UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Faculdade de Engenharia e Ciências de Guaratinguetá

MATHEUS VINÍCIUS RESENDE NASCIMENTO

Análise e predição de falhas em aerogeradores utilizando *deep learning*

Guaratinguetá
2023

Matheus Vinícius Resende Nascimento

Análise e predição de falhas em aerogeradores utilizando *deep learning*

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia elétrica da Faculdade de Engenharia e Ciências do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia elétrica .

Orientador: Profº Dra. Paloma Maria Silva Rocha Rizol

Guaratinguetá
2023

Nascimento, Matheus Vinícius Resende
N244a Análise e predição de falhas em aerogeradores utilizando deep learning / Matheus Vinícius Resende Nascimento– Guaratinguetá, 2023.
66 f : il.
Bibliografia: f. 65-66

Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2023.
Orientadora: Profª. Drª. Paloma Maria Silva Rocha Rizol

1. Turbinas eólicas. 2. Energia eólica. 2. Parque eólico.
4. Energia - Fontes alternativas. I. Título.

CDU 620.91

Luciana Máximo
Bibliotecária/CRB-8 3595

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE GUARATINGUETÁ

MATHEUS VINÍCIUS RESENDE NASCIMENTO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE **"GRADUANDO EM ENGENHARIA ELÉTRICA "**

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Profº Dr. DANIEL JULIEN BARROS DA SILVA SAMPAIO
Coordenador

BANCA EXAMINADORA:

Profº Dra. Paloma Maria Silva Rocha Rizol
Orientador/UNESP-FEG

Profº Dr. Daniel Julien Barros da Silva Sampaio
UNESP-FEG

Profº Mestrando Cristóvão José Dias da Cunha
Membro Externo

Janeiro , 2023

DADOS CURRICULARES

MATHEUS VINÍCIUS RESENDE NASCIMENTO

NASCIMENTO 23/01/1999 - Taubaté / SP

FILIAÇÃO Daniela Resende Nascimento
Rogério de Melo Nascimento

À minha família, que sempre me apoiou e não mediu esforços para que eu chegasse até esta etapa da minha vida. À minha namorada, que sempre me ajuda, não importam as circunstâncias. A todos os professores da FEG-UNESP pela contribuição na minha formação. Aos meus amigos que a graduação proporcionou, que serão levados para toda a vida.

“A tarefa não é tanto ver aquilo que ninguém viu,
mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.“
(Arthur Schopenhauer)

RESUMO

Estamos vivendo em um período de transição energética, motivado por vários indicadores que comprovam que as ações humanas estão afetando o clima terrestre, no qual governos então reunindo esforços a fim de aumentar a participação de fontes renováveis de energia elétrica em suas matrizes energéticas, para reduzir as emissões de gases de efeito estufa. Neste contexto, o Brasil possui um grande potencial de energia eólica a ser explorado. Ao longo dos últimos 5 anos, a participação da energia eólica cresceu 53,67%, sendo responsável por 11,4% da capacidade instalada da matriz energética brasileira. Nesse cenário, este trabalho propõe o desenvolvimento de um modelo preditivo para auxiliar a operação e manutenção de parques eólicos localizados no Brasil. O objetivo é encontrar um modelo de *deep learning* capaz de prever falhas em aerogeradores utilizando dados de vibração e potência captados pelos sensores internos do aerogerador, além de comparar outras técnicas de *machine learning*. A resolução do problema foi dividida em três etapas sendo exploração, escolha do modelo e previsão. Para a criação do modelo, foram avaliadas algumas técnicas de aprendizado não supervisionado e semi supervisionado, como *isolation forest*, *one class support vector machine* e os *autoencoders*. Todos os modelos gerados foram avaliados utilizando as métricas de acurácia, precisão, *recall*, F_1 *score* e AUC, sendo que o modelo que apresentou melhor resultado foi o *deep autoencoder* esparso, uma variação do *autoencoder* tradicional.

PALAVRAS-CHAVE: *Deep learning*; Vibração; *Autoencoder*; Aerogerador.

ABSTRACT

We are living in a period of energy transition, motivated by several indicators that prove that human actions are affecting the Earth's climate, in which governments are joining efforts in order to increase the share of renewable sources of electricity in their energy mixes, to reduce greenhouse gas emissions. In this context, Brazil has a great potential for wind energy to be explored. Over the past 5 years, the share of wind energy has grown by 53.67%, accounting for 11.4% of the installed capacity of the Brazilian energy matrix. In this scenario, this work proposes the development of a predictive model to assist the operation and maintenance of wind farms located in Brazil. The objective is to find a deep learning model capable of predicting failures in wind turbines using vibration and power data captured by the internal sensors of the wind turbine, in addition to comparing other machine learning techniques. The resolution of the problem was divided into three steps: exploration, model choice and prediction. To create the model, some unsupervised and semi-supervised learning techniques were evaluated, such as isolation forest, one class support vector machine and autoencoders. All generated models were evaluated using the metrics of accuracy, precision, recall, F_1 score and AUC, and the model that presented the best result was the sparse deep autoencoder, a variation of the traditional autoencoder.

KEYWORDS: Deep learning; Vibration; Autoencoder; Wind turbine.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

UNESP	Universidade Estadual Paulista
CMS	Condition Monitoring System
O&M	Operation and Maintenance
API	Application Programming Interface
IA	Inteligência Artificial
ML	Machine Learning
ANN	Redes Neurais Artificiais
DNN	Redes Neurais Profundas
SVM	Máquina de Vetores de Suporte
OCSVM	One Class Support Vector Machines

SUMÁRIO

1 INTRODUÇÃO

A energia elétrica se tornou um recurso indispensável no último século para a humanidade e seu crescimento seguiu um ritmo acelerado nos últimos anos. No Brasil, a Empresa de Pesquisas Energéticas (EPE) destaca um crescimento de 4,2% no consumo de 2021 em relação ao ano anterior. Contudo, ainda há um grande desafio na geração da energia elétrica, buscando o uso de recursos renováveis que seguem os direcionamentos mundiais para a redução da emissão dos gases de efeito estufa.

O Brasil possui uma matriz energética majoritariamente composta por fontes renováveis, sendo a maior contribuição oriunda dos recursos hídricos. Outras fontes renováveis como a geração fotovoltaica e a eólica estão ganhando destaque devido às incertezas relacionadas à escassez hídrica no país ao longo dos anos.

Direcionando o escopo do trabalho ao estudo da energia eólica no Brasil, é importante ressaltar que houve um crescimento da geração de 53,67% nos últimos 5 anos. Em 2021, o Brasil foi o 4º país que mais produziu energia eólica no mundo[1] apenas atrás da China, Estados Unidos e Alemanha. Nesse mesmo ano, a geração total foi de 72,286 GWh[2], responsável por 11,4% da capacidade total instalada no Brasil. Os estados brasileiros com maior geração são Rio Grande do Norte, Bahia, Piauí, Ceará e Rio Grande do Sul. De acordo com o último boletim anual da ABEEólica (ABEEÓLICA, 2022), o ano de 2021 terminou com um número igual a 795 parques eólicos instalados.

A grande quantidade de parques necessitam de um amplo esforço por parte das empresas para garantir a operação e manutenção dos aerogeradores. Segundo Antonio Romeiro Camacho (ROMERO et al., 2016), os custos relativos à operação e manutenção (O&M) constituem uma grande parcela dos custos anuais de um aerogerador e, para uma nova turbina, os custos podem facilmente representar de 20% a 25% dos custo nivelado total por kWh produzido durante todo o ciclo de vida da turbina.

De acordo com James Carroll (CARROLL et al., 2018), as *gearboxes* (caixas de velocidade) são um dos maiores causadores de tempo de inatividade e de custos de reposição de todos os componentes de aerogeradores.

> As *gearboxes* de aerogeradores tendem a falhar mais prematuramente em comparação com outras aplicações. Uma *gearbox* nem sempre alcança a desejada vida útil de 20 anos, falhando prematuramente entre 2 e 11 anos. (ROMERO et al., 2016, p. 1)

---

[1] Our World in Data, **Renewable energy**. 2022. Disponível em: <https://ourworldindata.org/renewable-energy>.

[2] Empresa de Pesquisa energética **BALANÇO ENERGÉTICO NACIONAL 2022**. 2022. Disponível em: <https://www.epe.gov.br/pt>.

> Dado que as falhas nos rolamentos das *gearboxes* levam a manutenções não planejadas dispendiosas e tempo de inatividade da turbina, há uma oportunidade de reduzir os custos de operação e manutenção em usinas eólicas, prevendo com precisão a probabilidade de falha e implementando soluções para reduzir a frequência dessas falhas. (OFFICE, 2020)

1.1 OBJETIVOS

O objetivo principal deste trabalho é desenvolver um modelo de previsão de falhas em aerogeradores utilizando algoritmos de *deep learning* para turbinas eólicas localizadas no Estado do Rio Grande do Norte, no Brasil, utilizando os dados de vibração e potência captados pelos sensores das turbinas. Durante as várias fases deste trabalho, outros objetivos serão atingidos como a comparação entre modelos de predição e técnicas de *machine learning* e, também, auxiliar na tomada de decisões sobre paradas para manutenções preventivas a fim de diminuir custos na produção de energia.

1.2 JUSTIFICATIVAS

Todos os anos, as mudanças climáticas estão trazendo consequências severas em todo o planeta. Dentre elas estão as ondas de calor, as inundações, o aumento do nível do mar, os incêndios frequentes, entre outros. O principal indicativo é que estes fenômenos estão sendo causados devido ao aquecimento global antropogênico, causado principalmente pelas altas emissões de gases de efeito estufa na atmosfera.

De acordo com o Painel Intergovernamental para Mudanças Climáticas (IPCC), a temperatura global já aumentou 1,2ºC desde a Revolução Industrial e deve atingir 1,5ºC entre os anos de 2030 e 2052. A Organização Meteorológica Mundial (OMM) destacou que nos últimos 50 anos o número de desastres como inundações e ondas de calor causados pelas mudanças climáticas aumentou em 5 vezes, matando mais de 2 milhões de pessoas em todo o mundo.

Diante destes fatos, a Conferência das Nações Unidas sobre Mudanças Climáticas (COP26) realizada em 2021 firmou medidas para frear o aquecimento global. Dentre as medidas, destaca-se o objetivo de que, até 2030, haja participação de 45 a 50% das energias renováveis na composição da matriz energética global.

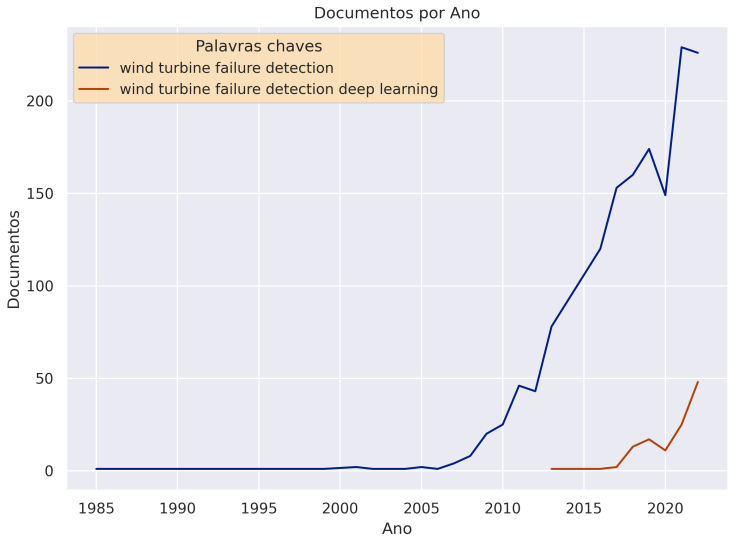
Aliados a esse objetivo, na Agenda 2030 da ONU realizada em setembro de 2015, foram definidos outros 17 objetivos a serem cumpridos até o ano de 2030 para um desenvolvimento sustentável (ODS) do planeta. Especificamente, o objetivo 7 tem como definição assegurar o acesso confiável, sustentável, moderno e a preço acessível à energia para todas e todos[3]. O item 7.2 dita que deve haver um aumento substancial da participação das energias renováveis na matriz energética global.

Este trabalho tem como justificativa principal a elaboração de um método que possa atender aos direcionamentos mundiais para a transição energética, focando em uma solução que utiliza inteligência

[3] <https://brasil.un.org/pt-br>

artificial para diminuir os custos de operação de aerogeradores, possibilitando uma energia elétrica mais barata e mais acessível.

Figura 1 – Pesquisa de trabalhos na plataforma Scopus

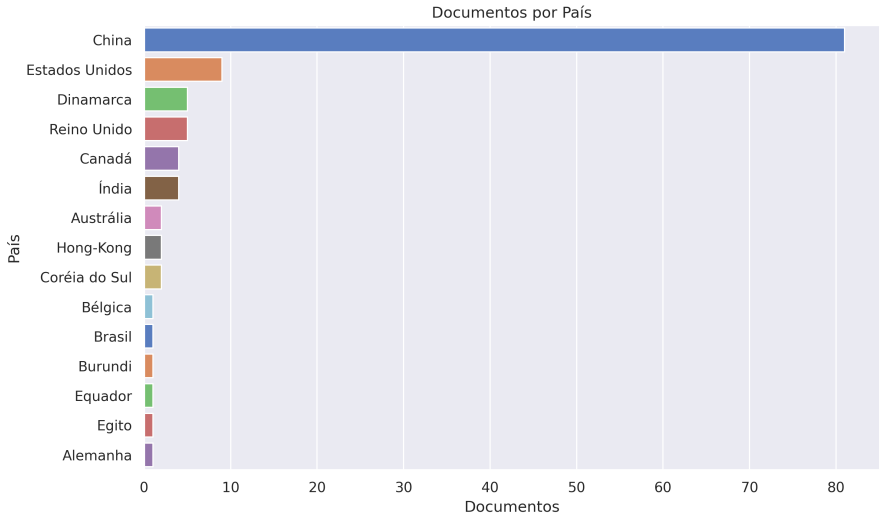


Fonte: Scopus (2022).

Na plataforma Scopus, foi efetuada uma pesquisa utilizando as palavras chaves: "*wind*, *turbine*, *failure*, *detection*" e foram encontrados 1645 resultados. Realizando uma pesquisa específica utilizando também os termos *deep learning* foram obtidos 119 resultados, como ilustra a Figura 1.

A Figura 2 indica a quantidade de documentos por país para a pesquisa específica. Fica evidente que a China é a maior produtora de trabalhos neste tema.

Figura 2 – Países com publicações com as palavras-chave "*wind turbine failure detection deep learning*"



Fonte: Scopus (2022).

Analisando os artigos mais citados da pesquisa específica na plataforma Scopus. O primeiro trabalho com o título *Multiscale Convolutional Neural Networks for Fault Diagnosis of Wind Turbine Gearbox* (JIANG et al., 2019) tem 421 citações e utiliza redes neurais convolucionais multiescala em

um conjunto de dados das vibrações puras para realizar uma extração de *features* e classificar a saída simultaneamente. Os resultados obtidos demonstraram uma superioridade deste tipo de modelo em comparação com redes neurais convolucionais tradicionais.

O trabalho com o título *Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis* (JIANG et al., 2017) possui 196 citações e propõem uma abordagem utilizando um modelo de redes neurais profundas chamado *stacked multilevel-denoising autoencoder* capaz de filtrar ruídos indesejados dos dados de espectro de frequência puro das vibrações e aprender as *features* mais importantes para diagnosticar falhas. Os resultados apontaram que esta abordagem permitiu aprender de forma mais robusta e discriminatória as representações das falhas atingindo uma melhor acurácia comparada às abordagens tradicionais.

O trabalho com o título *Anomaly detection and fault analysis of wind turbine components based on deep learning network* (ZHAO et al., 2018) possui 160 citações aborda um método utilizando redes *deep autoencoder* em auxílio com controle de operação supervisionada e aquisição de dados (SCADA) dos aerogeradores. Também é utilizado o algoritmo da Máquina de Boltzmann restrita como entrada para o modelo da rede neural.

1.3 DELIMITAÇÕES DA PESQUISA

Dentro do contexto apresentado, este trabalho foi delimitado ao uso do método de aprendizado semi supervisionado e não supervisionado, não sendo utilizadas outras técnicas de *machine learning*.

Os estudos desenvolvidos neste trabalho serão aplicados a uma empresa brasileira no segmento de geração e comercialização de energia eólica e, portanto, os resultados não serão generalizados para outros segmentos. Além disso, as limitações técnicas serão as bibliotecas livres disponíveis para a linguagem de programação Python.

1.4 ESTRUTURA DO TRABALHO

O trabalho está estruturado em 4 capítulos. O primeiro capítulo constitui a introdução do trabalho. No capítulo 2, será apresentado o referencial teórico contendo os temas pertinentes ao desenvolvimento do trabalho. O capítulo 3 abordará a descrição e a modelagem do problema e, por fim, o capítulo 4 apresentará as conclusões e possíveis continuações do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo abordará os conceitos da linguagem de programação Python aplicada à ciência de dados e *frameworks* disponíveis para *machine learning* (ML), os algoritmos de detecção de anomalias e, também, as métricas relevantes para a avaliação dos modelos obtidos.

2.1 PYTHON APLICADA A CIÊNCIA DE DADOS

A Linguagem Python é atualmente uma das principais linguagens de programação utilizadas para a ciência de dados (*data science*) devido à facilidade em sua sintaxe de alto nível, uma grande quantidade de bibliotecas disponíveis para as mais diversas aplicações como *data science*, *web applications* e *game development* e uma grande comunidade que desenvolve soluções e disponibiliza-as na internet.

Os índices de popularidade indicam o Python como a linguagem mais popular nos últimos anos, como ilustra a Figura 3 e a Figura 4,devido ao grande sucesso de algumas bibliotecas, tais como:

- **numpy**: O numpy[1] é uma biblioteca de código aberto (*open-source*) para computação científica que disponibiliza objetos do tipo vetor com n-dimensões e vários objetos derivados como vetores e matrizes, além de rotinas de ordenação para rápidas operações com vetores, incluindo operações matemáticas, lógicas, manipulação de dimensão, entre outros. É muito utilizada como sub pacote de outras bibliotecas por oferecer uma alta performance. O numpy também foi uma importante parte da pilha de *softwares* responsáveis pela descoberta das ondas gravitacionais e da primeira imagem de um buraco negro (HARRIS et al., 2020).

- **pandas**: O pandas[2] é uma biblioteca que provê rápidas, flexíveis e expressivas estruturas de dados modeladas para trabalhar com dados relacionais ou classificados de maneira fácil e intuitiva. O pandas é indicado para trabalhar com diferentes tipos de dados, como dados tabulares com heterogenia dos dados (como tabelas SQL e planilhas), séries temporais ordenadas ou não ordenadas, matrizes de dados arbitrários com linhas e colunas, quaisquer outras formas de dados observacionais/estatísticos.

  Os dois tipos de estruturas primárias são as *Series* (1 dimensão) e o *DataFrame* (2 dimensões). As funcionalidades implementadas para essas estruturas permitem fazer uma análise exploratória e o pré-processamento dos dados.

