

DESENVOLVIMENTO DA INTELIGÊNCIA COMPUTACIONAL PARA A DETECÇÃO DE FALHAS EM TURBOGERADORES DE PLATAFORMAS OFFSHORE

NAYANE MOURA NERIS¹, SYMONE GOMES SOARES ALCALÁ², MÁRIO MARCONDES MACHADO³

¹ Mestranda Engenharia Elétrica e da Computação, UFG, Goiânia-GO, nayanemoura.eng@gmail.com;

² Dr.^a Engenharia Elétrica e da Computação, UFG, Goiânia-GO, symone@ufg.br

³ Dr. Engenharia de produção, PETROBRAS, E&P, Rio de Janeiro - RJ, mmarcondes@petrobras.com.br

Apresentado no
Congresso Técnico Científico da Engenharia e da Agronomia – CONTECC
4 a 6 de outubro de 2022

RESUMO: A redução de custos se tornou uma questão essencial para empresas se manterem competitivas no mercado, principalmente para aquelas com uma logística complexa, como é o caso da área do Pré-Sal no Brasil. Ser capaz de otimizar processos e detectar falhas se tornou indispensável, desta forma a aplicação de algoritmos de aprendizado de máquina tem cada vez mais ganhado espaço. Este trabalho apresenta uma análise comparativa entre classificadores e propõe uma arquitetura de detecção de falhas em programas de manutenção preventiva, por meio de técnicas de aprendizado de máquina, no ambiente operacional *offshore* de uma indústria de petróleo e gás. Inicialmente realizou-se uma mineração no banco de dados com o objetivo de gerar características (*features*) adequadas para os classificadores. Com o banco de dados construído, foram treinados seis classificadores e realizada uma análise comparativa com outros trabalhos. Foi possível identificar a necessidade de aumentar o banco de dados, principalmente os dados de falhas, para verificar se o desempenho dos classificadores pode ser melhorado.

PALAVRAS-CHAVE: manutenção preventiva, aprendizado de máquina, turbogerador.

DEVELOPMENT OF COMPUTATIONAL INTELLIGENCE FOR THE FAILURE DETECTION IN TURBOGENERATORS OF OFFSHORE PLATFORMS

ABSTRACT: Cost reduction has become an essential issue for companies to remain competitive in the market, especially for those with complex logistics, such as the Pre-Salt area in Brazil. Being able to optimize processes and detect failures has become indispensable, so the application of machine learning algorithms has increasingly gained space. This paper presents a comparative analysis between classifiers and proposes a fault detection architecture in preventive maintenance programs, through machine learning techniques, in the offshore operating environment of an oil and gas industry. Initially, a mining was carried out in the database with the objective of generating characteristics appropriate for the classifiers. With the database built, six classifiers were trained and a comparative analysis was performed with other studies. It was possible to identify the need to increase the database, especially the failure data, to verify that the performance of the classifiers can be improved.

KEYWORDS: preventive maintenance, machine learning, turbogenerator.

INTRODUÇÃO

A necessidade de evitar prejuízos econômicos e operacionais se tornou uma questão essencial para as aplicações de engenharia, principalmente quando a redução de custos de logística é obrigatória, como é o caso da área do Pré-Sal no Brasil. Para um sistema ter um bom desempenho, ele tem de ser capaz de detectar e diagnosticar falhas em tempo real, bem como, corrigi-las ou determinar o processo de correção para cada uma delas.

