Aplicação de Aprendizagem de Máquina para previsão de Energia Eólica Gerada

Mauricio Gerhardt and Carine Geltrudes Webber

***Resumo*—A inteligência artificial (IA) tem experimentado avanços significativos nos últimos anos, impulsionando seu crescente campo de aplicação em diversas áreas, entre as quais se destaca a aplicação em sistemas preditivos. Nesses sistemas, ocorre o mapeamento e rotulagem de dados previamente coletados, sendo esses dados utilizados como entrada para algoritmos de IA, que visam prever possíveis valores futuros. Para o presente trabalho foi escolhida uma área, que vem se destacando no Brasil e no mundo, que trata da geração de energia através de geradores eólicos. O presente trabalho explora os conceitos e funcionamento dos modelos preditivos e propõe-se a realizar uma pesquisa de natureza exploratória, a fim de identificar os principais métodos que são utilizados, por meio de uma pesquisa bibliográfica e uma revisão sistemática. A partir desta pesquisa, propõe-se a implementação de dois modelos preditivos mais utilizados na área de problemas de regressão que se tratam dos modelos *Long Short Term Memory* (LSTM) e o *Extreme Gradient Boosting* (XGBoost). Após a implementação e treinamento dos modelos propostos, são apresentados os resultados quantificados por métricas, conhecidas e amplamente utilizadas para avaliar os valores obtidos na previsão da energia gerada por estes modelos.**

***Index Terms*—Aprendizado de máquina. Previsão de Dados. Geração de Energia. Inteligência Artificial. Redes Neurais.**

I. Introdução

EM 2021 o Brasil passou por uma das piores crises energéticas dos últimos anos, podendo ser comparada com a crise energética de 2001 que ficou conhecida como o apagão de 2001, as crises energéticas podem se basear em diferentes motivos entre eles política, econômica e ambiental [1]. Uma das principais características que pode-se analisar em meio a uma crise energética é a falta de garantia de segurança na capacidade de abastecimento para sua população.

Analisando informações pode-se notar como o Brasil é fortemente baseado em sua produção de energia hídrica tendo mais de 50% de sua produção proveniente desta origem enquanto todas as outras fontes de energia representam os outros 50% como energia eólica, solar, proveniente de gás natural ou biocombustíveis, o que mostra como o Brasil deve buscar outros modos de produzir energia para diminuir a dependência e correr menor risco de sofrer outra crise [2]. Um destes modos de produzir energia é a energia eólica que já vem crescendo muito nos últimos anos.

Em 2021 o Brasil passou a ocupar o 6º lugar no ranking internacional em capacidade de produção de energia eólica estando atrás apenas de Estados Unidos, China, Alemanha, Índia e Espanha. Em 2012 ocupava o 12º lugar neste mesmo ranking, demonstrando como o progresso nesta área está tomando evidência e força [3]. Em 2022 obteve seu recorde em produção de energia eólica sendo capaz de atender toda demanda da região nordeste do país e tendo excedente de 23.2%, o que mostra o potencial de produção deste meio [4].

Uma das principais questões na produção de energia eólica é a produção estar totalmente a mercê de questões ambientais como a direção e intensidade do vento, por esta questão os principais parques eólicos se encontram em regiões propícias a esta condição como é o caso do nordeste brasileiro onde possui uma região mais propícia para receber ventos. Porém, mesmo que uma região seja mais propícia, ainda possui-se um problema que é estabelecer quanta energia será produzida por uma turbina eólica ou por um parque eólico, o que pode trazer mais confiança em um investimento ou em um acordo. Sistemas preditivos podem auxiliar neste processo.

Na área da Computação, a Inteligência Artificial (IA) é o campo de estudo que se interessa por conceber e implementar sistemas com capacidade analítica e de tomada de decisão em várias áreas do conhecimento [5]. Em se tratando de sistemas preditivos, diversos modelos de IA estão sendo aplicados nas áreas de finanças, saúde e também na geração de energia.

Não se pode negar que a inteligência computacional está presente no dia a dia de cada um de nós, como em carros inteligentes, assistentes digitais em seu celular, porém muitas vezes não percebemos a real importância desta. Neste trabalho iremos analisar e entender o funcionamento de diversas etapas que constroem uma inteligência computacional e será apresentado uma possível solução para um problema real na previsão de energia gerada na área de energia eólica.

Nos estudos sobre sistemas preditivos aplicados à geração de energia, observa-se que o processo de Aprendizagem de Máquina surge como uma solução que vem sendo aplicada. Ele parte de etapas de coleta de informações, tais como a direção e a velocidade do vento, para inferir quanta energia pode ser produzida em uma faixa de tempo.

A inteligência computacional consiste em uma área da inteligência artificial que busca representar, por meio de algoritmos, diversas características humanas, incluindo habilidades como raciocínio lógico, reconhecimento de padrões e aprendizado. Tal ramo tem sido objeto de estudo e pesquisa graças à sua capacidade de simular, em certa medida, o comportamento humano e fornecer soluções para problemas complexos em diversas áreas do conhecimento [5]. Com estas características busca solucionar problemas que sempre estão se adequando às mudanças.

Algumas técnicas da área da inteligência computacional são a lógica *fuzzy*, máquinas de vetores de suporte, aprendizagem de máquina (*Machine Learning*) e aprendizagem profunda (*Deep Learning*) [5].

Dentre estas técnicas, a que mais tem sido empregada em

Figura 1: Parque eólico da cidade de João Câmara (RN).

sistemas inteligentes é sub-área de aprendizagem de máquina. Entende-se por aprendizagem de máquina, de acordo com Mitchell [6], o processo em que uma máquina, realizando tarefas T, e tendo seu desempenho mensurado por D, adquire experiência E, se o seu desempenho D melhora conforme adquire experiência E, ao longo do tempo.

Pode-se dizer que aprendizagem de máquina é o campo de estudo onde damos ao computador a capacidade de aprender sem necessariamente termos que programar todas as possibilidades, trazendo as características humanas de raciocinar e encontrar padrões para obter resultados cada vez melhores.

O objetivo geral deste trabalho é avaliar modelos preditivos aplicados à área de geração de energia. Para que isto seja possível, os seguintes objetivos específicos foram definidos:

1) Identificar métodos aplicáveis ao desenvolvimento de modelos preditivos, para estimar a geração de energia.
2) Compreender o funcionamento dos métodos preditivos, aplicados para a área da geração de energia.
3) Identificar métricas para se avaliar o desempenho dos métodos preditivos.
4) Coletar, pré-processar e transformar dados de geradores de energia, a fim de constituir um conjunto de dados.
5) Implementar o treinamento de modelos preditivos a partir dos dados de geradores de energia.
6) Comparar, avaliar e apresentar os resultados obtidos em testes com os modelos preditivos.

