
**ANÁLISE COMPARATIVA DE MODELOS DE MACHINE LEARNING NA
PREDIÇÃO DE DEMANDA PARA INSTALAÇÃO DE VARAIS**

**COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS IN PREDICTING
DEMAND FOR CLOTHESLINES INSTALLATION**

Bruno Grizzo Marchese¹

Bruna Sozzo²

RESUMO

Este artigo tem como objetivo comparar a acurácia do modelo de machine learning Prophet e da rede neural recorrente Long Short-Term Memory (LSTM) na predição de demanda de um serviço de instalação de varais com base em um histórico de instalações realizadas no período de 2004 ao fim de 2021. Para realizar esta avaliação, são criados modelos LSTM e Prophet para a predição da demanda a partir das informações obtidas. Por fim, são analisadas as métricas de desempenho dos modelos, como o Erro Quadrático Médio (MSE), a Raiz Quadrada do Erro Médio (RMSE) e o Erro Médio Absoluto (MAE), a fim de comparar a acurácia dos dois métodos.

209

Palavras-chave: machine learning; long short-term memory; prophet; predição de demanda.

ABSTRACT

This article aims to compare the accuracy of the Prophet machine learning model and the Long Short Term Memory (LSTM) recurrent neural network in predicting the demand for a clothesline installation service based on a historical record of installations from 2004 to the end of 2021. To conduct this evaluation, LSTM and Prophet models are created to forecast the demand based on the gathered data. Finally, performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are analyzed to compare the accuracy of the two methods.

Keywords: machine learning; long short-term memory; prophet; demand forecast.

INTRODUÇÃO

A predição de demanda é uma atividade crucial para empresas que buscam

¹ Centro Universitário Filadélfia de Londrina - UniFil

² Centro Universitário Filadélfia de Londrina - UniFil

otimizar seus processos e recursos, pois a estimativa correta da demanda permite que as decisões relacionadas à produção, logística e materiais sejam tomadas de maneira mais assertiva, resultando em redução de custos e aumento da eficiência do negócio como um todo. Nesse sentido, a automação dessa previsão utilizando métodos de *machine learning* (ML) oferece grande diferencial competitivo. Dessa forma, ao aplicar algoritmos de aprendizado de máquina, a empresa encontra maior potencial para obter previsões mais precisas e confiáveis, o que pode levar a melhoras significativas na gestão de seus recursos e aumento da margem de lucro.

Com base em Ludemir (2021), o conhecimento do computador são hipóteses geradas a partir de dados. Essa abordagem do aprendizado de máquina é particularmente interessante para empresas que possuem dados sobre seus negócios ao longo de vários anos, já que podem ser usados para criar modelos de ML capazes de prever a demanda.

Os modelos de séries temporais em ML são técnicas que utilizam dados históricos a fim de prever valores de uma série temporal. Esses modelos, geralmente, envolvem a análise de padrões de comportamento da série temporal e a identificação de componentes como tendência, sazonalidade, variação aleatória e outros fatores que podem influenciar os dados. Dentre estes modelos está o *Prophet*, desenvolvido pela empresa *Meta* e projetado para prever séries temporais com sazonalidade anual, semanal e diária, sendo também capaz de lidar com mudanças abruptas nos dados, tendências não lineares, feriados e eventos especiais. Adicionalmente, será explorada a arquitetura LSTM (*Long Short-Term Memory*), que é um tipo de Redes Neurais Recorrentes (RNN) conhecido por sua capacidade de processar dados sequenciais, como são as séries temporais, permitindo a captura de padrões complexos e relações de longo prazo nos dados.

O trabalho proposto tem como objetivo principal a aplicação de técnicas de ML para previsão de demanda de um serviço de instalação de varais utilizando o modelo *Prophet* e a RNN LSTM. Para tanto, será realizado um estudo de caso em uma loja específica, onde serão coletados dados históricos de instalações realizadas e outras informações relevantes, que serão utilizados para treinar e validar os modelos propostos. A partir disso, busca-se desenvolver modelos para previsão utilizando as técnicas citadas anteriormente, a fim de realizar uma análise

comparativa dos resultados e definir o método mais acurado e confiável, capaz de auxiliar de forma mais efetiva a loja na tomada de decisões estratégicas e na otimização de seus processos de gestão de estoque e produção.

A organização deste trabalho está estruturada em quatro seções principais. A primeira seção aborda o estado da arte, fornecendo uma visão abrangente dos principais conceitos relacionados à pesquisa, como ML, séries temporais, LSTM e *Prophet*. A segunda seção é dedicada à metodologia, onde serão descritos os procedimentos e técnicas empregados na condução da pesquisa. Em seguida, será exposto o desenvolvimento do trabalho, onde serão apresentadas as implementações práticas das abordagens selecionadas, contemplando a coleta, preparação e tratamento dos dados, bem como a utilização de *Prophet* e LSTM para a predição da série temporal, destacando as considerações e decisões tomadas durante a implementação. A quarta seção se concentra nos resultados obtidos, são apresentadas as predições geradas, bem como uma análise crítica da performance de cada abordagem avaliadas por meio de múltiplas métricas. Por fim, a seção de conclusão fornecerá a síntese do trabalho, resumindo os principais pontos, destacando as contribuições do estudo e indicando as direções futuras da pesquisa.

211

ESTADO DA ARTE

Machine Learning

Mitchell (1997) descreve ML como qualquer programa de computador que melhora sua performance na realização de uma tarefa por meio da experiência. O método de ML parte da formulação de hipóteses e regras de acordo com os dados oferecidos ao sistema como exemplos. As hipóteses e regras são geradas a partir da indução que, dado um conjunto de treinamento, os valores dos atributos de entrada são relacionados a ele para gerar os atributos de saída que serão comparados ao conjunto de dados de teste.