Segundo Santos *et al.* (2015), fazem-se necessárias avaliações de criticidade; coleta e análise de dados de campo; identificação, isolamento e modelagem de mecanismos de degradação; diagnóstico e prognóstico. Essas atividades exigem o envolvimento de diferentes grupos e habilidades

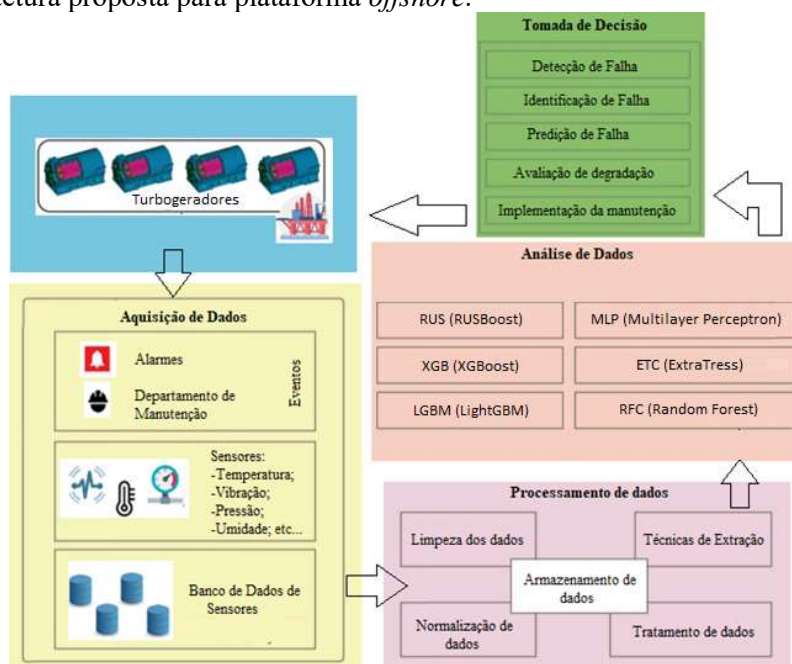
de maneira colaborativa para alcançar os benefícios de uma mentalidade e decisão proativas em toda a organização. A manutenção preditiva tornou-se um termo amplamente utilizado na área industrial e na pesquisa acadêmica e está sendo desenvolvido constantemente por engenheiros e pesquisadores com base no monitoramento de dados históricos, modelagem, simulação e probabilidades de falha para prever falhas e deterioração do sistema sobre sua vida útil (RAN *et al.*, 2019).

Devido a abrangência do processo de diagnóstico de falhas, este trabalho busca propor uma arquitetura que detecte falhas em plataformas *offshore* com soluções em tempo real. Para tal, foi realizada uma pesquisa bibliográfica em diversos temas que irão compor o presente projeto, previsão de falhas, manutenção baseada em condições e métodos de detecção de falhas. Além disso, apresenta uma análise comparativa dos classificadores com os trabalhos de Santos *et al.* (2015).

MATERIAIS E MÉTODOS

O monitoramento de condições para instalações *offshore* é certamente um desafio, especialmente quando se trata de qualidade e análise de dados. Pensando nisso, este trabalho desenvolveu uma metodologia baseada na arquitetura proposta da Figura 1. Ela é composta por uma camada de aquisição de dados, uma camada de processamento de dados e uma camada de tomada de decisão de manutenção.

Figura 1 - Arquitetura proposta para plataforma *offshore*.



Fonte: Elaborado pelos autores.

Para dar início a metodologia foi necessário identificar a base de dados por meio dos registros de todos os eventos relacionados às falhas e paradas normais, período de duração do evento, período de manutenção, detalhamento da falha e em qual turbogerador ela aconteceu. Neste trabalho, a base de dados utilizada pertence ao principal sistema de geração de energia de uma unidade flutuante de produção, armazenamento e descarga (FPSO - *floating unit of production, storage and unloading*). A equipe da Petrobras forneceu registros de manutenção dos equipamentos, relatórios de operação, tabelas de equipamentos e análise financeira. Dentre essas informações pôde-se analisar o custo de manutenção para cada subsistema e determinar que diante de todos os subsistemas atribuídos à plataforma *offshore* da Bacia de Campos, o sistema de combustível é um dos que potencialmente exige um cuidado constante da manutenção, o que onera todo o sistema. Ele possui 4 turbogeradores, cada um constituído por uma turbina a gás derivada de aerogerador com capacidade nominal de 25.000 kW, acionando um gerador elétrico. A ênfase principal é dada aos motores de turbina a gás. A carga necessária da plataforma *offshore* é de aproximadamente 35-45 (MW); em seguida, cada gerador é

classificado para aproximadamente 12-15 (MW). Portanto, pelo menos três geradores devem ser operados para atender aos requisitos da plataforma *offshore* (PETROBRAS, 2019).

