PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Previsão do Avanço da Corrosão em Instalações Industriais a partir de Dados Climáticos, de Área e de Percentual Corroído

Arthur Xavier Tavares

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Graduação em Ciência da Computação

Rio de Janeiro, Novembro, 2024

Arthur Xavier Tavares

Previsão do Avanço da Corrosão em Instalações Industriais a partir de Dados Climáticos, de Área e de Percentual Corroído

Relatório de Projeto Final, apresentado ao curso de Ciência da Computação como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Paulo Ivson Netto Santos

Rio de Janeiro

Novembro de 2024

Agradecimentos

Primeiramente, gostaria de agradecer à minha família por todo o apoio durante a minha jornada na graduação. Agradeço, sobretudo, aos meus pais, pelos valores que carrego comigo e por terem possibilitado toda a minha formação acadêmica e pessoal desde a minha mais tenra idade.

Faço um agradecimento especial a Marina Rios e Paulo Ivson pela oportunidade que me deram e por toda a assistência que me proporcionaram na produção deste trabalho.

Também agradeço à toda equipe do Instituto Tecgraf, por ter viabilizado a elaboração deste estudo. Em particular, agradeço imensamente a Hugo Neves, que me auxiliou em questões externas do projeto.

Agradeço a todos os meus amigos por permitirem que a minha caminhada se tornasse menos íngreme, por meio apenas de sua amizade. Em especial, agradeço a Alexandre Abrahão, Breno Gomes e Renan Moreira, pelo suporte direto que me deram em momentos difíceis da graduação.

Por fim, agradeço ao Botafogo de Futebol e Regatas, pela abrasadora felicidade que vivi ao longo deste ano e por ter tornado 2024 ainda mais glorioso.

Resumo

Tavares, Arthur, Ivson, Paulo. Previsão do Avanço da Corrosão em Instalações Industriais a partir de Dados Climáticos, de Área e de Percentual Corroído. Rio de Janeiro, 2024. 34 páginas. Relatório Final de Projeto Final – Centro Técnico Científico, Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

A corrosão externa é um dos principais causadores de falhas em equipamentos em instalações industriais, ocasionando em manutenções altamente custosas. Este estudo apresenta uma abordagem de aprendizado de máquina para prever taxas de corrosão com base em dados climáticos, de área e percentual corroído. O modelo utiliza o algoritmo de aprendizado supervisionado *Random Forest*, aproveitando um conjunto de dados de medições de corrosão coletados ao longo do tempo. Além disso, buscou-se incluir novos dados e variáveis ao modelo e avaliar o impacto gerado na performance da previsão. Dessa forma, o principal objetivo deste projeto é, a partir do algoritmo trabalhado, permitir o planejamento de manutenções prescritivas, que possam garantir a segurança operacional e reduzir os custos.

Palavras-chave

Corrosão externa, machine learning, random forest, previsão, manutenção

Abstract

Tavares, Arthur, Ivson, Paulo. Predicting the Progress of Corrosion in Industrial Facilities from Climate, Area and Percentage Corroded Data. Rio de Janeiro, 2024. 34 páginas. Relatório Final de Projeto Final – Centro Técnico Científico, Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

External corrosion is one of the main causes of equipment failures in industrial facilities, leading to highly costly maintenance. This study presents a machine learning approach to predict corrosion rates based on climatic data, area, and percentage of corrosion. The model employs the supervised learning algorithm Random Forest, leveraging a dataset of corrosion measurements collected over time. Additionally, the study aims to incorporate new data and variables into the model and evaluate their impact on prediction performance. Thus, the primary objective of this project is to enable prescriptive maintenance planning through the developed algorithm, ensuring operational safety and reducing costs.

Keywords

External corrosion, machine learning, random forest, prediction, maintenance

Sumário

Lista de Figuras

1. Introdução

Estratégias de manutenção figuram como protagonistas em atividades offshore, uma vez que apresentam alto risco ambiental, social e financeiro. Nesse setor, estima-se que somente os custos de manutenção representam 40% dos custos totais e a maior parte deles está ligada a atividades de manutenção planejada inadequada ou não cientificamente (ELMAS et al., 2023). Portanto, destaca-se que, em virtude desses fatores e por se tratar de um setor intensivo em ativos, compostos por centenas de equipamentos, que operam continuamente, a gestão da manutenção é dita altamente crítica (HAMEED et al., 2019).

As paradas de manutenção, portanto, representam um processo estratégico importante para o planejamento da manutenção dentro de uma planta industrial, sendo assim, fundamentais na gestão de equipamentos que requerem vistorias mais prolongadas e aprofundadas (CAIADO; LIMA; QUELHAS, 2015). Diante disso, a tarefa certa no momento certo com o equipamento certo é essencial para garantir que a instalação permaneça em uma condição operacional confiável (HAMEED et al., 2019).

Dentro desse contexto, a corrosão atmosférica é um dos principais causadores de falhas em equipamentos, ocasionando em reposições altamente custosas (ELMAS et al., 2023). Por se tratar de um dado atmosférico, o percentual de corrosão depende de inúmeros fatores e não pode ser tratado em laboratório, tal que se faz necessário o acompanhamento de informações históricas para a análise desse problema. Dessa forma, a previsão do comportamento da corrosão em instalações industriais offshore torna-se uma tarefa extremamente desafiadora, de maneira que se mostra essencial a realização de uma análise detalhada para cada caso individualmente.

O monitoramento do avanço da corrosão externa é de suma importância no contexto das manutenções prescritivas. Essas manutenções geram planos de pintura visando a previsão de falha e avanço do dano. Dessa forma, conseguir realizar um acompanhamento eficiente do avanço da corrosão é essencial para poder priorizar as atividades de manutenção pela condição ao longo do tempo.

2. Objetivos

O propósito deste projeto de pesquisa é construir uma solução que envolva mecanismos computacionais para agregar dados reais e de simulações, visando a manutenção e integridade de instalações de superfície. Para isso, será realizado o desenvolvimento de melhorias em um modelo de aprendizado máquina utilizado para auxiliar no monitoramento da corrosão externa de instalações industriais.

Devido à disponibilidade e quantidade dos dados disponíveis para o treinamento, não foi possível realizar o desenvolvimento de uma série temporal, que seria o mais adequado pela natureza do problema. Dessa forma, modelos de regressão apresentaram-se como a solução mais apropriada nesse contexto. A Tabela 1 apresenta os valores das métricas do resultado do último treinamento do modelo analisado, anterior aos experimentos realizados neste trabalho.

RMSE treino	**RMSE validação**	**R^2 treino**	**R^2 validação**
0.023072	0.057804	0.913878	0.539043

Tabela 1 – Resultados do último treinamento do modelo.

A principal finalidade do esforço de implementação empenhado neste trabalho é conseguir realizar melhorias no desempenho do modelo de *Random Forest* central do projeto. Para alcançar esse objetivo, buscou-se aumentar a base de dados utilizada no treinamento, a partir de novos dados que foram devidamente tratados e preparados, e inserir novas variáveis para serem levadas em consideração pelo modelo. Além disso, foi feita a atualização da versão da biblioteca *scikit-learn*, utilizada para realizar o treinamento do modelo, o que também pode trazer benefícios ao desempenho do algoritmo.

