Na literatura são encontradas diversas métricas para esse tipo de cenário, no entanto, algumas das mais utilizadas são *Mean Absolute Percentage Error* (MAPE), *Symmetric Mean Absolute Percentage Error* (SMAPE) e *Mean Absolute Error* (MAE) (MYTTENAERE et al., 2016; ARMSTRONG; FORECASTING, 1985; WILLMOTT; MATSUURA, 2005).

Mean Absolute Error (MAE) é uma métrica simétrica e que também é amplamente utilizada como função de *loss* em técnicas de regressão, que avalia o erro absoluto do valor

estimado quando comparado com o valor real. Sendo, dado pela Equação 2.16, onde P_t é o valor predito e A_t o valor real.

$$MAE = \frac{1}{n} \sum_{t=1}^n |P_t - A_t| \quad (2.16)$$

Mean Absolute Percentage Error (MAPE) é uma métrica para erro percentual, onde o resíduo gerado entre o valor real e o predito é escalado em função do valor real. Logo, o MAPE é definido segundo a Equação 2.17, onde P_t é o valor predito e A_t é o valor real.

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \frac{|P_t - A_t|}{|A_t|} \quad (2.17)$$

Symmetric Mean Absolute Percentage Error (SMAPE) é uma métrica para erro percentual que, assim como o MAPE, permite a comparação de erros em diferentes escalas. Além disso, o SMAPE é simétrico, o que significa que ambos, o valor predito e real, possuem a mesma importância para o cálculo da métrica. Logo, diferente de outras métricas de erro percentual, como o MAPE, o SMAPE não é sensível a valores pequenos, conforme demonstrado por Kim e Kim (2016). Além disso, o SMAPE gera valores entre 0 e 100%, o que pode simplificar a comparação em alguns cenários.

Sendo assim, SMAPE é definido segundo a Equação 2.18, onde P_t é o valor predito e A_t é o valor real.

$$SMAPE = \frac{100\%}{n} \sum_{t=1}^n \frac{|P_t - A_t|}{|P_t| + |A_t|} \quad (2.18)$$

3 TRABALHOS RELACIONADOS

Nessa seção são apresentados trabalhos que abordam a estimação individual de consumo de energia elétrica em séries de curta, média e longa duração, além de abordar otimização de redes neste contexto, sendo eles [Lu, Niu e Jia \(2004\)](#), [Rodrigues et al. \(2019\)](#), [Dong e Grumbach \(2018\)](#), [Li et al. \(2018b\)](#), [Kong et al. \(2017a\)](#), [Lara-Benítez et al. \(2020\)](#) e [Kong et al. \(2017b\)](#).

[Lu, Niu e Jia \(2004\)](#) propuseram um modelo híbrido que combina *Autoregressive Integrated Moving Average* (ARIMA) e *Artificial Neural Network* (ANN) para realizar a predição de consumo. A base de dados utilizada referente ao consumo diário de cada cidade de Hebei, na China, entre o período de 1 de Março até 30 de Junho do ano de 2003, onde registros de leitura eram capturadas a cada 15 minutos. Os autores fazem uso de um modelo híbrido, uma vez que, a base de dados é complexa, com observações variadas, incluindo partes lineares e não-lineares. Assim, visto que o ARIMA possui características que possibilitam lidar com predições lineares e o ANN com predições não-lineares, os autores decidiram então combinar as técnicas em uma única abordagem. Os resultados obtidos demonstram que a proposta obteve resultados satisfatórios, um MSE de 0.063 comparado à 0.113 utilizando apenas o ARIMA, obtém uma melhora de 44%.

No trabalho de [Rodrigues et al. \(2019\)](#) tem-se o objetivo de identificar inconsistências e erros no processo de leitura para reduzir custos atrelados as análises de leituras e possíveis processos. Para tanto, foi aplicado o método de *Empirical Bayes* ([MARITZ; LWIN, 2018](#)) e uma estratégia para remoção de sazonalidade, utilizando essa característica em favor da predição de consumo. Os dados utilizados são de uma companhia de energia elétrica do Brasil, com cerca de 35 milhões de leituras de 1,3 milhões de clientes. As leituras correspondem a um período de Fevereiro de 2014 à Novembro 2016. Os autores avaliam o trabalho exclusivamente pela análise de casos que foram enviados para verificação. Os resultados obtidos demonstram uma redução de até 23% de casos considerados como inconsistência e que necessitam ser enviados para análise, além de manter a satisfação dos clientes.

Já [Dong e Grumbach \(2018\)](#) propuseram um modelo híbrido baseado em uma *Long short-term memory* (LSTM) para estimação de demanda anual de alimentadores de distribuição esperada de uma companhia. Foi usado *Principal Component Analysis* para extrair características socioeconômicas e de consumo. Por fim, foi construído um modelo LSTM para prever demanda. Os experimentos foram conduzidos baseados em informação de consumo da região urbana ocidental do Canadá, durante o verão e inverno. O *dataset* era composto por dados dos últimos 14 anos de 298 centros de distribuição

elétrica. Foram obtidos Erro Médio Absoluto Percentual (Mean Absolute Percentage Error - MAPE) de 6,77% para o verão e 4,87% para o inverno.

Li et al. (2018b) desenvolveram um modelo para prever consumo de energia de curta duração, aplicando como base *Empirical Mode Decomposition* (EMD), *Particle Swarm Optimization* (PSO) e LSTM, que os autores denominaram de EMD-PSO-LSTM. O EMD é utilizado para decompor a base de dados em componentes de frequência semelhante. Já o LSTM é responsável pelo treinamento e previsão do consumo, sendo o mesmo como seus hiperparâmetros otimizados pelo PSO. Foi utilizado o *dataset* EUNITE, que é composto de 8.760 leituras obtidas entre 00:00 e 23:00, em cada dia do ano de 1998. O experimento foi conduzido usando 335 dias como treinamento e o 336º dia foi reservado para teste. Além disso, as séries foram submetidas a uma etapa de pré-processamento para suavizar, seguida de uma decomposição utilizando EMD. O melhor resultado obtido foi um MAPE de 1,89%.

Uma rede LSTM também foi empregada em Kong et al. (2017a) para previsão de curto tempo em consumo de energia em casas individuais. Foi utilizado um *dataset* criado pelo governo Australiano no projeto Smart Grid Smart City (SGSC), que monitorou o consumo de 10.000 consumidores de New South Wales durante 4 anos. Foi utilizado para avaliar os resultados obtido usando MAPE, com erro de 8,18% na previsão de consumo.

Em Lara-Benítez et al. (2020) foi utilizada uma rede TCN para realizar previsão de séries temporais relacionadas à energia. Os dados utilizados são referentes à dois tipos de dados: demanda de eletricidade e energia de estações de carregamento de veículos elétricos. O *dataset* de demanda elétrica incluí dados de 02/01/2014 até 01/11/2019, com leituras feitas a cada 10 minutos, totalizando 306.721 registros. Nos experimentos realizados, os autores demonstram que a rede TCN obteve melhores resultados quando comparado à LSTM, tanto em acurácia, quanto ao tempo necessário para realizar as previsões. Os resultados também mostram uma erro absoluto percentual ponderado (do inglês, *Weighted Absolute Percentage Error - WAPE*) de 0,4224, uma melhora de desempenho de cerca de aproximadamente 4% em relação ao LSTM.

Já no contexto de otimização de hiperparâmetros, Kong et al. (2017b) descreve os resultados obtidos aplicando um *framework* de otimização de hiperparâmetros foram apresentados para a previsão de consumo usando *neural networks*. A base de dados utilizada é a mesma de Kong et al. (2017a), composto de leituras residenciais. Os autores aplicaram *Tree-structured Parzen Estimator* (TPE) para realizar a otimização dos hiperparâmetros de uma rede LSTM. Com um MAPE de 22,94%, o *framework* proposto se mostrou eficaz para a tarefa designada, com uma melhora de 10%.

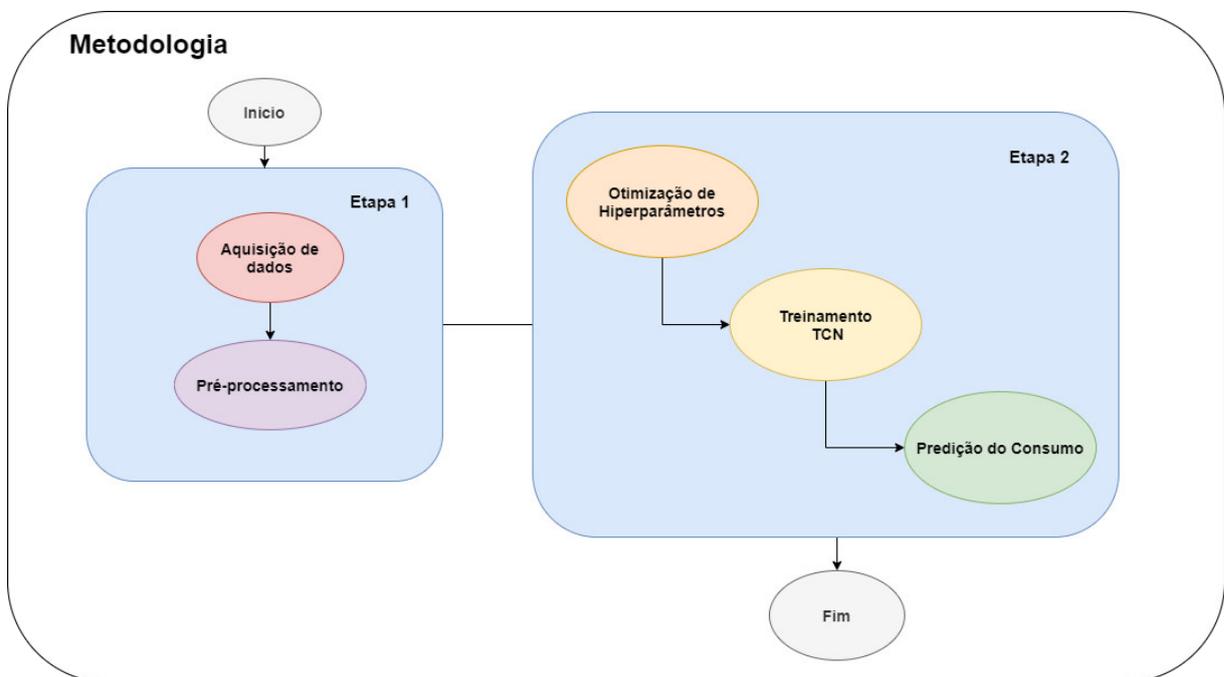
Os trabalhos citados possuem bases de dados com diferentes características relacionadas à previsão de consumo de energia. A maioria dos trabalhos usam registros de leituras realizadas por minuto ou hora (Li et al. (2018b); Li et al. (2018b); Lara-Benítez

et al. (2020); Kong et al. (2017a); Kong et al. (2017b)), e alguns desses usam medidores inteligentes ou *SmartMeter*. Já Rodrigues et al. (2019) é o único que realiza predição mensal de consumo. Este trabalho visa abordar uma base mais abrangente de classes de consumo, variando em diferentes perfis de consumo entre alta e baixa tensão. Os trabalhos apresentados obtiveram bons resultados, porém os *dataset* são, em geral, menores e mais controlados. Outra diferença são as categorias de equipamentos utilizados no registro de consumo, sendo realizada em alguns dos trabalhos citados de forma automática utilizando medidores inteligentes, porém neste trabalho a leitura é realizada de forma manual, o que aumenta as chances de problemas nos registros da base de dados. Este trabalho se aproxima dos objetivos de Rodrigues et al. (2019). Contudo, diferente deste trabalho, os autores não utilizam métricas de predição para avaliar os resultados obtidos. Já o Lara-Benítez et al. (2020) faz uso da TCN, que se alinha à este trabalho por utilizar a mesma abordagem de predição. Portanto, esta pesquisa busca preencher estas lacunas na literatura, realizando a predição mensal, criando um modelo para cada cliente analisado e utilizando uma abordagem com a rede TCN com otimização de hiperparâmetros.