Segundo Fradkov (2020), que apresenta a evolução da área de ML sob uma perspectiva histórica, a origem do ML moderno remonta ao fim da década de 1950 e está relacionada ao psicólogo Frank Rosenblatt, da Universidade de Cornell. Ele

criou o *Perceptron* (Rosenblatt, 1958), uma máquina que reconhecia letras do alfabeto usando sinais analógicos e discretos e que se tornou o protótipo das redes neurais artificiais, tendo semelhanças com os modelos de aprendizagem animal e humana. Na década de 1960, houve um grande interesse no desenvolvimento de algoritmos de ML e, após dois "invernos da IA", períodos nos quais houve um baixo interesse no desenvolvimento do campo, o ML volta a ganhar notoriedade no século XXI, devido à ascensão do tema *big data*, a redução do custo da computação paralela e o surgimento de algoritmos de aprendizagem profunda.

Atualmente, o ML é uma das tecnologias mais promissoras do século XXI e está presente em uma grande variedade de aplicações. Sistemas de recomendação, detecção de fraudes, realização de diagnóstico médico, reconhecimento facial e de voz, tradução de idiomas e realização de previsões são alguns dos exemplos da utilização de ML. Com isso, a versatilidade dessa tecnologia lhe confere um papel importante na transformação digital das empresas e na resolução de problemas complexos em diversas áreas.

Com a história e evolução do ML em mente, é possível entender como os algoritmos são estruturados e qual sua aplicabilidade. Conforme Murphy (2012), os três tipos de ML, supervisionado, não supervisionado e por reforço, foram desenvolvidos para solucionar diferentes problemas e alcançar diferentes objetivos. O primeiro tem como objetivo aprender uma correspondência entre *inputs* e *outputs*, sendo usado em algoritmos de classificação e regressão. O segundo é o que procura encontrar padrões interessantes nos dados sem uma informação clara de quais padrões observar ou qual métrica de erro usar. O terceiro é o aprendizado por reforço, que é útil para aprender a agir ou se comportar quando sinais ocasionais de recompensa ou punição são dados. Cada tipo de ML é adequado para diferentes tipos de problemas, logo, é importante escolher o tipo certo para o problema em questão a fim de obter os melhores resultados. Uma outra observação a ser considerada é a qualidade do conjunto de exemplos disponibilizado para a máquina, o que poderá afetar o desempenho do modelo de ML. Como a qualidade dos dados não é garantida, é preciso aplicar técnicas que a aprimorem (Ludemir, 2021).

Séries Temporais

Série temporal é uma ferramenta matemática e estatística que, no entendimento de Brockwell e Davis (2016) e Shumway e Stoffer (2017) se refere a um conjunto de observações x_t , cada qual sendo registrada em um determinado tempo t , formando uma sequência de dados relacionados temporalmente. Para Metcalfe e Cowpertwait (2009), quando uma variável é medida sequencialmente ao longo do tempo, os dados resultantes dessas medições formam uma série temporal. Essa ferramenta permite identificar padrões e tendências de um determinado dado ao longo do tempo, sendo útil em diversas áreas de análise de dados.

A análise de séries temporais é utilizada em diversas áreas e permite a realização de previsões, cuja acurácia depende da abrangência dos dados avaliados. A utilização no tempo t de observações disponíveis de uma série temporal para prever seu valor em algum momento futuro $t + l$ pode fornecer uma base para planejamento econômico e empresarial, planejamento de produção, controle de inventário e produção e controle e otimização de processos industriais (Box *et al.*, 2019).

Além disso, a análise de séries temporais tem ganhado destaque em outras áreas, sendo aplicada em diversas situações como na previsão de número de casos de determinada doença, níveis de poluentes no ar, fluxo de tráfego, entre outras situações. Assim, a ferramenta tornou-se essencial para tomadas de decisão em diferentes setores.

Long Short-Term Memory

As RNNs são uma categoria especial de redes neurais que se destacam por sua capacidade de processar dados sequenciais, como são, segundo Aggarwal (2018), séries temporais, textos e dados biológicos. A característica principal das RNNs é a conexão cíclica, que permite a ela possuir a capacidade de atualizar o estado atual com base em estados passados e dados de entrada atual (Yu *et al.*, 2019).

LSTM é um tipo de RNN, acompanhado de um algoritmo de aprendizado

baseado em gradientes, inicialmente proposto por Hochreiter e Schmidhuber (1997). Esta solução foi concebida para enfrentar os desafios associados à dissipação ou explosão dos gradientes durante o treinamento de RNNs, especialmente ao aplicar técnicas como *Back-Propagation Through Time* (BPTT) e *Real-Time Recurrent Learning* (RTRL). Na visão de Aggarwal (2018), estes obstáculos são inerentes a redes neurais que confiam unicamente em atualizações multiplicativas sucessivas sobre a matriz de pesos, demonstrando eficácia apenas para aprendizado em sequências curtas, resultando em uma boa capacidade de memória de curto prazo e revelando-se insuficiente para manter uma memória efetiva a longo prazo. Por contraste, a rede LSTM demonstra a habilidade de preservar informações relevantes ao longo de intervalos de tempo substanciais sem comprometer sua capacidade de retenção de informações de curto prazo (Hochreiter; Schmidhuber, 1997).

Tal capacidade desta rede neural se deve ao seu mecanismo de portas, desenvolvidos para substituir as funções de ativação clássicas. As células LSTM, unidade básica de processamento que compõem a rede neural, possuem três portas: uma de entrada, uma de esquecimento e uma de saída. Estas portas permitem realizar alterações em um vetor de estado da célula que é propagado iterativamente para capturar dependências de longo prazo. Isto permite à rede memorizar múltiplas dependências temporais com diferentes características (Lindemann et al., 2021).

