

DESENVOLVENDO EM ESCALA INDUSTRIAL MODELOS PARA MANUTENÇÃO PREDITIVA NA SIDERURGIA*

Marcelo Magalhães do Carmo¹
Ricardo Pimenta de Menezes²
Severino Albani Junior³
Fabio Feu Rosa Valente⁴
Sergio Valle⁵

Resumo

Este trabalho apresenta um estudo de caso que permitiu escalar a produtividade nos processos de desenvolvimento de soluções de manutenção preditiva, utilizando técnicas e ferramentas de aprendizado de máquina. Ao comparar o esforço para desenvolver um protótipo em Python com o uso de ferramenta visual para desenvolvimentos analíticos, conseguimos acelerar a modelagem e a escalabilidade das soluções em mais de 10 vezes, alcançando resultados de modelo semelhantes. Os ganhos de produtividade e gerenciamento de soluções permitiram o desenvolvimento e implantação de 44 modelos em apenas 9 meses, quando comparado a apenas um modelo protótipo em seis meses usando bibliotecas de código aberto com Python. Os ganhos foram percebidos pelas equipes de modelagem e especialistas em manutenção envolvidos, acelerando a adoção da tecnologia preditiva e permitindo economias financeiras.

Palavras-chave: Manutenção preditiva; Detecção de anomalias.

LARGE-SCALE DEVELOPMENT OF MODELS FOR PREDICTIVE MAINTENANCE IN STEEL INDUSTRY

Abstract

This work presents a case study that allowed productivity in the scaling process of developing predictive maintenance solutions, using machine learning techniques and tools. By comparing the effort to develop a prototype in Python with the use of low-code/no-code analytics tool, we were able to accelerate the modeling and scalability of solutions by more than 10 times, achieving similar model results. The productivity gains and solution management allowed the development and deployment of 44 models in just 9 months, when compared to just one prototype model in six months using open-source libraries with Python. These gains were noticed by the modeling teams and maintenance experts involved, accelerating the adoption of predictive technology while enabling savings.

Keywords: Predictive maintenance; Anomaly detection; Outlier detection.

¹ Eng. de Computação pela UFES, Pós-graduado em Ciência de Dados pelo IFES, Mestre em Computação Aplicada com ênfase em Inteligência Artificial, IFES. Especialista em Automação e Modelagem de Processo, ArcelorMittal Tubarão. Vitória, ES, Brasil.

² Eng. Mecânico pela UFES, Mestre em Controle e Automação pelo IFES, Pós-graduado em Engenharia de Confiabilidade pela UTFPR, Especialista em Confiabilidade e Gestão de Ativos. ArcelorMittal Tubarão, Vitória, ES, Brasil.

³ Eng. Mecânico pela UFES, Pós-graduado em Engenharia de Manutenção pela UNIVIX, Pós-graduado em Gestão de Projeto pela FGV, Pós-graduado em Engenharia de Confiabilidade pela UTFPR, Especialista em Confiabilidade e Gestão de Ativos, ArcelorMittal Tubarão, Vitória, ES, Brasil.

⁴ Formado em Ciência de Computação pela UNICAMP, Mestre em Informática pela UFES com certificação PMP (Project Management Professional), Professor do MBA de Gerenciamento de Projetos na UVV (Universidade de Vila Velha) durante 8 anos. Analista de TI, ArcelorMittal Tubarão. Vitória, ES, Brasil.

⁵ Bacharel em Engenharia Elétrica com Habilitação em Computação, Pós-Graduado em Engenharia Elétrica com ênfase em Sistemas Inteligentes para Automação. Especialista em Automação e Modelagem de Processo, ArcelorMittal Tubarão. Vitória, ES, Brasil.

1 INTRODUÇÃO

A ArcelorMittal Tubarão é uma planta siderúrgica integrada com capacidade produtiva anual de 7,5 milhões de toneladas de aço, distribuídos em produtos de placas e bobinas a quente [1]. Por se tratar de uma empresa que faz uso intenso de ativos físicos para gerar valor em sua cadeia, a ArcelorMittal Tubarão possui um programa extenso de manutenção e gestão de ativos para manter sua planta operacional e eficiente. Dentro do programa, uma disciplina bastante aplicada é a Manutenção baseada na condição (sigla CBM do inglês *Condition-Based Maintenance*), ou também conhecida por manutenção preditiva.