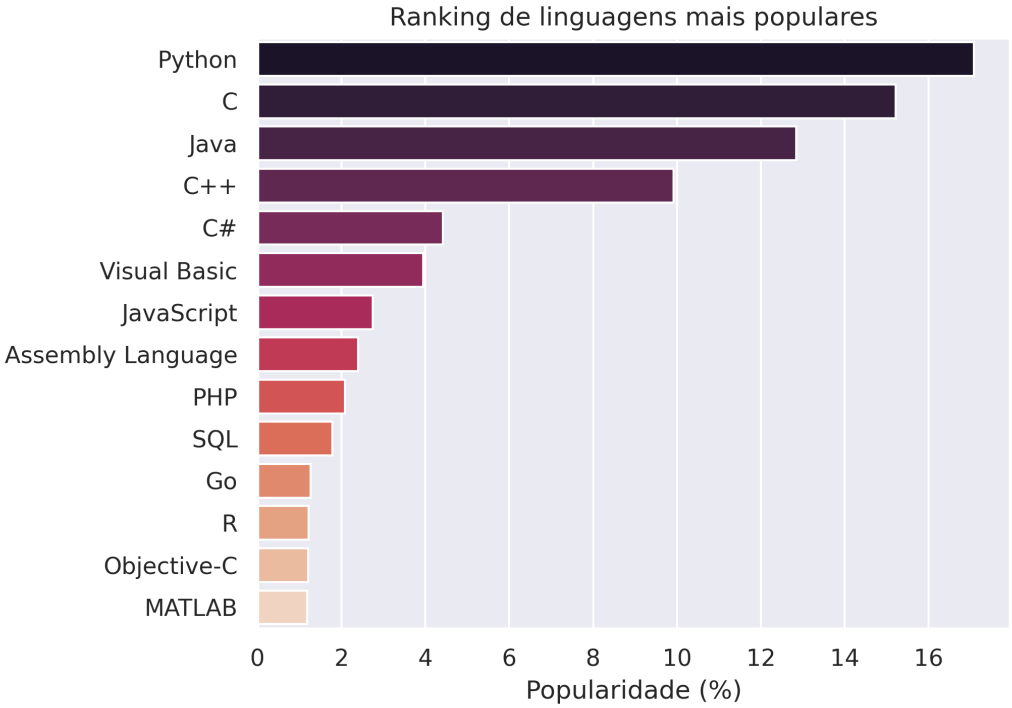
- **matplotlib** e **seaborn**: O matplotlib[3], como as outras bibliotecas mencionadas, é *open-source* e faz parte do conjunto de ferramentas que tornam o Python uma ferramenta completa para ciência de dados. O matplotlib oferece uma API completa para traçar gráficos, e em conjunto com a biblioteca seaborn[4], disponibiliza uma interface de fácil utilização para visualização de dados estatísticos.

[1] <https://numpy.org>
[2] <https://pandas.pydata.org>
[3] <https://matplotlib.org>
[4] <https://seaborn.pydata.org>

- **scikit-learn**: O scikit-learn[5] é uma biblioteca de *machine learning open-source* que suporta modelos supervisionados e não supervisionados. Ela também possui uma variedade de ferramentas para treinamento de modelos, pré-processamento de dados, seleção de modelos, avaliação de modelos e outras utilidades.

- **keras**: A biblioteca Keras[6] é uma API *open-source* de *deep learning* escrita em Python, rodando sobre a plataforma de *machine learning* TensorFlow. Ela possibilita uma fácil construção de modelos complexos de *deep learning* utilizando o conceito de camadas em um modelo sequencial. O Keras é amplamente utilizado pela comunidade acadêmica e pela indústria. Somente no ano de 2021 houve mais de um milhão de usuários individuais.

Figura 3 – Classificação das linguagens mais populares



Fonte: TIOBE Index (2022).

A combinação de uma comunidade muito ativa e uma extensa seleção de bibliotecas fazem com que o python seja uma das melhores linguagens atualmente para trabalhar com *machine learning*.

2.2 INTELIGÊNCIA ARTIFICIAL APLICADA A DETECÇÃO DE ANOMALIAS

Essa seção abordará o que é inteligência artificial e seus subcampos e, também, o significado de anomalias e como detectá-las.

[5] <https://scikit-learn.org>
[6] <https://keras.io>

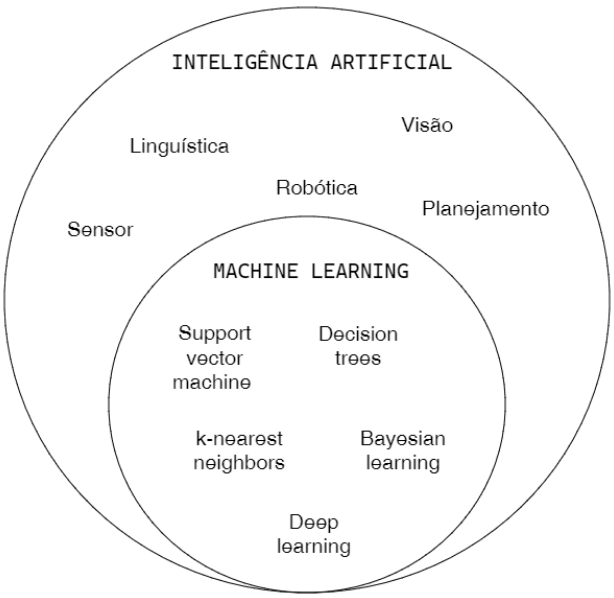
Figura 4 – Classificação do Python nos últimos anos

Fonte: TIOBE Index (2022).

2.2.1 Inteligência artificial e *machine learning*

Existem várias definições de inteligência artificial (IA). Uma das mais aceitas é a de John McCarthy que diz ”É a ciência e a engenharia de fabricar máquinas inteligentes, especialmente programas de computador inteligentes. Ela está relacionada à tarefa semelhante de usar computadores para entender a inteligência humana, mas a IA não precisa se limitar aos métodos biologicamente observáveis” (MCCARTHY, 2007). Uma das principais capacidades que pode ser considerada ”inteligente” é a tomada de decisões com base em um conjunto de entradas. O objetivo da IA é simular a inteligência humana a fim de resolver problemas complexos.

Figura 5 – Subconjuntos da IA



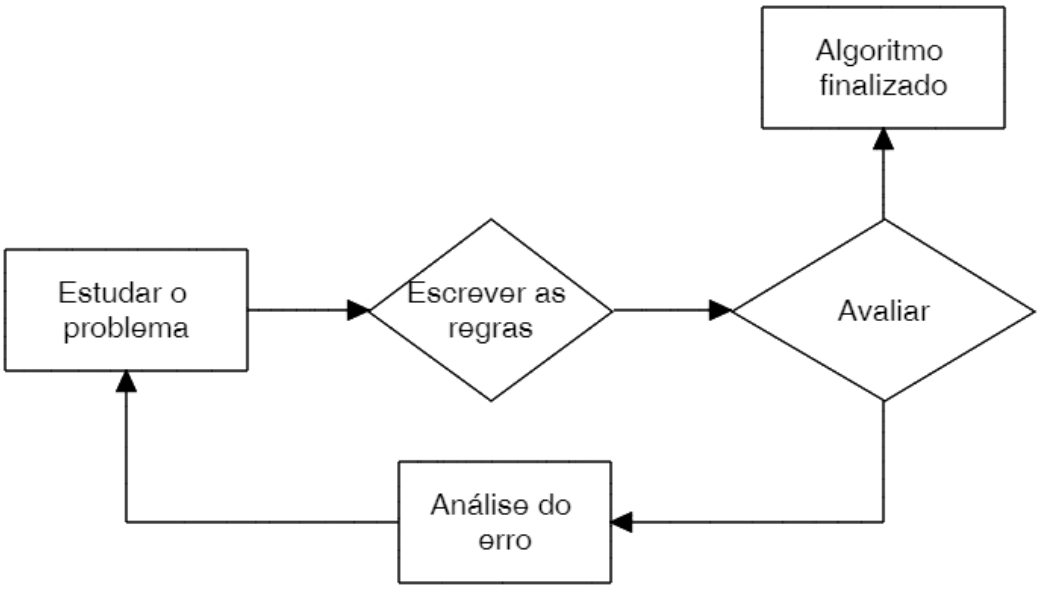
Fonte: Próprio autor.

O *machine learning* (ML) é um subconjunto de IA, como ilustrado na Figura 5, no qual o objetivo é aprender com base em dados e ser capaz de prever resultados quando exposto a novos dados ou descobrir padrões ocultos em dados não rotulados. Esse tipo de algoritmo não utiliza a denominada programação tradicional, em que o programador deve prever todas as regras e condições que o algoritmo irá executar, e portanto, permite a resolução de problemas complexos que demandariam grande esforço e tempo do programador para desenvolvê-lo de forma mais simples. Considerando um exemplo de classificação de e-mails maliciosos, utilizando a programação tradicional, o programador irá:

1. Analisar como e-mails maliciosos são caracterizados, por exemplo, eles podem conter as palavras-chave: cartão de crédito, grátis, gratuito. Também pode haver algum padrão no assunto ou no remetente deste e-mail.
2. Escrever regras para cada padrão de e-mail malicioso encontrado.
3. Testar e avaliar os passos anteriores até obter um resultado suficientemente bom.

A Figura 6 ilustra os passos descritos acima.

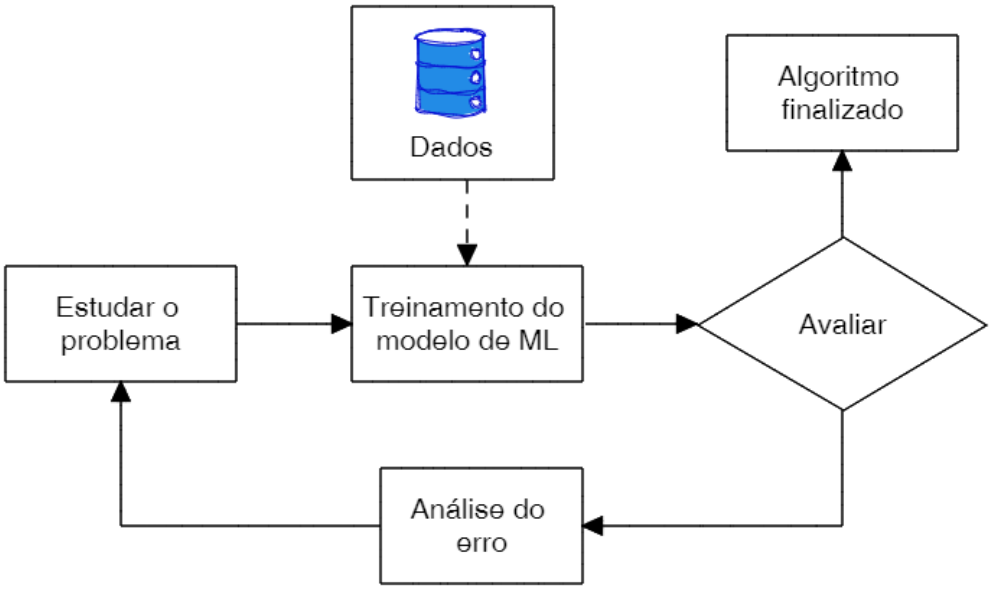
Figura 6 – Abordagem de programação tradicional



Fonte: Adaptado de GÉRON (2019) .

Este mesmo problema pode ser resolvido utilizando técnicas de ML que automaticamente irão encontrar os padrões e quais palavras são relevantes para caracterizar um e-mail malicioso (Figura 7). Essa solução resultará em um programa com menos linhas de código, de fácil manutenção e mais preciso. Outra vantagem de utilizar ML é a possibilidade de automatizar o processo de treinamento do modelo à medida que mais dados sejam obtidos, desta forma, garantindo que o modelo atual seja o mais preciso.

Figura 7 – Abordagem de programação utilizando ML



Fonte: Adaptado de GÉRON (2019) .

2.2.2 Classificação das técnicas de ML

Existem diversos tipos de sistemas de ML. De acordo com (GÉRON, 2019), eles podem ser classificados baseados em:

- Quando o treinamento utiliza supervisão humana ou não (Supervisionado, não supervisionado, semi supervisionado e aprendizado por reforço);
- Quando pode aprender incrementalmente com os dados ou não (aprendizado online ou em lote);
- Quando o sistema irá apenas comparar novos dados com dados já conhecidos ou irá identificar padrões nos dados de treinamento e construir um modelo de previsão. (baseado em instância ou baseado em modelo)

Os critérios não são exclusivos, podendo um sistema combinar da maneira que for necessária para a resolução do problema.

Os sistemas de ML são classificados de acordo com a quantidade e o tipo dos dados. As quatro principais categorias são: aprendizado supervisionado, não supervisionado, semi supervisionado e por reforço.

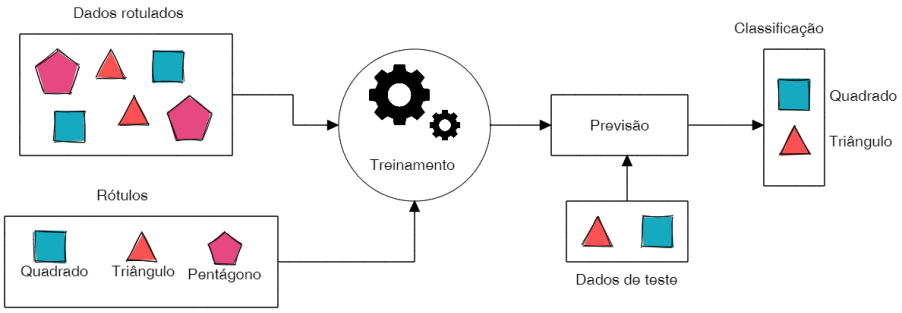
2.2.2.1 Aprendizado supervisionado

No aprendizado supervisionado, os dados para treinamento possuem a resposta esperada pelo sistema, o rótulo. Uma tarefa típica de um sistema supervisionado é a classificação. Esse tipo de sistema pode ser utilizado para detectar anomalias em dados, como na Figura 8.

Alguns dos algoritmos mais importantes de aprendizado supervisionado são:

- k-nearest neighbor

Figura 8 – Sistema de ML supervisionado



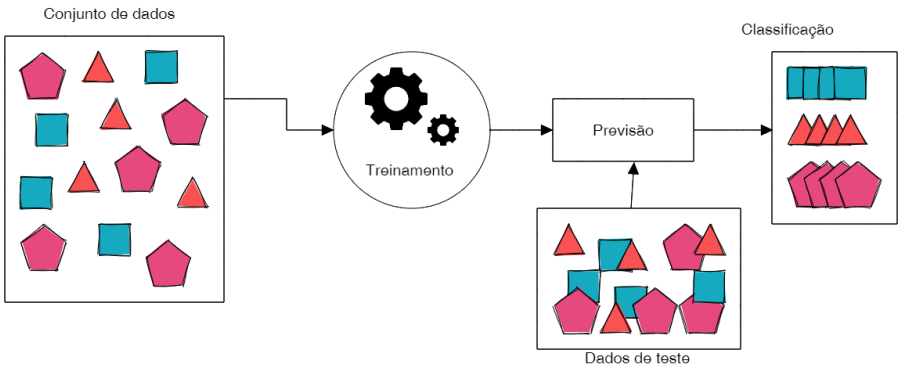
Fonte: Próprio autor.

- Regressão linear
- Regressão logística
- Support Vector Machines (SVMs)
- Decision Tree e Random Forest
- Redes Neurais Artificiais[7]

2.2.2.2 Aprendizado não supervisionado

De modo contrário ao aprendizado supervisionado, os dados para o aprendizado não supervisionado não possuem rótulos. O sistema busca extrair padrões dos dados (*features*) para agrupá-los para, posteriormente, separá-los. A Figura 9 demostra este processo.

Figura 9 – Sistema de ML não supervisionado



Fonte: Próprio autor.

Alguns dos algoritmos de aprendizado não supervisionado mais importantes são:

- Agrupamento (clustering)
  - K-Means
  - DBSCAN

[7] Válido para modelos de classificação

  - Hierarchical Cluster Analysis (HCA)

- Detecção de anomalia e detecção de novidade

  - Isolation Forest

A diferença de detecção de anomalias para a detecção de novidades é que a detecção de novidades espera receber apenas dados normais durante o treinamento, enquanto a detecção de anomalias aceita uma pequena porcentagem de *outliers* durante o treinamento.

2.2.2.3 Aprendizado semi supervisionado

Os algoritmos semi supervisionados conseguem tratar dados parcialmente rotulados.Usualmente, utiliza-se muitos dados em rótulos e uma pequena quantidade de dados rotulados. O aprendizado semi supervisionado em sua maioria são combinações de algoritmos supervisionados e não supervisionados.

Alguns dos algoritmos semi supervisionados são:

- One-class SVM
- *Deep belief networks* (DBNs)
- Máquina de Boltzmann restrita (RBMs)

2.2.2.4 Aprendizado por reforço

A aprendizagem por reforço é um treinamento no qual o modelo precisa enfrentar uma série de decisões para atingir um objetivo. O sistema funciona por tentativa e erro, que a cada acerto o modelo recebe uma recompensa, ou uma penalidade pelos erros. O objetivo do modelo é maximizar a recompensa total.

2.2.3 Definição de anomalia

Uma definição para anomalia pode ser um resultado ou um valor que foge do esperado, mas o critério exato para classificar uma anomalia vai depender de situação para situação.

De acordo com (ALLA; ADARI, 2019) as anomalias podem ser classificadas em três categorias.

- Anomalias baseadas em pontos de dados
- Anomalias baseadas em contextos
- Anomalias baseadas em padrões

2.2.3.1 Anomalias baseadas em pontos de dados

Um exemplo para este tipo de anomalia pode ser um conjunto de dados de hemogramas. A maioria dos valores de plaquetas no conjunto irão indicar uma quantidade saudável de plaquetas. Nesse caso, valores anômalos irão indicar algum tipo de enfermidade. Esses valores não são necessariamente *outliers*, mas poderiam, já que eles têm uma pequena probabilidade de existirem, levando em consideração a quantidade de dados saudáveis.

2.2.3.2 Anomalias baseadas em contexto

As anomalias baseadas em contexto são dados que aparentemente são normais, porém são considerados anomalias em certo contexto. Para exemplificar, um alto volume de compras é esperado ocorrer durante algumas datas, como a *black friday* e, portanto, são dados normais. Caso um alto volume de compras ocorra no meio de abril, avaliando o contexto das datas, possivelmente serão anomalias.

2.2.3.3 Anomalias baseadas em padrões

As anomalias baseadas em padrões são padrões ou tendências que desviam do esperado para o histórico dos dados. Um exemplo desse tipo de anomalia pode ser a quantidade de chuvas respectivas para um determinado mês. Caso o volume de chuvas no mês de janeiro na região sudeste seja pequeno, pode-se tratar de uma anomalia pois, de acordo com o histórico de chuvas, no mês de janeiro na região sudeste é esperado um alto volume de chuvas.

2.2.4 Detecção de anomalias

A detecção de anomalias se baseia em algoritmos avançados que identificam dados ou padrões que possam ser anômalos. Associadas à detecção de anomalias, existem outras atividades como a detecção de *outliers*, a detecção de novidades (*novelty detection*) e a remoção de ruídos.

2.2.4.1 Detecção de *outliers*

A detecção de *outliers* é uma técnica para identificar pontos anômalos em um conjunto de dados. Existem três métodos que podem ser aplicados para a detecção de *outliers*: realizar o treinamento em dados normais e identificar *outliers* através de altos erros de reconstrução, utilizar um modelo de distribuição de probabilidade no qual os *outliers* estão associados a probabilidades muito baixas de existirem ou treinar um modelo com dados rotulados para ensinar ao modelo as características dos dados normais e as de anomalias.

2.2.4.2 Remoção de ruídos

A remoção de ruído é aplicada quando há um constante ruído em um determinado conjunto de dados que precisa ser filtrado. Um modelo pode aprender com precisão como representar os dados e, portanto, pode reconstruí-los sem os ruídos presentes nos dados originais.