Após a divulgação desta rede neural, inúmeras contribuições ao modelo original foram propostas por acadêmicos ao longo dos anos, como o *Bidirectional LSTM*, *Stacked LSTM*, *Multidimensional LSTM*, *Convolutional LSTM*, entre outros.

Prophet

O *Prophet* é uma biblioteca de previsão de séries temporais desenvolvida internamente na empresa *Meta Platforms*, antigo *Facebook*, por Sean J. Taylor e Benjamin Letham em 2017. Foi disponibilizado como software *open source* para as linguagens de programação Python e R juntamente com o artigo "*Forecasting at Scale*" (Taylor; Letham, 2017), que descreve sua metodologia e motivações.

Taylor e Letham (2017) identificaram dois problemas comuns enfrentados

pelos analistas responsáveis pelas tarefas de ciência de dados. Em primeiro lugar, as técnicas de predição automatizadas são inflexíveis e difíceis de ajustar, o que pode levar a resultados imprecisos. Em segundo lugar, a maioria dos analistas não têm treinamento em previsão de séries temporais, o que dificulta ainda mais a tarefa. Para resolver esses problemas, o *Prophet* foi projetado para gerar previsões de alta qualidade com poucas especificações e também ser altamente personalizável e flexível. Assim, mesmo usuários sem experiência em previsão de séries temporais podem obter previsões precisas e ajustadas às suas necessidades. Com esses objetivos em mente, o *Prophet* foi projetado para ser escalável para muitas séries temporais, flexível o suficiente para uma ampla variedade de séries temporais relevantes para negócios, possivelmente idiossincráticas, e ao mesmo tempo intuitivo (Auffarth, 2021). Seu modelo pode ser descrito como:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (1)$$

Conforme descrito por Taylor e Letham (2017), em (1), $y(t)$ é a previsão gerada pelo modelo, $g(t)$ é a função de tendência que modela mudanças não periódicas, $s(t)$ é a sazonalidade que representa mudanças periódicas e $h(t)$ representa os feriados, que são considerados como efeitos que ocorrem em programações potencialmente irregulares. O termo ϵ_t representa o erro relacionado a mudanças idiossincráticas não explicadas pelo modelo, além disso, ele é assumido como normalmente distribuído.

Com base nas subseções apresentadas neste capítulo, foi possível observar que a literatura existente sobre predição de demanda com ML engloba uma ampla variedade de setores e serviços, mas há uma escassez de estudos que se concentram na aplicação deste trabalho. Portanto, o presente trabalho busca contribuir com a comunidade científica e empresarial ao aplicar o modelo *Prophet* e a arquitetura de RNN LSTM no contexto de instalações de varais, fornecendo uma nova perspectiva de aplicação do ML e resultados relevantes que podem ser aplicados em outros setores empresariais.

METODOLOGIA

A condução deste estudo para prever a demanda do serviço de instalação de varais envolveu uma série de etapas bem definidas. Inicialmente, os dados históricos foram adquiridos a partir do banco de dados da empresa Casa dos Varais.

Uma vez obtidos os dados, procedeu-se à fase de pré-processamento. Essa etapa visa preparar os dados para a aplicação dos modelos de ML, garantindo a qualidade e a coerência necessárias para as previsões. Dessa forma, após a seleção da variável de interesse para orientar a previsão, foram implementadas estratégias para lidar com dados ausentes, principalmente em situações como domingos, feriados e períodos de férias, assim como para tratar *outliers*. Em seguida, determinou-se os melhores hiperparâmetros para cada modelo. Para a Rede Neural LSTM, foram conduzidos testes com o apoio de técnicas de *tuning* envolvendo a configuração do número de camadas e unidades, a seleção de funções de ativação e otimizadores mais adequados. Já para o *Prophet* foram experimentados inúmeros valores para seus componentes.

216

Com os hiperparâmetros definidos, os modelos foram treinados utilizando o conjunto de dados de treinamento, permitindo-lhes aprender padrões e relações complexas presentes na série temporal. Posteriormente, foram empregados para realizar previsões sobre o conjunto de teste, fornecendo estimativas da demanda de instalação de varais.

Finalmente, para avaliar e comparar a performance dos modelos, os resultados das previsões foram confrontados com os valores reais. Para analisar com mais critério o desempenho da rede neural LSTM e do modelo *Prophet*, métricas relevantes, como Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE) e Erro Absoluto Médio (MAE), foram empregadas para quantificar a acurácia das previsões.

DESENVOLVIMENTO

Nesta seção, serão apresentados todos os procedimentos e etapas realizados neste trabalho com o objetivo de prever a demanda de um serviço de

instalação de varais, empregando a rede neural LSTM e o modelo *Prophet*. Inicialmente, serão apresentados o ambiente computacional e as ferramentas escolhidas para o desenvolvimento. Em seguida, será detalhada a coleta e preparação dos dados, seguidas das explicações referentes à implementação dos modelos citados. Por fim, a predição realizada pelos modelos será avaliada.

Ambiente e Ferramentas

Todo o projeto foi realizado utilizando o ambiente do *Google Colaboratory*, uma plataforma de *notebook* interativo baseada na nuvem, que oferece suporte à linguagem de programação Python em sua versão 3.10.6.

As principais bibliotecas utilizadas e sua utilidade neste trabalho são:

- *Prophet* (versão 1.1.4): Implementa o modelo *Prophet* para predição de séries temporais.
- *Tensorflow* (versão 2.13.0): É uma biblioteca de código aberto para ML. Foi utilizada para implementar o modelo LSTM.
- *Pandas* (versão 1.5.3): *Pandas* é uma biblioteca de análise de dados que fornece estruturas de dados flexíveis para sua manipulação e análise.
- *Matplotlib* (versão 3.7.1): É uma biblioteca de visualização de dados que oferece uma variedade de ferramentas para a criação de gráficos.
- *Scikit-learn*: Biblioteca que oferece diversos algoritmos para a realização de tarefas com ML. Neste trabalho, foi utilizado o pacote *metrics* para realizar a avaliação dos modelos.