3. Trabalhos Relacionados

3.1. Técnicas de *Machine Learning*

De maneira geral, o aprendizado de máquina (do inglês *Machine Learning*) é a evolução de algoritmos computacionais capazes de mimetizar a inteligência humana através do aprendizado proporcionado pelo ambiente (EL NAQA; MURPHY, 2015). O *Machine Learning* emerge da interseção entre ciência da computação e estatística, e está no centro da inteligência artificial e da ciência de dados (ELMAS et al., 2023).

Dentro desse contexto, os sistemas de aprendizado de máquina geralmente são divididos em três categorias: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço. O aprendizado supervisionado geralmente inclui entradas e saídas rotuladas, de maneira que é feita a previsão de novas saídas e entradas. Já no aprendizado não-supervisionado, o algoritmo não aprende com dados rotulados, mas tenta encontrar padrões no conjunto de dados analisados. Assim como o aprendizado supervisionado, o aprendizado por reforço se utiliza de entradas e saídas rotuladas; porém, o algoritmo não aprende com a saída rotulada, mas apresenta um feedback sobre a decisão tomada (RAJENDRA; GIRISHA; GUNAVARDHANA NAIDU, 2022).

3.2. Técnicas de *Machine Learning* para previsão

O presente estudo busca enfatizar o uso das técnicas de aprendizado de máquina para previsão. Dentre os modelos analisados, destacaram-se as redes neurais artificiais (*Artificial Neural Networks - ANN*), o algoritmo *Support Vector Regression (SVR)*, o algoritmo *Random Forest* e os algoritmos de *Grandient Boosting*, com foco no *XGBoost*.

As Redes Neurais Artificiais (*Artificial Neural Networks - ANN*) são modelos computacionais que imitam a forma como as redes neurais biológicas processam informações (GROSAN; ABRAHAM, 2011). Uma *ANN* é normalmente composta por um grande número de nós de processamento interconectados, também chamados de neurônios, que trabalham em conjunto para resolver problemas específicos (GROSAN; ABRAHAM, 2011). As redes neurais são particularmente hábeis na identificação de padrões e tendências em dados, tornando-as úteis para a tarefa de previsão (ISAAC ABIODUN et al., 2018). No entanto, as redes neurais apresentam limitações ao lidar com bases de dados não confiáveis (CAI et al., 2000; CAO et al., 2018). Dados incompletos, incorretos ou com ruído podem impactar significativamente o desempenho de uma *ANN*, pois a rede poderá aprender padrões incorretos (ENNETT; FRIZE; WALKER, 2001).

Support Vector Regression (SVR) é um tipo de algoritmo de aprendizado de máquina baseado nos princípios das Máquinas de Vetores de Suporte (Support Vector Machines - SVM) (ROY; MANNA; CHAKRABORTY, 2019). É utilizado para problemas de regressão (ZHANG; O'DONNELL, 2019), o que significa que foi projetado para prever resultados contínuos. O SVR funciona encontrando uma função que aproxima a relação entre os recursos de entrada e a variável de saída (ZHANG; O'DONNELL, 2019). Essa função é escolhida para ter no máximo um desvio predefinido dos valores reais de saída para a maioria dos dados de treinamento, que é controlado por um parâmetro conhecido como épsilon (ROY; MANNA; CHAKRABORTY, 2019). Além disso, o SVR tenta minimizar a complexidade do modelo, mantendo a função o mais plana possível, controlada por outro parâmetro chamado parâmetro de regularização (ROY; MANNA; CHAKRABORTY, 2019). No entanto, os modelos SVR são sensíveis à qualidade dos dados de entrada (SABZEKAR; HASHEMINEJAD, 2021), o que os torna particularmente inadequados para lidar com bases não confiáveis. Esse cenário pode fazer o modelo capturar ruído como se fosse um sinal verdadeiro, levando a um fraco desempenho de generalização em dados novos e invisíveis e dificultando o processo de ajuste dos seus parâmetros, o que causa um aumento no risco de *overfitting* (LIU; ZIO, 2016).

O algoritmo *Random Forest Regression* é uma técnica de aprendizado de máquina versátil e poderosa que ganhou ampla popularidade devido à sua capacidade de lidar com uma variedade de tarefas de regressão com alta precisão (SCORNET; BIAU; VERT, 2015; ZHANG; NETTLETON; ZHU, 2019). Para formar uma previsão mais precisa e robusta, ele opera construindo diversas árvores de decisão durante a fase de treinamento e gerando a previsão média das árvores individuais (SCORNET; BIAU; VERT, 2015). Ao lidar com espaços de recursos de alta dimensão e estruturas de dados complexas, o algoritmo se mostra particularmente eficaz (BIAU; SCORNET, 2015; SCORNET; BIAU; VERT, 2015). É um método de aprendizagem conjunto que beneficia da agregação de previsões de diversas árvores de decisão aleatórias (*Bagging*), o que ajuda a reduzir o *overfitting* e a melhorar a generalização do modo (BIAU; SCORNET, 2015; SCHONLAU; ZOU, 2020; SCORNET; BIAU; VERT, 2015). O algoritmo não é apenas reconhecido pela sua precisão, mas também pela sua capacidade de lidar com amostras pequenas e pela sua simplicidade em termos de ajuste de parâmetros (BIAU; SCORNET, 2015; SCORNET; BIAU; VERT, 2015). Além disso, o algoritmo *Random Forest* foi estendido para enfrentar desafios específicos, como riscos concorrentes na análise de sobrevivência, onde pode ser adaptado para trabalhar com pseudovalores na presença de dados censurados (MOGENSEN; GERDS, 2013). Também foi generalizado para se ajustar a qualquer quantidade de interesse identificada como solução para um conjunto de equações de momentos locais, oferecendo uma abordagem flexível para uma ampla gama de tarefas

estatísticas (ATHEY; TIBSHIRANI; WAGER, 2019). A capacidade do algoritmo *Random Forest* de incorporar relações conhecidas entre a resposta e os preditores e de fornecer previsões confiáveis mesmo em problemas de extrapolação onde as previsões são necessárias fora do domínio do conjunto de dados de treinamento demonstra ainda mais sua versatilidade e robustez (ZHANG; NETTLETON; ZHU, 2019).

Gradient Boosting Regression é um conjunto de poderosos algoritmos de aprendizado de máquina (NATEKIN; KNOLL, 2013). Esses algoritmos constroem um conjunto de modelos de previsão fracos, geralmente árvores de decisão, para produzir um modelo preditivo mais preciso e robusto (BENTÉJAC; CSÖRGŐ; MARTÍNEZ-MUÑOZ, 2021). Esse algoritmo funciona adicionando iterativamente modelos ao conjunto, o que permite que cada novo modelo seja treinado para corrigir os erros cometidos pelos seus predecessores (BENTÉJAC; CSÖRGŐ; MARTÍNEZ-MUÑOZ, 2021). Este processo envolve a otimização de uma função de perda a partir do cálculo de gradientes (BENTÉJAC; CSÖRGŐ; MARTÍNEZ-MUÑOZ, 2021), o que explica a nomenclatura do algoritmo. A eficácia dessa técnica de aprendizado de máquina é justificada pela sua capacidade de combinar os pontos fortes de vários modelos “aprendizes” (NATEKIN; KNOLL, 2013) e de poder lidar com recursos heterogêneos, dados ruidosos e dependências complexas (JARDIM et al., 2024). Ela também é flexível em termos das funções de perda que pode otimizar (NATEKIN; KNOLL, 2013), tornando-a aplicável a uma ampla gama de problemas de regressão. A família dos algoritmos de *Gradient Boosting* foi expandida com diversas variantes notáveis, como *XGBoost*, *LightGBM* e *CatBoost* (BENTÉJAC; CSÖRGŐ; MARTÍNEZ-MUÑOZ, 2021).