4 METODOLOGIA

Esta seção apresenta a metodologia proposta para a predição do consumo mensal utilizando uma rede TCN (Seção 2.3). Na Figura 8 tem-se um esquema do fluxo das etapas da metodologia. Há duas macro-etapas, a primeira destinada às etapas de aquisição e tratamento dos dados, já a segunda inclui o processo de otimização de hiperparâmetros, treinamento da rede TCN e, por fim, a predição individual de consumo. As etapas desta metodologia são aplicadas para cada cliente.

Figura 8 – Metodologia proposta para predição.



Fonte: Acervo do autor.

4.1 Aquisição de dados

A base utilizada é privada, fornecida por uma companhia de distribuição de energia elétrica de um estado brasileiro. O *dataset* foi disponibilizado contendo um total de 69.656.857 registros de consumo mensal, formando 2.702.738 séries temporais, no qual cada série corresponde ao histórico de consumo de energia de um cliente da companhia. As leituras selecionadas são referentes ao período de Janeiro de 2018 até Junho de 2020, totalizando 30 meses de registro de consumo de energia. A base está dividida em oito classes de consumo, conforme classificação da ANEEL ⁵, sendo elas:

⁵ AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Tarifas Consumidores**. Brasília: ANEEL, 2021. Disponível em: https://www.aneel.gov.br/tarifas-consumidores/-/asset_publisher/zNaRBjCLDgBE/content/classe/654800. Acesso em: 25 mar. 2021.

- **Residencial:** nesta categoria, enquadram-se as unidades consumidoras com fim residencial;
- **Industrial:** são as unidades consumidoras em que são desenvolvidas atividades industriais;
- **Comercial, serviços e outras atividades:** nesta, enquadram-se as unidades consumidoras onde são desenvolvidas as atividades de prestação de serviços e demais não previstas nas outras classes;
- **Serviço público:** enquadram-se as unidades consumidoras que se destinem, exclusivamente, ao fornecimento para motores, máquinas e cargas essenciais à operação de serviços públicos de água, esgoto, saneamento e tração elétrica urbana ou ferroviária, explorados diretamente pelo Poder Público ou mediante concessão ou autorização;
- **Consumo próprio:** enquadram-se as unidades consumidoras de titularidade das distribuidoras;
- **Rural:** enquadram-se as unidades consumidoras que desenvolvem atividades de agricultura, pecuária ou aquicultura;
- **Poder público:** enquadram-se as unidades consumidoras de responsabilidade de consumidor que seja pessoa jurídica de direito público, independentemente da atividade desenvolvida, incluindo a iluminação em vias e semáforos, radares e câmeras de monitoramento de trânsito, exceto aqueles classificáveis como serviço público de irrigação rural, escola agrotécnica, iluminação pública e serviço público;
- **Iluminação pública:** serviço público que tem por objetivo exclusivo prover claridade aos logradouros públicos, de forma periódica, contínua ou eventual.

Dentre elas, a classe Residencial compõem a maior parte dos clientes da base de dados, com uma representatividade de 88,21%. Na Tabela 1, pode-se observar a distribuição dos clientes, o percentual para cada classe, a média de consumo registrado em cada classe e o desvio padrão.

Tabela 1 – Distribuição quantitativa, percentual, a média e o desvio padrão de clientes por cada classe.

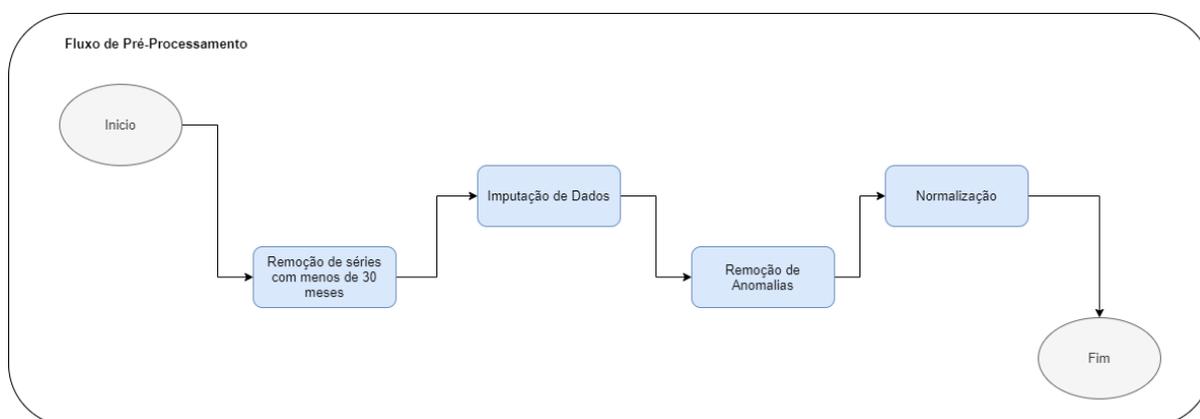
Classe	Nº Clientes	Percentual	Média	Desvio Padrão
Comercial	140.697	5,21%	1.101,78	8.401,35
Consumo Próprio	11.573	0,43%	3.478,77	15.323,70
Iluminação Pública	1.221	0,05%	2.427,03	2.583,60
Industrial	7.505	0,28%	4.956,51	25.317,97
Poder Público	25.955	0,96%	2.115,62	13.339,80
Residencial	2.383.956	88,21%	137,34	206,98
Rural	124.098	4,59%	472,79	7.155,48
Serviço Público	7.309	0,27%	5.280,30	24.866,76

Fonte: Acervo do autor.

4.2 Pré-processamento

Após aquisição da base de dados, realiza-se o pré-processamento a fim de preparar os dados e mitigar alguns dos problemas que podem ser encontrados nas séries temporais. A etapa de pré-processamento se subdivide em 5 subetapas: (1) Remoção de séries com menos de 30 meses; (2) Imputação de dados; (3) Remoção de anomalias e (4) Normalização. A Figura 9 ilustra o fluxo de execução do pré-processamento.

Figura 9 – Fluxo de execução do pré-processamento.



Fonte: Acervo do autor.

Inicialmente, são eliminados os clientes que não possuem os 30 de consumo registrado, sendo um por mês. Isso foi feito, uma vez que, via o experimento descrito na Seção 5.1 percebeu-se que à medida que as séries possuíam mais meses de consumo registrado, o modelo preditivo alcançado reduzia o erro de predição. Em casos de ausência de registro de consumo ao longo da série, é aplicada uma interpolação linear para o preenchimento dos vazios, como demonstrado em Noor et al. (2015). Também há problemas com registros de valores ruidosos (*outliers*). Para mitigar esse tipo de problema, a companhia utiliza regras para identificar os ruídos no consumo registrados para os clientes de baixa e alta tensão ⁶.

⁶ AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Regulação dos Serviços de Distribuição**. Bra-

Para clientes de baixa tensão, registros de um mês de consumo superior à 50.000 kWh são tratados como prováveis *outliers*. Já em caso de clientes de alta tensão, o limiar de corte é de valores acima de 1.000.000 kWh. Desta forma, excluí-se os registros que estão acima do limiar descrito, e em seguida aplica-se uma interpolação linear, da mesma forma que foi aplicado na etapa de imputação de dados. A base é então dividida em treino e teste, com 29 dos 30 meses reservados para treino e o último para teste. Desta forma, é possível aumentar a semelhança com o procedimento aplicado no dia-a-dia de uma empresa de distribuição de energia. Por fim, é aplicada uma normalização dos valores entre 0 e 1.

Conseqüentemente, o número de séries que seguiram para serem processadas reduziu para 1.780.675, após o pré-processamento, como pode ser visto na Tabela 2. Isso ocorre já que há muitos clientes com histórico de consumo inferior à 30 meses. Essa decisão foi tomada baseando-se nos experimentos demonstrados na Seção 5.1, onde pode-se observar a queda de desempenho à medida que há menos informação para treinamento.

Tabela 2 – Quantidade de instalações na base final após a etapa de pré-processamento para filtrar apenas séries com 30 meses de consumo registrado, média de consumo registrado por classe e o desvio padrão.

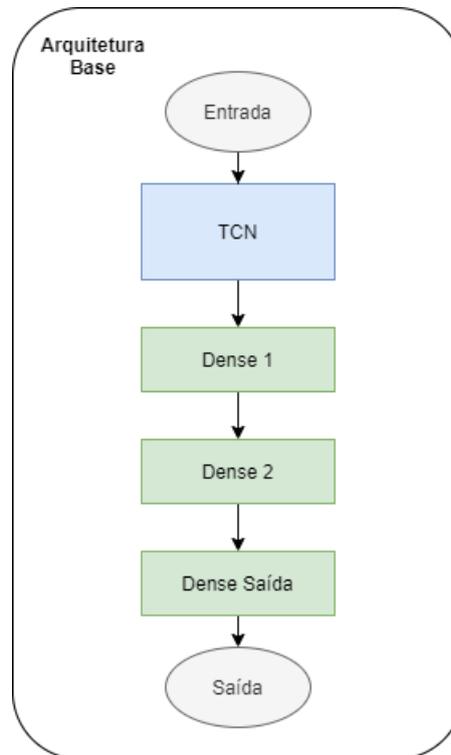
Classe	Quantidade	Porcentagem
Comercial	102.697	3,79%
Consumo Próprio	9.683	0,35%
Iluminação Pública	1.128	0,04%
Industrial	7.505	0,02%
Poder Público	21.754	0,08%
Residencial	1.535.621	56,81%
Rural	95.469	3,53%
Serviço Público	6.818	0,25%
Total	1.780.675	65,88%

Fonte: Acervo do autor.

4.3 Otimização de Hiperparâmetros

Nesta etapa é realizada a otimização dos hiperparâmetros da rede. Esse processo é realizado de forma individualizada, ou seja, para cada série temporal de consumo. O objetivo dessa etapa é obter o conjunto ótimo de hiperparâmetros que minimiza o erro de predição SMAPE (Seção 2.5) para o espaço de busca definido. Para isso, utiliza-se uma rede TCN (Seção 2.3) com arquitetura definida conforme ilustrado na Figura 10. Nela há uma camada TCN, duas camadas Densas intermediárias e uma camada Densa de saída. A escolha pela utilização de duas camadas Densas intermediárias foi baseada nos resultados do experimentos descritos em 5.2. Já a última camada Densa, gera a saída da rede.

Figura 10 – Arquitetura base utilizada para a otimização dos hiperparâmetros e o treinamento da rede.



Fonte: Acervo do autor.

O espaço de busca definido inclui hiperparâmetros da camada TCN e das camadas Densa 1 e 2. Na camada TCN tem-se o fator de dilatação e a taxa de *dropout*. Já para a camada Densa 1 e 2, são otimizados os seus respectivos números de unidades. Por fim, na camada Densa de saída foram utilizados os regularizadores l1 e l2 para o *kernel regularizer*, *bias regularizer* e *activity regularizer*. Na Tabela 3 pode-se ver os valores do espaço de busca definido. O número de dilatações corresponde à uma sequência incremental de potências de 2, valor esse baseado no experimento realizado por Bai, Kolter e Koltun (2018). A otimização de hiperparâmetros utiliza o Hyperopt (BERGSTRA; YAMINS; COX, 2013), biblioteca em Python que possui o TPE (Seção 2.4.1) implementado nativamente. O Hyperopt é a aplicação utilizada para avaliar os modelos gerados ao longo de 50 iterações.

4.4 Treinamento e Predição

Após a otimização de hiperparâmetros ser realizada, tem-se então o conjunto de hiperparâmetros que obteve melhores resultados para uma determinada série temporal. Assim, esses hiperparâmetros são aplicados ao treinamento da rede TCN. A rede treinada segue a mesma estrutura da imagem 10 que foi utilizada para a otimização dos hiperparâmetros. Tendo em vista que as séries temporais possuem 30 meses de consumo, a rede é alimentada com 29 meses para treinamento. Além disso, modelo é treinado então com as

Tabela 3 – Espaço de busca utilizado para otimização de hiperparâmetros.