217

Coleta e Tratamento dos Dados

Os dados utilizados nesta pesquisa foram obtidos do banco de dados da empresa Casa dos Varais, a qual oferece o serviço de instalação de varais. Para a coleta desses dados, foi acessada a base de dados por meio do sistema *Microsoft Access*. A fonte desses dados é a tabela denominada *IdServico_tbl*, a qual registra as instalações de varais realizadas no período entre fevereiro de 2004 e dezembro de 2021. A tabela inclui as colunas *IdServico*, *DataPedido*, *DataInstalacao*, *IdPeriodo*, *Sequencia*, *IdEdificio*, *Apartamento*, *Bloco*, *IdCliente* e *Observacao*.

Para possibilitar a manipulação e análise dos dados, a tabela foi exportada no formato de planilha, permitindo assim sua importação e conversão em uma estrutura tabular conhecida como *DataFrame*, por meio da biblioteca Pandas em ambiente Python. Esse processo marcou o início da fase de tratamento dos dados para a aplicação dos modelos de predição.

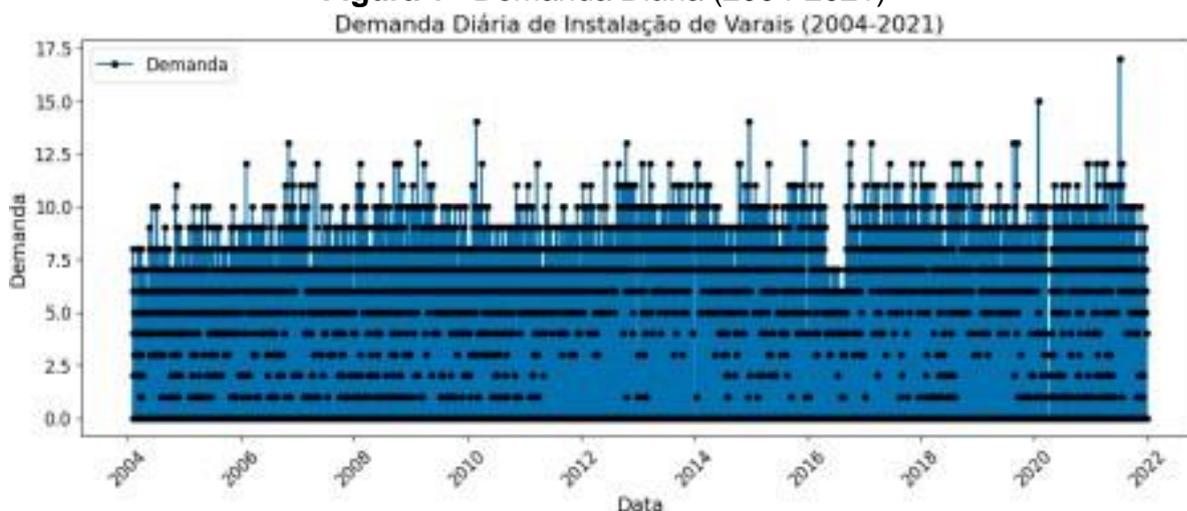
Inicialmente, procedeu-se à seleção das colunas pertinentes para a realização da predição, resultando na retenção exclusiva da coluna *DataInstalação*, que representa a data em que o serviço de instalação foi realizado. Tal seleção decorre da premissa estabelecida de que a informação fundamental para antecipar a demanda do serviço reside na quantidade de instalações de varal efetuadas. Através dessa etapa, foram obtidas as quantidades diárias de instalações, levando em consideração a frequência na qual as respectivas datas emergem no conjunto de dados.

Em seguida, foi observado que existe um grande número de dias que não estão presentes no conjunto de dados, sendo em sua maior parte domingos, feriados e período de férias. Devido à característica do modelo *Prophet* de considerar feriados e outros eventos especiais que ocorrem regularmente e devido a testes utilizando dados de todos os dias do ano, foi considerado propício adicionar tais datas faltantes, tendo como quantidade de demanda o valor zero.

O gráfico 1 exibe a demanda diária do serviço ao longo dos anos, de fevereiro de 2004 a dezembro de 2021, da loja Casa dos Varais.

218

Figura 1 - Demanda Diária (2004-2021)



Fonte: Autoria própria

No contexto do tratamento de *outliers*, adotou-se uma abordagem fundamentada em técnicas de análise estatística visando tanto à identificação quanto à correção destes valores discrepantes. A estratégia empregada para detecção de *outliers* baseou-se no método de estabelecimento de limites por meio do intervalo interquartil (IQR). Nesse procedimento, os dados foram ordenados em ordem crescente e divididos em quartis, sendo o primeiro quartil representativo dos 25% dos dados inferiores, o segundo quartil correspondente à mediana do conjunto e o terceiro quartil contendo os 25% superiores das observações. O cálculo do IQR, obtido pela diferença entre o terceiro e o primeiro quartil, permitiu a definição de um limiar, situado a 1,5 vezes o valor do IQR, acima e abaixo do qual os dados foram considerados como *outliers*. Esta abordagem possibilitou, assim, a identificação efetiva dos *outliers*, que posteriormente foram removidos do conjunto de dados para não influenciar o resultado final das previsões.

Por fim, foi realizada a segmentação dos dados em conjuntos de treinamento e teste. Os registros de fevereiro de 2004 ao fim de 2017 foram atribuídos ao conjunto de treinamento, enquanto os dados dos anos de 2018 a 2021 foram designados para o conjunto de teste. Essa divisão foi executada visando a obtenção de uma proporção de aproximadamente 80% de dados de treinamento e 20% de dados de teste, contudo, com a atenção voltada para a manutenção dos anos completos em ambos os conjuntos. Essa estratégia garante a representatividade temporal e preserva a continuidade dos padrões anuais ao longo das fases de treinamento e teste dos modelos de previsão.