Existem vários tipos de falhas dentro de um turbogerador, seja no sistema de combustível, na própria turbina, no compressor, no sistema de exaustão e outros. Os eventos de máquina priorizados neste trabalho seguem a linha de pesquisa dos trabalhos: Santos *et al.* (2015) e Machado *et. Al.* (2014), que contemplam o sistema de combustível e todos os dados foram obtidos diretamente da Interface Homem Máquina (IHM) do sistema Turbo REM (sistema de registro de manutenção de equipamentos).

Identificado os dados, partiu-se para a etapa de processamento, na qual foram aplicadas técnicas de limpeza e normalização dos dados, seguindo duas abordagens: seleção de dados e de eventos. Mediante aos critérios da seleção de cada evento, totalizou-se um conjunto com 225 dados, sendo 193 paradas normais (NS – normal stop) e 32 eventos de falha. Identificadas as falhas de todos os turbogeradores, foi necessário relacioná-las com os dados dos sensores a fim de apontar qual era o estado da máquina (falha ou não falha) para cada medição de temperatura. Desta forma, o conjunto de dados foi composto por 87 colunas, sendo 85 (17 sensores x 5 características) de temperaturas normalizadas entre 0 e 1, 1 coluna indicativa de falha, sendo 0 não falha e 1 falha; e 1 coluna com as amostras/minuto (coluna de data desejada). Totalizando um conjunto de dados com 151 datas (amostras), sendo 133 NS e 17 falhas de ignição OIF (OVERFUEL TO IGNITION FAILURE SHUTDOWN). O intuito foi comparar com os estudos publicados por Santos *et. al.* (2015) e Machado *et. al.* (2014);

Em seguida, realizaram-se testes a fim de determinar qual das *features* melhor atenderia ao modelo, contemplando dados de Mínimo, Máximo, Média, Desvio Padrão, Mediana e suas combinações:

- Teste 1: *feature* chamada de **Total**, contemplando dados de todas as *features*, ou seja, Mínimo, Máximo, Média e Desvio Padrão;
- Teste 2: *feature* chamada de **Média**, contemplando dados apenas da *feature* Média;
- Teste 3: *feature* chamada de **MinMax**, contemplando dados das *features* Mínimo e Máximo;
- Teste 4: *feature* chamada de **MedMinMax**, contemplando dados das *features* Mínimo, Máximo e Média;
- Teste 5: *feature* chamada de **Desvio**, contemplando dados apenas da *feature* Desvio Padrão;
- Teste 6: *feature* chamada de **Mediana**, contemplando dados apenas da *feature* Mediana;

Para a etapa de análise dos dados, foi desenvolvido um algoritmo de classificação, em *Python*, que realizou o treinamento do conjunto de dados. Entretanto, antes de iniciar o treinamento, as amostras foram embaralhadas de uma forma balanceada e em seguida separadas, sendo que 90% foram utilizados para o treinamento, 10% utilizado para validação dos dados com o objetivo de avaliar o desempenho do modelo. Para cada um dos testes, foram implementados seis classificadores: XGB (*XGBoost*), LGBM (*LightGBM*), RFC (*Random Forest*), ETC (*ExtraTress*), RUS (*RUSBoost*) e MLP (*Perceptron* de várias camadas). E para cada classificador, o processo abaixo foi repetido 20 vezes (20 *runs*):

Dividiu-se o conjunto do experimento em n blocos/*folders* ($n=5$ ou 10).

F1, F2, ... Fn; onde cada Fi é utilizada para teste.