Hiperparâmetro	Espaço
Dropout	0 à 0.4
Fator de dilatação	4, 5, 6, 7 e 8
Nº de unidades Densa 1	8, 16, 32, 64, 128 e 256
Nº de unidades Densa 2	8, 16, 32, 64, 128 e 256
L1 kernel regularizer	10^{-2} , 10^{-3} , 10^{-4} e 10^{-5}
L2 kernel regularizer	10^{-2} , 10^{-3} , 10^{-4} e 10^{-5}
L1 bias regularizer	10^{-2} , 10^{-3} , 10^{-4} e 10^{-5}
L2 bias regularizer	10^{-2} , 10^{-3} , 10^{-4} e 10^{-5}
L1 activity regularizer	10^{-2} , 10^{-3} , 10^{-4} e 10^{-5}
L2 activity regularizer	10^{-2} , 10^{-3} , 10^{-4} e 10^{-5}

Fonte: Acervo do autor.

seguintes configurações: MAPE como função de loss, otimizador Adam, ativação ReLu e 100 iterações.

Ao fim do treinamento, é realizada a predição do último mês de consumo da série temporal. O valor predito é então desnormalizado para que sejam calculadas as métricas de avaliação, processo esse definido pela Equação 4.1, onde X é o valor desnormalizado, X' o valor normalizado, X_{max} e X_{min} os intervalos máximo e mínimo, respectivamente.

$$X = X' \cdot (X_{max} - X_{min}) + X_{min} \quad (4.1)$$

Tanto o treinamento quanto a predição são realizados através do Keras (CHOLLET et al., 2015), *framework* que possui uma boa sinergia com o Hyperopt. O Pseudo-algoritmo 2 apresenta sequência de etapas da metodologia proposta. Dada uma lista A de séries temporais a serem processadas, para cada séries na lista são realizadas as seguintes passo: na linha 3 a série é passada pela etapa de pré-processamento e o resultado do processo é armazenado em S ; na linha 4 a série é separada em dados de treino, *labels* de treino e dados de teste, x , y e x' , respectivamente; na linha 4 ambos x e y são utilizados para a realização da otimização de hiperparâmetros, que resulta em um conjunto Z de hiperparâmetros; a linha 5 mostra o conjunto de *labels* treino y e o conjunto de hiperparâmetros Z a serem utilizados para o treinamento, o que resulta em um modelo final X ; em seguida, na linha 7 tem-se o modelo X e os dados para teste x' sendo utilizados para a predição do consumo.

Algorithm 2 Treinamento e Predição

ht!

```
1:  $A = \text{lista\_series\_temporais}$ 
2: for  $i$  em  $A$  do
3:    $S = \text{pre\_processamento}(i)$ 
4:    $x, y, x' = S[i]$ 
5:    $Z = \text{otimizacao}(x, y)$ 
6:    $X = \text{treinamento}(Z, x, y)$ 
7:    $y' = \text{predicao}(X, x')$ 
8: end for
```

5 EXPERIMENTOS E DISCUSSÃO

Neste capítulo são descritos os experimentos realizados e os resultados obtidos nas diferentes etapas deste trabalho. A Seção 5.1 descreve o experimento que avalia o impacto da quantidade de meses de consumo. Já a Seção 5.2 apresenta uma análise do ganho de desempenho ao estender o número de camadas presentes na arquitetura base utilizada. Por fim, na Seção 5.3 apresenta os resultados obtidos em cada etapa da metodologia proposta, bem como os resultados gerais.

5.1 Experimento 1: Análise de quantidade de meses

Este experimento teve como objetivo analisar e identificar o impacto da quantidade de registros de consumo mensal utilizada durante o treinamento. É conhecido que quanto maior a quantidade de dados disponíveis para treinamento, maior é o desempenho das técnicas de aprendizado de máquina (RAUDYS; JAIN et al., 1991). Porém, no contexto de consumo mensal de energia elétrica, há uma grande variedade na quantidade de leituras registradas. Portanto, fez-se necessário identificar o número de meses do histórico de consumo que disponibiliza dados suficientes para treinamento, minimizando o número de clientes que deixam de ser processados. O experimento observa o comportamento da predição de consumo e o número de clientes que deixam de ser processados à medida que o requerimento mínimo de meses de consumo aumenta.

Foram analisadas 101.063 séries temporais de clientes de um município de brasileiro. São utilizadas quantidades incrementais em intervalos semestrais, formando um conjunto de clientes possuindo pelo menos 6, 12, 18, 24 e 30 meses de consumo. A base de dados é dividida de tal forma que o último mês é reservado para teste e os demais meses para treinamento. Foram utilizados os métodos ARIMA (NELSON, 1998), SES (HOLT, 2004), SGD (BOTTOU, 2012), HOLT (CHATFIELD, 1978), LSTM (Seção 2.3) e TCN (Seção 2.3), para comparar e avaliar o comportamento dos diferentes tipos de técnicas na predição de séries com variação de quantidade de dados. Em ambos a rede LSTM e TCN utilizam uma configuração de arquitetura simples, com uma camada de entrada e uma densa de saída. Por fim, para a avaliação do desempenho de predição foi utilizada a métrica SMAPE (Seção 2.5), visto que, esta possui uma maior capacidade de comparar dados em diferentes escalas. Para critério de comparação, os resultados aqui apresentados refletem o mesmo conjunto de clientes. Isso se dá, pois, à medida que o número de meses de treinamento aumenta, a quantidade de clientes disponíveis diminui.

5.1.1 Resultados

Para 6 meses de consumo houve um corte de 3.079 clientes, restando 97.984 séries para serem processadas nesse experimento. A Tabela 4 apresenta o resultado da métrica SMAPE para cada método de predição. O menor erro foi obtido pelo ARIMA, destacado na tabela em verde, com erro de 23,31%. Em laranja, pode-se ver o pior resultado de previsão, obtido pelo LSTM, com erro de 26,57%. Com 6 meses de consumo, há uma grande abrangência das séries que são estimadas, porém o erro nas predições é alto em todos os métodos, com métricas acima de 23%. Isso ocorre, pois neste cenário os modelos tem poucos dados para treinamento, o que consequentemente tende a aumentar os erros de predição.

Tabela 4 – Resultados de predição para 6 meses de consumo.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN
Comercial	31,76%	36,31%	39,55%	45,27%	38,85%	32,37%
Consumo Próprio	9,22%	9,63%	11,37%	8,48%	8,79%	10,38%
Iluminação Pública	19,45%	18,84%	20,16%	20,37%	21,80%	20,81%
Industrial	30,33%	32,14%	34,63%	32,73%	35,06%	31,34%
Poder Público	42,30%	38,13%	38,89%	36,68%	32,75%	43,86%
Residencial	22,29%	24,15%	26,39%	23,06%	26,76%	23,36%
Rural	23,75%	25,58%	27,47%	26,64%	26,09%	25,65%
Serviço Público	19,69%	19,37%	21,58%	22,39%	18,71%	19,24%
Total	23,31%	24,60%	26,25%	24,74%	26,57%	24,36%

Fonte: Acervo do autor.

Já a Tabela 5 mostra o resultado das predições para séries com 12 meses de consumo, destacando o menor erro em verde e o maior em laranja. Os resultados mostram que todos os métodos melhoraram a previsão, reduzindo-se o erro com o aumento da quantidade de consumos mensais utilizados no treinamento. Neste experimento, o ARIMA obteve uma pequena melhora na métrica SMAPE quando comparado com o último conjunto com 6 meses. Porém, no geral, o mesmo resultou na média mais alta de erros, 23,45%. Também pode-se observar que a rede TCN, neste cenário, obteve o menor erro dentre as demais técnicas de predição, com 21,36%. Quanto ao corte de clientes no experimento com 12 meses, 8.810 clientes não foram processados, totalizando uma base restante de 92.253.

O mesmo comportamento de redução gradativo dos erros de previsão pode ser percebido nas Tabelas 6 , 7 e 8, para os demais experimentos com 18, 24 e 30 meses, respectivamente. A partir de 18 meses, o TCN passa a ter o menor erro dentre todas as demais técnicas, com 19,48% para séries com 18 meses, 17,82% para 24 meses e 17,62% com 30 meses. Os melhores e piores resultados em cada um dos experimentos estão destacados em verde e laranja. Nesses mesmos experimentos, houve uma queda expressiva de desempenho do SGD, apresentando o maior erro nas estimações.

Quanto ao número de clientes que deixam de ser processados, para 18 meses, 13.741

Tabela 5 – Resultados de predição para 12 meses de consumo.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN
Comercial	34,93%	33,56%	34,62%	38,81%	33,77%	28,18%
Consumo Próprio	8,27%	8,65%	9,38%	7,48%	7,81%	8,54%
Iluminação Pública	18,37%	17,58%	19,74%	18,92%	18,51%	18,01%
Industrial	31,18%	29,83%	30,69%	28,37%	30,15%	27,26%
Poder Público	28,07%	35,18%	35,46%	32,58%	27,28%	38,42%
Residencial	22,57%	21,41%	23,38%	20,75%	22,28%	20,43%
Rural	23,36%	23,17%	24,48%	23,65%	22,49%	21,92%
Serviço Público	16,55%	17,62%	19,19%	19,47%	16,23%	16,79%
Total	23,45%	22,17%	23,36%	21,68%	22,78%	21,36%

Fonte: Acervo do autor.

Tabela 6 – Resultados de predição para 18 meses de consumo.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN
Comercial	35,56%	33,83%	33,11%	37,61%	33,55%	27,34%
Consumo Próprio	7,12%	8,38%	9,77%	7,35%	7,74%	8,24%
Iluminação Pública	16,76%	17,09%	18,69%	17,85%	17,78%	16,86%
Industrial	26,32%	29,42%	30,82%	28,36%	29,58%	26,81%
Poder Público	30,17%	33,82%	35,17%	32,77%	27,09%	37,87%
Residencial	19,12%	21,78%	22,66%	19,64%	21,87%	19,21%
Rural	21,06%	23,19%	23,56%	22,91%	21,93%	20,75%
Serviço Público	18,06%	17,72%	19,07	19,08%	16,01%	16,78%
Total	20,12%	22,02%	23,16%	20,93%	22,46%	19,48%

Fonte: Acervo do autor.

Tabela 7 – Resultados de predição para 24 meses de consumo.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN
Comercial	28,67%	33,75%	32,95%	37,44%	31,85%	26,94%
Consumo Próprio	8,12%	8,17%	9,48%	7,12%	7,76%	7,19%
Iluminação Pública	15,27%	16,48%	18,50%	17,36%	17,46%	16,79%
Industrial	25,46%	29,36%	30,65%	27,86%	29,38%	26,43%
Poder Público	29,38%	33,71%	35,50%	32,35%	26,89%	30,68%
Residencial	19,60%	21,49%	22,42%	19,17%	20,20%	17,76%
Rural	20,29%	22,79%	22,70%	22,92%	21,94%	19,86%
Serviço Público	15,86%	17,69%	18,97%	18,92%	15,87%	15,82%
Total	19,28%	21,87%	22,86%	20,88%	21,88%	17,82%

Fonte: Acervo do autor.

clientes deixaram de ser processados, totalizando uma base de 87.322. Já para 24 meses, 82.396 clientes foram processados, o equivalente à um corte de 18.667 séries de consumo. Por fim, o experimento com 30 meses de consumo gera um corte total de 21.475, sendo então processados 79.588 clientes. Na Figura 11 pode-se analisar o progresso de quantos novos clientes deixam de ser processados a cada novo corte. Inicialmente, partindo da base total, o corte inicial de 6 meses de consumo resultou em 3079 clientes que não foram processados. O próximo corte, de 12 meses, gerou um aumento de 5.731 clientes em relação