Cabe ressaltar que, em meio à análise, foi avaliada a possibilidade de excluir registros referentes aos anos de 2020 e 2021 caso se constatasse uma alteração significativa na demanda em decorrência das medidas de restrição para combate à pandemia da COVID-19. Nesse contexto, é importante mencionar que ocorreu uma marcante redução no início de 2020. Contudo, essa tendência logo cedeu lugar à retomada da quantidade de demanda observada em anos anteriores e, portanto, optou-se por manter os dados que dizem respeito ao período observado.

Implementação da rede neural *Long Short-Term Memory*

Com os dados devidamente coletados e tratados, deu-se o desenvolvimento da rede neural LSTM para realizar a predição da demanda do serviço observado. Todavia, antes de adentrarmos na configuração do modelo, é importante destacar o processo de normalização dos dados de treino e teste. Utilizou-se a técnica conhecida como Min-Max Scaler, amplamente empregada em ML e estatística. Essa técnica tem por objetivo padronizar os valores do conjunto de dados para um intervalo específico, comumente entre 0 e 1, permitindo que os algoritmos de ML os interpretem de maneira mais eficaz.

Visando encontrar o modelo mais adequado ao problema, foram testadas diversas funções de ativação e otimizadores. Dentre as funções estudadas estão ELU, tanh, ReLU, swish, SELU, sigmoid e GELU (Rasamoelina *et al.*, 2020) (Hendrycks; Gimpel, 2023), que afetam a capacidade da rede de modelar relações complexas entre os dados de entrada e saída determinando como a informação é transmitida entre as diferentes partes da rede. Da mesma forma, a escolha do otimizador, entre opções como Adam, Nadam, AdaDelta, Adamax, AdaGrad, SGD e RMS Prop (Ruder, 2017), influencia a eficiência do processo de aprendizado ao ajustar os pesos da rede com o objetivo de minimizar os erros durante o treinamento.

Além disso, foram empregadas técnicas de *tuning* para determinar uma configuração otimizada de número de camadas, quantidade de neurônios para cada uma dessas camadas e o período de *lookback*. Este período define quantos passos temporais anteriores são considerados para a predição do próximo passo temporal e desempenha um papel crucial na captura de padrões de longo prazo.

O modelo desenvolvido foi estruturado de forma que inicialmente, é empregada uma camada de entrada composta por um único neurônio e leva em consideração o período de *lookback* de 7 passos anteriores que o modelo utiliza para prever o próximo valor, permitindo uma análise mais abrangente das dependências temporais. Essa camada é responsável pela recepção das informações iniciais. Em seguida, são incorporadas quatro camadas ocultas de LSTMs, onde ocorrem cálculos que auxiliam a rede a encontrar a saída adequada. A

primeira delas possui 225 unidades, a segunda, 100, a terceira, 50 e a última delas, 225, sendo todas ativadas pela função eLU, que se demonstrou eficaz para aprender padrões dos dados em questão.

Finalizando, a saída do modelo é processada por uma camada densa composta por uma única unidade, resultando na predição do próximo valor na série temporal, consolidando as aprendizagens realizadas ao longo da rede.

O modelo é treinado utilizando o otimizador Adam e a métrica de perda MSE. Durante o treinamento, o modelo é submetido a 1300 épocas, com um tamanho de lote de 32 exemplos que são apresentados à rede neural em uma única iteração durante o processo de treinamento.

Implementação do modelo *Prophet*

Dando sequência à análise e implementação dos modelos de predição de séries temporais, foi explorada a adaptação e aplicação do modelo *Prophet* para antecipar padrões e tendências na série temporal da demanda.

221

No desenvolvimento, foram realizados diversos testes para otimizar os parâmetros e obter uma previsão de demanda precisa e confiável. Foram ajustados os seguintes parâmetros: *changepoint_prior_scale*, *seasonality_prior_scale*, *yearly_seasonality*, *weekly_seasonality* e *daily_seasonality*.

Um dos recursos do *Prophet* são os *changepoints*, pontos onde a tendência da série temporal pode se alterar. Esses *changepoints* são identificados e inseridos pelo próprio *Prophet*, mas também há a opção de serem adicionados pelo programador, o que não foi necessário no presente trabalho. O parâmetro *changepoint_prior_scale* controla a sensibilidade a essas mudanças. No caso, o valor padrão de 0,005 foi ajustado para 0,043, resultando em uma menor flexibilidade na representação da tendência.

No que diz respeito ao parâmetro *seasonality_prior_scale*, que controla a flexibilidade da sazonalidade, um valor mais alto possibilita que a sazonalidade se ajuste a variações mais amplas. A alteração foi de 10, que implica em pouca alteração significativa na sazonalidade, para 30.

No contexto deste estudo, os parâmetros *yearly_seasonality*,

weekly_seasonality e daily_seasonality foram personalizados para capturar padrões recorrentes anuais, semanais e diários. De acordo com Spadini e Alencar (2022), a sazonalidade é definida através da soma parcial da Série de *Fourier*, resultando no fato de que o número de termos nessa soma parcial determina a rapidez com que a sazonalidade pode mudar. No modelo em questão, foi desabilitada a sazonalidade diária, enquanto a sazonalidade semanal e anual foram ajustadas com 2 e 20 termos de *Fourier*, respectivamente. Os parâmetros que não foram citados permaneceram com seus valores padrão.

Além dos parâmetros mencionados, foram adicionados ao modelo os feriados brasileiros. A inclusão desses feriados foi realizada para que o *Prophet* leve em consideração eventos especiais que podem impactar a demanda.

Após a otimização dos parâmetros, o modelo *Prophet* foi treinado com o conjunto de dados de treino, e a previsão foi gerada para o período contido nos dados de teste.