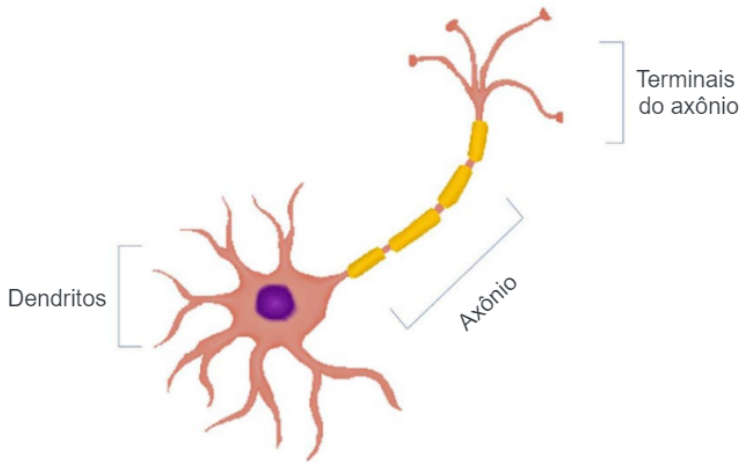
2.2.4.3 Detecção de novidades

A detecção de novidade é similar à detecção de *outliers*. A diferença principal entre os dois tipos de detecção é que no caso da detecção de *outliers* o conjunto de dados pode ou não conter dados anômalos. Já para a detecção de novidades, o conjunto de dados possui apenas dados normais e o modelo irá tentar classificar anomalias em um novo conjunto de dados nunca visto pelo modelo.

2.3 REDES NEURAIS ARTIFICIAIS E *DEEP LEARNING*

As redes neurais artificiais (ANN) são um subcampo de ML, no qual teve inspiração pela estrutura e funcionamento biológico do cérebro. A Figura 10 ilustra um neurônio biológico, no qual a entrada de impulsos é realizada através dos dendritos, que irão decidir se vão ativar o neurônio ou não. Após ativar, os sinais são conduzidos pelo axônio até seus terminais que irão transmitir para outros neurônios, essa transferência de sinais é chamada de sinapse.

Figura 10 – Exemplo de um neurônio biológico



Fonte: Traduzido de ALLA; ADARI (2019) .

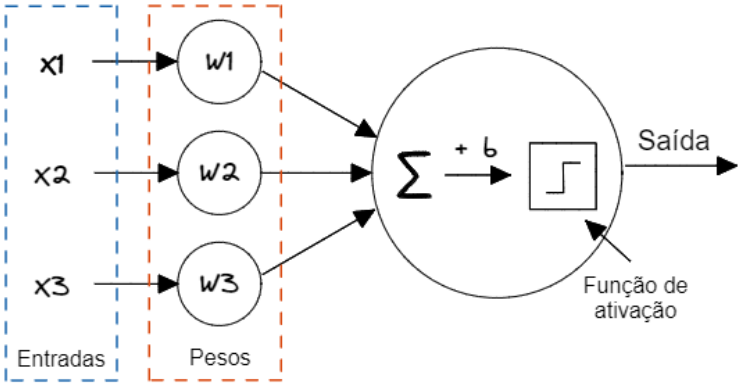
As ANNs são compostas por camadas com nós interconectados, esses nós são chamados de neurônios artificiais. Os neurônios artificiais possuem um funcionamento similar ao biológico, como mostra a Figura 11.

Em um neurônio artificial, é realizado o produto matricial do vetor de entrada **X** com o vetor de pesos **W**. O resultado do produto é somado e pode ser acrescido de **b** (chamado de *bias*). Assim, os valores são passados para a função de ativação, que irá decidir se o neurônio será ativado ou não. As funções de ativação mais utilizadas são a *ReLu* (Equação 2) e a *softmax* (Equação 3).

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

$$f(x) = max(0, x) \tag{2}$$

Figura 11 – Exemplo de funcionamento de um neurônio artificial

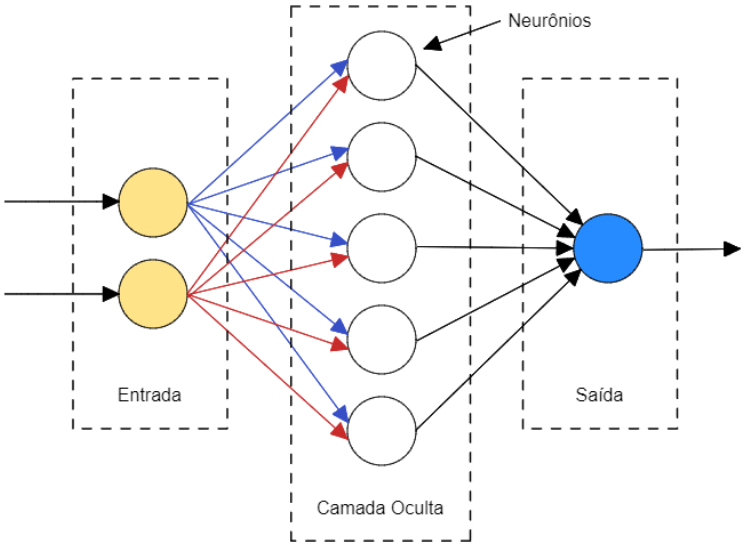


Fonte: Próprio autor.

$$f(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{3}$$

Uma ANN simples é composta de 3 camadas, sendo elas: a entrada, a camada oculta e a saída, como observado na Figura 12. A denominação *deep learning* é dada quando uma ANN possui mais de uma camada oculta, como na Figura 13.

Figura 12 – Exemplo de um modelo de ANN simples



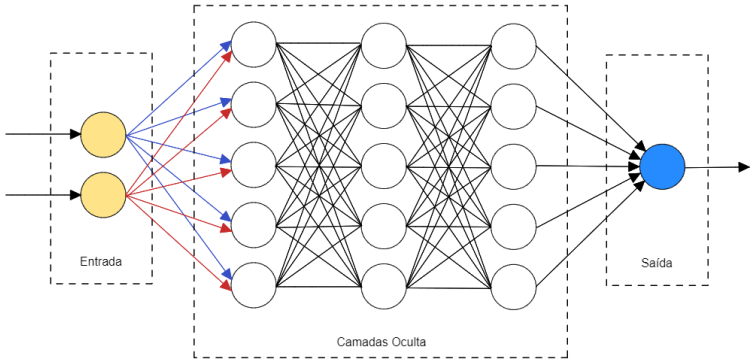
Fonte: Próprio autor.

2.4 AUTOENCODERS

Os *autoencoders* são um tipo de rede neural profunda (DNN) que possuem a habilidade de descobrir representações de baixa dimensão de dados com altas dimensões e são capazes de reconstruir os dados de entrada (ALLA; ADARI, 2019).

Os *autoencoders* são compostos por duas partes, o *encoder* e o *decoder*. O *encoder* reduz a dimensionalidade de uma entrada com alta dimensão enquanto o *decoder* realiza o processo inverso, de expandir a dimensionalidade. A saída do *encoder* é chamada de gargalo (*bottleneck*), pois representa

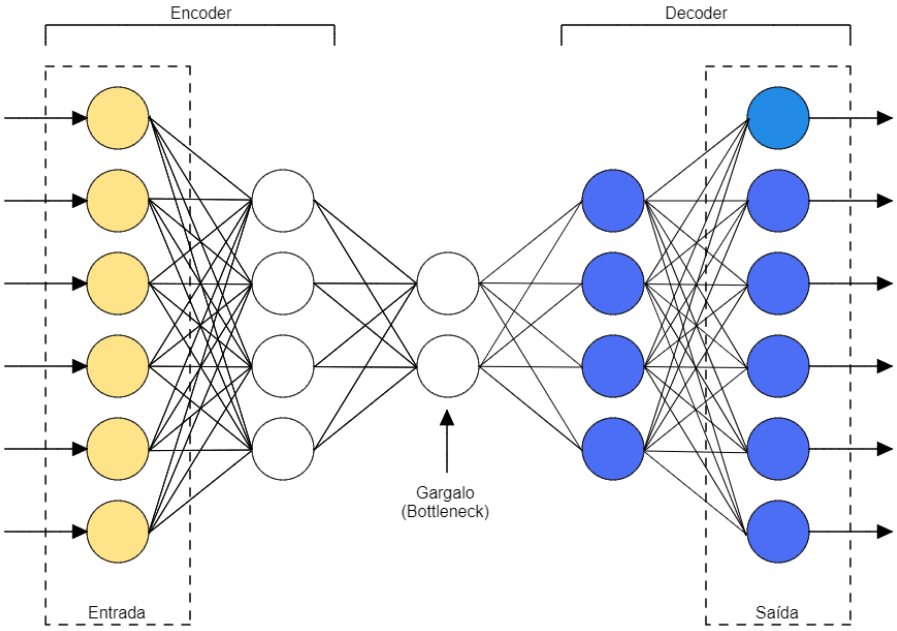
Figura 13 – Exemplo de modelo de *deep learning*



Fonte: Próprio autor.

a menor dimensionalidade que o conjunto de dados pode ser reduzido mantendo suas características principais. A Figura 14 ilustra as camadas de um *autoencoder* simples.

Figura 14 – Exemplo de um *autoencoder* simples



Fonte: Próprio autor.

2.5 MÉTRICAS PARA AVALIAÇÃO DOS MODELOS DE DETECÇÃO DE ANOMALIAS

De acordo com a literatura, existem diversas métricas para avaliar a performance de um modelo e compará-los com outros modelos. Nesse trabalho, serão utilizadas a matriz de confusão e suas métricas derivadas.

2.5.1 Matriz de confusão

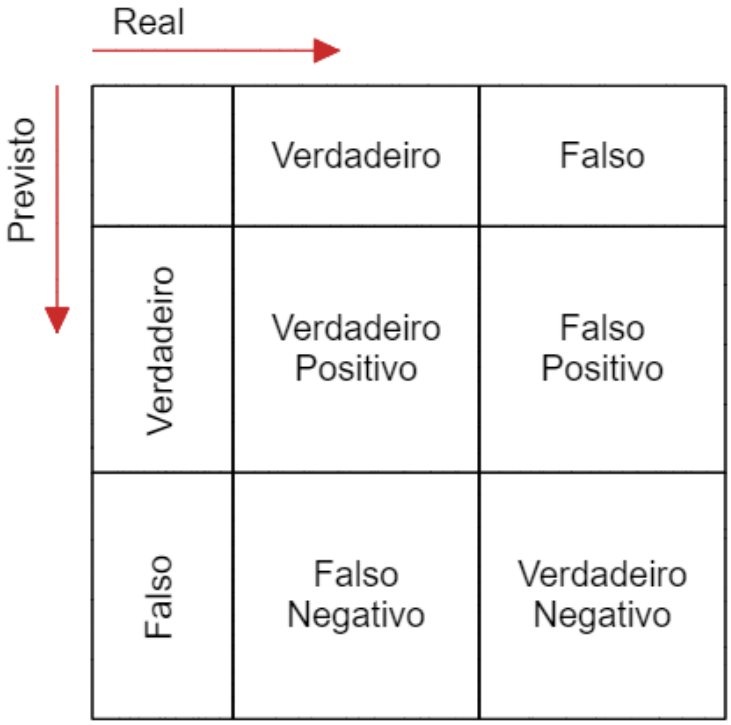
Uma das maneiras de avaliar a performance de um modelo é utilizando uma matriz de confusão. Ela é composta pela quantidade de vezes que o modelo previu um resultado para determinada entrada.

Por exemplo, um modelo classifica quando um animal testa positivo para uma doença ou não. Se um animal doente é classificado como positivo (significa que o animal possui a doença), este caso é um **verdadeiro positivo**. Caso um animal saudável seja classificado como negativo (significa que o animal não possui a doença) este caso é um **verdadeiro negativo**. Porém, existem os casos em que o modelo erra a previsão, se um animal saudável é classificado como doente, ou seja, como positivo, este caso é um **falso positivo**. O último caso é se um animal doente for classificado como saudável, ou seja, como negativo, então este caso é chamado de **falso negativo**.

- **Verdadeiro positivo**: Quando a condição é verdadeira, e a previsão também é verdadeira.
- **Verdadeiro negativo**: Quando a condição é falsa, e a previsão também é falsa.
- **Falso positivo**: Quando a condição é falsa, mas a previsão é verdadeira.
- **Falso negativo**: Quando a condição é verdadeira, mas a previsão é falsa.

A Figura 15 representa o modelo de uma matriz de confusão.

Figura 15 – Matriz de confusão



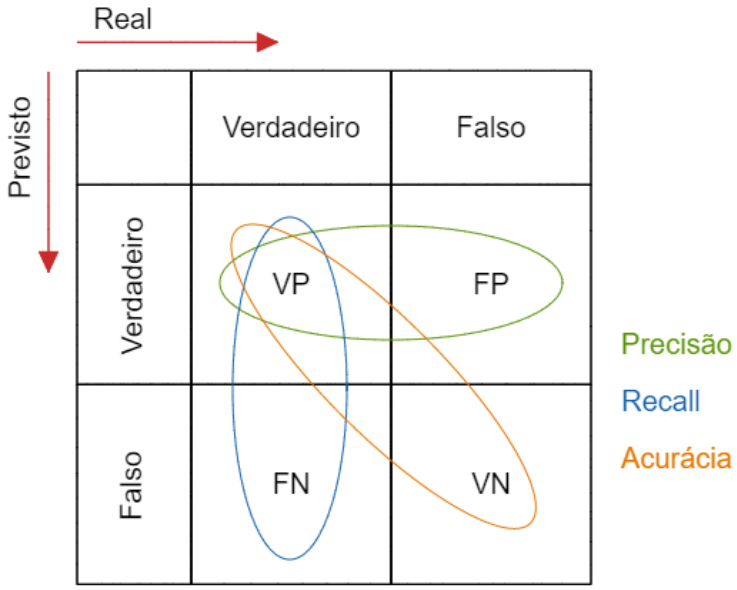
Fonte: Próprio autor.

Com os valores obtidos da matriz (Figura 16), pode-se obter algumas métricas, como a acurácia (equação 5), a precisão (equação 4) e o *recall* (equação 6).

$$Precisão = \frac{VP}{VP + FP} \tag{4}$$

$$Acurácia = \frac{VP + VN}{VP + FP + FN + VN} \tag{5}$$

Figura 16 – Matriz de confusão e métricas associadas



Fonte: Próprio autor.

$$Recall = \frac{VP}{VP + FN} \tag{6}$$

- **Precisão** é a medida que descreve quantas das previsões verdadeiras são realmente verdadeiras. Em outras palavras, considerando todas as previsões verdadeiras, quanto dados o modelo previu corretamente.
- **Acurácia** é a medida que descreve quantas previsões estão corretas no conjunto de dados inteiro. Em outras palavras, quantas previsões corretas o modelo fez entre positivas e negativas.
- ***Recall*** é a medida que descreve quantas das previsões verdadeiras de todo o conjunto de dados que realmente são verdadeiros. Em outras palavras, considerando todos os dados verdadeiros do conjunto de dados, quantos dados o modelo previu corretamente.

Geralmente, é conveniente combinar a precisão e o *recall* em um única métrica chamada F_1 *score*. O F_1 *score* consiste na média harmônica da precisão e do *recall* (equação 7) e, portanto, para obter valores altos do F_1 *score*, é preciso que tanto a precisão quanto o *recall* sejam altos.

$$F_1 score = 2 * \frac{Precisão * Recall}{Precisão + Recall} \tag{7}$$

2.5.2 Curva ROC

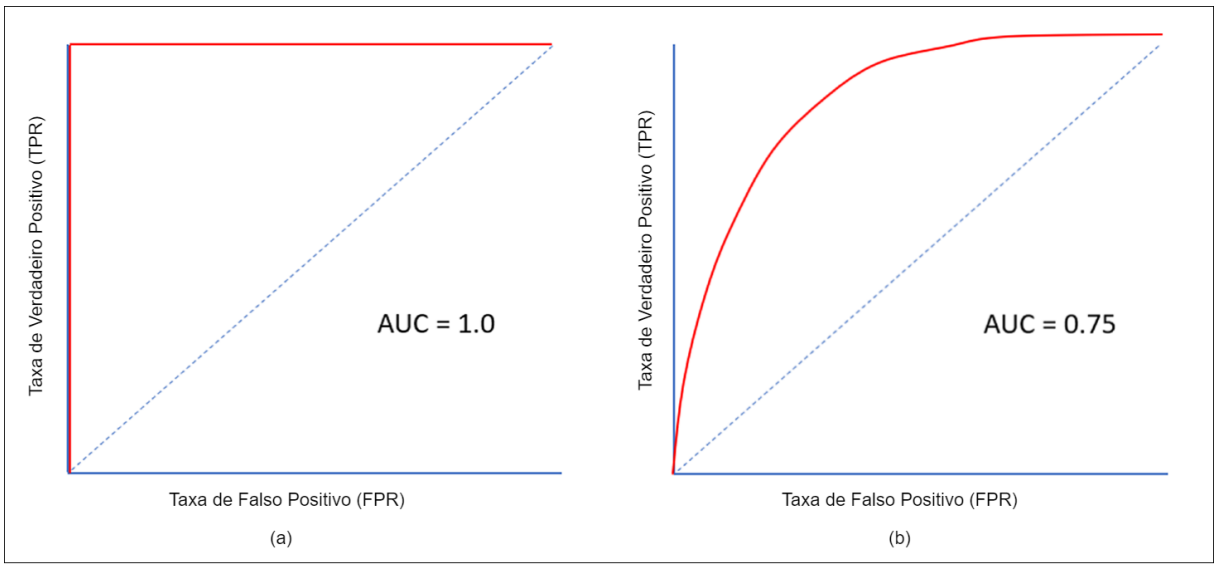
A curva Característica de Operação do Operador (*receiver operating characteristic* - ROC) é uma ferramenta muito utilizada com classificação binária. Ela é muito similar à curva precisão/*recall*, mas em vez de traçar a precisão pelo *recall*, a curva ROC traça a taxa de verdadeiro positivos (*true positive rate - TPR*) = ***recall*** = **sensibilidade** contra a taxa de falso positivos (*false positive rate* - FPR). O FPR

(equação 8) é a razão entre instâncias negativas que foram incorretamente classificadas como positivas e todas as instâncias analisadas (GÉRON, 2019).

$$FPR = 1 - especificidade = \frac{FP}{FP + VN} \tag{8}$$

A Figura 17 ilustra dois exemplos de curvas ROC.

Figura 17 – Exemplos de Curvas ROC: (a) AUC=1; (b) AUC=0,75



Fonte: ALLA; ADARI (2019) .

Uma maneira para comparar a performance entre modelos é realizando o cálculo da área abaixo da curva (*area under the curve* - AUC). Um modelo perfeito possui uma AUC igual a 1. Quando o modelo classifica os dados de forma aleatória, a AUC é igual a 0,5, sendo o pior resultado.

2.6 TRANSFERÊNCIA DE APRENDIZADO

A transferência de aprendizado pode ser aplicada em modelos de ANN e consiste na técnica de extrair *features* já treinadas de um problema, e aplicá-las em outro problema similar. Por exemplo, um modelo treinado para para identificar bolas de futebol em imagens pode ser usado como pontapé inicial para um modelo para identificar bolas de basquete.

A transferência de aprendizado é realizada travando os pesos dos nós de um modelo já treinado e criando outro modelo copiando as camadas. No novo modelo então é adicionada uma nova camada ao final para que o modelo a treine e seja capaz de resolver o novo problema.

3 MATERIAIS E MÉTODOS

Neste capítulo será apresentada a aplicabilidade de sistemas baseados em IA para a resolução de problemas reais da indústria, objetivando implementar um modelo de previsão de falhas em aerogeradores para a tomada de decisões sobre manutenções preventivas a fim de reduzir custos de operação. Além disso, este capítulo irá detalhar as etapas ilustradas na Figura 18.