XGBoost, que significa *eXtreme Gradient Boosting*, é uma implementação avançada de algoritmos de aumento de gradiente. Esse algoritmo ganhou popularidade devido à sua eficiência e eficácia em diversas tarefas de aprendizado de máquina, como a previsão. Um novo algoritmo de reconhecimento de dispersão é utilizado nesta técnica de aprendizado de máquina para lidar com dados esparsos, bem como um esboço de quantil ponderado para aprendizado aproximado de árvore. Quando combinados com otimizações para padrões de acesso ao cache, compactação de dados e fragmentação, esses recursos permitem que o *XGBoost* seja dimensionado para lidar com conjuntos de dados consideráveis com eficiência (CHEN; GUESTRIN, 2016).

3.3. Modelos de *Machine Learning* Selecionados

Uma das principais questões do problema tratado por este estudo é a confiabilidade dos dados disponíveis. A base de dados a ser trabalhada possui uma infinidade de problemas em relação à qualidade dos seus dados. Dessa forma, mostra-se necessário realizar uma seleção cautelosa de quais técnicas de *machine learning* são mais adequadas para serem utilizadas nesse cenário.

Anteriormente, vimos que as Redes Neurais Artificiais (*ANNs*) e a *Support Vector Regression* (*SVR*) são métodos de *machine learning* extremamente sensíveis à qualidade dos dados que serão utilizados. Esses algoritmos demonstraram uma incapacidade em realizar previsões precisas quando os dados disponíveis possuem inconsistências. Portanto, esses algoritmos não são particularmente adequados para atender às necessidades do nosso problema.

Por outro lado, as técnicas *Random Forest* e de *Grandient Boosting*, com destaque para o *XGBoost*, mostraram-se particularmente apropriados para lidar com dados dos tipos mais diversos, incluindo dados de bases não confiáveis. Esses algoritmos destacaram-se pela sua robustez e versatilidade, sendo hábeis ao processar dados altamente dimensionais e complexos e sendo adaptáveis a mudanças nas condições dos dados.

3.4. Métricas de Avaliação

As métricas de avaliação são componentes essenciais dos modelos preditivos baseados em aprendizado de máquina. Elas podem ser definidas como construções lógicas e matemáticas designadas à predição do quão perto o resultado previsto está do resultado real (BOTCHKAREV, 2019). Algumas métricas podem acabar apresentando inconsistências em sua performance, o que ressalta a importância da escolha das métricas corretas para a obtenção de previsões precisas em *machine learning* e modelos de regressão de uma forma geral (PLEVRIS et al., 2022).

Nesse contexto, as métricas de regressão destacam-se como as mais apropriadas para a avaliação de um modelo de previsão baseado em aprendizado de máquina para resolver o problema visto no presente estudo. Dentre as métricas de regressão estão o Erro Absoluto Médio (MAE - *Mean Absolute Error*), o Erro Quadrático Médio (MSE - *Mean Squared Error*), a Raiz do Erro Quadrático Médio (RMSE - *Root Mean Squared Error*) e o R^2 (Coeficiente de Determinação).

O erro médio absoluto (MAE - *Mean Absolute Error*) é definido como a média das diferenças absolutas entre os valores previstos pelo modelo e os valores reais (BOTCHKAREV, 2019; SAXENA et al., 2008). Também é conhecido como precisão dependente da escala, pois calcula erros em observações feitas na mesma escala (BOTCHKAREV, 2019).

O erro quadrático médio (MSE - *Mean Squared Error*) é definido como a expectativa do desvio quadrático dos valores previstos em relação aos valores reais (BOTCHKAREV, 2019; PLEVRIS et al., 2022; SAXENA et al., 2008). Seu resultado é sempre não negativo e valores próximos de zero são melhores (PLEVRIS et al., 2022).

A raiz do erro quadrático médio (RMSE - *Root Mean Squared Error*) é calculado como a raiz quadrada da média das diferenças quadradas entre os valores previstos e observados (BOTCHKAREV, 2019; SAXENA et al., 2008). É uma boa medida de precisão, mas apenas para comparar erros de previsão para uma variável específica, pois, assim como o MAE, depende da escala (BOTCHKAREV, 2019; PLEVRIS et al., 2022). Além disso, o RMSE tem sido utilizado como uma métrica estatística padrão para medir o desempenho de modelos em estudos de meteorologia, qualidade do ar e pesquisa climática (CHAI; DRAXLER, 2014).

O R^2, também denominado como Coeficiente de Determinação, é definido como o quadrado do coeficiente de correlação entre os valores observados e previstos da variável dependente (ALEXANDER; TROPSHA; WINKLER, 2015). Ele indica a proporção de pontos de dados que estão dentro da linha criada pela equação de regressão, tal que um valor mais alto de R^2 é desejável, pois indica melhores resultados. Embora o R^2 seja uma medida útil, muitas vezes é recomendado relatar métricas adicionais, como o RMSE ou medidas equivalentes de dispersão, que normalmente são de maior importância prática (ALEXANDER; TROPSHA; WINKLER, 2015).

3.5. Estado da arte para previsão da corrosão

O uso de sistemas de aprendizado e máquina para a previsão do comportamento da corrosão é um objeto de estudo em ascensão nos últimos anos. Dessa forma, foi possível encontrar diversas propostas de abordagens sobre esse assunto.

Um modelo de Redes Neurais Artificiais (*Artificial Neural Network* - ANN) foi desenvolvido para prever a taxa de corrosão atmosférica do aço carbônico, utilizando fatores meteorológicos e químicos como variáveis de entrada. Essa abordagem possui um alto coeficiente de determinação e uma pequena raiz de erro quadrático médio, tornando-o uma ferramenta prática para previsão (TRAN et al., 2021).

Os algoritmos *Random Forest*, árvores de decisão (*Decision Trees*) e *Support Vector Machine* (SVM) têm sido utilizados para classificação na previsão do comportamento da corrosão de aços inoxidáveis em ambientes à base de ácido láctico. Esses algoritmos alcançaram alta precisão de treinamento e teste, e mostraram-se confiáveis para prever a degradação por corrosão nas condições citadas (POURRAHIMI et al., 2023).

Foi realizado o desenvolvimento de um modelo de redes neurais para prever a corrosão do CO2 em altas pressões parciais. O modelo construído mostrou-se confiável a partir dos resultados dos testes, também sendo validado por meio do método de validação cruzada *Leave-One-Out* (LOOCV) (ABBAS; NORMAN; CHARLES, 2018).

Um modelo de *Random Forest* foi desenvolvido para prever a taxa anual de corrosão em plataformas offshore FPSO na indústria de petróleo e gás. São utilizados dados climáticos e outros dados relevantes para prever tendências de corrosão com base em variáveis selecionadas (ELMAS et al., 2023).

A partir da análise das tecnologias citadas, o modelo de *Random Forest* descrito em (ELMAS et al., 2023) obteve um maior destaque em relação aos outros trabalhos referidos, pois lida exatamente com o acompanhamento do avanço da corrosão em uma instalação industrial como um todo, de maneira que os demais modelos possuem aplicações voltadas para materiais específicos. Dessa forma, o modelo selecionado será aproveitado durante o processo de desenvolvimento do projeto, tal que ele será o ponto de partida da implementação a ser construída.