A manutenção preditiva é a aplicação de métodos de medição, como termografia, monitoramento de vibrações e tribologia, para avaliar a condição atual dos ativos monitorados e, com base nessa condição, aplicar as atividades de manutenção necessárias [2]. O seu conceito baseia-se no acompanhamento histórico das medições e verificação de tendências de aumento [3]. No entanto, para que essas medições sejam comparáveis ao longo do tempo, elas precisam ser realizadas em condições similares e bem definidas. Portanto, a aplicação de manutenção preditiva em equipamentos que possuem vários modos de operação pode ser muito complexa ou até inviável [4].

Uma abordagem para superar essa dificuldade é a aplicação de algoritmos de Inteligência Artificial (IA). Nesse contexto, várias técnicas de IA na categoria de detecção automática de anomalias foram estudadas e desenvolvidas pela comunidade acadêmica [5][6][7] e por bibliotecas *open-source* [8][11] bem como produtos comerciais disponíveis no mercado. No entanto, devido à diversidade de opções de técnicas disponíveis que podem ser utilizadas em cada caso de uso, é necessária a presença de um especialista em IA para interpretar o problema e desenvolver a modelagem. Além disso, para cada caso de uso, todas as disciplinas de desenvolvimento, treinamento, validação, testes e implementação precisam ser executadas. Normalmente, em projetos típicos, uma equipe de especialistas em processo, análise de dados e desenvolvimento de sistemas é necessária para criar uma solução completa e gerenciada.

A escassez de profissionais capacitados em análise de dados de equipamentos industriais é um problema que tem sido relatado em vários setores. Segundo [9], nos Estados Unidos, espera-se a criação de 11,5 milhões de vagas de trabalho em ciência de dados, tornando essa disciplina uma das mais competitivas do mercado de trabalho. Além disso, de acordo com um estudo da IBM [10] de 2022, 35% das empresas já estão utilizando soluções de Inteligência Artificial (IA) em seus negócios, enquanto outras 42% estão explorando oportunidades. O estudo também aponta que essa demanda vem crescendo ao longo dos anos, aumentando a necessidade de talentos em disciplinas relacionadas à IA. Isso se torna um grande desafio na adoção desta tecnologia para 34% das empresas entrevistadas. Portanto, é importante que as empresas invistam em treinamento e desenvolvimento de seus funcionários ou em parcerias com instituições de ensino para preencher essa lacuna e aproveitar ao máximo os dados que coletam. No entanto, mesmo com esses investimentos, é crucial a adoção de ferramentas de produtividade para acelerar a entrega de resultados aos negócios.

Considerando o contexto apresentado, o objetivo do presente trabalho foi aplicar ferramentas analíticas que possam ampliar a capacidade de monitoramento de ativos industriais de maneira eficaz e escalável, com o mínimo de recursos humanos especializados necessários.

A escrita proposta deste trabalho está dividida em 3 capítulos. O Capítulo 1 apresenta a introdução do problema e contextualização geral, trazendo uma proposta de trabalho e seus objetivos. O Capítulo 2 apresenta o desenvolvimento do trabalho, com sua fundamentação

teórica, métodos e técnicas de modelagem, como também experimentos e resultados. O Capítulo 3 por sua vez, apresenta as conclusões finais e propostas de trabalhos futuros.

2 DESENVOLVIMENTO

2.1 FUNDAMENTAÇÃO TEÓRICA – ATIVOS MONITORADOS

Este trabalho visou monitorar ativos críticos da ArcelorMittal, selecionados por sua grande importância para o negócio, devido às possíveis consequências adversas em caso de falhas ou indisponibilidade. Os ativos escolhidos são bem instrumentados e historiados, o que permite uma grande quantidade de dados a serem trabalhados. A escolha de ativos com diferentes regimes de operação trouxe complexidade à análise, devido à dificuldade de estabelecer níveis adequados de operação para cada situação, além de frequentes operações fora das faixas determinadas pelo fabricante, exigindo que especialistas identifiquem anormalidades com base em sua experiência.