- **Etapa 1** - Exploração: Consiste na parte inicial do trabalho, com a definição do escopo do trabalho e a identificação do problema, visto que, é necessário compreender o contexto do processo estudado, além de identificar e entender o comportamento das variáveis que influenciam o problema. Em seguida, coleta-se o conjunto de dados e inicia um processo de análise exploratória, buscando identificar padrões presentes nos dados e relações entre as variáveis.

- **Etapa 2** - Modelos de *machine learning*: Trata-se da criação e treinamento dos modelos de ML. Os dados já preparados serão divididos em três conjuntos, o de treinamento, validação e teste. Nessa etapa serão analisados alguns tipos de modelos, como *isolation forest*, OCSVM e modelos de redes neurais artificiais, como os *autoencoders*, avaliando qual modelo possui uma melhor performance no problema em questão.

- **Etapa 3** - Previsão: Com base no modelo que possui a melhor performance, será avaliado novos conjuntos de dados para gerar alertas caso a previsão indique uma falha no aerogerador.

3.1 DESCRIÇÃO DO PROBLEMA

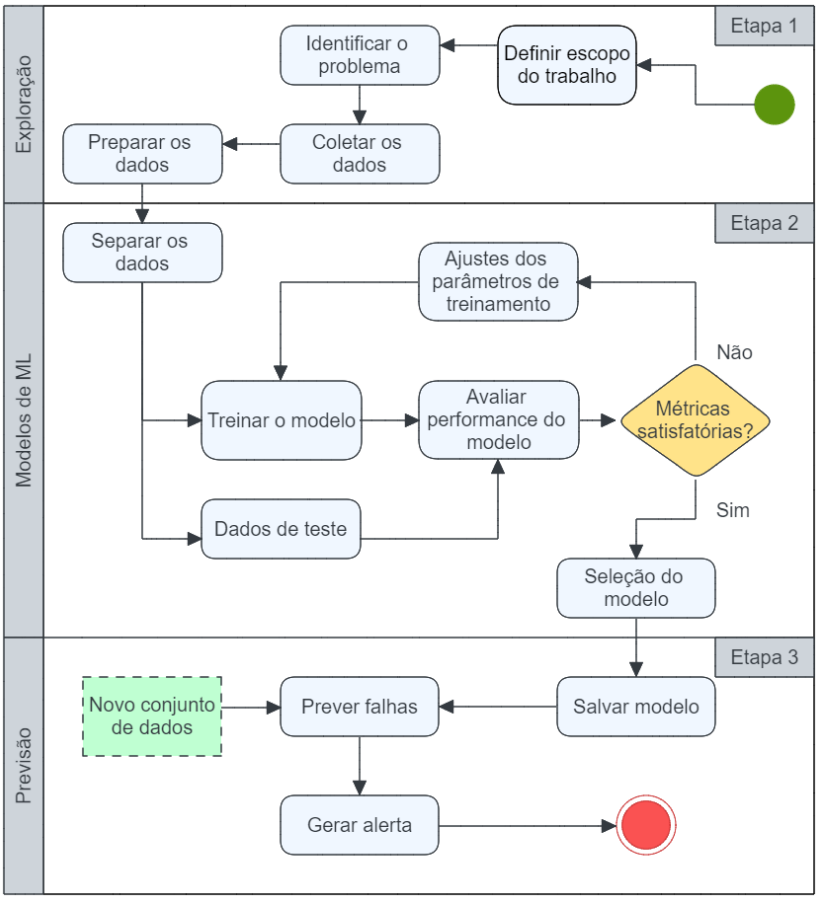
Estamos vivendo na era digital, mais precisamente na era do *Big Data*, na qual empresas de todos os tamanhos e setores buscam armazenar a maior quantidade de dados relevantes ao negócio possível de forma digital. Os dados sem um tratamento não agregam valor, portanto, é necessário um ferramental que possibilite a descoberta de novos *insights* e otimizações para o negócio.

Uma das abordagens para o tratamento deste grande volume de dados é a utilização de ML para extrair informações úteis. Hoje em dia os modelos de ML são utilizados em diversos cenários como para a detecção de transações fraudulentas, detecção de vazamento de dados, análise de qualidade em plantas de manufatura, na identificação de anomalias em exames médicos, etc (GÉRON, 2019).

A aplicabilidade dessa abordagem é comprovada em um problema real, em uma empresa brasileira do ramo de geração e comercialização de energias renováveis. Assim, é estudado um problema da área de detecção de anomalias, especificamente, nos valores de vibração dos sensores de aerogeradores visando construir um modelo preditivo para estabelecer faróis quando alguma falha está prestes a ocorrer, diminuindo o risco de falhas inesperadas que acarretam em maiores custos financeiros na operação.

O estudo da problemática começa analisando falhas que já ocorreram na operação dos parques eólicos, nas quais, dependendo do componente que apresentou a falha, devido a fatores logísticos, são

Figura 18 – Fluxograma de execução da pesquisa



Fonte: Próprio autor.

necessários vários meses para realizar a substituição e permitir que o aerogerador volte a operar. Os principais componentes mecânicos que apresentam falhas são os rolamentos e as caixas de velocidade (*gearboxes*) (ROMERO et al., 2016).

A empresa disponibilizou um conjunto de dados, para auto-preservação, contendo os dados de vibração de apenas três aerogeradores de um de seus parques eólicos. Os dados são compostos de 135 variáveis, com a intensidade das vibrações dividida em intervalos de frequência, temperatura e potência nos terminais de saída. Como os sensores do aerogerador possuem tempos de aquisição diferentes, os valores de cada variável são agrupados em intervalos de 10 minutos, extraindo os valores máximos no intervalo. A Tabela 1 descreve todas as variáveis do conjunto dados, contendo informações da localização dos sensores e a grandeza que cada variável representa.

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(continua)

Identificação	**Nome da variável**	**Tipo**	**Descrição**	**Local**
Nível de vibração	Variável 1 - x_1	Numérico	Magnitude de 3º ordem	Nacele Eixo X
Nível de vibração	Variável 2 - x_2	Numérico	Magnitude de 3º ordem	Nacele Eixo Z
Nível de vibração	Variável 3 - x_3	Numérico	Magnitude de 1º ordem	Nacele Eixo X
Nível de vibração	Variável 4 - x_4	Numérico	Magnitude de 1º ordem	Nacele Eixo Z
Nível de vibração	Variável 5 - x_5	Numérico	Valor residual	Gearbox 2º Planetary Stage
Nível de vibração	Variável 6 - x_6	Numérico	Valor residual	Gearbox High Speed Stage
Nível de vibração	Variável 7 - x_7	Numérico	Valor residual	Gearbox High Speed Stage
Nível de vibração	Variável 8 - x_8	Numérico	Valor residual	Gearbox High Speed Stage
Nível de vibração	Variável 9 - x_9	Numérico	Valor residual	Gearbox Intermediate Stage
Nível de vibração	Variável 10 - x_{10}	Numérico	Valor residual	Gearbox High Speed Stage
Nível de vibração	Variável 11 - x_{11}	Numérico	Magnitude de 2º ordem	Generator Drive End
Nível de vibração	Variável 12 - x_{12}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Nacele Eixo Z
Nível de vibração	Variável 13 - x_{13}	Numérico	Magnitude de 1º ordem	Generator Drive End
Nível de vibração	Variável 14 - x_{14}	Numérico	Magnitude de 3º ordem	Generator Drive End
Nível de vibração	Variável 15 - x_{15}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Gearbox 1º Planetary Stage
Nível de vibração	Variável 16 - x_{16}	Numérico	Magnitude de 2º ordem	Gearbox Intermediate Stage
Nível de vibração	Variável 17 - x_{17}	Numérico	Banda passante 300Hz-700Hz	Generator Drive End
Nível de vibração	Variável 18 - x_{18}	Numérico	Magnitude de 4º ordem	Generator Drive End
Nível de vibração	Variável 19 - x_{19}	Numérico	Magnitude de 3º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 20 - x_{20}	Numérico	Magnitude de 2º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 21 - x_{21}	Numérico	Magnitude de 5º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 22 - x_{22}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Main Bearing

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(continuação)

Identificação	Nome da variável	Tipo	Descrição	Local
Nível de vibração	Variável 23 - x_{23}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Gearbox Intermediate Stage
Nível de vibração	Variável 24 - x_{24}	Numérico	Frequência de 1º ordem nos dentes da engrenagem,	Gearbox 1º Planetary Stage
Nível de vibração	Variável 25 - x_{25}	Numérico	Frequência nas barras do rotor	Generator Non Drive End
Nível de vibração	Variável 26 - x_{26}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Gearbox Rotor Bearing
Nível de vibração	Variável 27 - x_{27}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Nacele Eixo X
Nível de vibração	Variável 28 - x_{28}	Numérico	2º frequência passante nas barras do rotor	Generator Drive End
Nível de vibração	Variável 29 - x_{29}	Numérico	Magnitude de 2º ordem	Generator Drive End
Nível de vibração	Variável 30 - x_{30}	Numérico	RMS de 0.1Hz a 10Hz - ISO 10816-21:2015	Nacele Eixo Z
Nível de vibração	Variável 31 - x_{31}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Gearbox 2º Planetary Stage
Nível de vibração	Variável 32 - x_{32}	Numérico	-	Generator Drive End
Nível de vibração	Variável 33 - x_{33}	Numérico	Magnitude de 1º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 34 - x_{34}	Numérico	Magnitude de 2º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 35 - x_{35}	Numérico	Frequência alta da banda passante	Generator Non Drive End
Nível de vibração	Variável 36 - x_{36}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Generator Non Drive End
Nível de vibração	Variável 37 - x_{37}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Generator Drive End
Nível de vibração	Variável 38 - x_{38}	Numérico	RMS de 10Hz a 2kHz - ISO 10816-21:2015	Gearbox Intermediate Stage
Nível de vibração	Variável 39 - x_{39}	Numérico	-	Generator Non Drive End
Nível de vibração	Variável 40 - x_{40}	Numérico	Filtro móvel da banda passante	Gearbox Intermediate Stage
Nível de vibração	Variável 41 - x_{41}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Generator Drive End
Nível de vibração	Variável 42 - x_{42}	Numérico	-	Generator Drive End
Nível de vibração	Variável 43 - x_{43}	Numérico	Frequência nas barras do rotor	Generator Drive End
Nível de vibração	Variável 44 - x_{44}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Generator Non Drive End

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(continuação)

Identificação	Nome da variável	Tipo	Descrição	Local
Nível de vibração	Variável 45 - x_{45}	Numérico	Frequência de 2º ordem do estator	Generator Non Drive End
Nível de vibração	Variável 46 - x_{46}	Numérico	Alta frequência fator Crest	Gearbox Intermediate Stage
Nível de vibração	Variável 47 - x_{47}	Numérico	Magnitude de 5º ordem	Generator Drive End
Nível de vibração	Variável 48 - x_{48}	Numérico	Magnitude de 4º ordem	Generator Drive End
Nível de vibração	Variável 49 - x_{49}	Numérico	Magnitude de 3º ordem	Generator Drive End
Nível de vibração	Variável 50 - x_{50}	Numérico	Frequência de 2º ordem do estator	Generator Drive End
Nível de vibração	Variável 51 - x_{51}	Numérico	Banda passante 100Hz-1kHz	Gearbox Rotor Bearing
Nível de vibração	Variável 52 - x_{52}	Numérico	Frequência alta da banda passante	Gearbox Rotor Bearing
Nível de vibração	Variável 53 - x_{53}	Numérico	Magnitude de 1º ordem	Generator Drive End
Nível de vibração	Variável 54 - x_{54}	Numérico	Magnitude de 6º ordem	Generator Drive End
Nível de vibração	Variável 55 - x_{55}	Numérico	Frequência de 3º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 56 - x_{56}	Numérico	Frequência alta da banda passante	Main Bearing
Nível de vibração	Variável 57 - x_{57}	Numérico	Frequência de 1º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 58 - x_{58}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox High Speed Stage
Nível de vibração	Variável 59 - x_{59}	Numérico	-	Gearbox High Speed Stage
Nível de vibração	Variável 60 - x_{60}	Numérico	Frequência de 3º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 61 - x_{61}	Numérico	Magnitude de 3º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 62 - x_{62}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 63 - x_{63}	Numérico	Frequência de 1º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 64 - x_{64}	Numérico	Filtro móvel da banda passante	Gearbox 2º Planetary Stage
Nível de vibração	Variável 65 - x_{65}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox 2º Planetary Stage
Nível de vibração	Variável 66 - x_{66}	Numérico	Frequência alta da banda passante	Gearbox 2º Planetary Stage

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(continuação)

Identificação	Nome da variável	Tipo	Descrição	Local
Nível de vibração	Variável 67 - x_{67}	Numérico	Banda passante de velocidade	Gearbox 2º Planetary Stage
Nível de vibração	Variável 68 - x_{68}	Numérico	Frequência de 3º ordem nos dentes da engrenagem,	Gearbox 2º Planetary Stage
Nível de vibração	Variável 69 - x_{69}	Numérico	Frequência de 1º ordem nos dentes da engrenagem,	Gearbox 2º Planetary Stage
Nível de vibração	Variável 70 - x_{70}	Numérico	Filtro móvel da banda passante	Gearbox 1º Planetary Stage
Nível de vibração	Variável 71 - x_{71}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox 1º Planetary Stage
Nível de vibração	Variável 72 - x_{72}	Numérico	Banda passante 20Hz-100Hz	Gearbox 1º Planetary Stage
Nível de vibração	Variável 73 - x_{73}	Numérico	Frequência de 3º ordem nos dentes da engrenagem,	Gearbox 1º Planetary Stage
Nível de vibração	Variável 74 - x_{74}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Gearbox 1º Planetary Stage
Nível de vibração	Variável 75 - x_{75}	Numérico	Filtro móvel da banda passante	Gearbox High Speed Stage
Nível de vibração	Variável 76 - x_{76}	Numérico	Magnitude de 1º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 77 - x_{77}	Numérico	Frequência de 1º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 78 - x_{78}	Numérico	Magnitude de 2º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 79 - x_{79}	Numérico	Frequência de 3º ordem nos dentes da engrenagem,	Gearbox Intermediate Stage
Nível de vibração	Variável 80 - x_{80}	Numérico	Magnitude de 3º ordem	Gearbox Intermediate Stage
Nível de vibração	Variável 81 - x_{81}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Gearbox Intermediate Stage
Nível de vibração	Variável 82 - x_{82}	Numérico	Frequência de 1º ordem nos dentes da engrenagem,	Gearbox Intermediate Stage
Nível de vibração	Variável 83 - x_{83}	Numérico	Magnitude de 1º ordem	Gearbox Intermediate Stage
Nível de vibração	Variável 84 - x_{84}	Numérico	Filtro móvel da banda passante	Gearbox High Speed Stage
Nível de vibração	Variável 85 - x_{85}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox High Speed Stage
Nível de vibração	Variável 86 - x_{86}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 87 - x_{87}	Numérico	Magnitude de 6º ordem	Generator Non Drive End
Nível de vibração	Variável 88 - x_{88}	Numérico	Magnitude de 1º ordem	Gearbox High Speed Stage

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(continuação)

Identificação	Nome da variável	Tipo	Descrição	Local
Nível de vibração	Variável 89 - x_{89}	Numérico	Filtro móvel da banda passante	Gearbox High Speed Stage
Nível de vibração	Variável 90 - x_{90}	Numérico	RMS de 0.1Hz a 10kHz - ISO 10816-21:2015	Nacele Eixo X
Nível de vibração	Variável 91 - x_{91}	Numérico	Frequência de 3º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 92 - x_{92}	Numérico	Magnitude de 3º ordem	Gearbox High Speed Stage
Nível de vibração	Variável 93 - x_{93}	Numérico	Frequência de 2º ordem nos dentes da engrenagem,	Gearbox High Speed Stage
Nível de vibração	Variável 94 - x_{94}	Numérico	Fator Crest	Nacele Eixo Z
Nível de vibração	Variável 95 - x_{95}	Numérico	Fator Crest	Nacele Eixo X
Nível de vibração	Variável 96 - x_{96}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Main Bearing
Nível de vibração	Variável 97 - x_{97}	Numérico	Alta frequência fator Crest	Main Bearing
Nível de vibração	Variável 98 - x_{98}	Numérico	-	Gearbox Intermediate Stage
Nível de vibração	Variável 99 - x_{99}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Generator Drive End
Nível de vibração	Variável 100 - x_{100}	Numérico	Banda passante 100Hz-500Hz	Gearbox 2º Planetary Stage
Nível de vibração	Variável 101 - x_{101}	Numérico	Frequência alta da banda passante	Gearbox High Speed Stage
Nível de vibração	Variável 102 - x_{102}	Numérico	-	Gearbox High Speed Stage
Nível de vibração	Variável 103 - x_{103}	Numérico	-	Gearbox 2º Planetary Stage
Nível de vibração	Variável 104 - x_{104}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox 2º Planetary Stage
Nível de vibração	Variável 105 - x_{105}	Numérico	Alta frequência fator Crest	Gearbox 2º Planetary Stage
Nível de vibração	Variável 106 - x_{106}	Numérico	Banda passante 20Hz-100Hz	Gearbox 2º Planetary Stage
Nível de vibração	Variável 107 - x_{107}	Numérico	-	Gearbox 1º Planetary Stage
Nível de vibração	Variável 108 - x_{108}	Numérico	Frequência alta da banda passante	Generator Drive End
Nível de vibração	Variável 109 - x_{109}	Numérico	Alta frequência fator Crest	Gearbox 1º Planetary Stage
Nível de vibração	Variável 110 - x_{110}	Numérico	Frequência alta da banda passante	Gearbox 1º Planetary Stage

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(continuação)

Identificação	Nome da variável	Tipo	Descrição	Local
Nível de vibração	Variável 111 - x_{111}	Numérico	-	Gearbox 1º Planetary Stage
Nível de vibração	Variável 112 - x_{112}	Numérico	-	Gearbox 1º Planetary Stage
Nível de vibração	Variável 113 - x_{113}	Numérico	Banda passante de velocidade	Gearbox 1º Planetary Stage
Nível de vibração	Variável 114 - x_{114}	Numérico	Banda passante 100Hz-500Hz	Gearbox 1º Planetary Stage
Nível de vibração	Variável 115 - x_{115}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox High Speed Stage
Nível de vibração	Variável 116 - x_{116}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox High Speed Stage
Nível de vibração	Variável 117 - x_{117}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox High Speed Stage
Nível de vibração	Variável 118 - x_{118}	Numérico	-	Gearbox High Speed Stage
Nível de vibração	Variável 119 - x_{119}	Numérico	Frequência alta da banda passante	Gearbox High Speed Stage
Nível de vibração	Variável 120 - x_{120}	Numérico	Banda passante 20Hz-100Hz	Gearbox Intermediate Stage
Nível de vibração	Variável 121 - x_{121}	Numérico	Banda passante 200Hz-2kHz	Gearbox Intermediate Stage
Nível de vibração	Variável 122 - x_{122}	Numérico	Banda passante de velocidade	Gearbox Intermediate Stage
Nível de vibração	Variável 123 - x_{123}	Numérico	Fator Crest	Gearbox Intermediate Stage
Nível de vibração	Variável 124 - x_{124}	Numérico	NaN	Gearbox Intermediate Stage
Nível de vibração	Variável 125 - x_{125}	Numérico	Frequência alta da banda passante	Gearbox Intermediate Stage
Nível de vibração	Variável 126 - x_{126}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox Intermediate Stage
Nível de vibração	Variável 127 - x_{127}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox Intermediate Stage
Nível de vibração	Variável 128 - x_{128}	Numérico	Banda passante 10Hz-1kHz	Gearbox Rotor Bearing
Nível de vibração	Variável 129 - x_{129}	Numérico	Alta frequência fator Crest	Gearbox Rotor Bearing
Nível de vibração	Variável 130 - x_{130}	Numérico	RMS de 10Hz a 1kHz - ISO 10816-21:2015	Gearbox Rotor Bearing
Nível de vibração	Variável 131 - x_{131}	Numérico	Banda passante 300Hz-700Hz	Generator Drive End
Nível de vibração	Variável 132 - x_{132}	Numérico	Graus Celsius	Temperatura Ambiente

Tabela 1 – Descrição das variáveis de entrada do conjunto de dados

(conclusão)

Identificação	**Nome da variável**	**Tipo**	**Descrição**	**Local**
Velocidade	Variável 133 - x_{133}	Numérico	RPM	Rotor
Potência	Variável 134 - x_{134}	Numérico	Potência [W]	Terminal 1
Inclinação	Variável 135 - x_{135}	Numérico	Graus	Pás

Fonte: Próprio autor.