O presente trabalho se encontra organizado como segue. A Seção II demonstra como foi realizada o estudo dos trabalhos relacionados bem como descreve os principais pontos observados nos trabalhos relacionados. Na Seção III são apresentados os materiais e o métodos, utilizados na implementação dos modelos. A Seção IV apresenta as etapas para realizar o desenvolvimento e o treinamento dos modelos propostos. A Seção V apresenta os resultados obtidos do experimento e a Seção VI conclui o trabalho, ressaltando as descobertas com o desenvolvimento e aplicação dos modelos.

II. Trabalhos Relacionados

O universo de métodos preditivos é amplo e complexo [6]. Identificar qual método se encaixa para prever um resultado em um domínio implica em testes e pesquisas aprofundadas. A revisão sistemática compreende uma sequência de etapas que visa auxiliar o pesquisador a mapear o estado da arte em uma área de estudo [7]. Sendo assim, propõe-se o uso desta revisão sistemática para identificar os principais trabalhos relacionados, descritos a seguir.

A. Revisão sistemática

A revisão sistemática é um método de pesquisa que utiliza a literatura como fonte de dados sobre um tema específico. Seu propósito é fornecer um resumo das evidências relacionadas a uma estratégia de intervenção específica, por meio da aplicação de métodos explícitos e sistematizados de busca, avaliação crítica e síntese das informações selecionadas.

Este artigo utilizou o processo de revisão sistemática a fim de encontrar as melhores e mais recentes fontes, relacionadas ao problema proposto. Uma revisão é composta de cinco passos [7]: definir a questão de pesquisa, buscar evidências, revisar e selecionar estudos, analisar o material e apresentar os resultados.

Uma revisão sistemática de qualidade exige uma pergunta ou questão bem formulada e clara. Para este trabalho em particular, a pergunta a ser respondida é: "Com base no que já foi proposto na literatura, quais são os métodos preditivos mais amplamente empregados e com resultados mais eficazes na previsão de energia eólica?"

Para esta revisão sistemática foi utilizada a plataforma online *ScienceDirect*[1], que oferece acesso a uma vasta gama de artigos científicos, revisões, livros e conferências em várias disciplinas acadêmicas. Operada pela *Elsevier*, ela é amplamente utilizada por pesquisadores, professores e estudantes para obter informações atualizadas e confiáveis. Através de pesquisa avançada, filtragem e recursos de download, os usuários podem acessar e explorar o conteúdo científico de forma eficiente.

A busca por evidências começa com a definição de termos ou palavras-chave, seguida das estratégias de pesquisa e da definição das bases e fontes de dados. Foram estabelecidos neste artigo os seguintes termos de busca, "*neural networks*", "*power generate*", "*forecast*"e "*wind power*".

Ao revisar e selecionar os estudos, é realizada a avaliação dos títulos e resumos (*abstracts*) identificados na busca inicial. Com o uso destas palavras chaves foram selecionados os 100 artigos mais relevantes para a leitura dos resumos, sendo esta ordenação realizada pela plataforma. Quando o título e o resumo não são esclarecedores, busca-se o artigo completo. Além disso, critérios de inclusão e exclusão são estabelecidos. Foram definidos nesta pesquisa os seguintes critérios de inclusão e exclusão:

- Trabalhos publicados nos últimos três anos;
- Que apresentaram maior relevância no ramo da energia eólica;
- Que obtiveram resultados satisfatórios;

Ao analisar a qualidade metodológica dos estudos, a validade dos estudos incluídos na revisão sistemática é um aspecto crucial. No presente estudo foi desenvolvida uma escala de pontuação com base na relevância de cada artigo em relação ao cenário estudado, utilizando uma escala de pontuação de 1 (menos relevante) a 5 (mais relevante).

Ao apresentar os resultados, é possível destacar as principais características dos artigos analisados em um quadro, incluindo informações sobre os autores, ano de publicação e principais resultados. Após a análise dos estudos, os artigos foram classificados por relevância e selecionaram-se os sete mais relevantes, que foram posteriormente estudados em detalhes. Os resultados dessa revisão sistemática dos sete artigos mais relevantes são apresentados na Tabela I.

B. Artigos Relacionados

Ao analisar os trabalhos relacionados foi visto que diversos deles possuíam o objetivo de buscar uma maneira de criar um método mais preciso e com a menor taxa de erro possível. Uma estratégia empregada pelos autores consiste em combinar métodos, compondo uma arquitetura complexa, normalmente se valendo de algoritmos tais como o *Long Short Term Memory* (LSTM) e o *Extreme Gradient Boosting* (XGBoost). Foi visto também que a maioria dos trabalhos se originam na China, país líder na geração de energia através de geradores eólicos. Observa-se contudo que os *datasets* dos trabalhos mencionados nem sempre se encontram disponíveis, o que gera uma dificuldade na comparação de seus resultados.

[1]https://www.sciencedirect.com/

Um dos artigos que obteve um resultado mais satisfatório em seu indicador de *Root Mean Squared Error* (RMSE) foi o artigo que utilizou uma arquitetura que mescla dois modelos preditivos, o LSTM e o BP [8]. Neste artigo de pesquisa foi utilizada uma arquitetura que destina a utilização de diferentes métodos de previsão de acordo com os valores de suas entradas, caso seus valores de entrada sejam uma sequencia de altas frequências, irá seguir para o LSTM e caso seja uma sequencia de baixas frequências, irá utilizar o BP.

Outro artigo que obteve resultados igualmente satisfatórios foi o artigo utilizando de uma arquitetura de XGBoost com a utilização de dados espaciais da região, para obter uma previsão com uma taxa de erro muito baixa. Para utilização do XGBoost foi utilizada a biblioteca de código aberto proposto por Carlos Guestrin e Tianqi Chen [14]. Para a implementação e validação do modelo foi utilizado um *dataset* proveniente da usina Eólica Datang Ruoqiang, localizada na província de Xinjiang, com capacidade de 49,5MW. Os dados utilizados foram dispostos ao longo de todo ano de 2018, com uma razão periódica de 15 minutos.