4. Geração da Base de Dados

4.1. Seleção dos dados

A primeira etapa no processo da geração da base de dados consiste na seleção de quais dados serão considerados nesse estudo. O modelo foi previamente treinado a partir de dados de inspeção de plataformas de petróleo de uma empresa de óleo e gás, localizadas em uma unidade de negócio (UN) específica. Neste caso, optou-se por inserir os dados das plataformas mais novas da unidade A (P1, P2, P3, P4), a partir de 2021, e das plataformas da unidade B (P5, P6, P7 e P8). Dessa forma, o dado de entrada utilizado no modelo é equivalente ao avanço real da corrosão em cada componente das plataformas, calculado a partir da subtração entre o percentual de corrosão inspecionado de um ano com o percentual do ano anterior. Vale ressaltar que a quantidade de dados disponível não é uniforme para todas as plataformas, a Tabela 2 apresenta os anos considerados para cada uma das plataformas.

Plataforma	**UN**	**Anos**	**Frequência**
P1	A	2021 até 2023	Anual
P2	A	2022 até 2024	Anual
P3	A	2021 até 2023-2	Anual até 2022; Semestral em 2023
P4	A	2021 até 2024	Anual
P5	B	2020 até 2022	Anual
P6	B	2019 até 2023	Bienal
P7	B	2021 até 2023	Bienal
P8	B	2021 até 2023	Bienal

Tabela 2 - Disponibilidade de dados por plataforma.

4.2. Tratamento dos dados

A segunda etapa consiste no levantamento e no tratamento dos dados selecionados. Os dados das plataformas da unidade A precisaram de pouca atenção nessa fase, pois já foram recebidos tratados. Por outro lado, foram necessários inúmeros ajustes nos dados da unidade B para colocá-los no padrão aceito pelo modelo de aprendizado de máquina abordado no presente estudo. Dentre esses ajustes estão:

- Transformação de Dados:
    - Define funções para transformar dados específicos da planilha:
        - *get_sheet_area*: Converte valores de área de *string* para *float*.

        - *get_caracteristica*: Ajusta os valores da coluna "Característica" para o padrão L, M e G.
        - *get_funcao*: Ajusta os valores da coluna "funções" para corrigir erros de preenchimento da planilha.
        - Aplica essas funções para transformar os dados da planilha.
    - Correção de Setores faltantes e IDs Quebrados da P7:
    - Define funções para corrigir setores faltantes e IDs quebrados da plataforma P7:
        - *fix_p7_missing_sectors*: Preenche setores faltantes da P7 com base em dados da P5.
        - *fix_p7_broken_ids*: Ajusta IDs quebrados da P7.
        - Aplica essas funções para corrigir os dados da P7.
- Remoção de Linhas Duplicadas:
    - Remove linhas duplicadas da planilha e redefine os índices.
- Importação de Planilha para Ajuste de Setores:
    - Lê uma planilha adicional que será usada para ajustar setores dos sistemas duplicados.
- Correção de IDs:
    - Define a função *get_correct_ids* para corrigir problemas nos IDs da planilha e ajustá-los para o padrão utilizado pelo modelo.
    - Aplica essa função para ajustar os IDs dos sistemas na planilha.
- Criação de Dicionário de IDs Duplicados:
    - Uma mudança interna na nomenclatura dos setores das plataformas da unidade B gerou IDs duplicados para sistemas diferentes nos dados fornecidos.
    - Cria um dicionário (*duplicated_ids_dict*) que guarda valores agregados de área, corrosão, elevação, exposição e característica dos IDs duplicados por plataforma e ano.
- Adição de Dados Calculados para IDs Duplicados:

- Define funções para adicionar dados calculados para IDs duplicados:
    - *get_corrosion_for_duplicated*: Adiciona a corrosão calculada.
    - *get_area_for_duplicated*: Adiciona a área calculada.
    - *get_elevation_for_duplicated*: Adiciona a elevação calculada.
    - *get_exposition_for_duplicated*: Adiciona a exposição calculada.
    - *get_characteristic_for_duplicated*: Adiciona a característica calculada.
    - Aplica essas funções para adicionar os dados calculados à planilha original.

- Manipulação de Dados de TVF:
    - Nos dados fornecidos, a informação de TVF veio separada em Tubulação e Flange. Dessa forma, foi necessário fazer a média ponderada desses dois dados a partir suas respectivas áreas.
    - Cria um *dataframe* (*tvf_df*) para manipular dados de TVF a partir dos dados de Tubulação e Área.
    - Define funções para calcular e inserir dados de TVF:
        - *get_tvf_corrosion*: Calcula a corrosão do TVF.
        - *get_tvf_area*: Calcula a área do TVF.
        - *insert_tvf_area*: Insere a área calculada do TVF na planilha original.
        - *insert_tvf_corrosion*: Insere a corrosão calculada do TVF na planilha original.
        - *insert_tvf_id*: Substitui o ID de 'Tubulação' por TVF.
        - *insert_tvf_system*: Substitui o dado da coluna 'Sistema' referente à 'Tubulação' por TVF.
        - Aplica essas funções para inserir os dados de TVF calculados na planilha original.

- Transformação Final dos Dados:
    - Transforma a coluna de corrosão em percentagem.
    - Renomeia as colunas selecionadas do *dataframe* para o padrão de input do modelo.
- Salvamento das Planilhas por Plataforma e Ano
    - Salva as planilhas com os dados de inspeção da unidade B por plataforma e ano em arquivos Excel separados.

Após os ajustes nos dados selecionados, foi feito o levantamento dos dados utilizados no treinamento anterior do modelo. Dessa forma, foi feita a remoção dos dados de 2020 da unidade A, pois houve uma mudança no arranjo dessas plataformas entres os anos de 2019 e 2020, o que, consequentemente, causou uma inconsistência nesses dados e trouxe a necessidade de removê-los da análise. Em seguida, foi feita a junção dos dados de treinamento antigos com os dados novos em um *dataset* unificado.

4.3. Adição de novas variáveis a serem inseridas no modelo

A terceira etapa da preparação da base de dados se refere à adição de novo fatores que podem afetar a corrosão. Após a avaliação, foram selecionados os fatores de plataforma, área e característica para serem levados em consideração pelo modelo.

A variável plataforma tem por objetivo incorporar especificidades de cada unidade de negócio, tanto referente à sua construção, como posicionamento e forma de gestão da corrosão.

Como o presente estudo trata da corrosão externa, há a possibilidade do percentual de corroído ocorrer em função do quanto a corrosão avança externamente. Portanto, a variável referente à área exposta foi escolhida para lidar com esse fator.

A variável de característica refere-se ao tipo de corrosão, relacionando-se com a distribuição da corrosão na superfície. Ela classifica a corrosão como localizada, moderada ou generalizada, conforme a classificação da norma ASTM D 610-1.

Após realizar a seleção dessas variáveis, elas foram incluídas ao *dataset* de treinamento. Dessa forma, elas juntaram-se às variáveis de idade, percentual corroído, sistema, temperatura, umidade, exposição, elevação, localização e função, as quais já estavam sendo consideradas anteriormente pelo modelo.