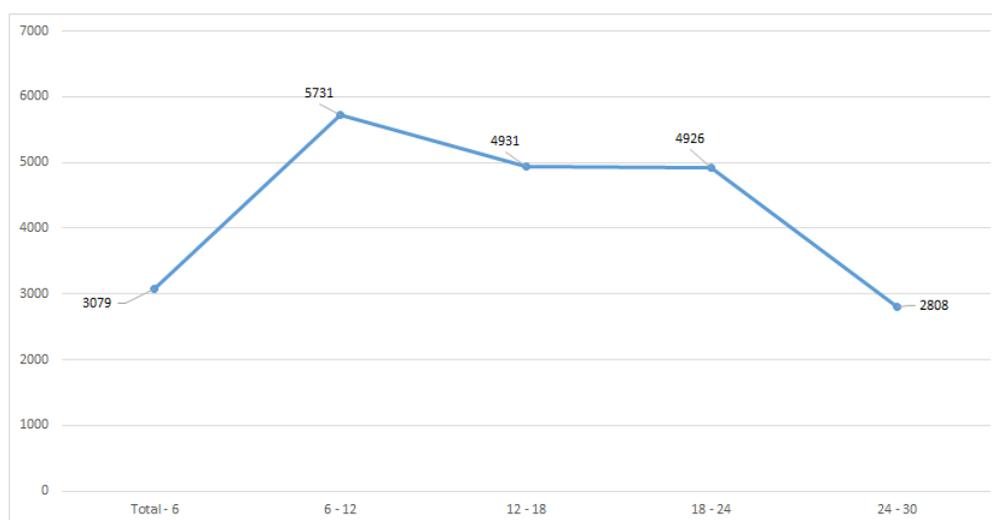
Tabela 8 – Resultados de predição para 30 meses de consumo.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN
Comercial	28,95%	30,35%	32,84%	36,83%	28,60%	26,61%
Consumo Próprio	7,44%	7,98%	9,03%	7,22%	5,33%	6,76%
Iluminação Pública	14,77%	15,85%	17,80%	17,03%	16,76%	14,81%
Industrial	25,08%	28,01%	30,36%	29,75%	29,01%	24,76%
Poder Público	30,15%	33,33%	35,22%	32,62%	28,54%	29,82%
Residencial	18,72%	19,98%	22,10%	19,87%	19,22%	16,93%
Rural	19,81%	20,57%	22,35%	22,95%	19,80%	19,14%
Serviço Público	15,18%	17,33%	18,66%	18,87%	15,46%	15,39%
Total	18,83%	20,62%	22,75%	21,44%	19,22%	17,62%

Fonte: Acervo do autor.

ao corte anterior e, a partir deste ponto em diante, esse número gradativamente diminui.

Figura 11 – Progresso do número de novos clientes deixam de ser processados pelos sucessivos cortes.

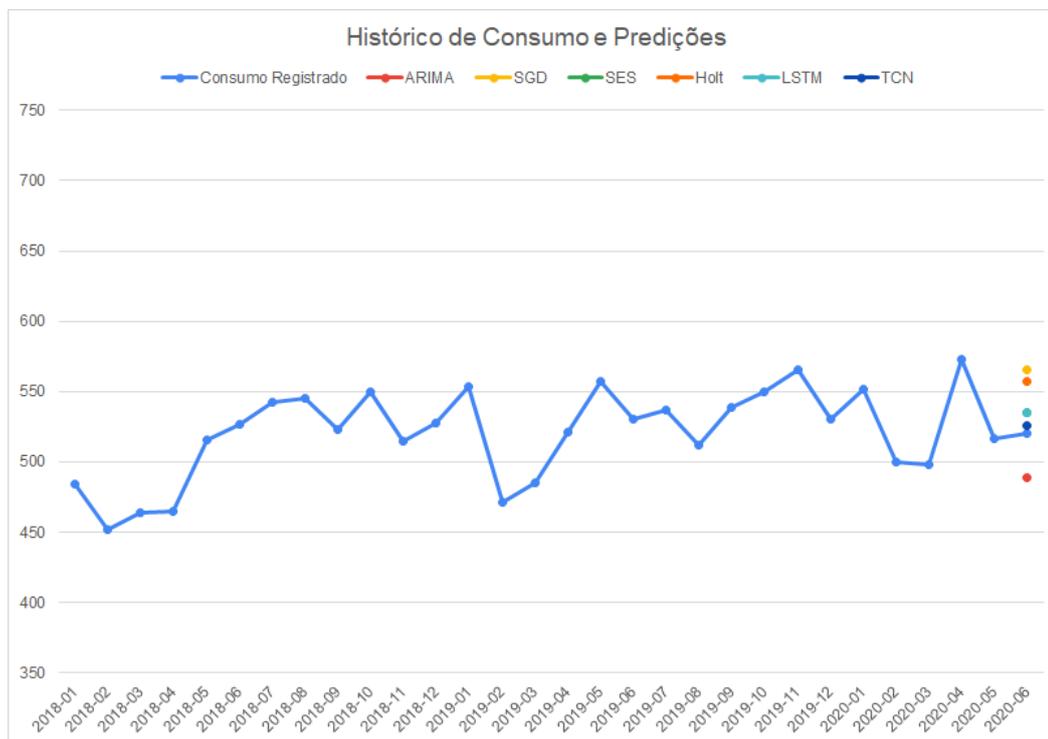


Fonte: Acervo do autor.

A Figura 12 apresenta o histórico de consumo de um cliente com série de tamanho 30 que foi estimado com sucesso. Analisando a série, pode-se perceber um comportamento de consumo centrado na faixa entre 450 kWh e 550 kWh, com consumo médio de 520,76 kWh e desvio padrão de 31,64 kWh. No último mês, pode-se ver o valor de consumo real a ser predito e o resultado obtido por cada um dos métodos comparados. No geral, todos obtiveram bons resultados, com predições que se aproximam do valor real. Para o 30º mês, o consumo real foi de 520 kWh e o método que mais se aproximou foi a rede TCN, que estimou 526 kWh, gerando um erro SMAPE de 0,57%. O pior resultado foi obtido pelo SGD, que previu 566 kWh, com erro de 4,23%, o que ainda é considerado um erro relativamente baixo.

Já na Figura 13 tem-se um exemplo de cliente com 30 meses de consumo que proporcionou uma maior dificuldade para os métodos de predição. Este cliente não possui

Figura 12 – Histórico de consumo de um cliente, um caso onde o último mês é estimado com sucesso por todos os métodos comparados.

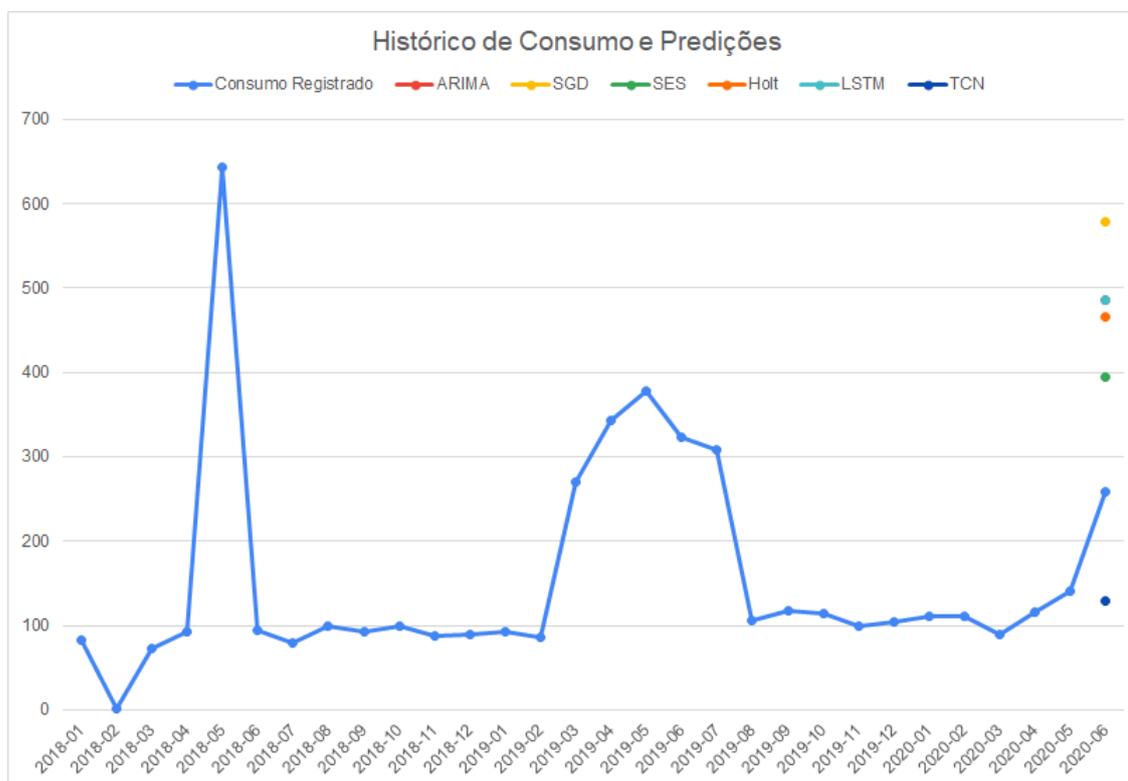


Fonte: Acervo do autor.

um padrão claro de consumo, o que dificulta a predição. A série de consumo possui uma média de consumo de 153,37 kWh e desvio padrão de 132,24 kWh. Neste caso, todos os métodos avaliados tiveram uma maior dificuldade de estimar o valor real. O método que mais se aproximou foi o SES, que estimou 394 kWh e obteve erro de SMAPE de 20,67%. Já o pior resultado foi obtido pela rede SGD, com uma estimativa de 578 kWh e erro de 38,11%. Em todos os casos, a predição gerada se distancia consideravelmente do valor de consumo real registrado, que foi de 259 kWh. Pode-se perceber que nos meses de Fevereiro e Abril de 2018 houve uma alta variação de consumo, saindo do padrão de consumo do cliente, e o mesmo voltou a ocorrer entre os meses de Março e Julho, onde o consumo do cliente dobrou em relação à média. Esses meses, nos quais o consumo registrado não acompanha o padrão do cliente, podem ter relação com o baixo desempenho dos métodos analisados.

Com os resultados obtidos nesse experimento, os cortes de 24 e 30 meses obtiveram os melhores resultados e, quanto menor o número de clientes processados, maior é o erro médio das predições. Isso confirma que à medida que há mais informações para treinamento, melhor o desempenho dos modelos. Já nesses dois casos, a quantidade de clientes processados ainda é expressiva, com 81,52% para 24 meses de consumo e 78,74% para 30 meses. Desta forma, ponderando os ganhos em desempenho e número de clientes que foram cortados, optou-se por utilizar o máximo de meses de consumo, visto que há

Figura 13 – Histórico de consumo de um cliente, um caso onde o último mês gera um alto erro na predição por todos os métodos comparados.



Fonte: Acervo do autor.

um impacto positivo nos resultados enquanto garante que a maior parte dos clientes sejam processados. Assim, os demais experimentos neste trabalho utilizam 30 meses de consumo como valor de corte para definição da base de treinamento.