C. Considerações Observadas

A partir da revisão sistemática foi identificado que é necessário seguir uma série de etapas para a construção de um modelo e a sua validação. Dentre estas etapas encontram-se as seguintes: coleta de dados, preparação destes dados, escolha do algoritmo, treinamento do algoritmo, avaliação dos resultados e ajustes de parâmetros. Por meio destas etapas busca-se obter um resultado satisfatório na implementação de um modelo de aprendizagem de máquina.

Conforme visto na revisão sistemática, o método preditivo mais mencionado, tanto para implementação quanto nas etapas de comparações de resultados foi o LSTM, visto que este modelo possui resultados muito satisfatórios para previsão de dados em séries temporais. Por outro lado, identificou-se um método que traz ótimos resultados em todas as áreas em que é utilizado, que se trata do XGBoost. Ao realizar a pesquisa de referencial teórico foi visto que um caminho a ser seguido para este trabalho seria a implementação destes dois modelos de sistemas preditivos, o LSTM e o XGBoost, para assim realizar uma comparação entre estes, utilizando o mesmo *dataset*.

Ao realizar-se a análise dos trabalhos relacionados foi constatado que cada trabalho possuía um *dataset* único, pois se tratavam de sistemas preditivos em relação as características climáticas e geográficas de uma região a ser prevista. O fato dos *datasets* não serem públicos, inviabiliza uma comparação direta deste trabalho com o estado da arte. Para realizar a avaliação do resultados precedentes foi comumente utilizada a métrica de *Mean Squared Error* (MSE) que é uma medida de erro absoluto que eleva os desvios obtidos ao quadrado para que assim seja evitado o cancelamento dos valores positivos e negativos. Esta métrica evidencia os erros maiores, buscando assim um resultado que possua o menor erro possível ao longo de toda a previsão.

III. Materiais e Métodos

Buscando alcançar o objetivo proposto para este trabalho foi realizada a implementação de modelos baseados nos al-

Tabela I: Revisão sistemática dos trabalhos relacionados.

Título	Autor(es) e ano	Algoritmo	RMSE
Short-term wind speed forecasting based on long short-term memory and improved BP neural network [8]	Gonggui Chen, Bangrui Tang 2022	LSTM-BP	0.1587
TS-XGB: Ultra-Short-Term Wind Power Forecasting Method Based on Fusion of Time-Spatial Data and XGBoost Algorithm [9]	Jiang Jiading, Wang Feng 2022	TS-XGB	4.8
A novel hybrid model based on nonlinear weighted combination for short-term wind power forecasting [10]	Jiandong Duan, Peng Wang 2022	LSTM/PSO-DBN	42.9055
An advanced short-term wind power forecasting framework based on the optimized deep neural network models [11]	Seyed Mohammad Jafar Jalali, Sajad Ahmadian 2022	Evol-CNN	0.0445
Wind power forecasting based on new hybrid model with TCN residual modification [12]	Jiaojiao Zhu, Liancheng Su 2022	TCN	134.2108
Short-term wind power probabilistic forecasting using a new neural computing approach: GMC-DeepNN-PF [13]	Qianchao Wang, Lei Pan 2022	GMC-DeepNN-PF	56.6893

goritmos LSTM e XGBoost, que se destacaram no estudo do estado da arte. Uma das motivações deste trabalho é produzir conhecimento sobre a situação atual no Brasil em relação a geração de energia eólica.

A. Descrição do Estudo de Caso

Atualmente um dos maiores problemas em relação a energia eólica é a previsão da geração de energia, pois esta depende totalmente das questões meteorológicas da região que se encontra o parque eólico.

Recentemente no dia 8 de julho de 2022 o Brasil teve seu recorde na produção de energia eólica estabelecendo uma produção de 14.167MW de energia o que foi capaz de atender toda região nordeste e ainda possuir um excedente de mais de 23.2% do total da energia produzida, isso só demonstra como esta fonte de energia é tão poderosa. O Brasil teve seu início na exploração desta energia em 2007 e vem crescendo a cada ano tendo sua capacidade atual de produção em 22.000MW desta capacidade 90% está localizada na região nordeste tendo como destaque os estados da Bahia, Rio Grande do Norte, Piauí e o Ceará. Somando a energia total gerada em 2022 estes estados representaram aproximadamente 84%. [15].

No momento, o Brasil ocupa a sexta posição no ranking global de capacidade de geração de energia eólica. Nesse ranking, países como China ocupam o primeiro lugar, com uma capacidade instalada de 310,6 GW, e os Estados Unidos estão em segundo lugar, com uma capacidade instalada de 134,3 GW. [16].

B. Descrição do Método

O presente trabalho consistiu em uma pesquisa de natureza exploratória, a qual visou investigar, compreender e aplicar as técnicas de aprendizado de máquina com os algoritmo de LSTM e o XGBoost no contexto da previsão de energia gerada através de geradores eólicos.

Um processo comumente usado para guiar a aplicação de algoritmos de aprendizagem de máquina foi proposto por Usama Fayyad [17] e é chamado de Processo *Knowledge Discovery and Data mining* (KDD). Este processo possui diversas etapas começando pelos dados que são a parte mais importante, seguindo pelo tratamento destes dados para remoção de erros, viés e registros duplicados, após isto é escolhido o modelo aplicado e assim realizado os treinamentos. Ao final do treinamento são realizadas as avaliações para identificar se o modelo aplicado esta tendo resultados satisfatórios e assim efetuar ajustes em seus parâmetros para melhorar seus resultados obtidos.

Para descrever o processo de aprendizagem de máquina, proposto por Fayyad [17], e reconhecido no estado da arte, apresenta-se a Figura 2, na qual os principais passos são: coleta de dados, preparação de dados, escolher um modelo, treinar este modelo, avaliar os resultados, ajustar parâmetros e obter uma previsão.

O aprendizado de máquina necessita, obrigatoriamente, de dados de treinamento. Os dados, por sua vez, dependem do tipo de problema que está sendo resolvido. Caso o problema seja de classificação ou regressão é necessário que os dados estejam devidamente rotulados. Caso o problema seja de agrupamento, não é necessária a rotulação. Portanto, a coleta de dados é a primeira etapa de um processo de aprendizado de máquina pois sem estes dados nada é possível.