3.2 BANCO DE DADOS

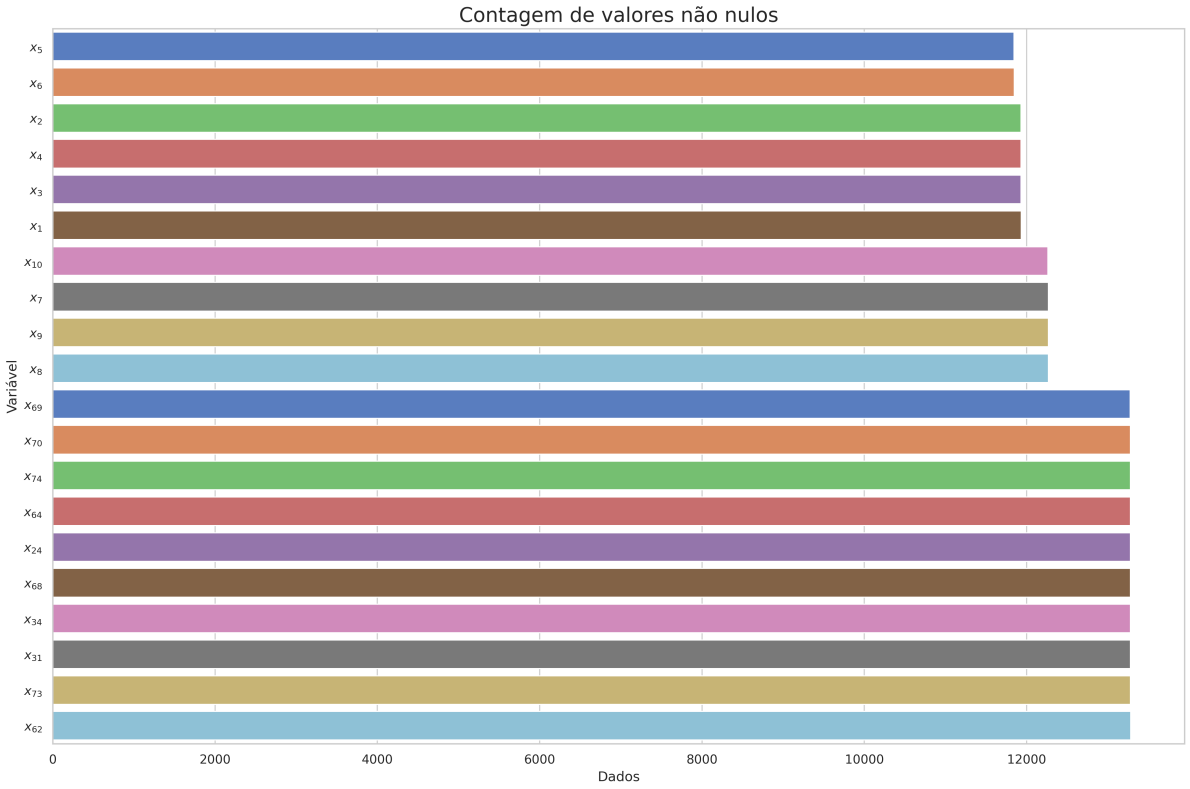
O conjunto de dados foi extraído do BigQuery, que consiste em um armazenamento de dados corporativo em nuvem totalmente gerenciado. A arquitetura do BigQuery permite utilizar consultas SQL para manipulação dos dados, e como conta com um mecanismo de análise distribuída e escalável, permite que consultas de terabytes sejam realizadas em poucos segundos. Os dados extraídos dos aerogeradores foram do período de agosto de 2021 até novembro de 2022. Dos três aerogeradores, apenas um apresentou falha no período, e por esse motivo, ele será utilizado para o treinamento e validação dos resultados. Os outros dois aerogeradores serão utilizados posteriormente com uma outra abordagem para realizar a predição de falhas.

Os dados apresentados a seguir são apenas para o aerogerador que apresentou falha durante o período de análise, pois ele será utilizado para realizar o treinamento e validação dos resultados.

O conjunto de dados fornecido pela empresa de estudo, além de possuir 135 variáveis de entrada, é formado por 16528 linhas de dados. Inicialmente, analisando o conjunto de dados, foram encontrados dados ausentes no conjunto. Esses dados no contexto do problema podem ter sido causados por diversos fatores, como problemas de conexão do aerogerador com a base de dados em nuvem.

No conjunto, foram encontrados 5825 linhas com pelo menos uma variável com dados ausentes, portanto, as linhas do conjunto com dados completos representam 64,76% do conjunto de dados. A Figura 19 ilustra a quantidade de dados não nulos para vinte variáveis ordenadas de forma crescente.

Figura 19 – Quantidade de valores não nulos por variável

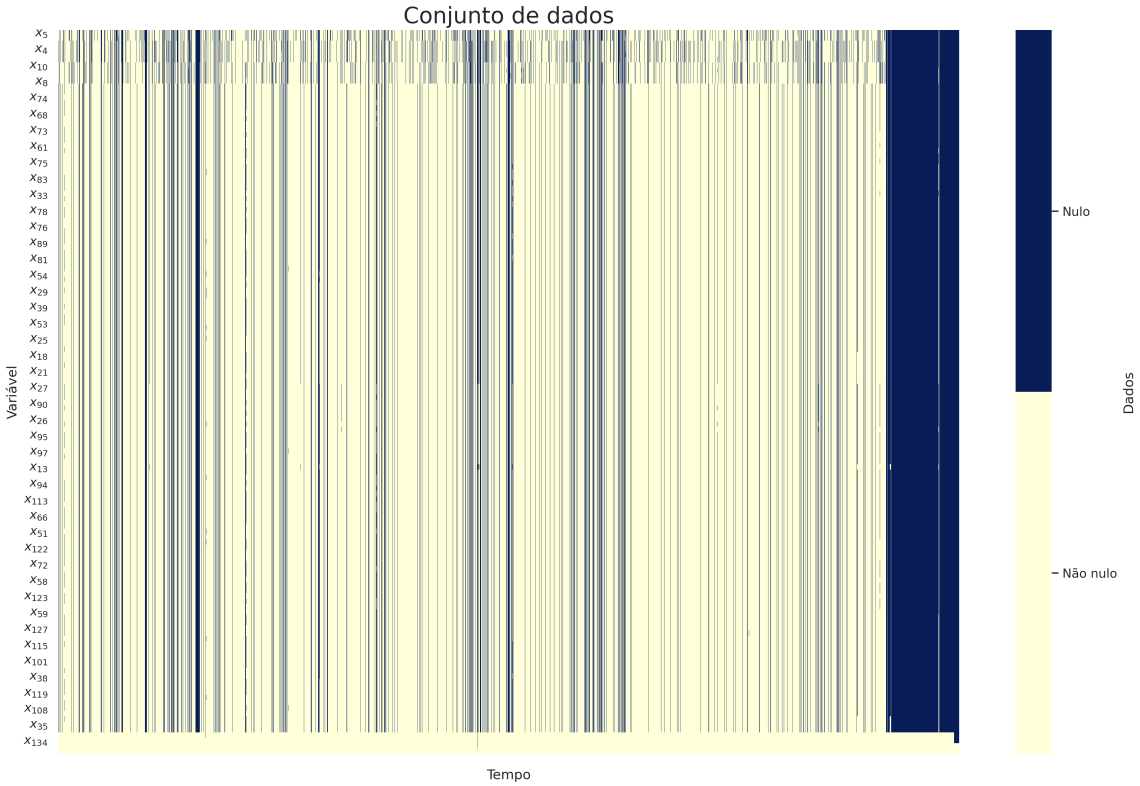


Fonte: Próprio autor.

Nota-se que a Figura 20 ilustra que 134 das 135 variáveis estão com valores nulos no final do conjunto de dados, desta forma, fica claro o momento que o aerogerador apresentou a falha e ficou

fora de operação para realizar a manutenção.

Figura 20 – Posição dos dados ausentes no conjunto de dados



Fonte: Próprio autor.

As linhas de dados com valores ausentes foram removidas para não prejudicar o treinamento dos modelos. A Tabela 2 apresenta um resumo dos dados utilizados para o desenvolvimento do trabalho, contendo um total de 10703 linhas de dados.

A Tabela 1 apresenta uma análise estatística de todas as variáveis. As métricas estatísticas utilizadas foram a média, o desvio padrão, valor mínimo e máximo e os quartis.

Tabela 3 – Análise estatística das variáveis do conjunto de dados

(continua)

Variável	**Média**	**Desvio Padrão**	**Mínimo**	**Quartil 1**	**Quartil 2**	**Quartil 3**	**Máximo**
x_1	1.08e-02	5.98e-03	2.05e-05	8.55e-03	1.15e-02	1.53e-02	2.41e-02
x_2	3.22e-02	1.57e-02	3.03e-05	3.23e-02	3.76e-02	4.23e-02	6.48e-02
x_3	3.93e-03	3.30e-03	3.83e-05	4.82e-04	3.71e-03	7.22e-03	1.12e-02
x_4	1.24e-02	8.52e-03	6.15e-05	3.69e-03	1.40e-02	2.03e-02	2.70e-02
x_5	6.90e-02	1.44e-02	3.86e-02	5.65e-02	7.70e-02	8.02e-02	1.10e-01
x_6	3.79e+00	2.40e+00	6.71e-01	1.63e+00	3.22e+00	5.59e+00	1.13e+01
x_7	1.76e+00	1.08e+00	3.42e-01	9.15e-01	1.35e+00	2.56e+00	5.63e+00
x_8	1.20e+00	6.21e-01	3.02e-01	6.82e-01	1.04e+00	1.78e+00	2.85e+00

Tabela 3 – Análise estatística das variáveis do conjunto de dados

(continuação)

Variável	Média	Desvio Padrão	Mínimo	Quartil 1	Quartil 2	Quartil 3	Máximo
x_9	2.88e+00	2.01e+00	4.32e-01	1.26e+00	2.16e+00	4.24e+00	9.30e+00
x_{10}	1.54e-02	2.31e-03	1.05e-02	1.37e-02	1.49e-02	1.65e-02	2.50e-02
x_{11}	3.49e-02	1.11e-02	1.05e-02	2.65e-02	3.36e-02	4.42e-02	6.75e-02
x_{12}	3.43e-02	1.98e-02	1.29e-04	2.13e-02	3.93e-02	5.05e-02	6.96e-02
x_{13}	2.77e-01	1.33e-01	5.21e-02	2.01e-01	2.21e-01	4.03e-01	7.24e-01
x_{14}	8.18e-03	3.68e-03	2.93e-03	4.28e-03	7.70e-03	1.15e-02	1.93e-02
x_{15}	3.93e-02	1.31e-02	9.13e-03	2.88e-02	4.13e-02	4.98e-02	6.68e-02
x_{16}	1.11e-02	4.12e-03	1.66e-03	7.33e-03	1.22e-02	1.43e-02	2.00e-02
x_{17}	1.93e-01	3.05e-02	9.93e-02	1.69e-01	1.98e-01	2.18e-01	2.60e-01
x_{18}	5.35e-03	2.03e-03	2.56e-03	3.78e-03	4.73e-03	6.75e-03	1.51e-02
x_{19}	1.75e-02	4.66e-03	6.94e-03	1.47e-02	1.81e-02	2.09e-02	2.54e-02
x_{20}	2.32e-02	5.88e-03	9.09e-03	1.94e-02	2.47e-02	2.82e-02	3.21e-02
x_{21}	7.19e-03	2.24e-03	3.23e-03	5.56e-03	6.71e-03	8.29e-03	1.78e-02
x_{22}	3.92e-02	1.35e-02	7.23e-03	2.85e-02	4.15e-02	5.01e-02	6.65e-02
x_{23}	6.14e-02	2.02e-02	1.35e-02	4.53e-02	6.47e-02	7.76e-02	1.04e-01
x_{24}	1.22e-02	3.20e-03	4.49e-03	1.00e-02	1.26e-02	1.46e-02	2.00e-02
x_{25}	5.86e+00	2.73e-01	5.26e+00	5.66e+00	5.77e+00	5.98e+00	6.74e+00
x_{26}	2.70e-02	9.21e-03	5.38e-03	1.97e-02	2.86e-02	3.44e-02	4.58e-02
x_{27}	2.64e-02	1.53e-02	1.13e-04	1.64e-02	3.04e-02	3.89e-02	5.40e-02
x_{28}	3.11e+00	2.62e-01	2.56e+00	2.85e+00	3.15e+00	3.31e+00	3.93e+00
x_{29}	3.55e-02	8.57e-03	1.11e-02	2.89e-02	3.50e-02	4.18e-02	6.56e-02
x_{30}	3.38e+00	1.93e+00	9.64e-03	2.18e+00	3.89e+00	4.94e+00	7.03e+00
x_{31}	5.16e-02	4.49e-03	3.47e-02	4.93e-02	5.15e-02	5.43e-02	6.99e-02
x_{32}	1.87e-01	1.34e-04	1.87e-01	1.87e-01	1.87e-01	1.87e-01	1.88e-01
x_{33}	7.62e-03	2.16e-03	1.90e-03	7.77e-03	8.12e-03	8.79e-03	1.19e-02
x_{34}	5.46e-03	1.53e-03	2.09e-03	4.10e-03	5.47e-03	6.67e-03	9.21e-03
x_{35}	6.65e+00	1.64e+00	3.16e+00	5.22e+00	6.37e+00	8.20e+00	1.04e+01
x_{36}	3.19e+00	2.97e-01	2.69e+00	2.92e+00	3.19e+00	3.38e+00	4.47e+00
x_{37}	1.61e+00	5.44e-01	4.76e-01	1.32e+00	1.43e+00	2.09e+00	3.64e+00
x_{38}	5.48e+00	2.00e+00	2.06e+00	3.81e+00	5.05e+00	7.07e+00	1.09e+01
x_{39}	5.90e-02	2.57e-02	1.66e-02	3.83e-02	5.33e-02	7.49e-02	1.70e-01
x_{40}	4.51e+00	2.24e+00	9.16e-01	2.58e+00	4.36e+00	6.38e+00	1.02e+01
x_{41}	3.82e-01	7.46e-02	1.52e-01	3.74e-01	4.01e-01	4.32e-01	4.99e-01
x_{42}	6.00e-02	2.38e-02	2.52e-02	4.14e-02	4.96e-02	7.63e-02	1.57e-01
x_{43}	5.77e+00	2.08e-01	5.39e+00	5.60e+00	5.66e+00	5.97e+00	6.32e+00
x_{44}	4.26e-01	9.63e-02	1.67e-01	4.01e-01	4.19e-01	4.90e-01	7.74e-01

Tabela 3 – Análise estatística das variáveis do conjunto de dados

(continuação)

Variável	Média	Desvio Padrão	Mínimo	Quartil 1	Quartil 2	Quartil 3	Máximo
x_{45}	3.11e+00	2.48e-01	2.81e+00	2.95e+00	3.02e+00	3.22e+00	4.28e+00
x_{46}	4.75e+00	3.26e-01	4.46e+00	4.69e+00	4.73e+00	4.76e+00	1.36e+01
x_{47}	7.54e-03	2.09e-03	3.74e-03	6.12e-03	7.19e-03	8.45e-03	2.01e-02
x_{48}	6.80e-03	1.80e-03	3.42e-03	5.38e-03	6.34e-03	8.22e-03	1.24e-02
x_{49}	7.99e-03	3.31e-03	2.84e-03	5.28e-03	6.99e-03	1.02e-02	2.14e-02
x_{50}	3.04e+00	1.19e-01	2.83e+00	2.96e+00	3.03e+00	3.12e+00	3.58e+00
x_{51}	3.63e-01	1.14e-01	1.14e-01	3.06e-01	3.58e-01	4.46e-01	7.86e-01
x_{52}	9.34e-01	3.51e-01	3.15e-01	6.37e-01	9.04e-01	1.23e+00	2.04e+00
x_{53}	1.92e-01	7.36e-02	2.88e-02	1.61e-01	1.78e-01	2.53e-01	4.33e-01
x_{54}	5.60e-02	3.87e-02	1.05e-02	1.40e-02	5.27e-02	9.48e-02	1.25e-01
x_{55}	1.15e+00	4.44e-01	2.11e-01	9.63e-01	1.12e+00	1.56e+00	2.03e+00
x_{56}	6.72e-02	1.97e-02	2.58e-02	5.24e-02	6.54e-02	8.56e-02	1.16e-01
x_{57}	7.33e-01	2.14e-01	2.30e-01	6.28e-01	7.29e-01	9.04e-01	1.27e+00
x_{58}	2.64e+00	7.61e-01	7.91e-01	2.29e+00	2.67e+00	3.19e+00	5.82e+00
x_{59}	3.44e+00	1.96e+00	3.92e-01	1.67e+00	3.10e+00	5.33e+00	7.42e+00
x_{60}	5.52e+00	4.10e+00	4.12e-01	1.28e+00	5.50e+00	1.03e+01	1.27e+01
x_{61}	6.39e-03	2.17e-03	1.98e-03	4.71e-03	6.36e-03	8.23e-03	1.12e-02
x_{62}	1.60e+00	7.66e-01	1.56e-01	1.01e+00	1.86e+00	2.24e+00	2.85e+00
x_{63}	1.95e+00	7.30e-01	3.32e-01	1.55e+00	2.06e+00	2.57e+00	3.63e+00
x_{64}	4.01e-01	9.21e-02	1.51e-01	3.93e-01	4.14e-01	4.46e-01	5.95e-01
x_{65}	1.22e+00	3.39e-01	3.35e-01	1.14e+00	1.29e+00	1.43e+00	2.04e+00
x_{66}	5.95e+00	2.65e+00	1.58e+00	3.38e+00	5.79e+00	8.37e+00	1.17e+01
x_{67}	1.97e+00	5.87e-01	7.07e-01	1.49e+00	2.06e+00	2.45e+00	3.18e+00
x_{68}	1.17e-01	4.78e-02	1.81e-02	7.87e-02	1.35e-01	1.58e-01	1.83e-01
x_{69}	1.13e-01	2.94e-02	4.29e-02	8.90e-02	1.12e-01	1.28e-01	2.05e-01
x_{70}	8.54e-02	2.98e-02	2.07e-02	6.62e-02	8.03e-02	1.02e-01	2.01e-01
x_{71}	4.63e-01	1.19e-01	1.64e-01	3.95e-01	4.81e-01	5.56e-01	8.77e-01
x_{72}	5.26e-02	9.60e-03	2.06e-02	4.44e-02	5.65e-02	6.02e-02	6.85e-02
x_{73}	2.17e-03	6.14e-04	1.38e-03	1.74e-03	1.90e-03	2.66e-03	4.13e-03
x_{74}	5.58e-03	1.35e-03	1.52e-03	4.42e-03	5.92e-03	6.72e-03	7.56e-03
x_{75}	6.88e+00	3.65e+00	1.10e+00	3.37e+00	6.82e+00	1.01e+01	1.49e+01
x_{76}	1.76e-02	2.74e-03	8.83e-03	1.57e-02	1.80e-02	1.98e-02	2.33e-02
x_{77}	1.19e+00	5.06e-01	2.87e-01	7.72e-01	1.22e+00	1.62e+00	2.37e+00
x_{78}	2.59e-02	7.65e-03	9.99e-03	1.96e-02	2.75e-02	3.28e-02	3.58e-02
x_{79}	2.33e+00	1.34e+00	3.00e-01	1.32e+00	2.07e+00	3.55e+00	5.19e+00
x_{80}	7.52e-03	2.63e-03	2.02e-03	4.91e-03	8.18e-03	9.40e-03	1.57e-02