Avaliação dos Modelos

222

Para mensurar a eficácia das abordagens de predição de séries temporais por meio de ML validando sua capacidade preditiva em relação aos valores reais, foram utilizadas as métricas Erro Quadrático Médio (MSE), Raiz do Erro Quadrático Médio (RMSE) e Erro Absoluto Médio (MAE). Em suas fórmulas, y_i representa o valor real na série temporal, enquanto \hat{y}_i denota a estimativa prevista e o termo n corresponde ao número de observações presentes no conjunto de dados.

O MSE é uma métrica que calcula a média dos quadrados das diferenças entre as previsões do modelo e os valores reais. Esta medida tende a ser fortemente influenciada pelos maiores erros, devido ao fato de ser elevada à segunda potência.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

O RMSE é constituído pela raiz quadrada do MSE, apresentando a medida na mesma escala dos dados fornecidos pela série temporal e tornando a interpretação mais intuitiva.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

O MAE calcula a média das diferenças absolutas entre as previsões do modelo e os valores reais.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Essas métricas de avaliação foram aplicadas a ambas as previsões realizadas. Isso permitiu uma análise abrangente da acurácia de cada abordagem em relação aos valores reais da série temporal.

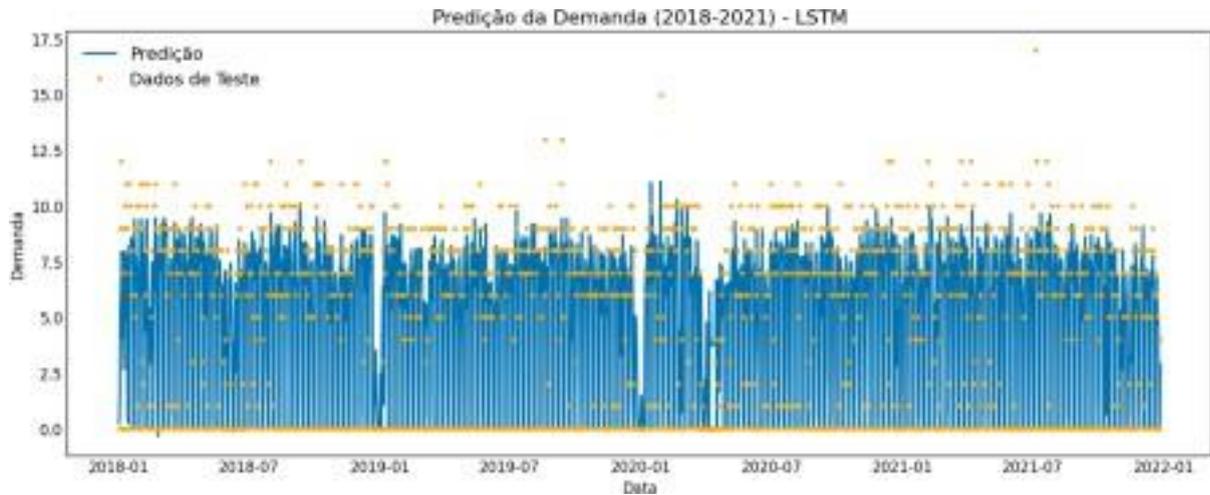
Ao longo deste capítulo, foi detalhado o processo, desde a coleta dos dados até a geração das previsões e a avaliação de sua acurácia utilizando as três métricas anteriormente mencionadas. Dessa maneira, os resultados obtidos através delas serão discutidos.

RESULTADOS

223

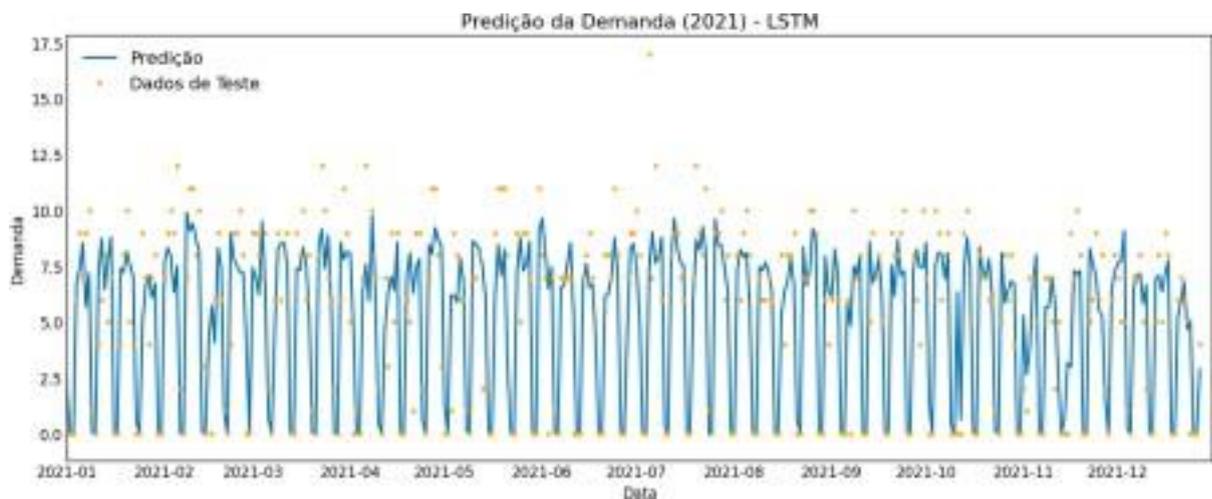
Na atual etapa, serão expostos e analisados os resultados provenientes das previsões geradas pelos modelos LSTM e *Prophet*. Os gráficos exibem tanto a demanda prevista quanto os dados reais coletados. Os gráficos apresentados nas figuras 2 e 3 correspondem à previsão realizada pela rede neural LSTM. A primeira abrange todo o período designado como teste, de 2018 a 2021, enquanto a segunda se concentra exclusivamente no ano de 2021, permitindo uma visualização mais detalhada das características da previsão gerada. Da mesma forma, os gráficos das figuras 4 e 5 representam as previsões do modelo *Prophet* para os períodos referidos anteriormente.