Os dados brutos sozinhos não são muito úteis devido a diversos problemas encontrados neles, tais como valores não normalizados, registros duplicados, valores incorretos e vieses. Portanto, é necessário que os dados sejam preparados na segunda etapa. A visualização dos dados pode ser usada para procurar padrões e valores discrepantes, para ver se os dados corretos foram coletados ou ainda se faltam dados.

Na terceira etapa, é crucial selecionar o modelo mais adequado para a situação em questão, considerando a ampla gama de modelos disponíveis para diversas finalidades. Ao realizar essa escolha, é fundamental garantir que o modelo esteja alinhado com os objetivos das regras de negócio. Além disso, é necessário levar em consideração o nível de preparação exigido pelo modelo, sua precisão e escalabilidade.

O treinamento do modelo constitui a fase central do aprendizado de máquina. Nessa etapa, utiliza-se os dados coletados e preparados para aprimorar gradualmente as previsões do modelo. No caso do aprendizado de máquina supervisionado, o modelo é construído com base em dados de amostra rotulados, permitindo a comparação entre os valores preditos e os valores reais em cada ciclo de treinamento. Por outro lado,

Tabela II: Revisão sistemática dos trabalhos relacionados.

Ano	Produção (GWh)	Acréscimo	Percentual de Produção
2007	645		0,14%
2008	837	+30%	0,18%
2009	1.238	+48%	0,27%
2010	2.177	+76%	0,42%
2011	2.705	+24%	0,51%
2012	5.050	+87%	0,91%
2013	6.576	+30%	1,15%
2014	12.208	+86%	2,1%
2015	21.626	+77%	3,7%
2016	33.488	+55%	5,8%
2017	42.373	+27%	7,2%
2018	48.475	+14,4%	8,1%
2019	55.986	+15,5%	8,9%

Figura 2: Passos de implementação de um algoritmo de aprendizado de máquina.

o aprendizado de máquina não supervisionado busca extrair padrões de dados não rotulados.

Após o treinamento do modelo, segue-se a etapa de avaliação, que envolve testar o modelo de aprendizado de máquina em um conjunto de dados não utilizado durante o treinamento, proporcionando um teste real do desempenho do modelo. Esse teste com dados de controle permite verificar se o modelo apresenta viés ou tendenciosidade, aspectos cruciais para a aplicação do modelo. Vale ressaltar que, em problemas com um maior número de variáveis, é essencial contar com conjuntos de treinamento e teste mais amplos.

Após avaliar o modelo, é necessário validar os parâmetros originalmente definidos no modelo. É possível aumentar o número de ciclos de treinamento para obter resultados mais precisos. No entanto, é essencial estabelecer critérios para determinar quando o modelo é suficientemente bom, evitando assim o ajuste excessivo que pode ocorrer quando o modelo se adapta totalmente aos dados de treinamento, ignorando outros casos não mapeados. Esse é um processo experimental que requer iterações repetidas para obter os melhores resultados.

Por fim, após passar pelas etapas de coleta de dados, preparação dos dados, seleção do modelo, treinamento, avaliação do modelo e ajuste de parâmetros, podemos afirmar que dispomos de uma máquina capaz de aprender e se adaptar a diferentes problemas, podendo ser aplicada em diversos tipos de previsões, como reconhecimento de imagem, reconhecimento de fala e previsão de dados.

C. Materiais

Para que fosse possível alcançar o resultado desejado, esta pesquisa fez uso das seguintes ferramentas e plataformas:

- A plataforma de codificação Colaboratory Google;
- *Dataset* com informações reais de um parque eólico localizado na Turquia;
- Python versão 3.10.11 foi usado como a linguagem de programação;
- API Keras versão 2.12.0, esta é uma API de alto nível que ajuda na construção e treinamento de redes neurais [18];
- Biblioteca Tensorflow versão 2.12.0, esta é uma biblioteca de inteligência artificial de código aberto.

IV. Desenvolvimento

Para realizar este desenvolvimento foram seguidos os passos já descritos e propostos por Fayyad [17] no contexto da descoberta de conhecimento (KDD). Ilustrados na Figura 2.

A. Etapa 1 - Obtendo os Dados

Seguindo as etapas propostas, iniciou-se pela obtenção de dados relacionados a área de geração de energia eólica. A partir de extensa pesquisa, selecionou-se o *dataset* proveniente da plataforma *Kaggle*[2] de disponibilização de *datasets* gratuitamente.

Os dados comumente encontrados possuem como principais informações a velocidade do vento no momento da coleta e a direção deste, assim como a energia gerada naquele instante. Os dados utilizados para implementação são provenientes do *Wind Turbine Scada Dataset*, sendo coletados em um parque eólico da Turquia e são dispostos em intervalos de 10 minutos ao longo do ano de 2018, possuindo assim uma variedade das informações ao longo das quatro estações do ano [19]. Neste *dataset* encontram-se 50.530 instâncias de informações.

Para utilização deste *dataset* os dados se encontram em um arquivo CSV onde cada coluna faz referência a um tipo de informação, sendo estas então definidas como, data e hora, energia gerada, velocidade do vento, energia gerada teórica e direção do vento.

[2]https://www.kaggle.com/

B. Etapa 2 - Pré-processando os Dados

Para realizar o pré-processamento destas informações foi necessário ajustar para que o dado referente a data e hora fosse separada em diferentes colunas sendo estas, dia, mês, hora e minuto. Também foram removidas as colunas data e hora e energia gerada teórica, por não oferecerem uma informação relevante ao processo de previsão ou por já estar sendo utilizada em novas colunas.

Referente aos dados numéricos foi utilizado o recurso de normalização por meio da classe *MinMaxScaler*, da biblioteca *Sklearn Preprocessing*[3]. Com ela foi possível realizar a normalização dos dados de treinamento e teste para uma faixa de 0 a 1. A normalização dos dados é uma etapa importante para garantir que todas as características tenham a mesma escala, o que pode melhorar o desempenho de algoritmos de aprendizado de máquina [20]. Dessa forma, tanto os dados de treinamento quanto os dados de teste são normalizados no intervalo especificado.

No estágio de pré-processamento, determinamos o número de passos a serem considerados ao prever informações futuras. Nesta implementação específica, tomamos a decisão de usar uma previsão avançada de 6 passos, com o objetivo de antecipar e fornecer informações com uma hora de antecedência em relação ao momento atual. Essa estratégia nos permite obter uma visão mais abrangente e precisa das condições futuras, auxiliando na tomada de decisões mais informadas.