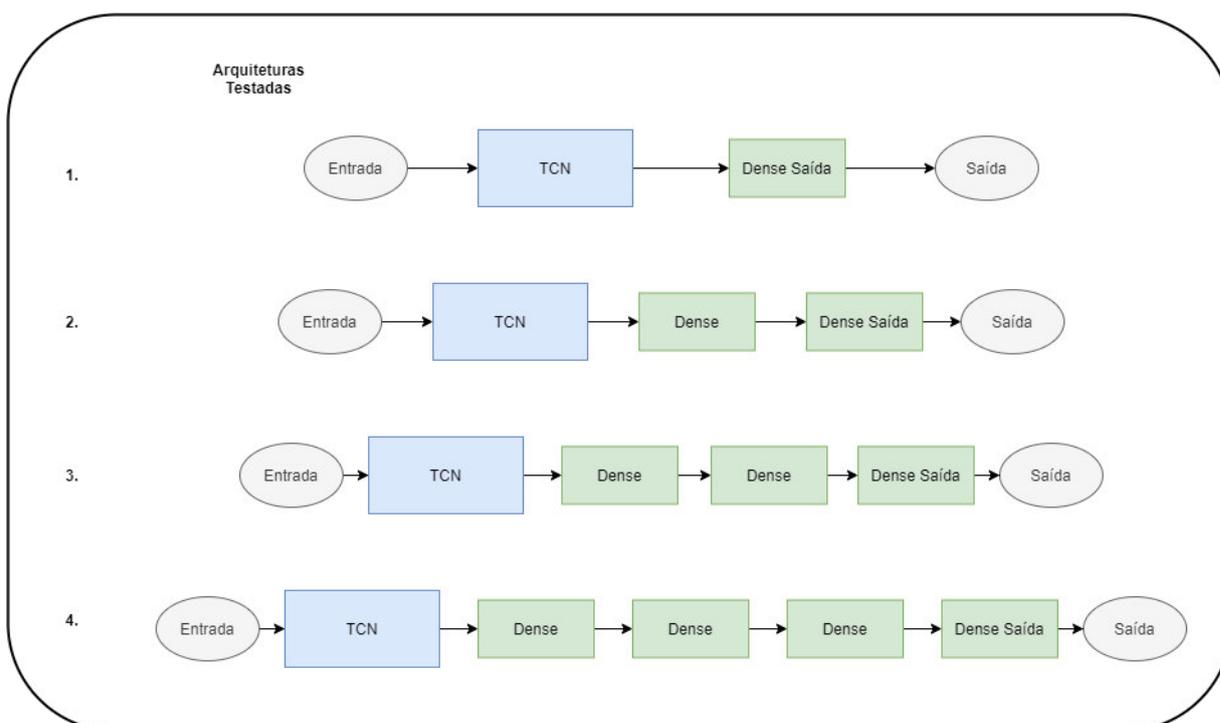
5.2 Experimento 2: Estudo do impacto das camadas densas

Houve a necessidade de identificar a configuração da arquitetura de rede que possui a melhor performance para a base de dados utilizada neste trabalho. Tinha-se como premissa que a arquitetura deveria ter no mínimo uma camada TCN e uma camada Densa de saída, porém também queria-se avaliar o impacto de outras camadas densas intermediárias para o desempenho dos modelos gerados. Assim, o experimento consistiu em aumentar o número de camadas densas entre as camadas TCN e a camada de saída.

Foram, então, avaliadas quatro configurações para a arquitetura dos modelos gerados: a primeira é composta da arquitetura mais simples onde há apenas uma camada TCN de entrada e uma Densa de saída que possui uma unidade; a segunda configuração parte do mesmo princípio da primeira porém inclui uma camada densa extra entre a camada TCN e a de saída; a terceira e quarta configurações seguem o mesmo padrão porém com duas e 3 camadas densas intermediárias. A Figura 14 apresenta graficamente a

representação de cada uma das configurações avaliadas. Para critérios de comparação em todos as diferentes configurações tem-se 128 unidades em cada camada Densa, sendo a única exceção a camada densa de saída, como mencionada anteriormente.

Figura 14 – Configurações de arquitetura testadas no experimento.



Fonte: Acervo do autor.

Para este experimento foram utilizadas 79.588 séries temporais de clientes de um município brasileiro, assim como o experimento descrito anteriormente (Seção 5.1). Foram utilizadas apenas séries que possuíam pelo menos 30 meses de consumo registrado. Por fim, a avaliação do desempenho é realizada utilizando a métrica SMAPE.

5.2.1 Resultados

A Tabela 9 apresenta os resultados obtidos pela utilização de cada uma das configurações descritas (Figura 14). A configuração número 3 obteve a menor média geral erro, com 15,98% de SMAPE, seguido da segunda configuração com 16,41%. Já a configuração 1, gerou o maior erro, com SMAPE médio de 17,62% e, por último, a configuração 4 obteve SMAPE de 16,88%. Quando analisado o erro médio por classe, pode-se perceber que a configuração 3 obteve um desempenho melhor em todas as classes que a configuração 2, porém, quando compara-se a configuração 4 com a configuração 1, observa-se que as classes: Comercial, Consumo Próprio, Poder Público, Rural e Serviço Público, geraram um erro maior. Apesar disso, dado que a classe Residencial corresponde à maior parte da base, resultou conseqüentemente em uma média geral inferior.

Assim, evidenciou-se, a partir da análise dos resultados, que o erro preditivo diminuiu à medida que as camadas densas foram adicionadas entre a camada TCN inicial e a camada densa de saída. No entanto, na configuração 4, na qual foram utilizadas três camadas densas intermediárias, percebeu-se que o ganho de desempenho deixou de ocorrer. Apesar disso, em todos os casos em que há camadas densas intermediárias houve redução no erro, quando comparados à configuração mais simples.

Tabela 9 – Comparativo entre o resultado das predições por classe para cada uma das configurações testadas. Em verde, destaca-se o melhor resultado obtido.

CLASSE	Config. 1	Config. 2	Config. 3	Config. 4
Comercial	26,61%	26,26%	25,59%	26,75%
Consumo Próprio	6,76%	6,74%	6,61%	6,93%
Iluminação Pública	14,81%	14,32%	14,17%	14,66%
Industrial	24,76%	24,64%	24,58%	24,72%
Poder Público	29,82%	29,67%	29,20%	30,22%
Residencial	16,93%	16,36%	15,93%	16,87%
Rural	19,14%	19,11%	19,03%	19,56%
Serviço Público	15,39%	15,35%	15,27%	15,42%
Total	17,62%	16,41%	15,98%	16,88%

Fonte: Acervo do autor.

Portanto, este experimento permitiu confirmar o efeito positivo que a inclusão das camadas densas teve ao erro gerado. Logo, considerando que o melhor resultado foi ocasionado com a utilização de duas camadas densas, definiu-se essa configuração a ser utilizada na geração dos modelos de predição.

5.3 Experimento 3: Aplicando o Método Proposto

Com os resultados obtidos nos experimentos descritos anteriormente foi então definido a configuração final do método proposto para este trabalho. Assim, esta seção apresenta os resultados alcançados ao aplicar o método proposto à base de dados completa, como mostrado no Capítulo 4. Nesta seção são apresentados os resultados obtidos na etapa de pré-processamento, onde são demonstrados os diferentes procedimentos realizados às séries temporais e seus efeitos. Na Otimização de Hiperparâmetros, descreve uma visão geral dos hiperparâmetros selecionados pelo método de otimização utilizado (Seção 2.4). Por fim, são apresentados os resultados gerais fornecidos pela predição, além de uma análise para validar se o método proposto entrega um desempenho preditivo satisfatório.

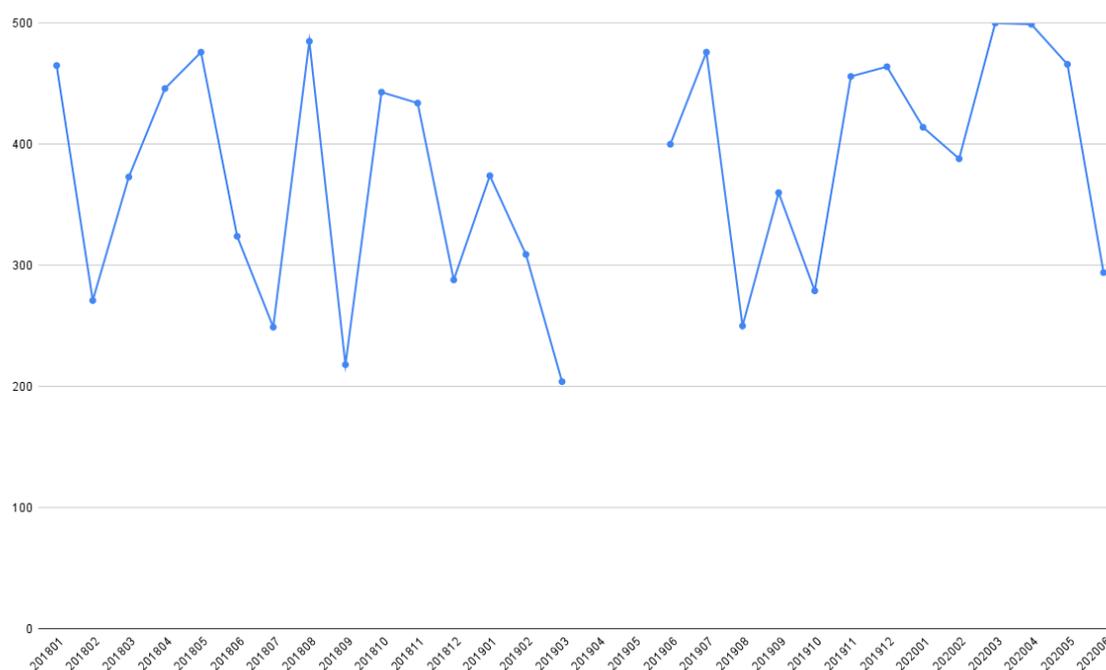
5.3.1 Resultados

Esta seção apresenta os resultados das etapas do método proposto, compostos pelos subtópicos: Pré-Processamento, Otimização de Hiperparâmetros e Predição.

5.3.1.1 Pré-processamento

Seguindo os passos do método proposto, na etapa de pré-processamento tem-se dois passos que são executados sob demanda, sendo eles o preenchimento de valores nulos e tratamento de anomalias. Primeiramente, na Figura 15 pode-se ver a série temporal de uma instalação denominada anonimamente de A. Neste exemplo, há a ausência de qualquer valor nos meses de Abril e Maio de 2019. Dada essa constatação, é aplicada então a interpolação linear com a finalidade de preencher essas lacunas com uma aproximação do que poderiam ser os valores reais.

Figura 15 – Histórico de consumo de um cliente, um caso onde o último mês gera um alto erro na predição por todos os métodos comparados.

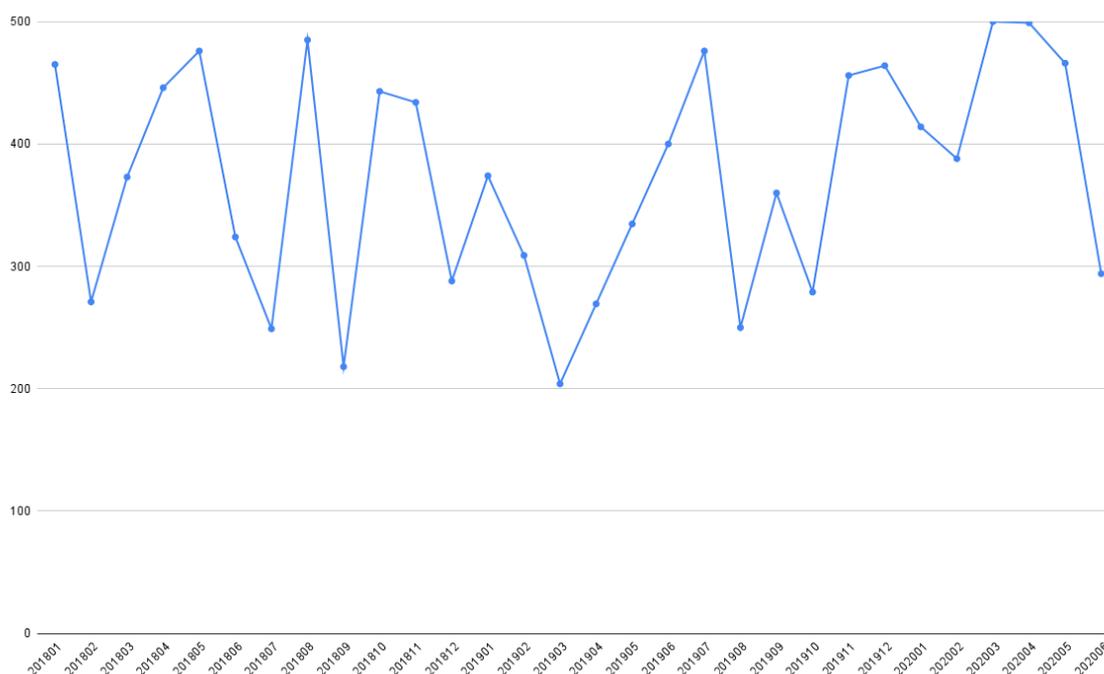


Fonte: Acervo do autor.

Com isso, a Figura 16 apresenta o resultado obtido ao aplicar a interpolação linear na instalação A. Pode-se ver que o procedimento realizado encontrou os pontos para os meses de Abril e Maio que fazem parte da reta determinada pelos pontos dos meses de Março e Junho. Isso gera uma melhor representação do comportamento de consumo de um cliente quando comparado com outras alternativas para preenchimento. Dentre as possibilidades, tem-se como as principais: a substituição por zero, média e até mesmo utilização de outros modelos preditivos. Aplicar um modelo mais específico apenas para esta tarefa, apesar de conseguir gerar uma melhor estimativa, também pode aumentar o tempo de processamento.