4.4. Limpeza da base de dados e tratamento de dados faltantes

A quarta etapa para a criação da base de dados foi a limpeza dos dados. Essa etapa se faz necessária pois em diversos casos foram identificados avanços de corrosão negativos ou zerados. Esse fenômeno não é possível de acontecer na prática, uma vez que a corrosão não regride de forma natural. Isso ocorre majoritariamente por dois motivos: (i) não é realizado um controle das áreas que foram pintadas no ano, de forma que o avanço negativo pode significar que determinado sistema foi pintado ao longo do ano; ou (ii) devido à subjetividade do processo, em que diferentes inspetores podem ter avaliações divergentes sobre um percentual de corrosão de um mesmo item. Dessa forma, optou-se por eliminar os dados negativos da base de dados. Observou-se que a base também apresentou instâncias com avanço nulo, o que pode ocorrer em casos que a pintura se apresente intacta. Logo, como o total de resultados nulos é muito representativo, optou-se por eliminar esses casos.

A quinta etapa consiste em tratar dados faltantes. A opção mais simples consiste em eliminar as instâncias com dados faltantes, o que levaria a uma perda considerável de dados. Outra opção seria tratar os dados faltantes como uma nova categoria, ou zerar os dados numéricos, porém isso pode levar o modelo a falsas interpretações, assumindo que a falta do valor é representativa para o problema. De forma a evitar que dados sejam perdidos por falta de informação de algum fator, optou-se por utilizar uma prática comum de preencher os dados faltantes com valores de média, para os atributos numéricos ou o valor mais frequente, no caso de variáveis categóricas (WITTEN; FRANK; HALL, 2005).

4.5. Pré-processamento da base de dados

Uma vez concluído o levantamento dos dados, a etapa seguinte é referente ao pré-processamento da base. Para a construção de uma base de dados a ser utilizada nos métodos de aprendizado de máquina, na maioria das vezes, os dados de entrada são variáveis categóricas, ou seja, descritas em formas de textos. O que ocorre, é que um texto não tem significado para o modelo, sendo necessário para tal, atribuir valores de importância aos textos para facilitar seu entendimento. Dessa forma, os métodos encoding como o *ordinal encoding*, que substitui variáveis categóricas por números inteiros ou *one hot encoding*, que transforma as variáveis categóricas em colunas binárias, fazem essa modificação. A decisão por cada um dos métodos não é direta. Ao transformar variáveis categóricas em números inteiros, o método de ordinal encoding naturalmente atribui uma ordem de importância às categorias que não necessariamente seguem uma ordem lógica. Por outro lado, o método *one hot encoder* pode gerar muitas colunas, especialmente em casos em que há grande variabilidade de categorias. O número excessivo de colunas pode representar um problema especialmente em modelos de floresta, como o *Random Forest* usado nesse estudo, pois a árvore se torna muito extensa e com muitas ramificações. Em bases de dados pequenas podem sobrar poucas instâncias em cada uma das folhas finais, levando ao *overfitting* (WITTEN; FRANK; HALL, 2005). Como as variáveis de localização, função e característica apresentam muitas categorias, foram feitos testes considerando as duas formas de *encoder* a fim de avaliar os benefícios de cada uma delas.

Os atributos considerados no problema são medidos em escalas diferentes, e o efeito de alguns atributos podem ser ofuscados por outros de maior escala, como por exemplo o atributo elevação varia de zero a 61.900, enquanto a variável de percentual varia de 0 a 0.7. Além disso, algumas variáveis podem apresentar pouca variabilidade, como é o caso da temperatura. Dessa forma, uma prática recomendada é a normalização dos atributos numéricos, garantindo que todos apresentem uma variação de 0 a 1 (WITTEN; FRANK; HALL, 2005). Para realizar essa normalização, foi escolhido o método *Min-max.* Portanto, foram feitos testes considerando a normalização dos dados ou não. A Tabela 3, apresenta resumidamente os tipos de pré-processamento realizados em cada uma das variáveis nos testes realizados.

A última etapa consiste na divisão da base de dados em treino e validação. Para isto, foi considerada a divisão de 80% para treino e 20% para teste, feita de forma aleatória. A fim de evitar variabilidade na divisão da base de dados para os diversos testes, foi considerado um valor de semente fixo

Teste	A < 2021	A >= 2021	B	Antigas	Plataforma	Idade	Percentual	Sistema	Temperatura	Umidade	Exposição	Elevação	Localização	Função	Área	Característica
1	Sim	Não	Não	Sim	Não	Sim	Sim	OHE	Sim	Sim	Sim	Sim	OE	OE	Não	Não
2	Sim	Sim	Sim	Sim	Não	Sim	Sim	OHE	Sim	Sim	Sim	Sim	OE	OE	Não	Não
3	Não	Sim	Sim	Não	Não	Sim	Sim	OHE	Sim	Sim	Sim	Sim	OE	OE	Não	Não
4	Sim	Sim	Sim	Sim	Não	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Não	Não
5	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Não	Não
6	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Min-max	Não
7	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OHE	OE	Min-max	Não
8	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OHE	Min-max	Não
9	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OHE	OHE	Min-max	Não
10	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Não	OHE
11	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Não	OE
12	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Min-max	OHE
13	Sim	Sim	Sim	Sim	OHE	Min-max	Min-max	OHE	Min-max	Min-max	Min-max	Min-max	OE	OE	Min-max	OE

Tabela 3 –Tabela relativa ao plano de testes de pré-processamento

5. Treinamento do algoritmo

O trabalho foi realizado em computadores com sistema operacional Windows, com diferentes configurações de hardware. As máquinas possuíam 16GB de memória RAM, SSDs de diferentes especificações e processadores AMD Ryzen 7 2700X, Intel i5 8400 e Intel i5 4690K, respectivamente. Os códigos-fonte do tratamento de dados, aferição e treinamento do modelo foram desenvolvidos usando a linguagem Python, na versão 3.12.7, com auxílio da plataforma Jupyter Notebook. O ambiente virtual criado para o desenvolvimento dos códigos utilizou as bibliotecas pandas, *scikit-learn*, *joblib*, *scipy*, *category-encoders*, *numpy*, *parse*, *openpyxl*, *xlsxwriter* e *ipykernel*.

Uma vez definida a base de dados, iniciou-se a etapa de treinamento do algoritmo de Random Forest. Como foi apresentado anteriormente, o dado de entrada do modelo é o avanço real da corrosão em cada componente das plataformas, tal que o dado de saída é o avanço previsto pelo modelo. Os parâmetros iniciais são apresentados na Tabela 4, onde *n_estimators* é referente ao número de árvores consideradas para a floresta, *max_depth* define a profundidade máxima da árvore em termos de camadas de nós de decisão, *max_features* é referente ao número de atributos considerados para definir o melhor parâmetro para a separação dos ramos, *criterion* é função utilizada para definir a qualidade de cada divisão, *min_sample_split* é o número mínimo de instâncias necessárias para dividir um nó interno à árvore, *min_impurity_decrease* refere-se ao grau de impureza para estabelecer se um nó deve ser dividido ou não e *bootstrap* define a metodologia de separação das amostras que serão utilizadas em cada árvore. Além disso, foi considerado um método para otimizar a definição dos parâmetros de entrada. Esse método é denominado *Grid Search CV*, em que avalia exaustivamente parâmetros em um determinado *grid*, e são avaliados pelo método de *cross-validation* com 5 *folds* (FABIAN PEDREGOSA et al., 2011). O *grid* utilizado nesse processo é apresentado na Tabela 4.