Tabela 3 – Análise estatística das variáveis do conjunto de dados

(continuação)

Variável	Média	Desvio Padrão	Mínimo	Quartil 1	Quartil 2	Quartil 3	Máximo
x_{81}	1.55e+00	7.25e-01	1.20e-01	9.23e-01	1.95e+00	2.13e+00	2.57e+00
x_{82}	6.88e-01	1.94e-01	1.85e-01	6.08e-01	7.11e-01	7.99e-01	1.27e+00
x_{83}	1.20e-02	4.14e-03	1.48e-03	9.15e-03	1.11e-02	1.56e-02	2.18e-02
x_{84}	2.71e+00	1.20e+00	6.49e-01	1.96e+00	2.46e+00	3.69e+00	6.53e+00
x_{85}	1.69e+00	4.72e-01	4.40e-01	1.46e+00	1.75e+00	2.02e+00	3.51e+00
x_{86}	1.03e+00	6.59e-01	9.73e-02	6.65e-01	8.30e-01	1.39e+00	2.49e+00
x_{87}	4.80e-02	3.06e-02	6.73e-03	2.25e-02	3.52e-02	7.85e-02	1.24e-01
x_{88}	2.56e-02	9.59e-03	5.79e-03	1.62e-02	2.66e-02	3.48e-02	4.05e-02
x_{89}	2.14e+00	1.09e+00	5.02e-01	1.14e+00	1.94e+00	3.16e+00	4.27e+00
x_{90}	2.19e+00	1.28e+00	7.15e-03	1.46e+00	2.45e+00	3.21e+00	4.87e+00
x_{91}	1.47e+00	1.15e+00	1.13e-01	4.73e-01	1.05e+00	2.98e+00	3.40e+00
x_{92}	2.58e-02	8.39e-03	8.59e-03	1.89e-02	2.80e-02	3.33e-02	3.73e-02
x_{93}	9.70e-01	7.09e-01	1.04e-01	3.35e-01	5.80e-01	1.76e+00	1.98e+00
x_{94}	7.54e+00	2.26e+00	4.82e+00	5.78e+00	7.06e+00	8.74e+00	2.00e+01
x_{95}	7.53e+00	1.79e+00	4.77e+00	6.06e+00	7.29e+00	8.59e+00	1.57e+01
x_{96}	8.12e-02	1.54e-02	3.39e-02	8.13e-02	8.59e-02	9.00e-02	1.18e-01
x_{97}	2.45e+01	7.93e+00	6.83e+00	1.92e+01	2.37e+01	2.83e+01	6.79e+01
x_{98}	2.02e+00	1.16e+00	2.58e-01	9.49e-01	1.80e+00	3.28e+00	4.21e+00
x_{99}	5.27e+00	6.90e-01	3.17e+00	4.69e+00	5.15e+00	5.78e+00	7.20e+00
x_{100}	5.94e-01	2.42e-01	1.51e-01	4.64e-01	5.17e-01	8.07e-01	1.20e+00
x_{101}	4.89e+00	2.17e+00	1.26e+00	2.91e+00	4.91e+00	7.04e+00	9.37e+00
x_{102}	4.30e-01	1.94e-01	1.07e-01	2.60e-01	3.94e-01	6.37e-01	8.33e-01
x_{103}	3.69e+00	2.59e-01	3.15e+00	3.43e+00	3.72e+00	3.90e+00	5.48e+00
x_{104}	4.35e+00	1.62e+00	1.32e+00	2.90e+00	4.29e+00	5.69e+00	8.67e+00
x_{105}	4.83e+00	6.34e-01	3.85e+00	4.71e+00	4.80e+00	4.89e+00	2.39e+01
x_{106}	5.94e-02	9.03e-03	3.53e-02	5.24e-02	5.65e-02	6.56e-02	8.86e-02
x_{107}	3.48e+00	4.76e-01	2.70e+00	3.09e+00	3.40e+00	4.09e+00	4.82e+00
x_{108}	6.08e+00	7.41e-01	3.68e+00	5.45e+00	5.99e+00	6.62e+00	8.22e+00
x_{109}	4.98e+00	6.63e-01	4.61e+00	4.78e+00	4.85e+00	4.97e+00	2.32e+01
x_{110}	9.42e-01	3.54e-01	3.13e-01	6.38e-01	9.18e-01	1.25e+00	1.93e+00
x_{111}	5.65e-02	2.29e-02	1.75e-02	3.77e-02	5.23e-02	7.65e-02	1.32e-01
x_{112}	2.69e-02	8.92e-03	7.95e-03	1.74e-02	3.22e-02	3.42e-02	3.74e-02
x_{113}	2.62e+00	8.57e-01	5.45e-01	1.88e+00	2.76e+00	3.29e+00	4.69e+00
x_{114}	2.47e-01	4.95e-02	1.00e-01	2.17e-01	2.66e-01	2.82e-01	3.81e-01
x_{115}	1.10e+00	2.89e-01	3.30e-01	8.89e-01	1.21e+00	1.31e+00	2.06e+00
x_{116}	2.47e+00	1.01e+00	6.73e-01	1.50e+00	2.39e+00	3.36e+00	4.55e+00

Tabela 3 – Análise estatística das variáveis do conjunto de dados

(conclusão)

Variável	Média	Desvio Padrão	Mínimo	Quartil 1	Quartil 2	Quartil 3	Máximo
x_{117}	6.08e-01	1.50e-01	2.48e-01	4.89e-01	6.48e-01	7.20e-01	1.00e+00
x_{118}	1.62e+00	1.14e+00	1.21e-01	6.37e-01	1.36e+00	2.59e+00	4.21e+00
x_{119}	9.17e+00	5.21e+00	1.67e+00	4.48e+00	7.95e+00	1.45e+01	1.98e+01
x_{120}	2.17e-01	6.44e-02	1.03e-01	1.68e-01	1.78e-01	2.80e-01	3.75e-01
x_{121}	5.47e+00	2.00e+00	2.02e+00	3.82e+00	5.03e+00	7.08e+00	1.10e+01
x_{122}	4.45e+00	1.43e+00	9.48e-01	3.28e+00	4.73e+00	5.55e+00	7.94e+00
x_{123}	3.69e+00	2.91e-01	2.44e+00	3.48e+00	3.79e+00	3.89e+00	4.67e+00
x_{124}	1.01e-01	1.84e-02	3.88e-02	9.13e-02	1.04e-01	1.14e-01	2.13e-01
x_{125}	1.20e+01	5.68e+00	3.01e+00	6.88e+00	1.08e+01	1.79e+01	2.31e+01
x_{126}	1.58e+00	4.12e-01	4.99e-01	1.32e+00	1.64e+00	1.88e+00	3.20e+00
x_{127}	7.66e-01	1.61e-01	3.19e-01	6.77e-01	8.02e-01	8.80e-01	1.25e+00
x_{128}	4.52e-02	1.06e-02	2.31e-02	3.65e-02	4.25e-02	5.21e-02	8.22e-02
x_{129}	9.16e+00	6.94e+00	4.73e+00	5.15e+00	5.87e+00	1.04e+01	6.73e+01
x_{130}	3.50e-01	1.04e-01	1.16e-01	3.00e-01	3.45e-01	4.24e-01	7.43e-01
x_{131}	2.25e-01	6.82e-02	8.22e-02	1.56e-01	2.54e-01	2.83e-01	3.80e-01
x_{132}	2.36e+01	2.55e+00	1.83e+01	2.17e+01	2.30e+01	2.53e+01	3.16e+01
x_{133}	8.98e+00	1.47e+00	0.00e+00	7.92e+00	9.49e+00	1.03e+01	1.04e+01
x_{134}	2.32e+03	1.24e+03	-3.88e+01	1.27e+03	2.23e+03	3.44e+03	4.30e+03
x_{135}	7.79e+00	1.93e+00	1.71e+00	6.49e+00	7.79e+00	8.90e+00	1.57e+01

Fonte: Próprio autor.

Além do tratamento dos valores ausentes, foi efetuada uma seleção dos dados de operação normal (significa que o aerogerador está operando sem nenhuma limitação) e de operação com anomalias, uma vez que os dados normais serão utilizados na entrada dos modelos de detecção de novidade, e também, para extrair métricas da performance dos modelos. Os dados com as informações das operações dos aerogeradores foram obtidos de outra fonte de dados também disponibilizada pela empresa, o conjunto de dados, da mesma forma que o conjunto anterior, possui as informações agregadas em intervalos de 10 minutos.

Para realizar o treinamento de modelos semi supervisionados e para obter métricas da performance dos modelos, foram considerados como dados anômalos os últimos 5 dias antes do aerogerador ficar inoperante. A Figura 21 indica que a falha aconteceu no dia 22 de outubro de 2022, portanto, os dados entre os dias 16 e 21 de outubro serão os dados anômalos.

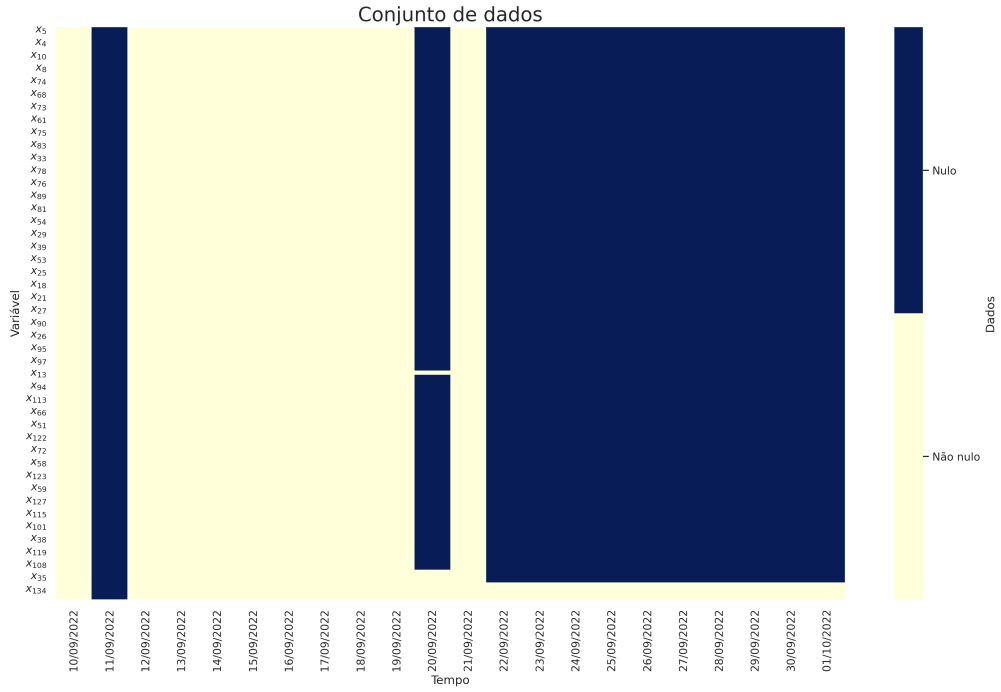
Por fim, os dados foram separados em dois conjuntos, o primeiro contendo apenas dados normais e o segundo contendo apenas dados considerados anomalias, como ilustra a Tabela 4.

Tabela 2 – Exemplo de alguns dados do conjunto em estudo

x_1	x_2	x_3	x_4	x_5	$\cdots$	x_{133}	x_{134}	x_{135}
2.36e-05	3.37e-05	4.25e-05	6.85e-05	7.44e-02	$\cdots$	8.11e+00	1.39e+03	6.67e+00
2.36e-05	3.39e-05	4.24e-05	6.88e-05	5.50e-02	$\cdots$	8.14e+00	1.41e+03	6.69e+00
2.36e-05	3.39e-05	4.25e-05	6.87e-05	5.50e-02	$\cdots$	7.61e+00	1.13e+03	5.98e+00
2.36e-05	3.40e-05	4.25e-05	6.87e-05	5.52e-02	$\cdots$	8.22e+00	1.44e+03	6.74e+00
2.29e-05	3.12e-05	4.23e-05	6.20e-05	5.56e-02	$\cdots$	8.22e+00	1.47e+03	6.75e+00
2.25e-05	3.11e-05	4.19e-05	6.52e-05	5.57e-02	$\cdots$	7.29e+00	9.66e+02	6.03e+00
2.36e-05	3.40e-05	4.02e-05	6.85e-05	5.11e-02	$\cdots$	7.55e+00	1.13e+03	6.34e+00
2.29e-05	3.14e-05	4.24e-05	6.54e-05	7.33e-02	$\cdots$	7.87e+00	1.21e+03	6.51e+00
2.36e-05	3.37e-05	4.16e-05	6.80e-05	5.61e-02	$\cdots$	8.08e+00	1.41e+03	6.78e+00
2.36e-05	3.35e-05	4.25e-05	6.79e-05	7.30e-02	$\cdots$	7.65e+00	1.10e+03	6.35e+00
2.24e-05	3.15e-05	4.16e-05	6.65e-05	5.62e-02	$\cdots$	7.67e+00	1.20e+03	6.49e+00
2.28e-05	3.11e-05	4.25e-05	6.15e-05	7.30e-02	$\cdots$	7.11e+00	9.64e+02	6.18e+00
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
2.25e-05	3.21e-05	4.15e-05	6.82e-05	5.63e-02	$\cdots$	8.75e+00	1.70e+03	7.33e+00
2.37e-05	3.30e-05	4.18e-05	6.98e-05	5.16e-02	$\cdots$	8.93e+00	2.10e+03	7.74e+00

Fonte: Próprio autor.

Figura 21 – Análise do período pré falha



Fonte: Próprio autor.

3.3 MODELAGEM DO PROBLEMA

Para criar os modelos de previsão de ML, foram testados alguns tipos de algoritmos semi supervisionados e não supervisionados. Não foram utilizados algoritmos supervisionados devido à baixa proporção de dados anômalos. Os algoritmos utilizados foram o *isolation forest*, o OCSVM e modelos de *deep learning* como os *autoencoders*. Nesta seção, são utilizadas a linguagem Python em conjunto

Tabela 4 – Conjunto de dados preparados

Conjunto de dados	Dimensão	
	Linhas	Colunas
Normal	10387	135
Anomalia	294	135

Fonte: Próprio autor.

com as bibliotecas livres e o conjunto de dados preparado na seção anterior.

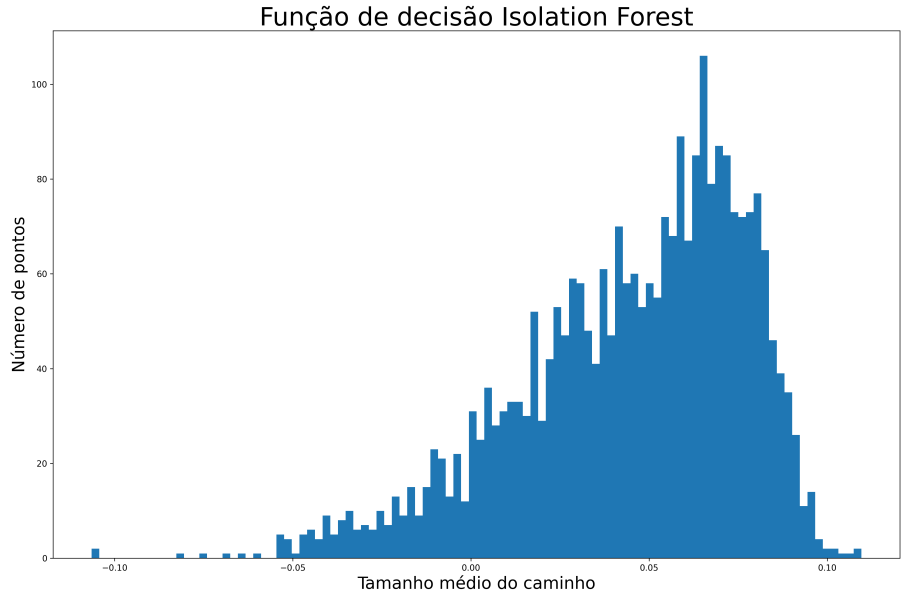
3.3.1 *Isolation Forest*

O algoritmo *isolation forest* é utilizado para detecção de anomalias, portanto, os conjuntos de dados normais e anômalos são concatenados, embaralhados de forma aleatória e divididos em 2 conjuntos, treinamento e teste, com as proporções 0,75 e 0,25, respectivamente.

Após o treinamento do modelo, foi possível obter a função de decisão que irá retornar o tamanho médio do caminho que cada ponto levou para ser isolado. Com isso, pôde-se estabelecer um valor limite (*threshold*) do tamanho e, desta maneira, determinar as anomalias. Isto foi possível devido ao fato de anomalias possuírem valores bem diferentes do considerado normal, e portanto, conseguem ser isoladas mais facilmente com um caminho menor (LIU; TING; ZHOU, 2008).

A Figura 22 ilustra o tamanho médio do caminho para o conjunto de dados de teste. A Tabela 5 mostra os valores das métricas dependendo do valor da variável limite (*threshold*) e a Figura 23 apresenta a matriz de confusão para o *threshold* de -0,75.

Figura 22 – Histograma do tamanho do caminho médio para os pontos de dados



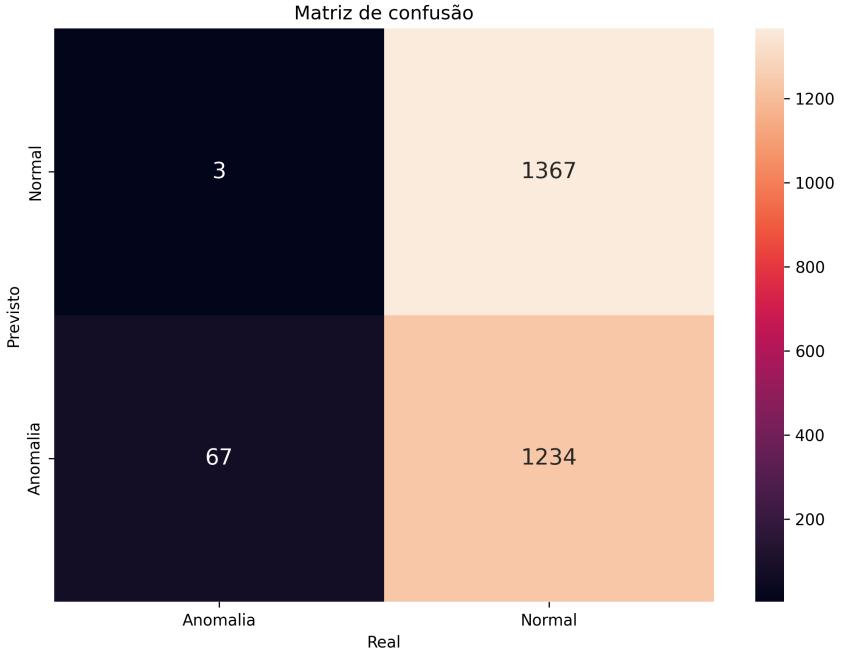
Fonte: Próprio autor.