Já ilustrando os cenários onde há uma anomalia no histórico de consumo, tem-se a Figura 17 que apresenta a instalação B. Neste caso, a instalação possui um consumo

Figura 16 – Histórico de consumo da instalação A após interpolação linear utilizada para preencher os valores nulos.

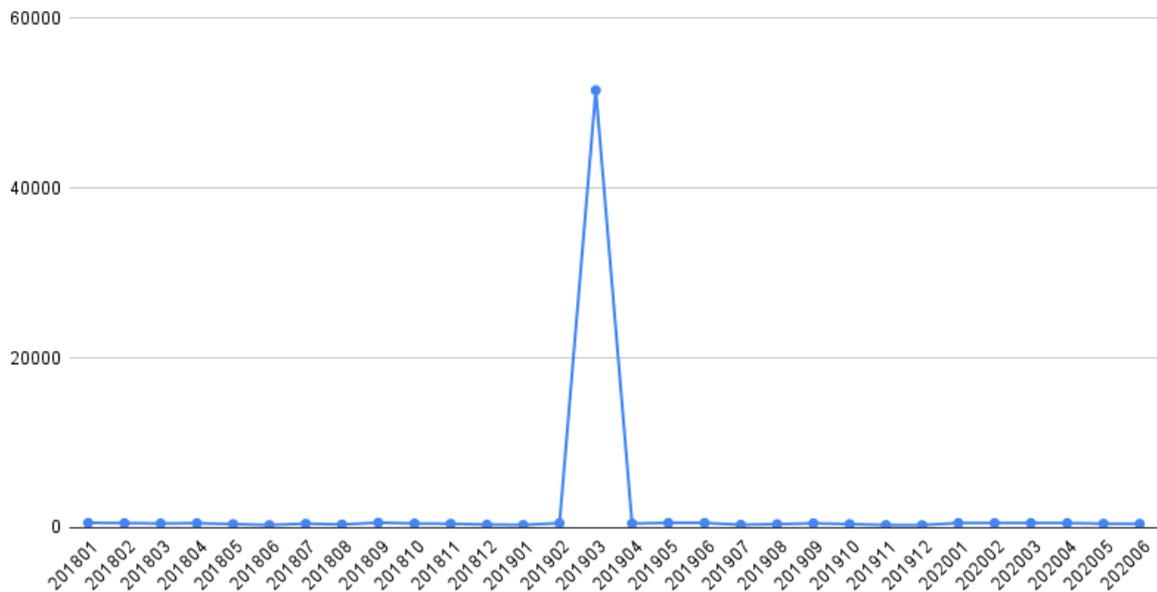


Fonte: Acervo do autor.

de 51.553 no mês de Março de 2019, valor este que ultrapassa o valor do limiar definido de 50.000 para instalações de baixa tensão. Desta forma, este registro é considerado uma anomalia e esses ruídos nos dados influenciam negativamente no desempenho dos modelos preditivos e por isso devem ser tratados. Na Figura 18 tem-se o resultado da remoção da anomalia na séries da instalação B. Como no preenchimento de valores nulos, é utilizada a interpolação linear para substituir a anomalia identificada. Realizando uma análise comparativa entre os dois gráficos da instalação B, fica evidente o impacto resultante da remoção da anomalia.

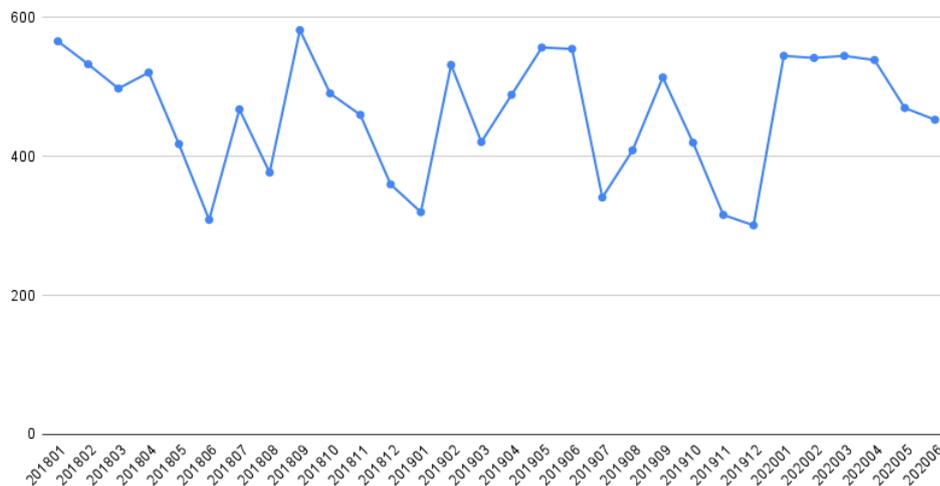
A última etapa de pré-processamento a ser realizada é a normalização da série, na qual os valores são representados em um padrão entre 0 e 1. A Figura 19 apresenta a série temporal da instalação C e na Figura 20 tem-se a mesma instalação C, após a aplicação da normalização. Comparando os dois gráficos pode-se identificar que com a normalização os picos e vales da série ficam evidenciados. Como também descritos nos trabalhos de Alshdaifat, Al-shdaifat e Aloqaily (2020), Pan, Zhuang e Fong (2016) e Nayak, Misra e Behera (2014), com a normalização as técnicas de predição tendem a ter uma maior facilidade a identificar padrões nas séries temporais.

Figura 17 – Histórico de consumo da instalação B que possui um valor que foge abruptamente do padrão de consumo.



Fonte: Acervo do autor.

Figura 18 – Histórico de consumo da instalação B após a remoção do registro de consumo considerado anomalia.

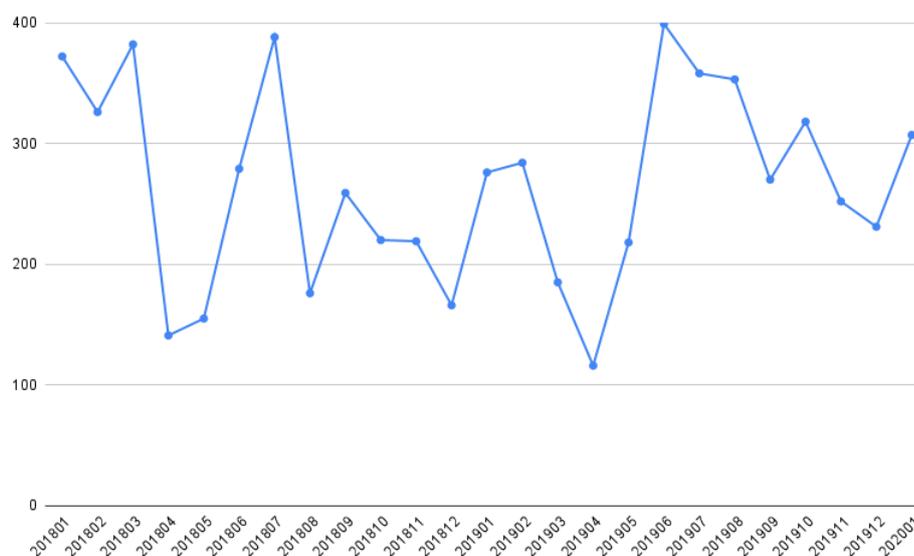


Fonte: Acervo do autor.

5.3.1.2 Otimização de Hiperparâmetros

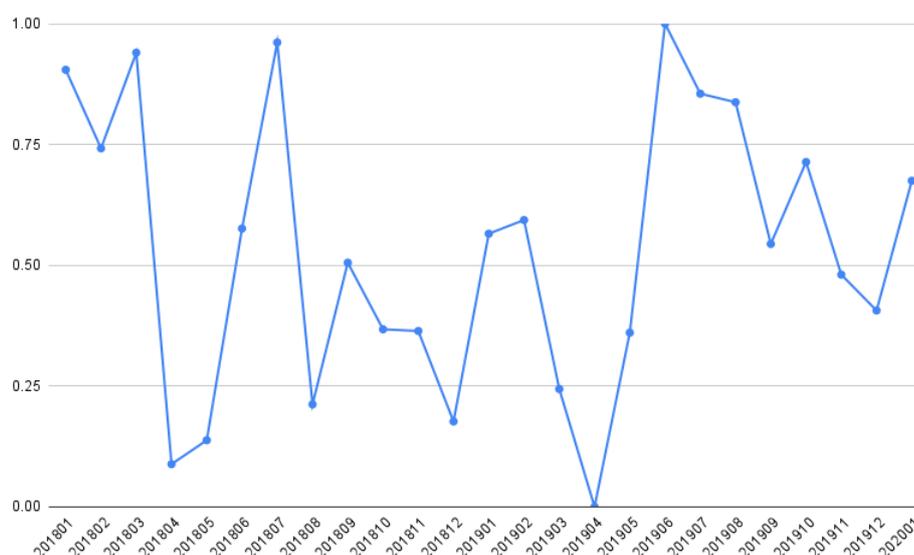
A aplicação desta etapa resulta num conjunto de hiperparâmetros que gera o melhor resultado para cada cliente processado. A análise dos resultados obtidos é realizada através do estudo dos hiperparâmetros mais selecionados durante o processo de otimização. A Tabela 10 apresenta o conjunto de hiperparâmetros que geraram os melhores resultados nos testes realizados. Iniciando pelos hiperparâmetros da camada TCN, tem-se o Dropout que teve o valor de 0,27 selecionado em 116.452 dos casos testados. Dentre esse quantitativo,

Figura 19 – Histórico de consumo da instalação C antes da normalização da série.



Fonte: Acervo do autor.

Figura 20 – Histórico de consumo da instalação C após a normalização da série.



Fonte: Acervo do autor.

corresponde, na sua maioria de clientes da classe Rural e Poder Público, com 23.685 e 17.803, respectivamente. Já a classe que menos utilizou este valor para Dropout foi a classe Residencial, com um total de 2.530, o que representa 2,17% do total. Todavia, quando analisado o fator de dilatação, o valor 6 foi mais vezes selecionado pela otimização realizada, totalizando 1.533.205. Neste caso, a proporção de seleções por classe representa bem a distribuição total da base de dados, com a classe Residencial, com um total de 82,13%, seguido das classes Comercial e Rural com 6,38% e 4,96%, respectivamente.

Já para as camadas densas intermediárias 1 e 2 foram obtidos os valores 128 e 32,

Tabela 10 – Relação dos valores de hiperparâmetros mais escolhidos pela otimização e suas respectivas quantidades.

Hiperparâmetro	Valor	Quantidade
Dropout	0,27	116.452
Fator de dilatação	6	1.533.205
Nº de unidades Densa 1	128	1.706.384
Nº de unidades Densa 2	32	1.139.046
<i>L1 kernel regularizer</i>	10^{-5}	701.584
<i>L2 kernel regularizer</i>	10^{-5}	829.129
<i>L1 bias regularizer</i>	10^{-4}	854.537
<i>L2 bias regularizer</i>	10^{-4}	933.537
<i>L1 activity regularizer</i>	10^{-4}	1.295.483
<i>L2 activity regularizer</i>	10^{-3}	1.035.815

Fonte: Acervo do autor.

respectivamente, como os os mais selecionados. Na camada densa 1, o valor de 128 foi selecionado 1.706.384 vezes pelo método de otimização. Deste total, 1.102.154 ou 64,59% eram da classe Residencial. Para a camada densa 2, o valor selecionados mais vezes foi 32, totalizando 1.139.046 dos clientes otimizados. Na camada Densa 2, mais uma vez a classe Residencial foi responsável pela maior parte do clientes que obtiveram o melhor resultado com 32 para o número de unidades da camada densa 2. Assim como esperado, este comportamento volta a ocorrer em todas os demais hiperparâmetros, uma vez que, por ser consideravelmente maior que todas as demais classes juntas, a classe Residencial tende a naturalmente se sobressair nas estatísticas gerais.