Também foi avaliado um segundo método para definir os parâmetros do modelo, denominado *Randomized Search CV*. Diferente do *Grid Search CV*, os parâmetros a serem avaliados são selecionados dentro do grid de forma aleatória, e o número de iterações realizadas é definido pelo usuário (FABIAN PEDREGOSA et al., 2011). No entanto, devido à sua aleatoriedade e à pequenas discrepâncias nos seus resultados em diferentes iterações, o uso desse método foi descartado.

Como o método *Grid Search CV* funciona através de uma avaliação exaustiva dos parâmetros do modelo, seu funcionamento é extremamente custoso em recursos computacionais, o que possui um impacto em sua performance. Dessa forma, foi possível rodá-lo apenas com os dados de pré-processamento do Teste 3 (Tabela 3), pois sua execução demandava entre 6 e 10 horas, aproximadamente, nas máquinas

disponíveis para o experimento. O *grid* utilizado na execução do *Grid Search CV* está detalhado na Tabela 4, tal que os melhores parâmetros selecionados pelo método são apresentados nas tabelas Tabela 5.

Parâmetro	**Valor**
n_estimators	[200, 400, 600, 800, 1000]
max_depth	[5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55]
max_features	["auto", "sqrt", "log2"]
criterion	["squared_error"]
min_samples_split	[2, 3, 4, 5, 6, 7, 8, 9, 10]
min_impurity_decrease	[0.0, 0.05, 0.1]
bootstrap	[True, False]

Tabela 4 - *Grid* para a seleção de parâmetros pelo método *Grid Search CV*.

Parâmetro	**Valor**
n_estimators	250
max_depth	25
max_features	"log2"
criterion	"squared_error"
min_samples_split	5
min_impurity_decrease	0.0
bootstrap	True

Tabela 5 - *Grid* para a seleção de parâmetros pelo método *Grid Search CV*.

Após a obtenção dos melhores parâmetros através do método *Grid Search CV*, foi realizada a rodada de treinamentos do modelo *Random Forest* utilizando-os, conforme o plano de testes detalhado na Tabela 3.

6. Resultados

6.1. Resultados da rodada de treinamentos

Os resultados obtidos para cada um dos testes descritos são apresentados na Tabela 6 na forma do R^2 calculado em cima da base de testes considerando toda a base com todas as plataformas das unidades A e B, e individualmente para cada uma das plataformas. Observa-se que o teste 3, que utilizava apenas os novos dados, apresentou o melhor resultado geral, de maneira que o teste 4 apresentou o melhor resultado dentre os testes que utilizavam a base de dados completa.

Teste	RMSE treino	RMSE validação	R^2 treino	R^2 validação
1	0.029444	0.038510	0.751226	0.550917
2	0.020243	0.031609	0.787957	0.515934
3	0.011805	0.019905	0.831865	0.574192
4	0.020243	0.031601	0.787976	0.516182
5	0.020910	0.031726	0.773753	0.512349
6	0.020750	0.031715	0.777202	0.512693
7	0.021054	0.031765	0.770628	0.511150
8	0.022241	0.032658	0.744044	0.483276
9	0.022001	0.032401	0.749549	0.491397
10	0.020823	0.031787	0.775640	0.510484
11	0.020725	0.031746	0.777742	0.511753
12	0.020867	0.031680	0.774703	0.513782
13	0.020713	0.031697	0.778002	0.513256

Tabela 6 - Resultados do modelo de *Random Forest* para os testes planejados.

6.2. Análise dos resultados do avanço

As Figuras 1 e 2 apresentam, respectivamente, as distribuições dos resultados preditos para os testes 3 e 4. Em um modelo perfeito todos os pontos deveriam estar sobrepostos à reta de 45°, em laranja. Como o teste 4 possui todos os dados agregados, é possível observar, na Figura 2, que os resultados do avanço são mais distribuídos. Isso acontece pelo fato dos dados antigos, isto é, da unidade A antes de 2021 e das plataformas antigas, serem mais contínuos. Dessa forma, o gráfico dos resultados do teste 3, representado na Figura 1, nos permite observar uma tendência exatamente oposta nos dados novos. É possível observar intervalos claros nos dados em relação ao eixo x do terceiro teste, o que revela a natureza discreta dos novos dados incluídos no modelo.

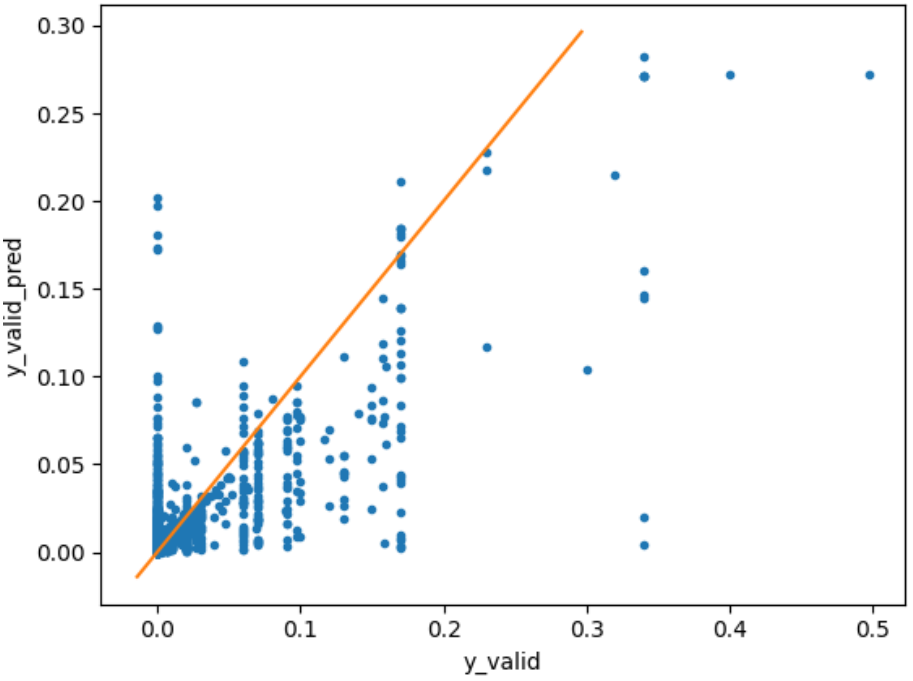

Figura 1 - Comparação entre resultados preditos e reais – teste 3.

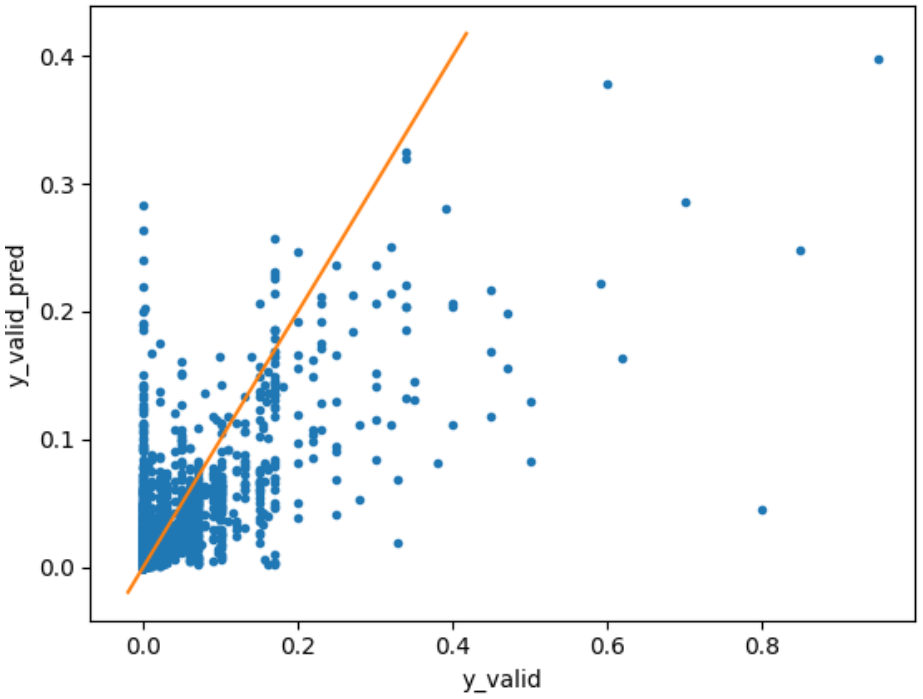

Figura 2 - Comparação entre resultados preditos e reais – teste 4.