Tabela 5 – Resumo métricas Isolation Forest

Isolation Forest						
Threshold	-0.05	-0.025	0	0.025	0.05	0.075
Acurácia	0.971	0.947	0.895	0.755	0.537	0.202
Precisão	0.975	0.977	0.981	0.986	0.998	1.000
Recall	0.995	0.968	0.910	0.759	0.526	0.181
F_1 Score	0.985	0.973	0.944	0.858	0.688	0.306
AUC	0.526	0.556	0.626	0.679	0.741	0.590

Fonte: Próprio autor.

Figura 23 – Matriz de confusão para o modelo Isolation Forest com *threshold* = -0,75



Fonte: Próprio autor.

3.3.2 OCSVM

O algoritmo OCSVM é utilizado para detecção de *outliers*, portanto, os dados de treinamento contêm apenas dados normais. O OCSVM foi utilizado através da biblioteca *scikit-learn* que o implementa de forma simples. O conjunto de dados possuem valores numericamente altos e algoritmos de ML não performam bem com entradas em escalas muito diferentes (GÉRON, 2019). Por esse motivo, é necessário aplicar técnicas de *feature scaling* e neste caso foi aplicada a estandardização (*Standardization*) para todas as variáveis.

A estandardização de s é calculada como na Equação 1, no qual u é a média dos valores e s é o desvio padrão.

$$z(x) = \frac{(x - u)}{s} \tag{1}$$

O conjunto de dados foi embaralhado aleatoriamente e separado com 75% dos dados para treinamento e 25% para testes. O modelo possui tipos de *kernel* diferentes e a Tabela 6 apresenta as métricas para o conjunto de dados de teste para alguns dos *kernels* disponíveis na biblioteca.

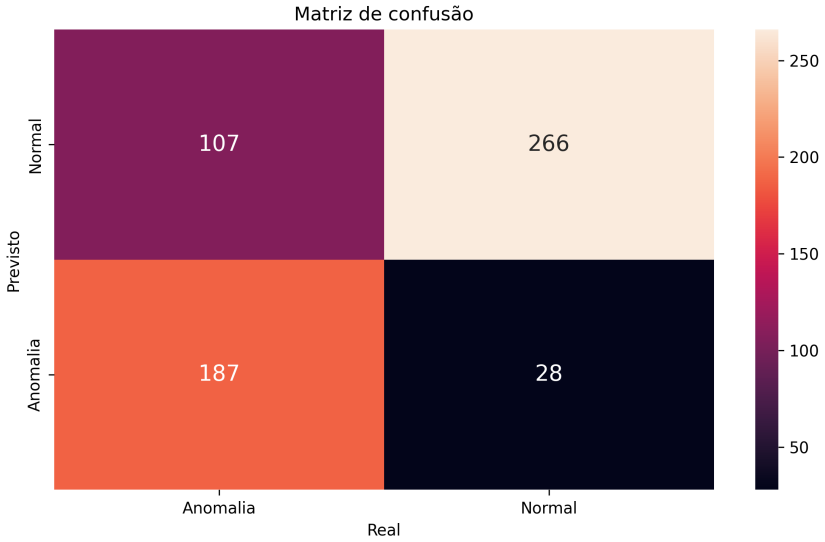
Tabela 6 – Resumo métricas OCSVM

One Class Support Vector Machine				
kernel	rbf	linear	sigmoid	poly
Acurácia	0.764	0.355	0.497	0.435
Precisão	0.712	0.403	0.498	0.465
Recall	0.884	0.602	0.884	0.857
F_1 Score	0.789	0.483	0.637	0.603
AUC	0.764	0.355	0.497	0.435

Fonte: Próprio autor.

O melhor resultado obtido para o OCSVM foi utilizando o *kernel* ”rbf”. A Figura 24 contém a matriz de confusão do melhor resultado.

Figura 24 – Matriz de confusão para o modelo OCSVM com *kernel* = ”rbf”



Fonte: Próprio autor.

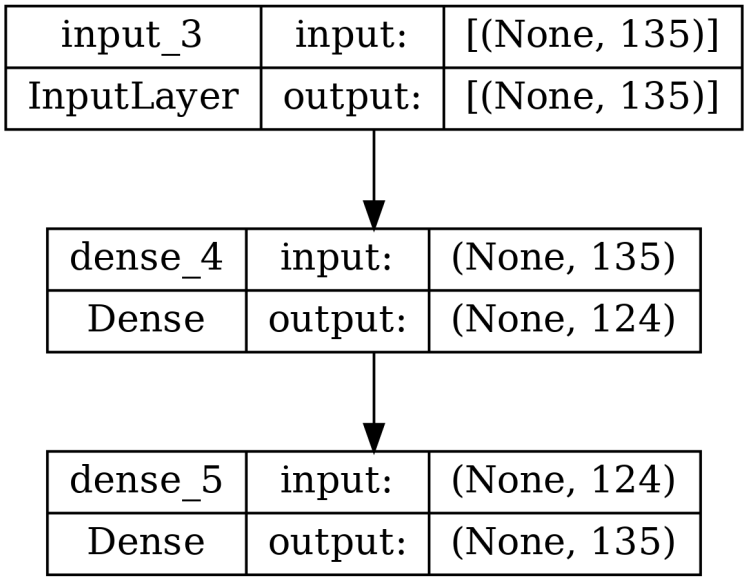
3.3.3 *Autoencoder* simples

O algoritmo de *autoencoder* é utilizado para detectar anomalias avaliando o erro de reconstrução. Uma anomalia é detectada quando o erro ultrapassa um valor predeterminado. O conjunto de dados normais e anômalos é concatenado e, da mesma forma dos modelos anteriores, é efetuado o pré-processamento do conjunto de dados (estandardização, embaralhamento e separação).

O modelo foi montado utilizando o objeto Sequential da biblioteca Keras e, para encontrar o *bottleneck* do modelo, foi utilizado o módulo keras-tuner que oferece métodos para efetuar a busca de hiper parâmetros ótimos para o modelo.

O treinamento do modelo utilizou como otimizador a função Adam com a métrica das perdas sendo o erro quadrático médio (*mean squared error - MSE*) e as métricas de avaliação sendo o erro absoluto médio (*mean absolute error - MAE*) e a acurácia. O modelo obtido foi ilustrado na Figura 25, possuindo 124 nós na camada de gargalo.

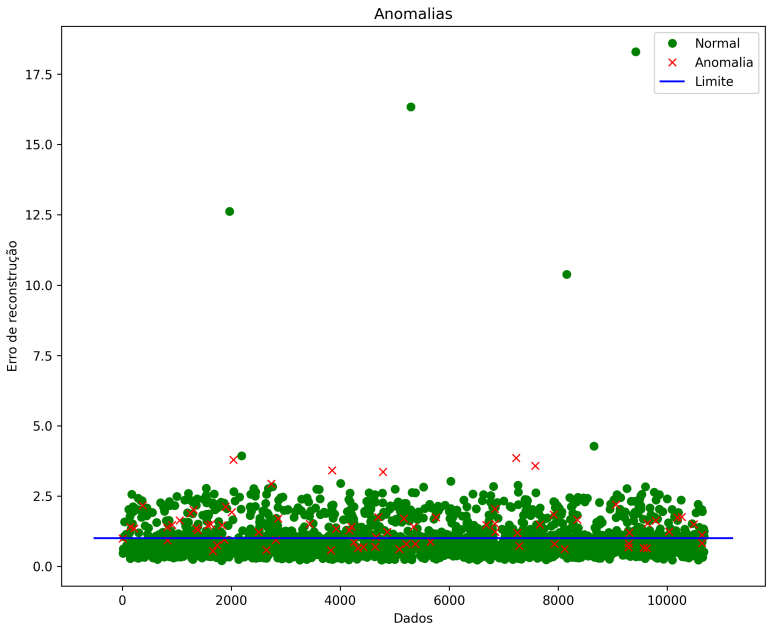
Figura 25 – Modelo *autoencoder* simples



Fonte: Próprio autor.

Com o modelo já treinado, calcula-se o erro de reconstrução para cada dado do conjunto de teste e é traçado um gráfico do erro para cada dado, como ilustrado na Figura 26. A Tabela 7 expõe os valores das métricas para diferentes valores de *threshold*.

Figura 26 – Separação das anomalias para o modelo de *autoencoder* simples



Fonte: Próprio autor.

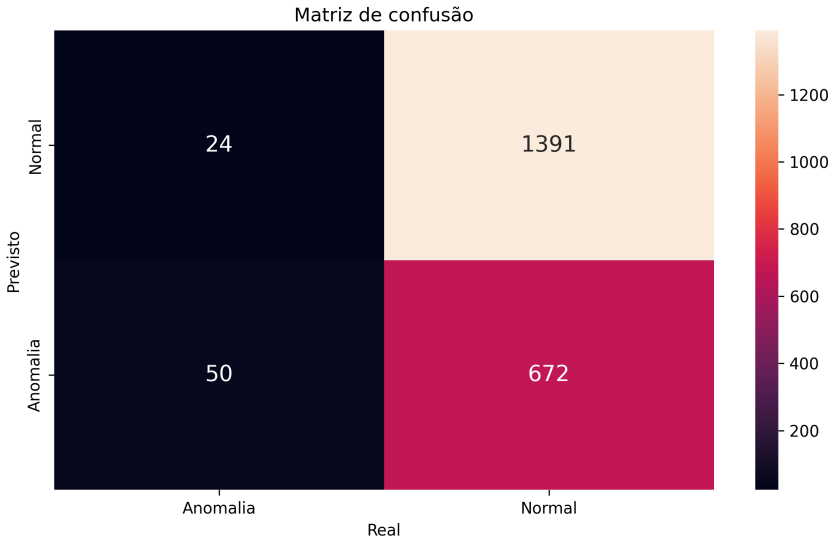
O melhor resultado obtido foi adotando um *threshold* de 1,0 e a Figura 27 demonstra a matriz de confusão.

Tabela 7 – Resumo métricas *autoencoder* simples

Autoencoder Simples					
Threshold	0.5	1.0	1.5	2.0	2.5
Acurácia	0.226	0.674	0.817	0.883	0.947
Precisão	1.000	0.983	0.973	0.968	0.967
Recall	0.199	0.674	0.834	0.910	0.978
F_1 Score	0.332	0.800	0.898	0.938	0.973
AUC	0.599	0.675	0.593	0.529	0.530

Fonte: Próprio autor.

Figura 27 – Matriz de confusão para o modelo de *autoencoder* simples com *threshold* = 1



Fonte: Próprio autor.

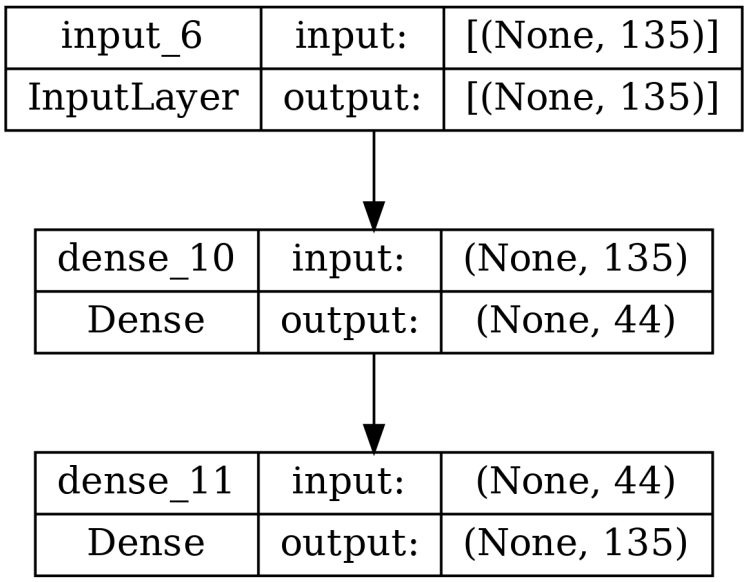
3.3.4 *Autoencoder* esparso

O algoritmo do *autoencoder* esparso possui uma estrutura similar a do *autoencoder* simples, sua diferença consiste apenas em uma regularização adicionada na camada de *bottleneck*. Essa regularização aplica uma penalidade na saída da camada que posteriormente é computada nas perdas do treinamento. Isso implica em menos nós sendo disparados, forçando o modelo a buscar representações mais compactas.

O modelo foi treinado do mesmo modo que no *autoencoder* simples e obteve o modelo ilustrado na Figura 28 contendo 44 nós na camada de *bottleneck*.

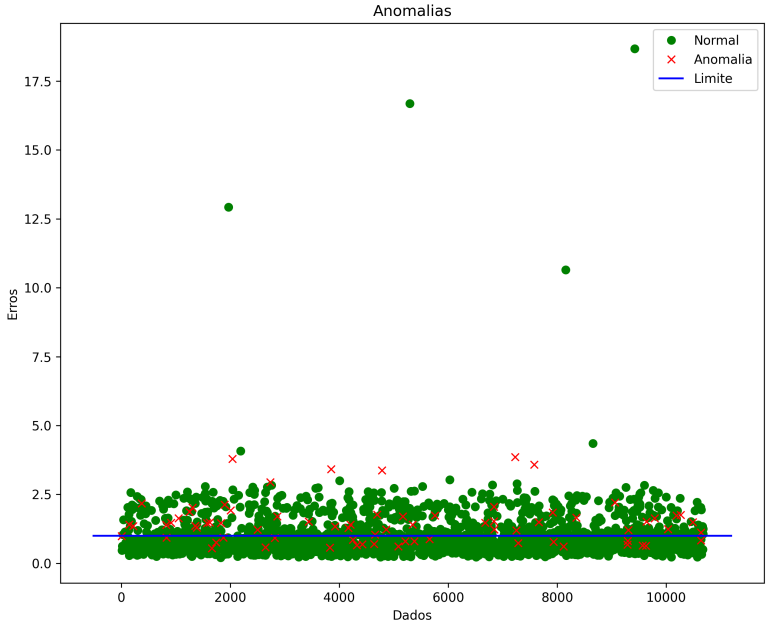
A Figura 29 ilustra o erro de reconstrução das previsões do modelo aplicado ao conjunto de dados de teste. A Tabela 8 apresenta as métricas com diferentes valores de *threshold*, sendo o melhor resultado obtido com o valor 1,0 e a metriz de confusão ilustrada na Figura 30.

Figura 28 – Modelo *autoencoder* esparso



Fonte: Próprio autor.

Figura 29 – Separação das anomalias para o modelo de *autoencoder* esparso



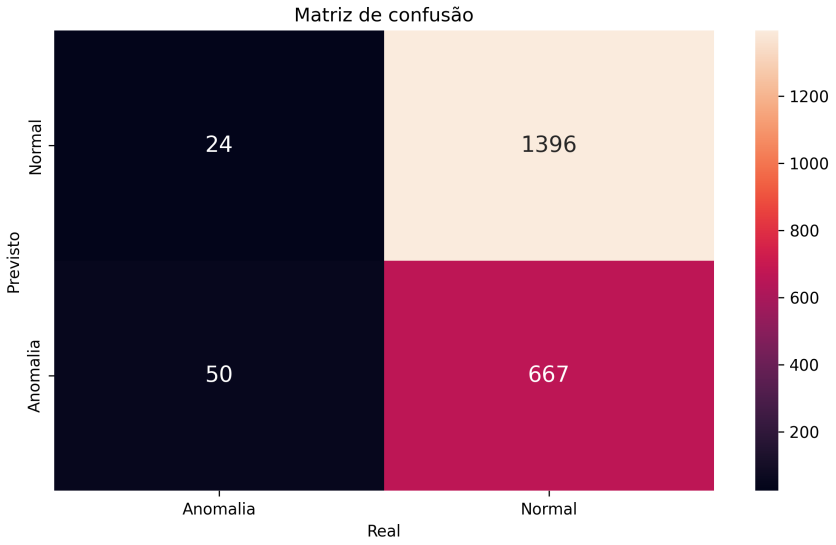
Fonte: Próprio autor.

Tabela 8 – Resumo métricas *autoencoder* esparso

Autoencoder Esparso					
Threshold	0.5	1.0	1.5	2.0	2.5
Acurácia	0.227	0.677	0.817	0.883	0.947
Precisão	1.000	0.983	0.973	0.968	0.967
Recall	0.199	0.677	0.834	0.910	0.978
F_1 Score	0.332	0.802	0.898	0.938	0.973
AUC	0.600	0.676	0.593	0.529	0.530

Fonte: Próprio autor.

Figura 30 – Matriz de confusão para o modelo de *autoencoder* esparso com *threshold* = 1



Fonte: Próprio autor.

3.3.5 *Deep autoencoder* esparso

O algoritmo do *deep autoencoder* esparso utiliza os conceitos dos *autoencoders* anteriores, porém adicionando mais camadas ocultas no modelo, aumentando a sua complexidade e o desempenho.

O conjunto de dados foi pré-processado aplicando a estandardização e os dados foram separados em 64% para treino, 16% para validação e 20% para testes.

O modelo obtido através da busca de parâmetros foi ilustrado na Figura 31 contendo 3 camadas ocultas de *encoder* com 116, 60 e 38 nós, respectivamente. O *decoder* possui as mesmas quantidades de nós. Também foi aplicada regularização nas camadas *dense_12* e *dense_13*.

As camadas do tipo Dropout definem aleatoriamente algumas entradas do nó para zero com uma frequência de 20%, isto ajuda a prevenir um *overfitting* do modelo.

O modelo foi treinado com 500 épocas e com as métricas da precisão durante a validação, e foi extraída a quantidade de épocas que gerou uma melhor precisão, e o modelo foi retreinado com este novo valor de épocas. A Figura 32 ilustra o modelo visualizado através do TensorBoard.

Utilizando o conjunto de dados de teste, foi traçado o gráfico do erro de reconstrução para cada dado, ilustrado na Figura 33.

A Tabela 9 apresenta os valores de algumas métricas para alguns valores de *threshold*. O melhor resultado obtido foi com o valor 4,9, como mostra a Tabela 10 e a Figura 34.