Quanto aos *kernel regularizer* L1 e L2, em ambos 10^{-5} foi o valor selecionado pela otimização na maioria das vezes. Já para os hiperparâmetros *bias regularizer* L1 e L2 e *activity regularizer*. L1 o valor de 10^{-4} , gerou os melhores resultados na maioria dos clientes processados. Por fim, para o *activity regularizer* L2 a otimização identificou que o valor 10^{-3} gerou os melhores resultados para 1.035.815 das instalações testadas. Como mencionado anteriormente, em todos esses hiperparâmetros a classe Residencial compreendia a maior parte dos casos.

5.3.1.3 Predição

Os resultados finais do método proposto são mensurados utilizando as métricas MAE, MAPE e SMAPE. A Tabela 11 apresenta o MAE para cada uma das classes obtido pelos métodos avaliados, destacando-se em verde os melhores resultados para cada classe. A metodologia proposta alcançou o menor erro médio em 4 das 8 classes disponível, sendo elas: Comercial, Industrial, Poder Público e Rural. Já as classes, Consumo Próprio e Serviço Público, tiveram o seu melhor resultado com a rede LSTM. Para a classe Iluminação Pública, o método com o melhor desempenho foi o ARIMA e para a classe Residencial o método SGD superou as demais opções. Como mostra a tabela, a média de erro é

consideravelmente alta, porém esta análise individual não fornece informação suficiente para identificar se a predição teve ou não uma boa performance. Isso se dá por que algumas dessas classes possuem uma grande média de consumo, como a Industrial. Assim, um erro de mais de 10.000 kWh em uma classe de altos valores de consumo passa a ser aceitável, como na classe Industrial. Ao mesmo tempo, um erro deste nível em uma classe com baixa média de consumo é indicativo de baixa qualidade na estimação. Analisando a última linha da tabela, pode-se ver que o menor erro absoluto médio total foi obtido pelo método proposto. Isso acontece, uma vez que, apesar de não ter alcançado o melhor resultado na classe Residencial, esta possui uma baixa média de consumo, principalmente quando comparado com outras classes com alta média de consumo. Assim, obter um bom equilíbrio na métrica MAE de cada classe foi o fator decisivo para que a média geral também fosse a menor.

Tabela 11 – Resultado da métrica MAE para as predições para cada uma das classes aplicando a metodologia proposta.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN	Proposta
Comercial	2.420.019,12	2.767.252,73	396.193,34	979.723,51	1.147.257,35	3.264,77	3.257,85
Consumo Próprio	5.498,90	5.946,78	2.888,53	3.562,97	2.234,46	3.790,91	3.395,24
Iluminação Pública	283,14	312,16	317,15	545,31	309,17	585,83	525,57
Industrial	94.542.635,91	99.434.500,64	66.030.758,82	59.380.634,48	69.537.681,84	181.024,16	179.683,11
Poder Público	15.676.736,76	15.945.655,78	14.109.361,82	12.461.043,88	14.589.805,79	53.710,22	46.715,43
Residencial	85,37	88,15	69,23	124,14	72,72	92,97	85,03
Rural	4.921.802,35	5.073.349,77	4.038.519,31	4.017.035,34	4.169.442,58	3.128.740,89	312.493,73
Serviço Público	11.621.336,34	11.647.788,89	11.467.159,57	13.009.214,19	11.414.513,78	13.282.821,53	13.253.947,33
Total	942.563,38	993.468,44	645.866,96	640.859,43	729.407,41	124.764,03	123.657,49

Fonte: Acervo do autor.

Já a Tabela 12 apresenta os resultados para a métrica MAPE obtido em cada uma das classes. Neste caso, olhando pela perspectiva do erro percentual, a situação muda um pouco. O método proposto obteve o melhor resultado nas classes: Comercial, Consumo Próprio, Poder Público, Residencial e Rural. Já nas demais classes: Iluminação Pública, Industrial e Rural, o menor erro foi gerado pelo ARIMA. Isso se dá parcialmente uma vez que essas classes possuem características sazonais e o ARIMA tem bom desempenho nesse tipo de série, o que acontece por que o mesmo captura a tendência e sazonalidade dos dados. Ainda que o método proposta tenha obtido o melhor resultado na principal classe, Residencial, é importante destacar que nas classes onde a proposta não obteve o melhor resultado o mesmo não foi a pior opção. Na classe Rural, apesar do ARIMA ter obtido o melhor resultado, a abordagem proposta gerou um erro percentual muito próximo, com uma diferença de aproximadamente 0.04%. Desta forma, a menor média geral também foi alcançada pelo método proposto, com 19,25% de erro percentual.

Baseando-se no trabalho de Yorucu (2003), em predições com MAPE com porcentagem abaixo de 10% são consideradas extremamente precisas, predições entre 10% e 20% são vistas como boas, predições entre 20% e 50% são consideradas razoáveis, e por fim, predições com mais 50% são consideradas como imprecisas. Levando isso em consideração, o resultado geral obtido pelo método proposto pode ser considerado como bom.

Tabela 12 – Resultado da métrica MAPE para as previsões para cada uma das classes aplicando o método proposto.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN	Proposta
Comercial	31,56%	31,24%	32,22%	37,57%	29,85%	27,96%	27,34%
Consumo Próprio	7,57%	9,71%	9,18%	8,54%	6,49%	6,89%	6,35%
Iluminação Pública	18,47%	21,60%	21,58%	20,13%	22,56%	22,69%	21,89%
Industrial	23,53%	29,19%	31,75%	26,38%	25,22%	25,73%	25,53%
Poder Público	28,78%	32,75%	32,58%	33,83%	31,67%	29,44%	28,48%
Residencial	26,65%	24,33%	25,74%	23,81%	24,15%	22,77%	20,40%
Rural	26,58%	31,16%	27,49%	28,47%	27,49%	26,84%	26,62%
Serviço Público	18,45%	22,16%	21,74%	20,26%	19,67%	18,63%	16,11%
Total	24,63%	22,38%	25,94%	22,75%	23,63%	22,21%	19,25%

Fonte: Acervo do autor.

Por fim, na Tabela 13 tem-se os resultados avaliados com a métrica SMAPE. O método proposto gerou o menor erro preditivo em 6 das 8 classes disponíveis, sendo elas: Comercial, Iluminação Pública, Industrial, Residencial, Rural e Serviço Público. As únicas classes que não foram melhor preditas pela proposta apresentada foram: Consumo Próprio e Poder Público, onde a LSTM obteve o melhor desempenho. Tendo isso em vista, o método proposto foi a segunda melhor opção em ambas as classes, e a diferença bruta para a LSTM é de aproximadamente 0,62% na classe Consumo Próprio e de 0,05% na classe Poder Público. Para esta última métrica, o método proposto também obteve a menor média total, 16,86% de erro. Isso se dá por que há uma maior importância ao usar métricas de erro percentual para obter um bom resultado nas classes que compõem a maior parte da base.

Tabela 13 – Resultado da métrica SMAPE para as previsões para cada uma das classes aplicando a metodologia proposta.

CLASSE	ARIMA	SES	SGD	Holt	LSTM	TCN	Proposta
Comercial	28,94%	30,34%	32,83%	36,82%	28,60%	26,61%	25,73%
Consumo Próprio	7,44%	7,98%	9,02%	7,21%	5,33%	6,75%	5,95%
Iluminação Pública	14,77%	15,85%	17,80%	17,03%	16,75%	14,81%	14,19%
Industrial	25,08%	28,00%	30,35%	29,74%	29,01%	24,76%	23,89%
Poder Público	30,14%	33,32%	35,21%	32,61%	28,53%	29,82%	29,48%
Residencial	18,72%	19,97%	22,09%	19,86%	19,22%	17,23%	16,73%
Rural	19,80%	20,57%	22,35%	22,95%	19,80%	19,14%	18,97%
Serviço Público	15,18%	17,33%	18,66%	18,86%	15,46%	15,38%	15,12%
Total	18,83%	20,62%	22,74%	21,43%	19,22%	18,02%	16,86%

Fonte: Acervo do autor.

Essas constatações evidenciam que a metodologia consegue entregar resultados satisfatórios em todas as classes. Isso fica ainda mais evidente quando analisamos os mesmos resultados da Tabela 13, comparando apenas a rede TCN com a proposta apresentada. Pode-se ver que a aplicação do método desenvolvido resultou em uma redução no erro nas previsões em todas as classes e, conseqüentemente, na média total, onde houve uma melhoria de 1.16%.

6 CONCLUSÃO

Este trabalho apresentou um novo método para predição de consumo mensal individual de energia elétrica. O método apresentado aplicou uma rede TCN para a estimação do consumo elétrico individual por instalação. O mesmo obteve bons resultados quando comparados com outros métodos preditivos já difundidos na literatura. Visando melhorar alguns experimentos com o objetivo de otimizar a definição do método utilizado. Um dos experimentos avaliou as possíveis variações de desempenho à medida que o número de meses de consumo registrado nas séries aumentava. Os resultados obtidos apontaram que havia um ganho notório de desempenho com o aumento das séries temporais e isso auxiliou a definir que, no método proposto o modelo apenas operaria com séries com 30 meses de consumo, visando maximizar a sua capacidade preditiva. O segundo experimento, visou identificar a quantidade ótima de camadas Densas que a arquitetura da rede proposta deve possuir. Neste, foi possível observar que à medida que mais camadas densas foram adicionadas, o desempenho preditivo do modelo também aumentou. Contudo, com a adição da terceira camada densa intermediária percebeu-se que não houve mais qualquer ganho de predição. Logo, isso possibilitou definir que a arquitetura de rede do método proposto é formada com duas camadas densas entre a camada TCN de entrada e a camada de saída.

O método proposto incluiu também uma etapa de Otimização de Hiperparâmetros para identificar o conjunto de hiperparâmetros que maximiza o poder preditivo do modelo gerado. Esta etapa se mostrou eficaz, sendo capaz de reduzir o erro para todas as classes quando comparada com a TCN originalmente testada sem qualquer otimização. Analisando sob uma ótica geral, o método proposto obteve a menor média de erro SMAPE em 6 das 8 classes existentes. Conseqüentemente, a proposta descrita alcançou a melhor média geral, com 16,86% de erro.

Dentre as etapas do método proposto, o pré-processamento se mostrou de grande relevância no tratamento dos dados utilizados. Isso se deu principalmente pela existência de diversos casos de dados com valores nulos e/ou ruidosos. Outro ponto que se destacou durante a análise dos resultados foi a importância que obter um bom resultado na classe Residencial causa na média geral. Para a TCN com e sem otimização de hiperparâmetros houve uma clara vantagem em relação aos demais modelos preditivos testados, visto que, eles obtiveram os menores erros médios para a classe Residencial. Nestes casos, a média geral tende a se aproximar da classe Residencial.

Desta forma, analisando o objetivo geral deste trabalho, acredita-se que o mesmo foi alcançado, visto que, a proposta apresentada obteve resultados que condizem com a tarefa de predição mensal de consumo. O mesmo vale-se dos objetivos específicos. Além

disso, destaca-se também a etapa de pré-processamento e os mecanismos de tratamento de dados nele utilizados como uma contribuição que não foi inicialmente objetivada por este trabalho, mas que foi de grande importância para os resultados obtidos.

6.1 Trabalhos Futuros

Ficou evidente também que apesar ser uma área de estudo bastante explorada, há diversas possibilidades a serem avaliadas que podem acarretar em melhorias nos resultados obtidos. Assim, destacam-se alguns pontos para serem experimentados como trabalhos futuros e complemento a este trabalho: (1) Avaliar a escalabilidade do método proposto em bases de dados do histórico mensal de outras companhias de energia elétrica; (2) Avaliar o desempenho da proposta apresentada em bases que registram o consumo elétrico por dia, hora e até minutos; (3) Estender a comparação com uma gama maior de métodos preditivos; (4) Realizar experimentos com outros hiperparâmetros que não foram testados; (5) Realizar experimentos com uma maior quantidade de meses de consumo registrado para a etapa de treinamento; (6) Analisar outros métodos de imputação de dados. Acredita-se que os itens 1, 2 e 3 são de suma importância para complementar e enriquecer ainda mais este trabalho. Já os itens 4 e 5 são pontos que espera-se que tenham impacto positivo no desempenho do método proposto gerando, assim, erros ainda menores.