6.3. Análise das importâncias dos parâmetros

O modelo de Random Forest é capaz de estimar o grau de importância de cada parâmetro no modelo decisório, em que os modelos com maior peso são contemplados nos primeiros nós da árvore em detrimento dos parâmetros de menor importância, localizados nos nós mais próximos às folhas. Esta análise foi feita para os testes 3, 4 e 12, tal que os valores dos 15 principais parâmetros dos testes 3 e 4 são apresentados, respectivamente, nas Tabelas 7 e 8; e todos os valores dos parâmetros do teste 12 estão representados na Tabela 9. Em paralelo, as Figuras 3, 4 e 5, apresentam os gráficos com os respectivos pesos das importâncias de forma decrescente dos testes descritos anteriormente.

Observa-se que o principal fator é o percentual de corrosão base, e que após o quinto parâmetro os valores de importância são consideravelmente menores que os quatro primeiros. Além disso, é possível observar que a corrosão possui muito mais influência que as demais variáveis no resultado do teste 3, cujo modelo foi treinado apenas com os dados novos, em comparação aos outros testes. Dessa forma, o fato dos dados novos serem mais discretos pode ser uma possível causa dessa tendência nas importâncias dos parâmetros do teste 3.

O teste 12 foi realizado levando em consideração os novos parâmetros inseridos na base; plataforma, área e característica. A partir dos dados exibidos pela Tabela 9 e pela Figura 5, podemos observar que os novos parâmetros não obtiveram um impacto significativo na performance do modelo. A variável nova de maior importância ocupa, apenas, a 12ª posição na tabela de importâncias. Dessa forma, a inserção desses novos parâmetros não se justifica, pois, se as novas variáveis não estão tendo uma importância considerável no resultado do modelo, o melhor a se fazer é não incluí-las, pois elas apenas aumentariam *overfitting* e deixariam o modelo mais pesado.

Variável	Valor
corrosao	0.456064
elevacao	0.098272
funcao	0.093852
temperatura	0.058672
direcao	0.043412
intensidade	0.040287
localizacao	0.037592
umidade	0.037496
idade	0.020290
exposicao	0.018014
sistema_Piso	0.017570
sistema_Suportes	0.015332
sistema_Estruturas	0.014739
sistema_Guarda-corpo	0.013826
sistema_TVF	0.011829

Tabela 7 - Principais parâmetros considerados no modelo – teste 3.

Variável	Valor
corrosao	0.312326
funcao	0.123569
elevacao	0.105310
temperatura	0.066138
intensidade	0.062527
umidade	0.050033
idade	0.049297
direcao	0.045898
localizacao	0.044463
sistema_Piso	0.027845
exposicao	0.022954
sistema_Suportes	0.020668
sistema_Guarda-corpo	0.019089
sistema_Estruturas	0.012282
sistema_TVF	0.012266

Tabela 8 - Principais parâmetros considerados no modelo – teste 4.

Variável	Valor
corrosao	0.267977
funcao	0.124403
elevacao	0.101855
intensidade	0.049460
temperatura	0.048935
localizacao	0.047599
direcao	0.042255
umidade	0.041982
idade	0.041775
sistema_Piso	0.032517
exposicao	0.024850
caracteristica_G	0.022769
sistema_Suportes	0.020788
sistema_Guarda-corpo	0.018501
plataforma_antiga3	0.015316
sistema_TVF	0.012477
sistema_Estruturas	0.012460
sistema_Escadas	0.010541
sistema_Equipamento	0.008401
plataforma_antiga2	0.008158
caracteristica_L	0.006253
plataforma_P1	0.005240
area	0.005175
caracteristica_M	0.004326
plataforma_antiga4	0.004103
plataforma_P1	0.003676
sistema_Antepara	0.003652
sistema_Teto	0.003457
plataforma_P2	0.003146
plataforma_antiga5	0.002870
plataforma_P3	0.002453
plataforma_P5	0.001056
plataforma_P6	0.000786
plataforma_P8	0.000701
plataforma_P7	0.000088

Tabela 9 – Todos os parâmetros considerados no modelo – teste 12.

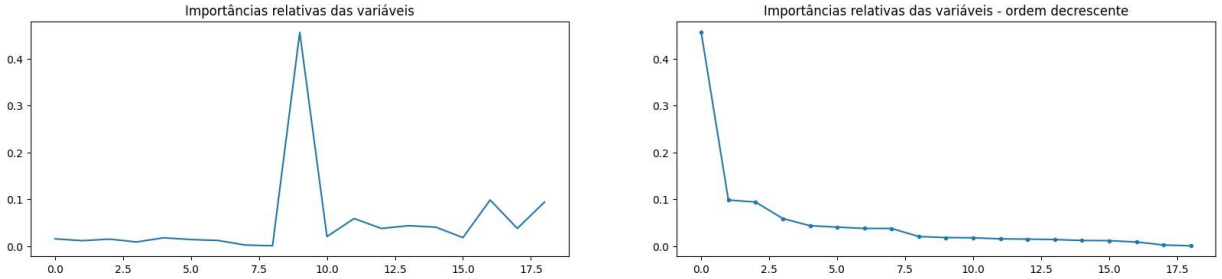

Figura 3 - Avaliação da importância das variáveis – teste 3

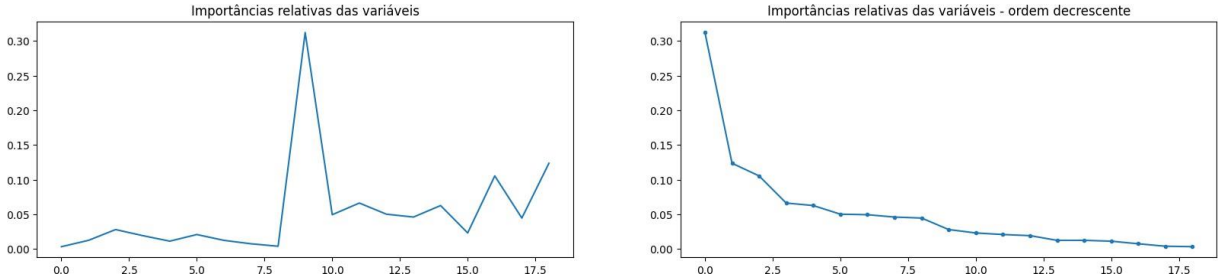

Figura 4 - Avaliação da importância das variáveis – teste 4.