Colocada as premissas, foram realizadas diversas conversas com as áreas produtivas para identificar estes candidatos para o trabalho em questão. Dos ativos selecionados para o projeto podemos destacar: Exaustores, sistemas de geração de energia, sistemas de controle ambiental, entre outros.

2.2 MÉTODOS E TÉCNICAS DE MODELAGEM

Certas situações exigem a detecção de comportamentos anômalos (termo comum em inglês sendo *outliers*), mas pode ser difícil ou impossível identificar previamente todos os tipos de anomalia. Para lidar com isso, são usadas técnicas de aprendizado de máquina para identificar modos de operação normais e gerar alertas quando o processo monitorado se desvia desses modos conhecidos. A detecção de comportamentos anômalos é complexa e requer abordagens adaptáveis para cada situação. É importante entender as diferentes técnicas disponíveis para escolher a abordagem mais adequada.

Como os equipamentos complexos possuem muitos sensores historiados, identificar os modos de operação da “normalidade” é um trabalho complexo e custoso em número de horas dos especialistas de processo. Estes dados geralmente são pré-requisitos para modelagem e gerá-los manualmente para cada equipamento individual pode ser um impeditivo no uso em larga escala das técnicas de IA na predição de eventuais problemas ou mudanças de comportamento dos ativos.

De forma geral, pelo padrão de treinamento não supervisionado, a maioria das técnicas consiste em tentar agrupar os modos de operação pelas suas características. Assim que identificados estes modos, os modelos treinados comparam novos dados amostrados em tempo real no processo e verificam o quanto distante eles estão de algum outro modo de operação conhecido. Os modos conhecidos precisam ser caracterizados como modos de normalidade, pois o que desviar destes modos pode significar um problema no processo (ou um novo modo desconhecido para o modelo de operação).

Existem várias técnicas para detecção de anomalias, onde muitas estão baseadas no processo de *autoencoder* (nome em inglês para codificação automatizada), ou seja, comparar o dado original com o mesmo dado após ser processado por um *encoder* (do inglês, codificador), que simplifica o espaço de informações contidos nos dados originais por alguma técnica, e um *decoder* (do inglês, decodificador) para reconstruir o sinal original a partir dos dados codificados. A diferença entre os resultados do dado original e seu análogo após o *autoencoder* será o ruído anômalo. Quanto maior for o ruído, mais anômalo os dados amostrados estão de algum modo conhecido que foi treinado pelos históricos.

Para exemplificar, o desenho da Figura 1 mostra com dados simulados como um sistema de *autoencoder* pode ser utilizado para detectar uma anomalia sistêmica em um ambiente com múltiplos sensores. Nesta figura, um processo de *encoder* converte os dados de dois sensores no tempo em modos de operação conhecidos, e depois reconstrói o dado original por um processo de *decoder*. Esses dois passos juntos funcionam como um filtro de ruídos dos sinais, e a diferença entre a saída do *autoencoder* e o sinal de entrada mede o nível de ruído, ou desvio, entre os modos de operação conhecidos e a situação atual do ativo. Quanto maior o desvio, mais anômalo está o sistema. A anomalia não significa um problema, pois pode indicar que o equipamento está entrando em um novo estado de operação ainda não conhecido pelo modelo, mas gerará um alerta para ser avaliado por um especialista.