Referências

- ALSHDAIFAT, E.; AL-SHDAIFAT, A. Z. A.; ALOQAILY, A. The impact of data normalization on predicting student performance: A case study from hashemite university. *International Journal*, v. 9, n. 4, 2020. Citado na página 48.
- ARMSTRONG, J. S.; FORECASTING, L.-R. From crystal ball to computer. *New York ua*, 1985. Citado na página 27.
- BAI, S.; KOLTER, J. Z.; KOLTUN, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. Citado 3 vezes nas páginas 23, 25 e 36.
- BAR-JOSEPH, Z.; GITTER, A.; SIMON, I. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, Nature Publishing Group, v. 13, n. 8, p. 552–564, 2012. Citado na página 18.
- BAYER, D. M.; CASTRO, N. d. R.; BAYER, F. M. Modelagem e previsão de vazões médias mensais do rio potiribu utilizando modelos de séries temporais. *Revista brasileira de recursos hídricos*, v. 17, n. 2, p. 229–239, 2012. Citado na página 18.
- BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, v. 13, n. 2, 2012. Citado na página 25.
- BERGSTRA, J.; YAMINS, D.; COX, D. D. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In: CITESEER. *Proceedings of the 12th Python in science conference*. [S.l.], 2013. p. 13–20. Citado na página 36.
- BERGSTRA, J. S. et al. Algorithms for hyper-parameter optimization. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2011. p. 2546–2554. Citado na página 26.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.: s.n.], 1992. p. 144–152. Citado na página 19.
- BOTTOU, L. Stochastic gradient descent tricks. In: *Neural networks: Tricks of the trade*. [S.l.]: Springer, 2012. p. 421–436. Citado na página 39.
- BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015. Citado na página 19.
- CHATFIELD, C. The holt-winters forecasting procedure. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Wiley Online Library, v. 27, n. 3, p. 264–279, 1978. Citado na página 39.
- CHO, K. et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. Citado na página 20.
- CHOLLET, F. et al. *Keras*. 2015. Disponível em: <<https://keras.io>>. Acessado em: 29 Janeiro 2021. Citado na página 37.

- DAVIS, H. T. *Analysis of economic time series*. [S.l.]: Principia Press, Bloomington, 1941. Citado na página 18.
- DEWANCKER, I.; MCCOURT, M.; CLARK, S. Bayesian optimization primer. URL https://app.sigopt.com/static/pdf/SigOpt_Bayesian_Optimization_Primer.pdf, 2015. Citado na página 27.
- DONG, M.; GRUMBACH, L. A hybrid long-term load forecasting model for distribution feeder peak demand using lstm neural network. *arXiv preprint arXiv:1812.04480*, 2018. Citado na página 29.
- DUCHON, C.; HALE, R. *Time series analysis in meteorology and climatology: an introduction*. [S.l.]: John Wiley & Sons, 2012. v. 7. Citado na página 18.
- EHLERS, R. S. Análise de séries temporais. *Laboratório de Estatística e Geoinformação. Universidade Federal do Paraná*, v. 1, p. 1–118, 2007. Citado na página 18.
- ELMAN, J. L. Finding structure in time. *Cognitive science*, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990. Citado na página 20.
- ENRIGHT, J. T. The search for rhythmicity in biological time-series. *Journal of theoretical Biology*, Elsevier, v. 8, n. 3, p. 426–468, 1965. Citado na página 18.
- FRAZIER, P. I. Bayesian optimization. In: *Recent Advances in Optimization and Modeling of Contemporary Problems*. [S.l.]: INFORMS, 2018. p. 255–278. Citado na página 26.
- FRIEDMAN, J. et al. *The elements of statistical learning*. [S.l.]: Springer series in statistics New York, 2001. v. 1. Citado na página 25.
- GONELLA, J. A rotary-component method for analysing meteorological and oceanographic vector time series. In: ELSEVIER. *Deep Sea Research and Oceanographic Abstracts*. [S.l.], 1972. v. 19, n. 12, p. 833–846. Citado na página 18.
- GOOIJER, J. G. D.; HYNDMAN, R. J. 25 years of time series forecasting. *International journal of forecasting*, Elsevier, v. 22, n. 3, p. 443–473, 2006. Citado na página 19.
- GRANGER, C. W. J.; NEWBOLD, P. *Forecasting economic time series*. [S.l.]: Academic Press, 2014. Citado na página 18.
- HAN, Z. et al. A review of deep learning models for time series prediction. *IEEE Sensors Journal*, IEEE, 2019. Citado na página 19.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 24.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 20.
- HOLLAND, J. H. Genetic algorithms. *Scientific american*, JSTOR, v. 267, n. 1, p. 66–73, 1992. Citado na página 25.
- HOLT, C. C. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, Elsevier, v. 20, n. 1, p. 5–10, 2004. Citado na página 39.

- HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Sequential model-based optimization for general algorithm configuration. In: SPRINGER. *International conference on learning and intelligent optimization*. [S.l.], 2011. p. 507–523. Citado na página 26.
- JORDAN, M. I. Serial order: A parallel distributed processing approach. In: *Advances in psychology*. [S.l.]: Elsevier, 1997. v. 121, p. 471–495. Citado na página 20.
- JR, E. S. G. Exponential smoothing: The state of the art. *Journal of forecasting*, Wiley Online Library, v. 4, n. 1, p. 1–28, 1985. Citado na página 19.
- JÚNIOR, W. J. de A. Métodos de otimização hiperparamétrica: um estudo comparativo utilizando árvores de decisão e florestas aleatórias na classificação binária. Universidade Federal de Minas Gerais, 2018. Citado na página 25.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Proceedings of ICNN'95-international conference on neural networks*. [S.l.], 1995. v. 4, p. 1942–1948. Citado na página 25.
- KIM, S.; KIM, H. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, Elsevier, v. 32, n. 3, p. 669–679, 2016. Citado na página 28.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *science*, American association for the advancement of science, v. 220, n. 4598, p. 671–680, 1983. Citado na página 25.
- KONG, W. et al. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, IEEE, v. 10, n. 1, p. 841–851, 2017. Citado 3 vezes nas páginas 29, 30 e 31.
- KONG, W. et al. Effect of automatic hyperparameter tuning for residential load forecasting via deep learning. In: IEEE. *2017 Australasian Universities Power Engineering Conference (AUPEC)*. [S.l.], 2017. p. 1–6. Citado 3 vezes nas páginas 29, 30 e 31.
- LANG, K. J.; WAIBEL, A. H.; HINTON, G. E. A time-delay neural network architecture for isolated word recognition. *Neural networks*, Elsevier, v. 3, n. 1, p. 23–43, 1990. Citado na página 19.
- LANGE, K.; HUNTER, D. R.; YANG, I. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, Taylor & Francis, v. 9, n. 1, p. 1–20, 2000. Citado na página 26.
- LARA-BENÍTEZ, P. et al. Temporal convolutional networks applied to energy-related time series forecasting. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 7, p. 2322, 2020. Citado 3 vezes nas páginas 29, 30 e 31.
- LI, S. et al. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 5457–5466. Citado na página 20.
- LI, T. et al. Short-term load forecasting using optimized lstm networks based on emd. *arXiv preprint arXiv:1809.10108*, 2018. Citado 2 vezes nas páginas 29 e 30.

- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3431–3440. Citado na página 23.
- LU, J.-C.; NIU, D.-X.; JIA, Z.-Y. A study of short-term load forecasting based on arima-ann. In: IEEE. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*. [S.l.], 2004. v. 5, p. 3183–3187. Citado na página 29.
- MARITZ, J. S.; LWIN, T. *Empirical bayes methods*. [S.l.]: Routledge, 2018. Citado na página 29.
- MEDSKER, L. R.; JAIN, L. Recurrent neural networks. *Design and Applications*, CRC press, v. 5, 2001. Citado na página 19.
- MYTTENAERE, A. D. et al. Mean absolute percentage error for regression models. *Neurocomputing*, Elsevier, v. 192, p. 38–48, 2016. Citado na página 27.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814. Citado na página 24.
- NAYAK, S.; MISRA, B. B.; BEHERA, H. S. Impact of data normalization on stock index forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, v. 6, n. 2014, p. 257–269, 2014. Citado na página 48.
- NELSON, B. K. Time series analysis using autoregressive integrated moving average (arima) models. *Academic emergency medicine*, Wiley Online Library, v. 5, n. 7, p. 739–744, 1998. Citado na página 39.
- NOOR, N. M. et al. Comparison of linear interpolation method and mean method to replace the missing values in environmental data set. In: TRANS TECH PUBL. *Materials Science Forum*. [S.l.], 2015. v. 803, p. 278–281. Citado na página 34.
- OLAH, C. Understanding lstm networks. 2015. Citado 3 vezes nas páginas 20, 21 e 22.
- OORD, A. v. d. et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. Citado 2 vezes nas páginas 23 e 24.
- PALIT, A. K.; POPOVIC, D. *Computational intelligence in time series forecasting: theory and engineering applications*. [S.l.]: Springer Science & Business Media, 2006. Citado na página 18.
- PAN, J.; ZHUANG, Y.; FONG, S. The impact of data normalization on stock market prediction: using svm and technical indicators. In: SPRINGER. *International Conference on Soft Computing in Data Science*. [S.l.], 2016. p. 72–88. Citado na página 48.
- QUINTANILHA, D. B. P. et al. Deteccao automatica de medidores elétricos em imagens utilizando uma combinacao de svm e cnn. 2017. Citado na página 14.
- RAUDYS, S. J.; JAIN, A. K. et al. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, v. 13, n. 3, p. 252–264, 1991. Citado na página 39.

- RODRIGUES, A. et al. Reducing power companies billing costs via empirical bayes and seasonality remover. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 81, p. 387–396, 2019. Citado 2 vezes nas páginas 29 e 31.
- RUMMELHART, D. E. et al. *Parallel Distributed Processing: explorations in the microstructure of cognition*. [S.l.]: MIT press Cambridge, 1986. Citado na página 19.
- SALGOTRA, R.; GANDOMI, M.; GANDOMI, A. H. Time series analysis and forecast of the covid-19 pandemic in india using genetic programming. *Chaos, Solitons & Fractals*, Elsevier, v. 138, p. 109945, 2020. Citado na página 18.
- SALIMANS, T.; KINGMA, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2016. p. 901–909. Citado na página 24.
- SAVI, M. A. *Dinâmica não-linear e caos*. [S.l.]: Editora E-papers, 2006. Citado na página 19.
- SINGH, S. P. et al. Machine translation using deep learning: An overview. In: IEEE. *2017 international conference on computer, communications and electronics (comptelix)*. [S.l.], 2017. p. 162–167. Citado na página 19.
- SORJAMAA, A. et al. Methodology for long-term prediction of time series. *Neurocomputing*, Elsevier, v. 70, n. 16-18, p. 2861–2869, 2007. Citado na página 19.
- SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 24.
- WANG, C. et al. Image captioning with deep bidirectional lstms. In: *Proceedings of the 24th ACM international conference on Multimedia*. [S.l.: s.n.], 2016. p. 988–997. Citado na página 19.
- WEBBY, R.; O’CONNOR, M. Judgemental and statistical time series forecasting: a review of the literature. *International Journal of forecasting*, Elsevier, v. 12, n. 1, p. 91–118, 1996. Citado na página 19.
- WEI, W. W. Time series analysis. In: *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. [S.l.: s.n.], 2006. Citado na página 18.
- WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, v. 30, n. 1, p. 79–82, 2005. Citado na página 27.
- WRIGHT, R. E. Logistic regression. American Psychological Association, 1995. Citado na página 19.
- YORUCU, V. The analysis of forecasting performance by using time series data for two mediterranean islands. *Review of Social, Economic & Business Studies*, v. 2, p. 175–196, 2003. Citado na página 52.
- YU, F.; KOLTUN, V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. Citado na página 23.

ZHANG, Y.; JIANG, Y.; TONG, Y. Study of sentiment classification for chinese microblog based on recurrent neural network. *Chinese Journal of Electronics*, IET, v. 25, n. 4, p. 601–607, 2016. Citado na página [19](#).