Para cada Fi: F1, F2, ... Fn

- Treinou-se o classificador com as $n-1$ *folders* (exceto Fi);
- Testou-se o classificador com Fi;
- Obteve-se o erro do classificador no treinamento: Ei_treino
- Obteve-se o erro do classificador no teste em Fi = Ei_teste

Calculou-se o Erro geral no teste = $(E1_teste + \dots + En_teste) / n$

E o Erro geral no treino = $(E1_treino + \dots + En_treino) / n$, obtendo-se a média geral das 20 vezes (20 *runs*). Por fim, gerou-se a matriz de confusão e algumas métricas para avaliar, quantitativamente, o algoritmo utilizado.

A matriz de confusão gerada possui as dimensões 2x2 representando as disposições do conjunto de dados para duas classes, obtendo quatro resultados possíveis apresentados na Tabela 1,

sendo (VP) número de verdadeiro positivos, (VN) número de verdadeiro negativos, (FP) número de falso positivos e (FN) número de falso negativos

Tabela 1 - Matriz de Confusão para Classificadores.

		Classe Predita (Modelo)	
		NEGATIVO	POSITIVO
Classe Verdadeira (Referência)	NEGATIVO	VN	FN
	POSITIVO	FP	VP

Fonte: Elaborado pela autora.

A partir da matriz de confusão, obteve-se a acurácia do sistema, que é uma métrica clássica e muito utilizada na literatura que indica uma medida percentual da quantidade de amostras corretamente classificadas, em relação ao total de amostras disponíveis. E será utilizada neste trabalho para fins de comparação dos resultados obtidos com outras publicações.

RESULTADOS E DISCUSSÃO

Ao analisar os testes identificou-se que a *feature* melhor desempenho é a **MedMinMax** (*feature* composta por média, mínimo e máxima). Desta forma, após executar o algoritmo de detecção de falhas obteve-se uma acurácia entre 78% a 92% para 5 *folds* e 78% a 89% para 10 *folds*. Os valores obtidos para cada classificador podem ser vistos na Tabela 2.

Tabela 2 - Acurácia geral dos classificadores.

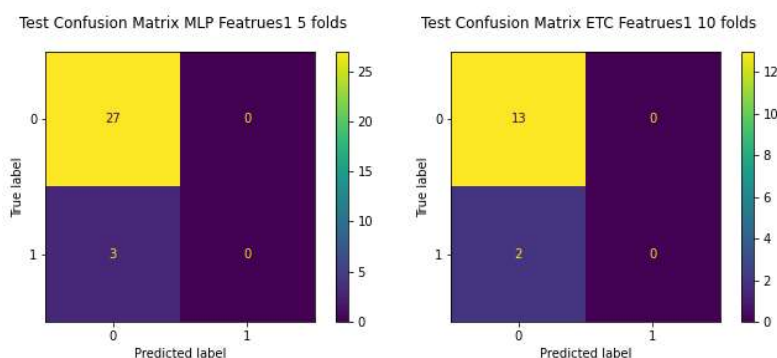
Classificador	XGB	LGBM	RFC	ETC	RUS	MLP
Média - <i>folds</i> 5	0,84	0,80	0,87	0,87	0,78	0,92
Média - <i>folds</i> 10	0,83	0,80	0,86	0,89	0,78	0,88

Fonte: Elaborado pela autora.

É possível perceber que os algoritmos RFC e RUS obtiveram acurácia de 87% e 78% respectivamente. Ao contrário do obtido em Santos *et. al.* (2015) em que o algoritmo RFC obteve bom desempenho (100% de acurácia) na fase de treinamento, mas alcançou resultados de identificação de OIF não satisfatórios no estágio de teste, apenas 1 em cada 9 falhas do OIF foram identificadas. Já o algoritmo RUS apresentou 96% de acurácia na etapa de treinamento e 5 em 9 falhas do OIF identificadas no procedimento de teste, desta forma Santos *et. al.* (2015) indica o algoritmo RUS como o mais adequado.

Neste trabalho, ao compará-los com outros classificadores, o algoritmo com melhor desempenho, para 5 *folds*, foi o MLP e para 10 *folds* foi o ETC. Na Figura 2, são esboçadas as matrizes de confusão destes dois classificadores.