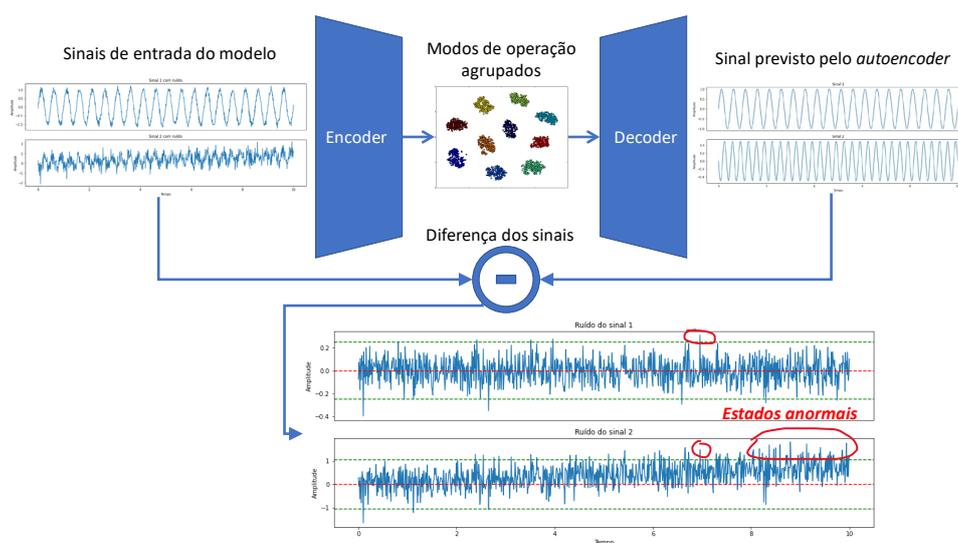


Figura 1. Exemplo de identificação de anomalias de processo utilizando *autoencoders*

Existem várias técnicas para codificar os modos de operação, indo de simples estatísticas de redução dimensional [11], como PCA (do inglês, *principal component analysis*) ou GRP (do inglês, *gaussian random projections*), passando por modelos de *clustering* (do inglês, agrupamento) [11] como *K-means*, Nearest Neighbors, *DBSCAN* e *OPiTICS*, até modelos mais complexos de *ensemble* e *deep-learning*.

Em experimentos preliminares, foram utilizados algoritmos em Python, por técnica *autoencoder* multivariada dos sinais, codificando por *ensemble* de GRP mais *Isolation Forest* (IF) e decodificando por técnica de *Random Forest* (RF), baseados nas bibliotecas *open-source* PyOD [8] e Scikit-Learn [11]. Figura 2 mostra a arquitetura do protótipo, e a Figura 3 os resultados preliminares dos eventos de anomalias identificadas em carta de controle multivariada do processo, além do indicador geral de saúde do conjunto com a probabilidade de anomalia no conjunto monitorado.

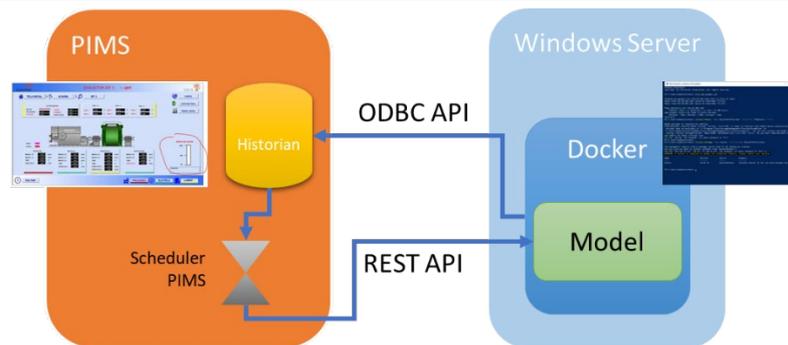


Figura 2. Arquitetura do protótipo de detecção de anomalia desenvolvido em Python¹



Figura 3. Resultados preliminares da detecção de anomalias no equipamento monitorado desenvolvido em plataformas *open-source* Python.

Os resultados foram positivos em detectar eventos de tendência no desgaste de rotores e infiltração de água misturada ao óleo, após análise das anomalias por especialistas, demonstrando a viabilidade técnica em aplicar IA ao processo. Entretanto, estes experimentos preliminares exigiram a expertise dedicada de especialistas de manutenção, tecnologia e modelagem por um período de seis meses para a criação de um único protótipo funcional. Replicar essas atividades para múltiplos equipamentos utilizando a mesma abordagem seria extremamente trabalhoso. Para simplificar esse processo, ferramentas e métodos foram aplicados a fim de reduzir o esforço necessário.