Ao final do pré-processamento foram constituídos dois conjuntos de dados para treino e teste, a partir do *dataset* original. O conjunto de treino possuía 70% dos dados, sendo que 10% destes foram utilizados para validação. O conjunto de teste possuía os 30% restantes dos dados.

C. Etapa 3 - Aplicando os Algoritmos

LSTM: Ao aplicar o algoritmo LSTM foram identificadas várias opções para sua construção, incluindo o número de camadas, o tipo de camada, o número de unidades lógicas por camada, a presença de *dropout* entre as camadas, bem como a taxa de *dropout*.

As camadas do tipo LSTM desempenham um papel crucial nas redes neurais recorrentes (RNNs) ao permitir a retenção de informações de longo prazo. Elas são constituídas por células LSTM que possuem um estado de célula interno e três portas (porta de esquecimento, porta de entrada e porta de saída), responsáveis por regular o fluxo de informações. A inclusão de camadas LSTM em um sistema preditivo viabiliza a captura de dependências temporais complexas presentes nos dados, permitindo a retenção de informações significativas a longo prazo.

Já as camadas densas, também conhecidas como camadas totalmente conectadas, desempenham um papel essencial ao mapear as representações de entrada para uma representação final. Cada nó em uma camada densa é conectado a todos os nós da camada anterior. A adição de camadas densas em um sistema preditivo em conjunto com camadas LSTM possibilita a combinação das informações extraídas pelas camadas LSTM e a geração da saída final do modelo, contribuindo para a realização de previsões ou classificações desejadas.

Dada as restrições deste trabalho a seleção da arquitetura apropriada envolveu a condução de uma série de exaustivos experimentos com o objetivo de identificar a configuração que proporcionasse o mais elevado índice de acurácia em relação às previsões desejadas. A Tabela III apresenta as configurações investigadas nos experimentos, que englobam uma variedade de números de camadas, seus tipos, quantidade de neurônios e a inclusão ou não de camadas de *dropouts*.

Tabela III: Valores utilizados na busca por melhor configuração.

Configurações	Valores Testados
Número de Camadas	1, 2, 3, 4, 5
Tipos de Camadas	LSTM, Densa
Número de Neurônios	10, 20, 30, 50, 100, 200
Existência de *Dropout*	Sim, Não

XGBoost: Para o algoritmo XGBoost foram identificados diversos hiperparâmetros que desempenham um papel fundamental na arquitetura de construção das árvores de decisão. Esses parâmetros são ajustados com o propósito de controlar o comportamento do modelo. O *max_depth* controla a profundidade máxima das árvores, enquanto o *learning_rate* determina a taxa de aprendizado do modelo. O parâmetro *gamma* define a redução mínima necessária da função de perda para realizar uma divisão na árvore. O *subsample* controla a proporção de dados de treinamento amostrados para cada árvore, e os parâmetros *colsample_bytree* e *colsample_bylevel* controlam a proporção de colunas selecionadas aleatoriamente para cada árvore e nível da árvore, respectivamente. A configuração adequada desses parâmetros é de extrema importância para alcançar um equilíbrio entre a complexidade do modelo e a capacidade de generalização.

É fundamental ressaltar que a configuração ideal desses parâmetros é intrinsecamente dependente do conjunto de dados em questão e da natureza específica do problema sendo abordado. Em geral, é necessário realizar ajustes e experimentos sistemáticos para encontrar a combinação ótima desses parâmetros que resulte no melhor desempenho do modelo.

Com o objetivo de realizar uma busca exaustiva pela melhor configuração de hiperparâmetros foi empregada a classe *RandomizedSearchCV*[4], proveniente da biblioteca *Sklearn Model_selection*[5]. Esta classe é responsável por realizar buscas randômicas em uma lista pré-definida de parâmetros fornecidos pelo programador. Dessa forma, é possível realizar uma busca cruzada entre diversas opções, visando identificar a combinação que apresenta o melhor desempenho em relação aos valores previstos nos dados de treinamento. A Tabela IV apresenta os hiperparâmetros investigadas nos experimentos.

D. Etapa 4 - Realizando o Treinamento

LSTM: A arquitetura construída para esta implementação se organiza em um formato sequencial, onde as camadas são

[3]https://scikit-learn.org/stable/modules/preprocessing.html

[4]https://encr.pw/RandomizedSearchCV

[5]https://scikit-learn.org/stable/model_selection.html

Tabela IV: Valores utilizados na busca por melhores hiperparâmetros.

Hiperparâmetro	Valores Testados
max_depth	2, 3, 4, 5, 6, 7
learning_rate	0.2, 0.1, 0.01
gamma	0.001, 0.01, 0.1, 1
subsample	0.1, 0.4, 0.7, 1
colsample_bytree	0.1, 0.4, 0.7, 1
colsample_bylevel	0.1, 0.4, 0.7, 1

adicionadas uma após a outra. A camada inicial é composta por uma unidade LSTM com 50 unidades de processamento, projetada para lidar com sequências de entrada de dados unidimensionais. Para regularização e evitar o *overfitting*, uma camada *dropout* com uma taxa de 20% é inserida após a camada LSTM, desativando aleatoriamente uma fração dos neurônios durante o treinamento. A seguir, três camadas densas são adicionadas, com 20, 10 e 5 unidades, respectivamente, permitindo que o modelo aprenda representações mais complexas e capturar correlações entre os dados. A camada final é uma camada densa com uma unidade de saída, ativada pela função *Rectified Linear Unit* (ReLU), apropriada para problemas de regressão para previsão de valores positivos.

O modelo foi compilado utilizando o otimizador Adam e a função de perda *Mean Squared Error* (MSE), comumente aplicada em problemas de regressão para medir o erro médio quadrático entre as previsões do modelo e os valores reais. A figura 3 apresenta um resumo do modelo sequencial LSTM empregado, destacando as camadas e o total de parâmetros de cada camada.