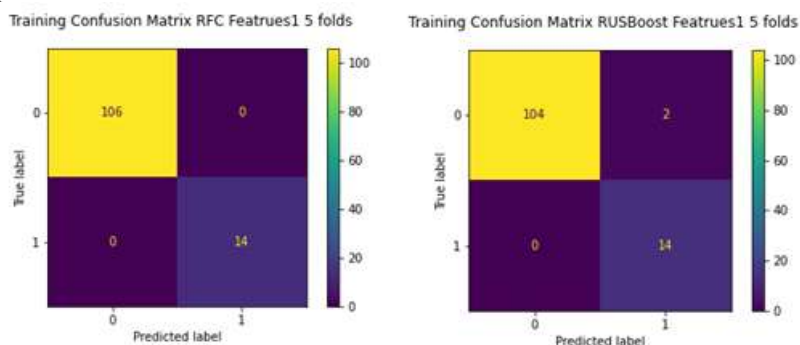
Figura 2 – Compilado das matrizes de confusão do teste.



Fonte: Elaborado pelos autores.

Nota-se que no classificador MLP (5 *folds*), das 27 amostras de não falha, todas foram classificadas corretamente como não falha. Já em relação às situações de falha, obteve-se 3 falhas no total e todas foram classificadas incorretamente como não falhas. Este fato acontece também com o classificador ETC e isso justifica-se pela quantidade de eventos de falhas do banco de dados ser pequena. Na Figura 3 é possível verificar que, neste trabalho, na fase de treino manteve-se a acurácia de 100% para o RFC e 99% para o RUS nos dados de teste.

Figura 3 – Compilado das matrizes de confusão do treino RFC e RUS.



Fonte: Elaborado pelos autores.

CONCLUSÃO

O presente trabalho apresentou um comparativo de resultados no processo de detecção de falhas entre um estudo anterior e uma nova arquitetura aplicada a turbogeradores, ambos utilizando como base os dados advindos dos sensores e dos eventos colhidos dentro do sistema. Para tal, foi necessária realizar algumas técnicas de pré-processamento no banco de dados a fim de obter *features* adequadas para a detecção de falhas nos turbogeradores.

Comparou-se os métodos desenvolvidos utilizando métricas de acurácia com trabalhos publicados, sendo apresentadas a acurácia de treinamento que indica a performance do modelo durante a etapa de aprendizado e a acurácia de teste que valida o desempenho do modelo durante uma avaliação. Vale ressaltar que apesar de se ter alcançado uma acurácia de 92% no teste com o modelo proposto, era esperado um resultado ainda superior. Isso pode ser justificado em função de um conjunto de dados com poucas informações de falhas (32 eventos), espera-se melhorar os resultados buscando aumentar a quantidade de amostras de falhas do banco de dados, como técnicas de *data augmentation*.

AGRADECIMENTOS

Os autores são gratos a Petrobrás por compartilhar seus conjuntos de dados.

REFERÊNCIAS

- MACHADO, M. M.; MANGUINHO, D. A. P. M.; VALLAND, A.; USHAKOV, S.; ROVERSO, D.; BEERE, W. H. A. RUL Modeling for Turbo Generators of a FPSO: Alternatives and Challenges. Proceedings of the Rio Oil & Gas Expo and Conference, Rio de Janeiro, Brazil, 2014.
- PETROBRAS (Rio de Janeiro). Bacias marítimas. Rio de Janeiro: Petrobrás, 2019. Disponível em: https://petrobras.com.br/pt/nossas_atividades/principais-operacoes/bacias/bacia-de-campos.htm. Acesso em: 1 nov. 2021.
- RAN, Y.; ZHOU, X.; LIN, P.; WEN, Y.; DENG, R. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. Pré-impressão arXivarXiv: 1912.07383. 2019.
- SANTOS, I. H. F. et al. Big Data Analytics for Predictive Maintenance Modeling: Challenges and Opportunities. Proceedings of the Offshore Technology Conference Brasil, Rio de Janeiro, Brazil, 2015.