Para escalar a iniciativa, uma ferramenta *low-code/no-code* (termos em inglês para ferramenta visual de desenvolvimento) foi escolhida neste projeto. O sistema é composto por vários componentes, sendo eles separados por funcionalidades, conforme Figura 4. Todos os dados utilizados no processo de treino e execução do modelo são obtidos do ambiente PIMS historiador da planta. Os modelos são executados no componente de *Predictive Analytics Server*, e desenvolvidos no *Predictive Analytics Client*. Para visualização a ferramenta disponibiliza um recurso web de acesso aos relatórios e visualizações em ambiente web, na camada de *Predictive Analytics Web Server*. Todas as configurações e dados para visualização são salvos em uma pequena instancia de banco de dados Microsoft SQL, e os dados coletados para treino dos modelos são salvos em um modulo de banco de dados temporal *Embedded Historian*.

Todas as previsões e alertas gerados podem ser integrados com ferramentas externas, ou tratadas internamente pela própria ferramenta para acionamento dos especialistas de manutenção. O ambiente utilizado possui capacidade de treinar e rodar 50 modelos simultâneos, sendo ele uma máquina virtual de 8 núcleos e 32 GB de RAM, o que é relativamente pouco para os padrões atuais de alocação de recursos em ambiente virtualizado.

¹ <https://github.com/svjr/windows-server-core-wfastcgi-flask-python>

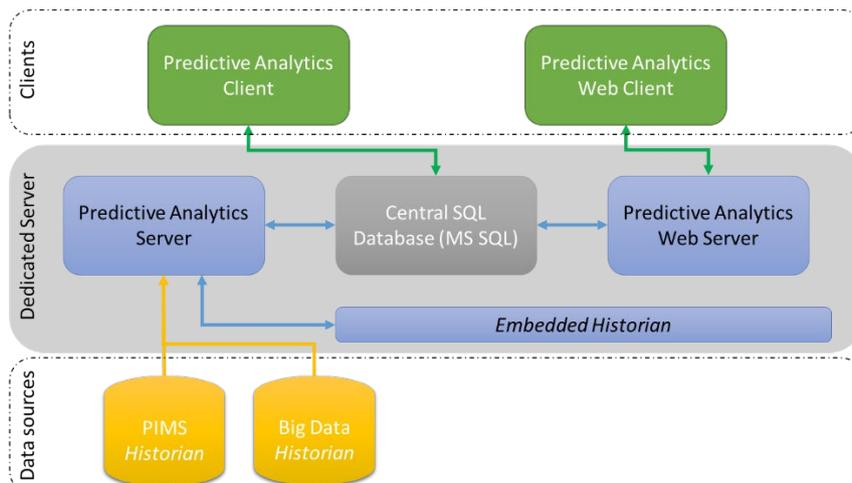


Figura 4. Arquitetura de componentes da solução analítica baseada em desenhos da documentação interna do produto²

A construção dos modelos seguiu as seguintes etapas apresentadas no diagrama da Figura 5. De forma macro, o processo se inicia pela pré modelagem, onde são levantados os dados dos ativos e definidos como as variáveis serão agrupadas em modelos. Existem várias formas de se fazer esse agrupamento, como ilustrado na Figura 6. Podem ser aplicados modelos por subconjunto do ativo, por exemplo um modelo para monitorar a turbina, outro para o gerador e mais um para o soprador. Ou então, um modelo monitorando todo o processo de vapor e ar soprado, e um modelo monitorando todas as variáveis de condição da parte mecânica (vibrações mancais, temperatura, lubrificação etc.). Essa prática vai depender do equipamento/processo e do objetivo do monitoramento.

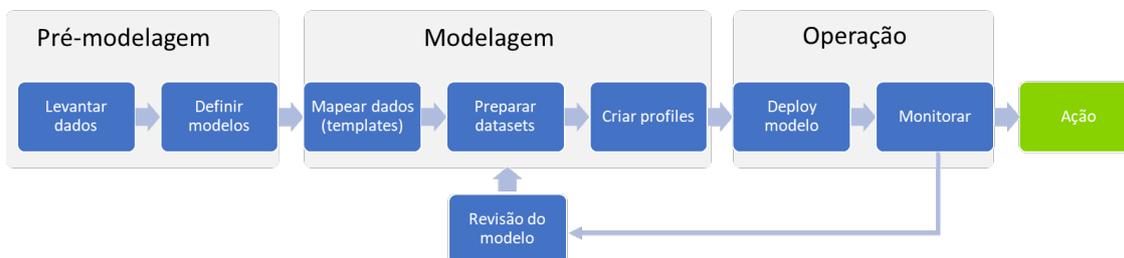


Figura 5. Diagrama com os passos macros para implementação dos modelos.