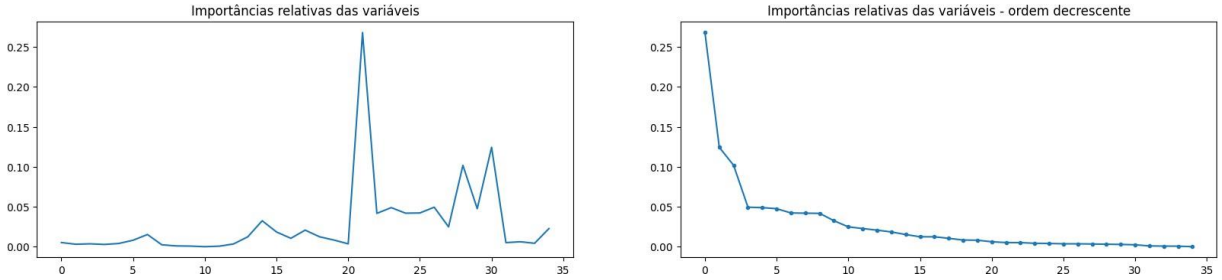

Figura 5 - Avaliação da importância das variáveis – teste 12.

7. Conclusão e Trabalhos Futuros

Após analisar os resultados do trabalho realizado, foi possível observar que houve uma melhoria na performance do modelo em relação ao treinamento anterior a partir das métricas dos resultados dos testes realizados, como apresentado nas Tabelas 1 e 6, respectivamente. Assim, percebe-se que a adição dos novos dados ao modelo teve um impacto consideravelmente maior em sua performance do que a inserção de novas variáveis. Dessa forma, optou-se por não incluir os novos parâmetros, plataforma, área e característica; ao modelo de *Random Forest* examinado neste estudo. O efeito dessas novas variáveis não atendeu às expectativas e sua inserção não gerou benefícios nos resultados do modelo.

Existem inúmeros caminhos interessantes para seguir a partir de onde o presente trabalho parou. Uma importante linha a ser adotada é a realização do treinamento do modelo a partir da divisão da base de dados por UNs e plataformas, de maneira que seria possível dimensionar o impacto que a adição de cada um dos dados à base trouxe ao modelo. Além dos novos parâmetros observados neste trabalho, também seria interessante realizar a inserção da variável relativa ao intervalo entre inspeções ao modelo, tal que seu impacto na performance seria avaliado da mesma forma que o das demais variáveis vistas. Por fim, também seria essencial avaliar o uso de outros modelos de aprendizado de máquina, como o *XGBoost*, e comparar os seus resultados aos do modelo atual de *Random Forest.*

8. Referências

ALEXANDER, D. L. J.; TROPSHA, A.; WINKLER, D. A. Beware of R2: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models. Journal of Chemical Information and Modeling, v. 55, n. 7, p. 1316–1322, 27 jul. 2015.

ATHEY, S.; TIBSHIRANI, J.; WAGER, S. Generalized random forests. Annals of Statistics, v. 47, n. 2, p. 1179–1203, 1 abr. 2019.

BENTÉJAC, C.; CSÖRGŐ, A.; MARTÍNEZ-MUÑOZ, G. A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, v. 54, n. 3, p. 1937–1967, 1 mar. 2021.

BIAU, G.; SCORNET, E. A Random Forest Guided Tour. 18 nov. 2015.

BOTCHKAREV, A. A new typology design of performance metrics to measure errors in machine learning regression algorithms. Interdisciplinary Journal of Information, Knowledge, and Management, v. 14, p. 45–76, 2019.

CAI, K.-Y. et al. On the neural network approach in software reliability modeling. Journal of Systems and Software, v. 58, n. 1, p. 47–62, ago. 2001.

CAO, W. et al. A review on neural networks with random weights. Neurocomputing, v. 275, p. 278–287, 31 jan. 2018.

CHAI, T.; DRAXLER, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. Geoscientific Model Development, v. 7, n. 3, p. 1247–1250, 30 jun. 2014.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery

and Data Mining. Anais...Association for Computing Machinery, 13 ago. 2016.

EL NAQA, I.; MURPHY, M. J. What Is Machine Learning? Em: Machine Learning in Radiation Oncology. Cham: Springer International Publishing, 2015. p. 3–11.

ELMAS, F. R. et al. Prediction of external corrosion rate in Oil and Gas platforms using ensemble learning: a Maintenance 4.0 approach. Brazilian Journal of Operations and Production Management, v. 20, n. 3, 24 ago. 2023.

ENNETT, C. M.; FRIZE, M.; WALKER, C. R. Influence of Missing Values on Artificial Neural Network PerformanceMedinfo. [s.l: s.n.].

GROSAN, C.; ABRAHAM, A. Artificial Neural Networks. Em: [s.l: s.n.]. p. 281–323.

HAMEED, A. et al. A decision support tool for bi-objective risk-based maintenance scheduling of an LNG gas sweetening unit. Journal of Quality in Maintenance Engineering, v. 25, n. 1, p. 65–89, 11 mar. 2019.

ISAAC ABIODUN, O. et al. State-of-the-art in artificial neural network applications: A survey. Heliyon, v. 4, p. e00938, 2018.

JARDIM, S. et al. Comparing Artificial Intelligence Classification Models to Improve an Image Comparison System with User Inputs. SN Computer Science, v. 5, n. 1, 1 jan. 2024.

LIU, J.; ZIO, E. An adaptive online learning approach for Support Vector Regression: Online-SVR-FID. Mechanical Systems and Signal Processing, v. 76–77, p. 796–809, ago. 2016.

MOGENSEN, U. B.; GERDS, T. A. A random forest approach for competing risks based on pseudo-values. Statistics in Medicine, v. 32, n. 18, p. 3102–3114, 15 ago. 2013.

NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. Frontiers in Neurorobotics, v. 7, n. DEC, 2013.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. 2 jan. 2012.

PLEVRIS, V. et al. Investigation of performance metrics in regression analysis and machine learning-based prediction models. 8th European Congress on Computational Methods in Applied Sciences and Engineering. Anais...CIMNE, 2022. Disponível em: <https://www.scipedia.com/public/Plevris_et_al_2022a>

RAJENDRA, P.; GIRISHA, A.; GUNAVARDHANA NAIDU, T. Advancement of machine learning in materials science. Materials Today: Proceedings, v. 62, p. 5503–5507, 2022.

ROY, A.; MANNA, R.; CHAKRABORTY, S. Support vector regression based metamodeling for structural reliability analysis. Probabilistic Engineering Mechanics, v. 55, p. 78–89, 1 jan. 2019.

SABZEKAR, M.; HASHEMINEJAD, S. M. H. Robust regression using support vector regressions. Chaos, Solitons and Fractals, v. 144, 1 mar. 2021.

SAXENA, A. et al. Metrics for Evaluating Performance of Prognostic Techniques. International Conference on Prognostics and Health Management. Anais...out. 2008.

SCHONLAU, M.; ZOU, R. Y. The random forest algorithm for statistical learning. Stata Journal, v. 20, n. 1, p. 3–29, 1 mar. 2020.

SCORNET, E.; BIAU, G.; VERT, J. P. Consistency of random forests. Annals of Statistics, v. 43, n. 4, p. 1716–1741, 1 ago. 2015.

WITTEN; FRANK; EIBE. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. [s.l: s.n.].

ZHANG, F.; O'DONNELL, L. J. Support vector regression. Em: Machine Learning: Methods and Applications to Brain Disorders. [s.l.] Elsevier, 2019. p. 123–140.

ZHANG, H.; NETTLETON, D.; ZHU, Z. Regression-Enhanced Random Forests. 23 abr. 2019.