Figura 3: Sumário do modelo LSTM empregado.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50)                10400

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 20)                1020

 dense_1 (Dense)             (None, 10)                210

 dense_2 (Dense)             (None, 5)                 55

 dense_3 (Dense)             (None, 1)                 6

=================================================================
Total params: 11,691
Trainable params: 11,691
Non-trainable params: 0
```

Durante o treinamento do LSTM foi empregado o método *EarlyStopping*[6], responsável por monitorar o progresso do modelo a cada época e interromper o treinamento de acordo com uma condição ou critério pré-estabelecido. Nesta implementação específica, configurou-se o *EarlyStopping* para encerrar o treinamento após 10 épocas consecutivas sem melhorias nos resultados. O erro mínimo aceito também compõe o conjunto de hiperparâmetros pré-definidos.

[6]https://l1nq.com/tensorflow-EarlyStopping

XGBoost: Para o treinamento do modelo XGBoost foi imprescindível determinar a combinação ótima de hiperparâmetros capaz de fornecer os resultados mais precisos em relação aos valores reais e previstos. Nesse sentido, utilizou-se a classe *RandomizedSearchCV*[7], a qual foi alimentada com um conjunto de valores conforme apresentado na Tabela IV. Por meio dessa classe, múltiplas simulações foram executadas a fim de identificar os hiperparâmetros que conduzissem ao menor erro. É relevante destacar que, devido ao elevado número de hiperparâmetros considerados foram geradas 4.608 possíveis combinações, tornando-se inviável para a classe examinar todas elas. Por essa razão foram selecionadas aleatoriamente combinações para serem testadas e, ao final, apenas um percentual pré-determinado dessas combinações foi avaliado.

Foram identificados para o presente modelo em questão os seguintes valores de hiperparâmetros, conforme apresentado na Tabela V, que resultaram na obtenção da menor taxa de erro durante o treinamento e avaliação do modelo.

Tabela V: Valores de hiperparâmetros com melhor resultado.

Hiperparâmetro	Valor Selecionado
max_depth	4
learning_rate	0.1
gamma	1.0
subsample	0.1
colsample_bytree	1.0
colsample_bylevel	1.0

E. Etapa 5 - Avaliando os Modelos

Para a etapa de avaliação do modelo utilizou-se como métrica o erro médio quadrático (MSE). Esta é uma métrica amplamente utilizada para avaliar o desempenho de modelos de regressão. Ela calcula a média dos quadrados das diferenças entre os valores preditos e os valores reais do conjunto de dados, evidenciando os erros maiores, fornecendo uma medida que reflete a qualidade geral da predição. Ao minimizar o MSE, durante o treinamento do modelo, busca-se melhorar as predições dos dados.

Como foi utilizado do mesmo *dataset* e aplicação do mesmo pré-processamento foi possível realizar a comparação direta entre os dois modelos construídos, aplicando assim a métrica MSE. A Figura 4 ilustra a comparação direta entre quatro modelos de LSTM treinados com arquiteturas distintas durante o processo de busca da arquitetura ótima.

V. Resultados

Buscando finalizar o estudo em relação aos dois modelos desenvolvidos são apresentado os resultados obtidos em relação aos dados de testes separados.

Os resultados obtidos demonstraram que o modelo LSTM registrou um MSE de 0.1254, ao passo que o modelo XGBoost obteve um valor de 0.1571. Esses resultados indicam que o modelo LSTM apresentou uma maior acurácia em relação ao modelo XGBoost. Os valores de MSE para validação e teste dos dois modelos são apresentados na Tabela VI, utilizando uma abordagem percentual para dados normalizados.

[7]https://encr.pw/RandomizedSearchCV

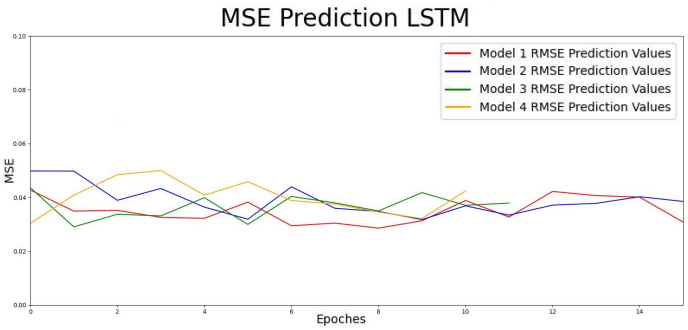


Figura 4: Comparação de MSE do treinamento de quatro diferentes arquiteturas para o modelo LSTM.

Tabela VI: Resultados obtidos nos valores de MSE através dos modelos treinados.

Modelo	Validação	Teste
Long Short Term Memory	2.59%	12.54%
Extreme Gradient Boosting	4.20%	15.71%

Na fase de treinamento e validação foi observada uma taxa de erro menor em comparação com a fase de testes. Isso pode ser atribuído ao comportamento do modelo que consistem em buscar uma compreensão completa dos dados fornecidos. Essa observação ressalta a importância de não utilizar os dados de teste nessa etapa, pois os modelos podem ser facilmente influenciados a se ajustarem especificamente a esses dados de teste, comprometendo a generalização do modelo.