Figura 2 - Predição da demanda com LSTM (2018-2021)



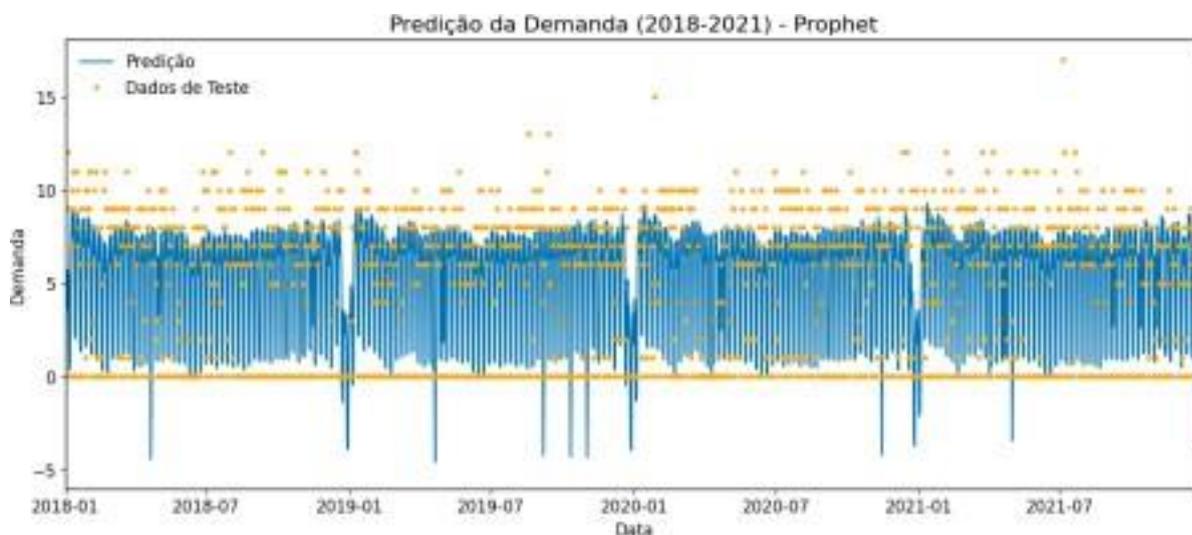
Fonte: Autoria própria

Figura 3 - Predição da demanda com LSTM (2021)



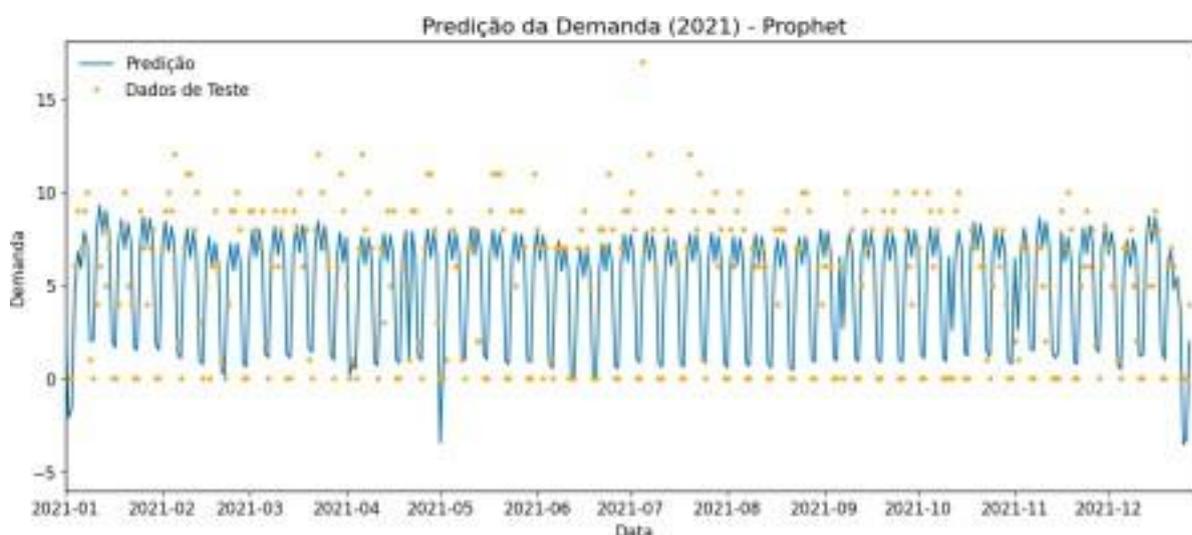
Fonte: Autoria própria

Figura 4 - Predição da demanda com Prophet (2018-2021)



Fonte: Autoria própria

Figura 5 - Predição da demanda com Prophet (2021)



Fonte: Autoria própria

Ao analisar o resultado gerado pelo modelo LSTM, apresentado nos gráficos das figuras 2 e 3, pode-se concluir que o modelo se destaca ao gerar predições melhores adaptadas à série temporal com padrões complexos. Por outro lado, a previsão de demanda gerada pelo *Prophet* assume valores negativos bastante distantes de zero dada a escala dos dados, uma ocorrência impossível no caso estudado. Além disso, o modelo não foi capaz de aprender o comportamento

complexo da demanda do serviço, prevendo padrões mais simples.

Apesar de terem sido realizadas tentativas para evitar a ocorrência de valores negativos em ambas as previsões, incluindo abordagens como a aplicação do crescimento logístico para estabelecer limites máximos e mínimos na previsão do *Prophet*, bem como a exploração de diferentes funções de ativação e otimizadores para o modelo LSTM, não foi possível eliminar completamente esse cenário.