Tabela 9 – Resumo métricas *deep autoencoder* esparso

Deep Autoencoder Esparso					
Threshold	1.0	2.0	3.0	4.0	5.0
Acurácia	0.034	0.087	0.503	0.726	0.923
Precisão	0.000	1.000	1.000	0.995	0.988
Recall	0.000	0.055	0.486	0.720	0.932
F_1 Score	0.000	0.104	0.654	0.835	0.959
AUC	0.500	0.527	0.743	0.804	0.806

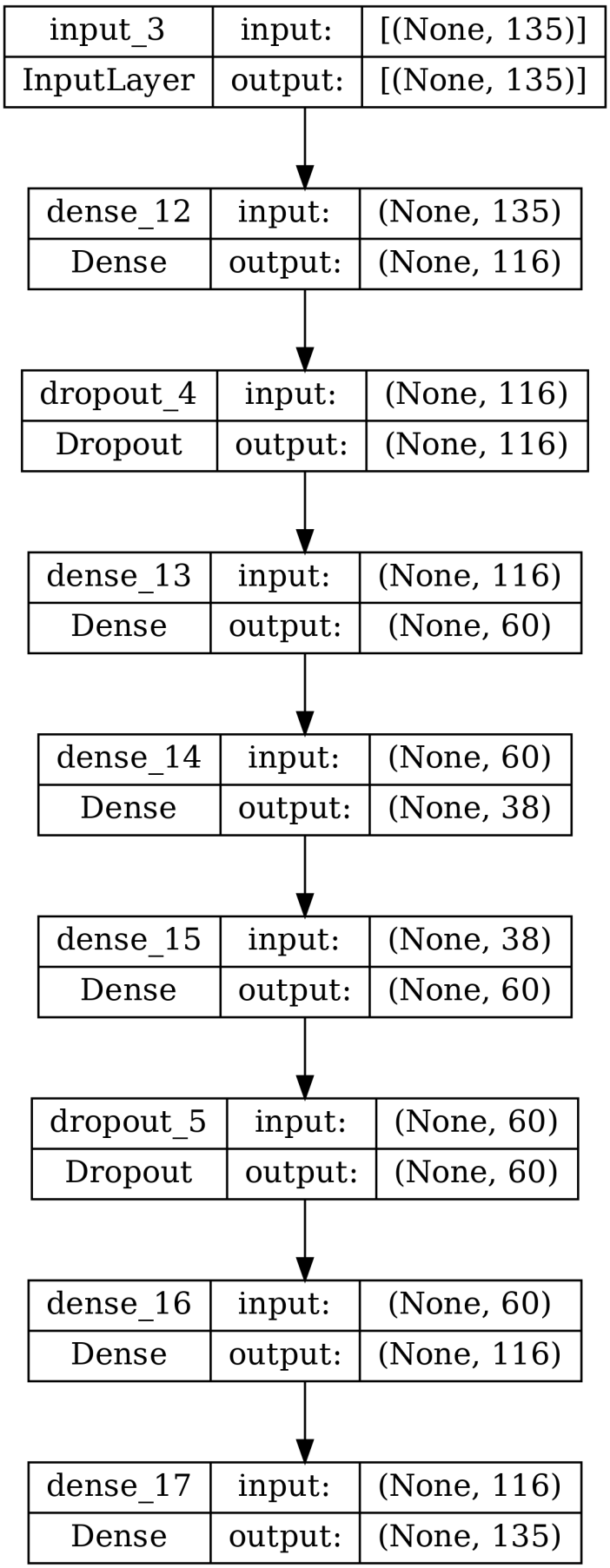
Fonte: Próprio autor.

Tabela 10 – Métricas *deep autoencoder* esparso com threshold de 4,9

Acurácia	0.901
Precisão	0.989
Recall	0.908
F_1 Score	0.947
AUC	0.815

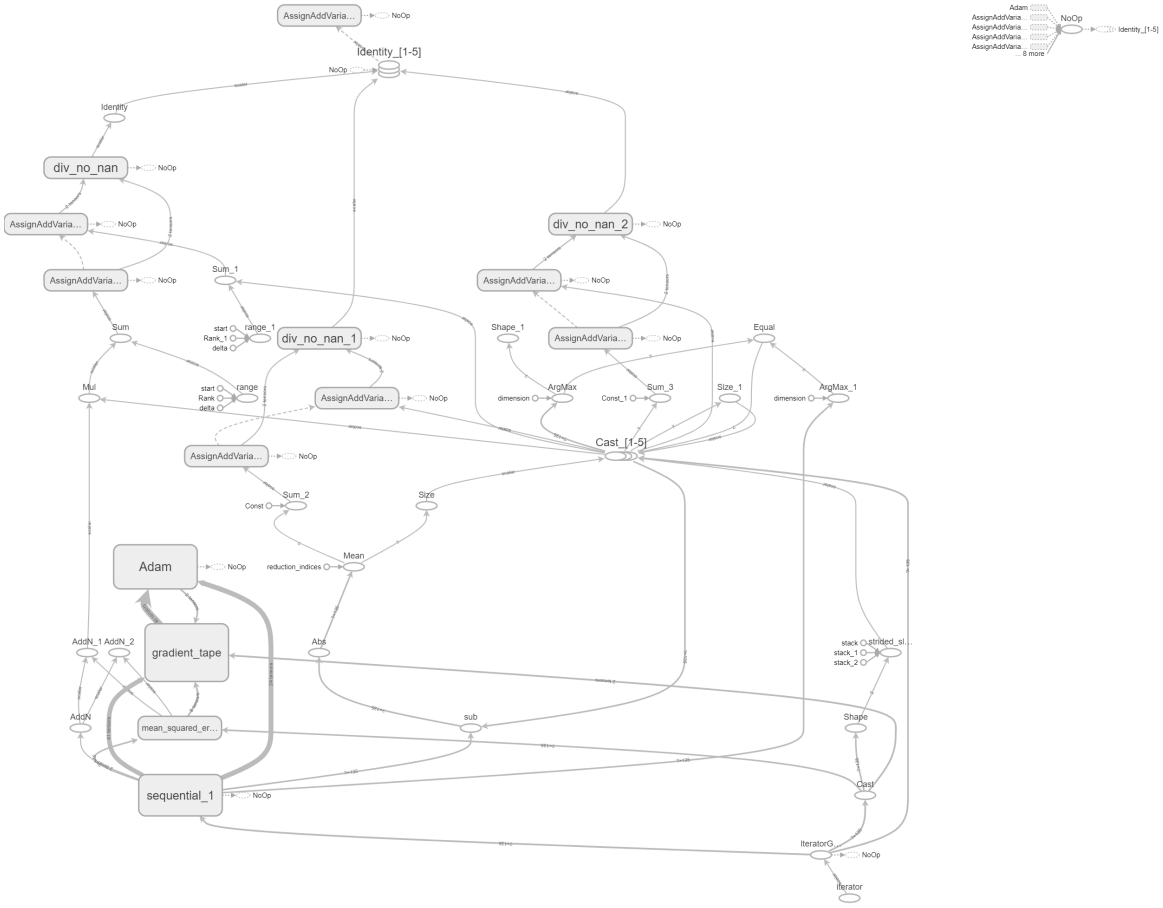
Fonte: Próprio autor.

Figura 31 – Modelo *deep autoencoder* esparso



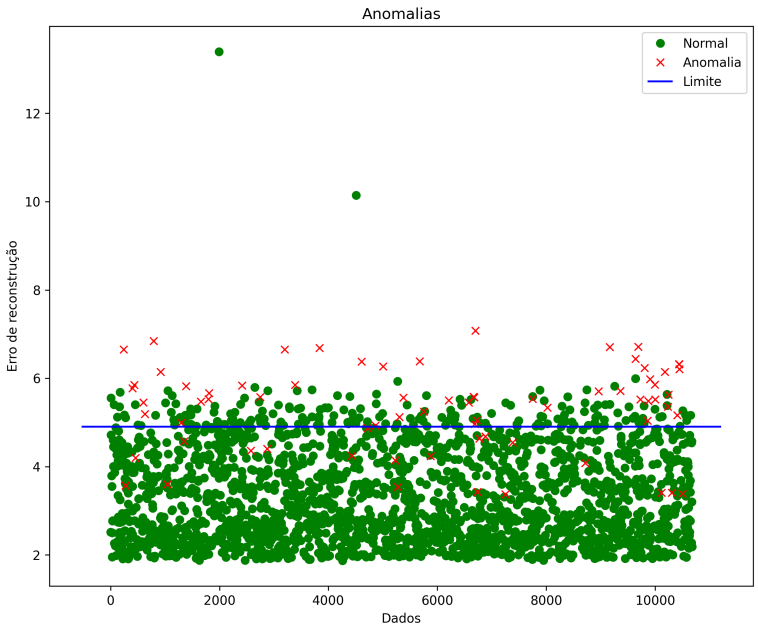
Fonte: Próprio autor.

Figura 32 – Visualização do modelo com TensorBoard



Fonte: Próprio autor.

Figura 33 – Separação das anomalias para o modelo de *deep autoencoder* esparso



Fonte: Próprio autor.

Figura 34 – Matriz de confusão para o modelo de *deep autoencoder* esparso com *threshold* = 4,9

Fonte: Próprio autor.

3.4 TRANSFERÊNCIA DE APRENDIZADO

Dos modelos estudados, o que apresentou uma melhor performance em classificar as anomalias foi o *deep autoencoder* esparso, que obteve um AUC de 0,815. Desta forma, ele será o modelo base para realizar a transferência de aprendizado para os dois aerogeradores que não apresentaram falhas no período de estudo.

A transferência de aprendizado para cada um dos aerogeradores consistiu em carregar o conjunto de dados e aplicar o mesmo pré-processamento realizado no treinamento do modelo base. O modelo base então é carregado, a última camada do modelo é removida e os pesos treinados do modelo são travados.

Um novo modelo para o aerogerador é criado utilizando o modelo base e adicionando uma nova camada no lugar da camada que foi removida. Deste modo, quando o modelo for treinado, apenas os pesos da última camada poderão ser alterados, como ilustra a Figura 35.

Figura 35 – Modelo para transferência de aprendizado compilado

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_12 (Dense)            (None, 116)               15776

 dropout_4 (Dropout)         (None, 116)               0

 dense_13 (Dense)            (None, 60)                7020

 dense_14 (Dense)            (None, 38)                2318

 dense_15 (Dense)            (None, 60)                2340

 dropout_5 (Dropout)         (None, 60)                0

 dense_16 (Dense)            (None, 116)               7076

 output (Dense)              (None, 135)               15795

=================================================================
Total params: 50,325
Trainable params: 15,795
Non-trainable params: 34,530
_________________________________________________________________
```

Fonte: Próprio autor.

O treinamento do modelo foi realizado utilizando 100 épocas, tamanho do lote de 16 e validação sendo 10% do conjunto de dados de treinamento. Posteriormente, foram destravados todos os pesos desse novo modelo e realizou-se um novo treinamento com apenas 10 épocas. Esse novo treinamento é chamado de ajuste fino do modelo.

Este processo foi realizado para os dois aerogeradores e obteve-se os resultados da Tabela 11 utilizando o valor de *threshold* igual a 4,9.

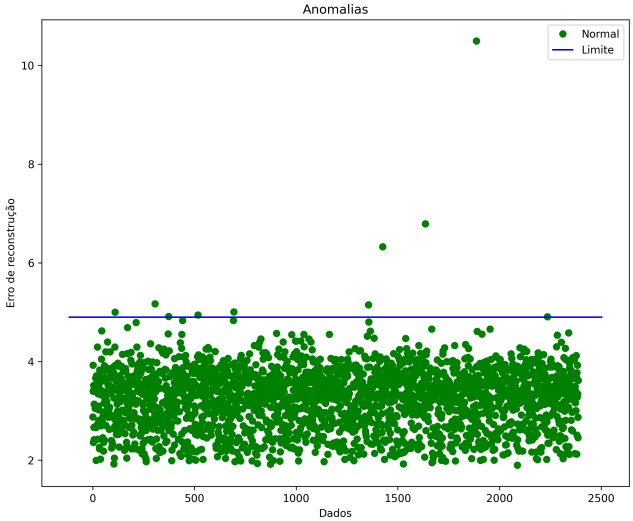
Tabela 11 – Métricas obtidas da transferência de aprendizado com *threshold* igual a 4,9

	Aerogerador 1	Aerogerador 2
Acurácia	0.996	0.997
Precisão	1.000	1.000
Recall	0.996	0.997
F_1 Score	0.998	0.999

Fonte: Próprio autor.

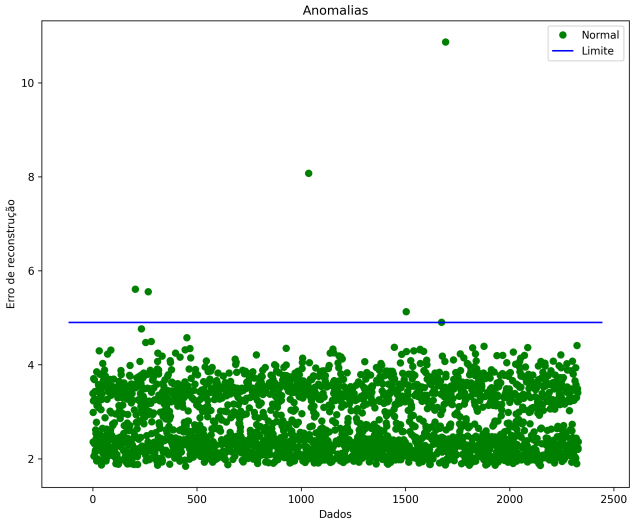
A Figura 36 e a Figura 37 ilustram a separação do dados para os aerogeradores que não apresentaram falhas no período de estudo.

Figura 36 – Separação das anomalias utilizando transferência de aprendizado do modelo base para o Aerogerador 1



Fonte: Próprio autor.

Figura 37 – Separação das anomalias utilizando transferência de aprendizado do modelo base para o Aerogerador 2



Fonte: Próprio autor.

3.5 ANÁLISE DOS RESULTADOS OBTIDOS

Analisando os melhores resultados de cada um dos modelos estudados (Tabela 12), ficou evidente que o *Deep autoencoder* esparso apresentou uma performance melhor em prever anomalias consideradas falhas no aerogerador. O modelo que obteve a menor performance foi o *Autoencoder* simples e esse resultado pode ter como origem a falta de complexidade do modelo que não foi capaz de aprender as *features* mais importantes para a reconstrução do conjunto de dados. Não foi possível obter a métrica AUC pois todos os dados reais foram considerados normais.

Tabela 12 – Resumo performance dos modelos estudados

Modelo	Acurácia	Precisão	*Recall*	F_1 *score*	AUC
Isolation Forest	0,537	0,998	0,526	0,688	0,741
OCSVM	0,764	0,712	0,884	0,789	0,764
***Autoencoder* simples**	0,674	0,983	0,674	0,800	0,675
***Autoencoder* esparso**	0,677	0,983	0,677	0,802	0,676
***Deep autoencoder* esparso**	0,901	0,989	0,908	0,947	0,815

Fonte: Próprio autor.

O modelo *Deep autoencoder* esparso obtido apresenta um bom resultado para auxiliar a tomada de decisão sobre as paradas de manutenção preventivas a fim de evitar falhas que causam um prejuízo financeiro à empresa.

4 CONCLUSÃO

A operação de parques eólicos no Brasil é um grande desafio, devido a complexidade do setor aliada a fatores como dificuldades de logística e cadeia de suprimentos e fornecedores de serviços qualificados.

Possuir sistemas capazes de prever falhas antes delas ocorrerem possibilita obter vantagens da manutenção preditiva.

> Algumas vantagens da manutenção preditiva para O&M de parques eólicos: permite melhor planejamento da manutenção - programação das atividades para períodos de baixo vento; viabiliza a otimização da logística das peças de substituição e facilita o acesso aos equipamentos auxiliares de manutenção - alguns componentes como as caixas multiplicadoras, as pás e alguns rolamentos possuem prazos elevados de entrega, e equipamentos auxiliares podem estar indisponíveis, impossibilitando o reparo ou aumentado seu custo e alongando o período de indisponibilidade; possibilita melhor coordenação das equipes de manutenção e facilita a contratação de serviços externos; permite a realização de diagnósticos e previsão de falhas sem adição extra de instrumentação, sem paradas e sem custos extras e potenciais implicações de garantia. (GONZÁLEZ et al., 2018, p. 3)

Nesse contexto, este trabalho descreve uma abordagem para a detecção de anomalias utilizando técnicas de *deep learning* para contribuir com a resolução de um problema real. Foi utilizado especificamente o modelo de *deep autoencoder* para obter os melhores resultados em prever falhas. Além disso, a execução do trabalho consistiu em três etapas principais (exploração do conjunto de dados, escolha do modelo de ML, previsão de falhas), e cada etapa foi composta de tarefas individuais que podem ser aplicadas a outros tipos de problemas nos quais se deseja detectar anomalias em um conjunto de dados.

4.1 PROPOSTA PARA PESQUISAS FUTURAS

Este trabalho abordou a criação de modelos capazes de prever falhas nos aerogeradores, deste modo, outras atividades podem ser aplicadas para contribuir com o trabalho.

- Aplicar treinamento em lote para cada aerogerador a fim de manter o modelo atualizado.
- Aplicar o modelo em um sistema utilizando arquitetura em nuvem para criar alertas sobre possíveis falhas.
- Aplicar a sistemática proposta para detectar falhas em componentes isolados do aerogerador.
- Aplicar a sistemática proposta para detectar falhas em outro tipo de sistemas mecânicos.

- Aplicar a metodologia separando as variáveis por componente do aerogerador para a detecção de falhas em componentes individuais.

REFERÊNCIAS

ABEEÓLICA. **Boletim de geração eólica**. 2022. Disponível em: <https://abeeolica.org.br/>. Acesso em: 16 nov. 2022.

ALLA, S.; ADARI, S. K. **Beginning anomaly detection using python-based deep learning**. Apress, 2019. Disponível em: <https://doi.org/10.1007/978-1-4842-5177-5>.

CARROLL, J. et al. Wind turbine gearbox failure and remaining useful life prediction using machine learning techniques. **Wind Energy**, Wiley, v. 22, n. 3, p. 360–375, nov. 2018. Disponível em: <https://doi.org/10.1002/we.2290>. Acesso em: 17 nov. 2022.

ELSEVIER. **Scopus**. 2022. Disponível em: <https://www.scopus.com/>. Acesso em: 29 dez. 2022.

GÉRON, A. **Hands-on machine learning with scikit-learn, KerSas, and TensorFlow**. Sebastopol, CA - EUA: O'Reilly Media, 2019.

GONZÁLEZ, M. et al. Operação e manutenção de parques eólicos do brasil: Desafios e oportunidades. **Brazil Windpower Conference and Exhibition**, 2018. Disponível em: <https://abeeolica.org.br/wp-content/uploads/2018/09/MARIO-GONZALES.pdf>. Acesso em: 20 dez. 2022.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>.

JIANG, G. et al. Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. **IEEE Transactions on Instrumentation and Measurement**, v. 66, n. 9, p. 2391–2402, 2017. Cited By :196. Disponível em: <www.scopus.com>. Acesso em: 12 mar. 2022.

JIANG, G. et al. Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox. **IEEE Transactions on Industrial Electronics**, v. 66, n. 4, p. 3196–3207, 2019. Cited By :421. Disponível em: <www.scopus.com>. Acesso em: 10 mar. 2022.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: **2008 Eighth IEEE International Conference on Data Mining**. IEEE, 2008. Disponível em: <https://doi.org/10.1109/icdm.2008.17>.

MCCARTHY, J. What is artificial intelligence. **Computer Science Department**, 2007. Disponível em: <http://jmc.stanford.edu/articles/whatisai/whatisai.pdf>. Acesso em: 13 nov. 2022.

OFFICE, W. E. T. **Zeroing in on the no. 1 cause of wind turbine gearbox failures**. 2020. Disponível em: <https://www.energy.gov/eere/wind/articles/zeroing-no-1-cause-wind-turbine-gearbox-failures>. Acesso em: 16 out. 2022.

ROMERO, A. et al. Vestas v90-3mw wind turbine gearbox health assessment using a vibration-based condition monitoring system. **Shock and Vibration**, Hindawi Limited, v. 2016, p. 1–18, 2016. Disponível em: <https://doi.org/10.1155/2016/6423587>. Acesso em: 15 jun. 2022.

TIOBE. **TIOBE index**. 2022. Disponível em: <https://www.tiobe.com/tiobe-index/>. Acesso em: 18 out. 2022.

ZHAO, H. et al. Anomaly detection and fault analysis of wind turbine components based on deep learning network. **Renewable Energy**, v. 127, p. 825–834, 2018. Disponível em: <www.scopus.com>. Acesso em: 12 mar. 2022.