Foi constatado que o modelo LSTM apresenta uma fase de treinamento significativamente mais lenta em comparação com o modelo XGBoost, podendo chegar a ser até 10 vezes mais demorada. No entanto, a fase de busca de hiperparâmetros do modelo XGBoost é extremamente mais lenta do que a do LSTM.

Por meio da análise da Figura 5, pode-se observar as previsões geradas pelo modelo baseado em LSTM. Verifica-se que, em geral, as previsões apresentam alta precisão ao longo de todas as etapas, com os valores reais em vermelho ficando muito próximos dos valores previstos em azul, com exceção de alguns momentos em que os valores reais exibiram níveis superiores. A Figura 6 apresenta um gráfico com uma menor quantidade de etapas, proporcionando uma visualização mais clara dos dados.

Ao analisar o modelo baseado em XGBoost, pode-se observar por meio da Figura 7 que as previsões geralmente apresentam uma maior frequência de divergência em relação aos valores reais. Os desvios entre os valores previstos e reais são mais pronunciados. Para uma melhor visualização dos dados, a Figura 8 exibe um gráfico com uma menor quantidade de etapas.

VI. Considerações Finais

O presente trabalho consistiu em uma pesquisa de natureza exploratória, a qual visou investigar, compreender e aplicar as técnicas de aprendizado de máquina. Foi realizado uma revisão sistemática de trabalhos relacionados para identificação do estado da arte. Com esta revisão foi possível identificar os algoritmo *Long Short Term Memory* (LSTM) e o *Extreme Gradient Boosting* (XGBoost) que possuíram maior influência no contexto da previsão de energia gerada através de geradores eólicos.

A partir da implementação e aplicação dos algoritmos investigados foi observado que os modelos preditivos fundamentados na LSTM demonstraram uma maior acurácia em relação à previsão de séries temporais. Este resultado foi considerado satisfatório em comparação ao apresentado no trabalho de Asati e Akshita [21] onde os resultados também demonstraram que o LSTM possui melhor precisão em relação ao XGBoost para séries temporais.

A criação do modelo preditivo baseado na LSTM foi facilitada pelo fato que existe abundância de conteúdo disponível sobre essa técnica, sendo criado o seu conceito em 1997 [22]. Por outro lado, o algoritmo XGBoost foi desenvolvido em uma data mais recente, especificamente em 2014 [23]. Portanto, é possível observar que a LSTM possui uma base de conhecimento estabelecida ao longo do tempo, com uma extensa literatura e recursos educacionais disponíveis, o que contribuiu para uma implementação mais eficiente e compreensão aprofundada do modelo. Por outro lado, o XGBoost é um algoritmo relativamente novo em comparação ao LSTM, o que implica em um período de tempo mais curto para o desenvolvimento de recursos e conhecimentos especializados relacionados a essa técnica específica.

Verificou-se que a construção de um modelo preditivo requer a adoção de etapas metodologicamente estruturadas, a fim de obter resultados satisfatórios. Enfrentaram-se diversas dificuldades relacionadas ao pré-processamento dos dados e à sua utilização no treinamento, resultando em ocasiões em que o modelo foi treinado com valores incorretos.

Foi constatada uma necessidade de maior poder computacional para a execução da busca pelos hiperparâmetros ótimos para o modelo XGBoost, podendo ter o custo de horas de processamento de cálculos nesta busca.

Como trabalhos futuros, sugere-se aumentar o número de instancias do conjunto de dados a fim de melhorar os resultados na previsão; aprofundar os estudo sobre modelos complexos onde exista a combinação de diferentes algoritmos e, assim, buscar melhores resultados; utilizar-se outras arquiteturas de *machine learning*, como a *AutoRegressive Integrated Moving Average* (ARIMA) e *Random Forest*, comparando os resultados com os encontrados neste estudo.

Referências

[1] “Crise energética no Brasil.” https://mundoeducacao.uol.com.br/geografia/crise-energetica-no-brasil.htm. Accessed: 2022-10-22.
[2] IEA, “Dados sobre energia no Brasil.” https://www.iea.org/countries/brazil, 2022.
[3] “Brasil sobe posição em ranking global de produção de energia eólica.” https://www.cnnbrasil.com.br/business/brasil-sobe-posicao-em-ranking-global-de-producao-de-energia-eolica/. Accessed: 2022-10-22.