Todavia, foi constatado que os valores negativos gerados pela rede neural LSTMsão mais próximos dos dados reais do que os apresentados pelo *Prophet*.

Na Tabela 1, apresenta-se os valores das métricas MSE, RMSE e MAE para cada uma das abordagens empregadas.

Tabela 1 – Métricas de avaliação dos modelos

	MSE	RMSE	MAE
LSTM	4,1500	2,0371	1,4341
<i>Prophet</i>	4,9839	2,2324	1,6887

226

Ao considerar a natureza dos dados, as métricas utilizadas e os gráficos apresentados, os resultados apontam claramente para o modelo LSTM como a escolha mais adequada para conjuntos de dados com essa característica, ou seja, séries temporais com características semelhantes, a implementação do LSTM é a recomendada. Vale ressaltar que diferentes aplicações podem apresentar particularidades nos dados, exigindo uma avaliação criteriosa para determinar o modelo mais apropriado. Dada a precisão notável do ajuste observado, o LSTM promete ser uma abordagem promissora para previsões de demanda no contexto do serviço em foco. Ao adotar esta rede neural para realizar previsões de demandas, a empresa poderá otimizar seus processos, minimizar custos, aprimorar a gestão de recursos e garantir uma resposta eficaz às necessidades dos clientes, fortalecendo assim sua posição no mercado.

CONCLUSÃO

Este trabalho teve como objetivo analisar e comparar os modelos LSTM e *Prophet* para a previsão da demanda de instalação de varais.

A RNN se destacou por aparentar maior adaptação às variações da série temporal, o que a torna mais recomendável para o caso estudado. Em contrapartida, o *Prophet* demonstrou dificuldade em capturar padrões complexos, provando ser inadequado.

Em perspectiva para trabalhos futuros, é considerada a implementação do modelo de *machine learning* no cotidiano do serviço a fim de avaliar seus impactos reais. Ademais, propõe-se a investigação de outros modelos e aprimoramento das técnicas atuais, visando um refinamento contínuo das previsões de demanda.

REFERÊNCIAS

AGGARWAL, C. C. **Neural Networks and Deep Learning: A Textbook**. 1 ed. New York: Springer Cham, 2018.

AUFFARTH, B. **Machine Learning for Time-Series with Python: Forecast, Predict, and Detect Anomalies with State-of-the-art Machine Learning Methods**. Birmingham: Packt Publishing Ltd., 2021.

BOX, G. E. P. et al. **Time Series Analysis: Forecasting and Control**. 5. ed. New Jersey: Wiley, 2016.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to Time Series Forecasting**. 3. ed. AG Switzerland: Springer Cham, 2016.

FRADKOV, A. L. Early History of Machine Learning. **IFAC-PapersOnLine**, [S. l.], v. 53, n. 2, p. 1385-1390, 2020. Disponível em: <https://doi.org/10.1016/j.ifacol.2020.12.1888>. Acesso em: 27 maio 2023.

HENDRYCKS, Dan; GIMPEL, Kevin. Gaussian Error Linear Units (GELUs). 2023. Disponível em: <https://arxiv.org/abs/1606.08415>. Acesso em: 14 out. 2023.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735-1780, 1997. DOI: 10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, v. 349, p. 255-260, 2015. DOI: 10.1126/science.aaa8415.

LINDEMANN, B.; MÜLLER, T.; VIETZ, H.; JAZDI, N.; WEYRICH, M. A survey on long short-term memory networks for time series prediction. **Procedia CIRP**, v. 99, p. 650-655, 2021. ISSN 2212-8271. DOI: 10.1016/j.procir.2021.03.088.

LUDERMIR, T. B. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. **Estudos Avançados**, [S. l.], v. 35, n. 101, p. 85-94, 2021. DOI: 10.1590/s0103-4014.2021.35101.007. Disponível em: <https://www.revistas.usp.br/eav/article/view/185035>. Acesso em: 19 mar.2023.

METCALFE, A. V.; COWPERTWAIT, P. S. P. **Introductory Time Series with R**. 1. ed. New York: Springer New York, 2009.

MITCHELL, T. M. **Machine Learning**. 1 ed. New York: McGraw-Hill, 1997.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Illustrated Edition. Cambridge: The MIT Press, 2012.

RASAMOELINA, A.D; ADJAILIA, F.; SINÁK, P. A Review of Activation Function for Artificial Neural Network. In: IEEE WORLD SYMPOSIUM ON APPLIED MACHINE INTELLIGENCE AND INFORMATICS, 18., 2020, Herlany. **Anais [...]** Herlany: IEEE, 2020. p. 281-286. DOI: 10.1109/SAMI48414.2020.9108717.

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p.386-408, 1958.

RUDER, Sebastian. **An overview of gradient descent optimization algorithms**. 2017. Disponível em: <https://arxiv.org/abs/1609.04747>. Acesso em: 14 out. 2023.

SKANSI, S. **Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence**. 1 ed. New York: Springer Cham, 2018.

SHUMWAY, R. H.; STOFFER, D. S. **Time Series Analysis and Its Applications With Examples in R**. 4 ed. New York: Springer Cham, 2017.

SPADINI, A.; ALENCAR, V. **Séries temporais com Prophet: Análise e previsão de dados com Python**. São Paulo: Casa do Código, 2022.

TAYLOR, S. J.; LETHAM, B. **Forecasting at scale**. PeerJ Preprints, v. 5, p. e3190v2, 2017. Disponível em: <https://doi.org/10.7287/peerj.preprints.3190v2>. Acesso em: 21 abr. 2023.

YU, Y.; SI, X.; HU, C.; ZHANG, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. **Neural Computation**, v. 31, n. 7, p.1235-1270, 2019. DOI: 10.1162/neco_a_01199.