[4] “Nordeste bate novo recorde de geração de energia eólica, aponta ONS.” https://www.cnnbrasil.com.br/business/nordeste-bate-novo-recorde-de-geracao-de-energia-eolica-aponta-ons/. Accessed: 2022-10-22.
[5] R. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*. USA: Academic Press Professional, Inc., 1996.
[6] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

Figura 5: Gráfico comparando valores de energia gerada reais e previstos através do modelo LSTM.

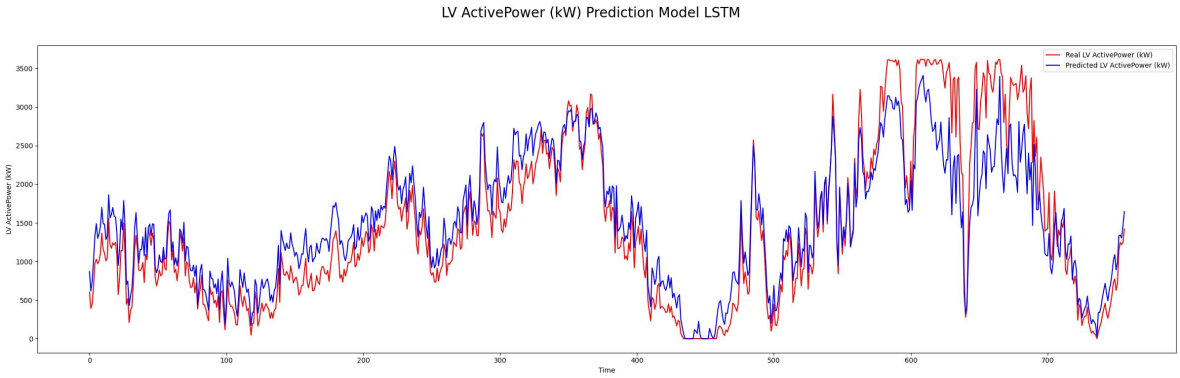


Figura 6: Gráfico comparando valores de ampliados energia gerada reais e previstos através do modelo LSTM.

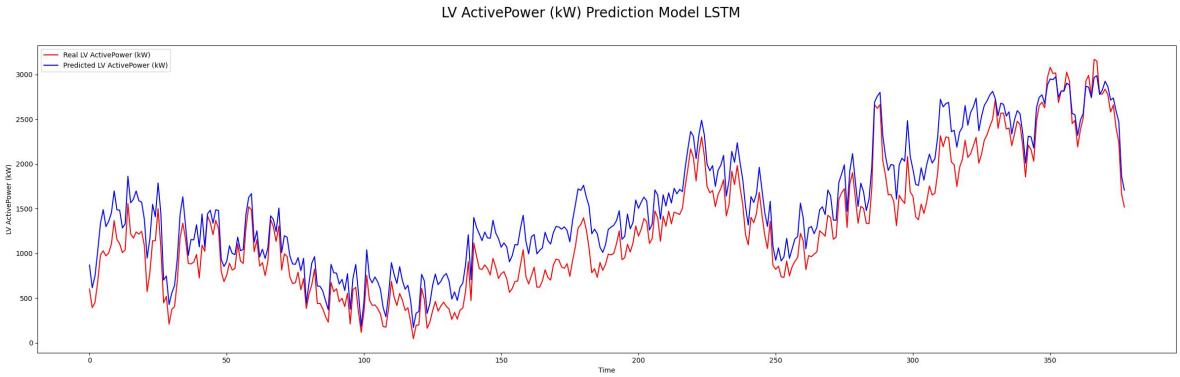


Figura 7: Gráfico comparando valores de energia gerada reais e previstos através do modelo XGBoost.

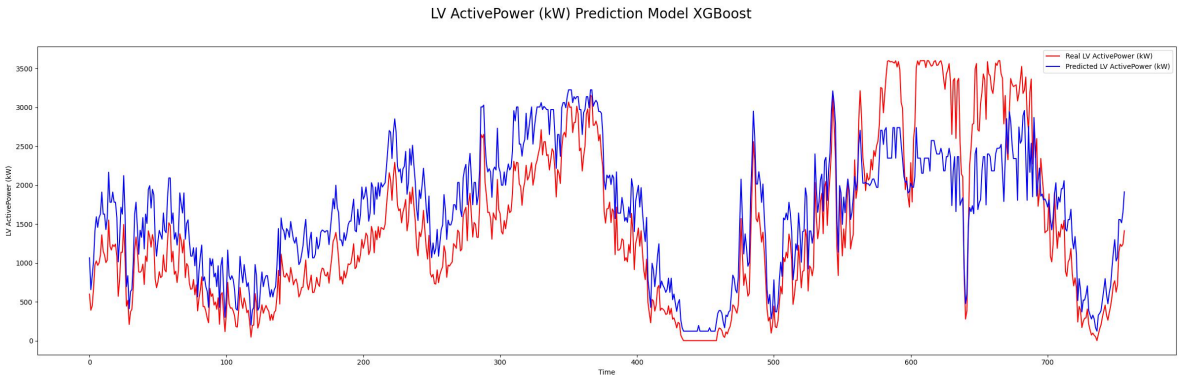
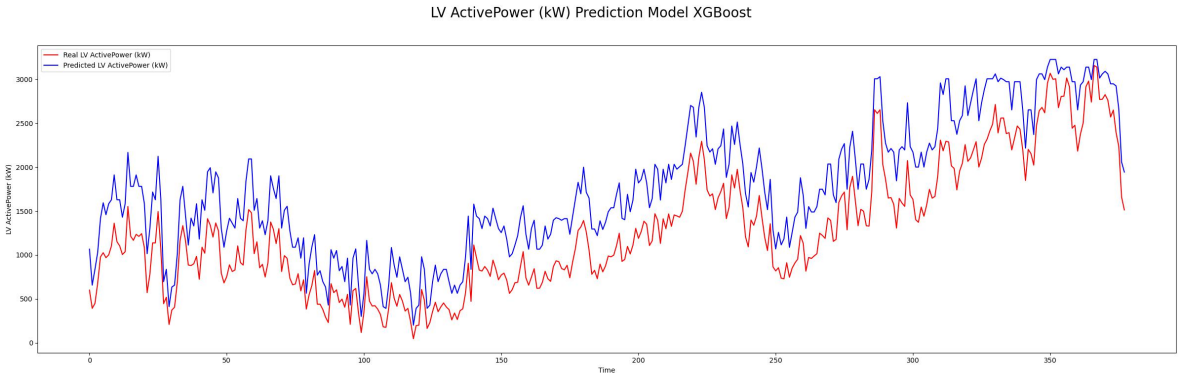


Figura 8: Gráfico comparando valores ampliados de energia gerada reais e previstos através do modelo XGBoost.



[7] R. Sampaio and M. Mancini, “Estudos de revisão sistemática: Um guia para síntese criteriosa da evidência científica,” *Brazilian Journal of Physical Therapy*, vol. 11, pp. 1–5, 2007.
[8] X. Z. P. Z. P. K. H. L. Gonggui Chen, Bangrui Tang, “Short-term wind speed forecasting based on long short-term memory and improved bp neural network,” *Science Direct*, vol. 134, no. 107365, 2022.
[9] T. R. Z. L. X. X. Jiang Jiading, Wang Feng, “Ts-xgb:ultra-short-term wind power forecasting method based on fusion of time-spatial data and xgboost algorithm,” *Science Direct*, vol. 199, pp. 1103–1111, 2022.
[10] W. M. S. F. Z. H. Jiandong Duan, Peng Wang, “A novel hybrid model based on nonlinear weighted combination for short-term wind power forecasting,” *Science Direct*, vol. 134, no. 107452, 2022.
[11] M. K. A. K. M. S.-k. S. N. J. P. C. Seyed Mohammad Jafar Jalali, Sajad Ahmadian, “An advanced short-term wind power forecasting framework based on the optimized deep neural network models,” *Science Direct*, vol. 141, no. 108143, 2022.
[12] Y. L. Jiaojiao Zhu, Liancheng Su, “Wind power forecasting based on new hybrid model with tcn residual modification,” *Science Direct*, vol. 10, no. 100199, 2022.
[13] H. W. X. W. Y. Z. Qianchao Wang, Lei Pan, “Short-term wind power probabilistic forecasting using a new neural computing approach: Gmc-deepnn-pf,” *Science Direct*, vol. 126, no. 109247, 2022.
[14] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.
[15] “Energia eólica registra primeiro recorde de geração instantânea de 2022.” https://www.gov.br/pt-br/noticias/energia-minerais-e-combustiveis/2022/08/energia-eolica-registra-primeiro-recorde-de-geracao-instantanea-de-2022. Accessed: 2022-10-22.
[16] “Eólicas ganham espaço e Brasil sobe em ranking internacional.” https://www.udop.com.br/noticia/2022/04/08/eolicas-ganham-espaco-e-brasil-sobe-em-ranking-internacional.html. Accessed: 2022-10-22.
[17] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, *et al.*, “Knowledge discovery and data mining: Towards a unifying framework.,” in *KDD*, vol. 96, pp. 82–88, 1996.
[18] “Keras: the python deep learning api.” https://keras.io/. Accessed: 2022-07-30.
[19] B. ERISEN, “Wind turbine scada dataset.” https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset, 2018.
[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[21] A. Asati, “A comparative study on forecasting consumer price index of India amongst XGBoost, Theta, ARIMA, Prophet and LSTM algorithms,” *OSF Preprints*, no. hyqsb, 2022.
[22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[23] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.

Mauricio Gerhardt é futuro bacharel em Engenharia da Computação pela Universidade de Caxias do Sul (2023).

Carine Geltrudes Webber é doutora em Ciência da Computação pela École Doctorale Mathématiques et Informatiques, da Universite de Grenoble I Joseph Fourier, França (2003). Mestre (UFRGS) e Graduada (UCS) em Ciência da Computação. Atua como Professora Titular na Área de Conhecimento de Exatas e Engenharias da UCS. Desenvolve projetos de pesquisa na área de Machine e Deep Learning, aplicada aos setores indústrias e de serviços. Integra o Programa de Pós-graduação em Ensino de Ciências e Matemática. Atua na área Informática aplicada ao Ensino e Inteligência Artificial. Participa da REDE AIRES para o uso da IA de forma ética, responsável e sustentável.