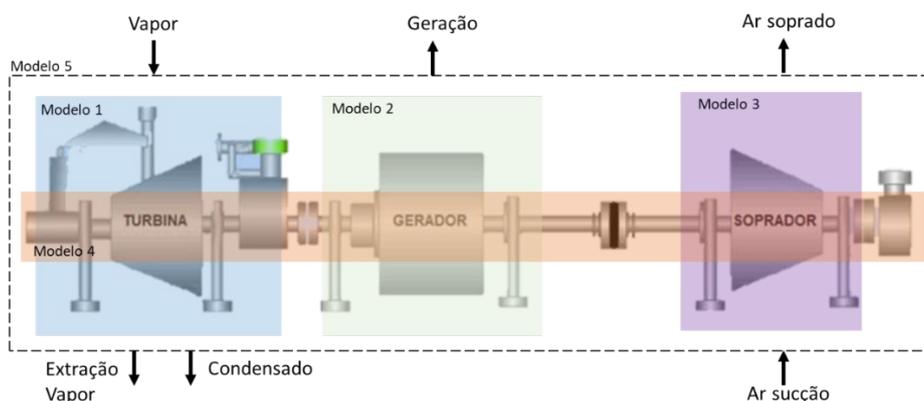


Figura 6. Exemplo de agrupamento das variáveis para criação dos modelos.

² <https://www.aveva.com/en/products/predictive-analytics/>

A etapa seguinte compreende a parte de modelagem do ativo. Nela são criados *templates* (do inglês, usado para referenciar um padrão) dos ativos a serem modelados. O *template* pode ser usado em diversos projetos de forma a acelerar o processo de criação de modelos. O *template* consiste na definição de forma genérica das variáveis que compõem os ativos. Por exemplo, para um motor elétrico seriam mapeadas as variáveis de corrente elétrica, temperatura dos enrolamentos, velocidade de rotação, tensão de alimentação etc. Assim que criado um *template*, ele pode ser aplicado de forma mais rápida para construir vários projetos para modelar motores elétricos.

Criado os *templates*, parte-se para o projeto do modelo. Todo projeto possui como base um *template*, as variáveis mapeadas, os *datasets* (do inglês para conjunto de dados) para treino e teste do modelo, o profile de operação, filtros para desligar o modelo, os modos de falhas conhecidos e anotações. Os *datasets* carregados no projeto podem ser utilizados para treino do modelo e para testes de seu funcionamento. O *dataset* de treino deve ser cuidadosamente escolhido e ajustado na ferramenta (removido outliers ou dados ruins). Esse *dataset* deve refletir o funcionamento normal do ativo. Ele será utilizado como a referência de normalidade durante o monitoramento do ativo. Neste passo do desenvolvimento, o acelerador está no tratamento visual de todos os dados, e cálculos automáticos das estatísticas durante a limpeza dos dados.

O próximo passo é a criação do profile de operação (treino do modelo). A ferramenta escolhida implementa nativamente alguns modelos de detecção de anomalias, e entre eles LSH-NN (Locality Sensitive Hashing Nearest Neighbor), o qual foi utilizado em praticamente todos os modelos. A ideia básica deste algoritmo é construir uma tabela *hash* onde cada modo de operação (denominado pela ferramenta de *operational profiles*) é representado por uma linha desta tabela. Uma vez construída a tabela *hash* (passo de treinamento do *encoder*), a previsão para novos dados será feita calculando o *hash* do novo sinal, pesquisando pelo *hash* mais próximo e decodificando o sinal pelo inverso da geração da *hash* encontrado na tabela (passo *decoder*). Como a ferramenta é proprietária, o algoritmo interno de *hashing* não pode ser modificado e funciona como uma caixa preta para o especialista de manutenção. Esta abordagem possui vantagens e limitações. Por um lado, acelera grandemente o processo de modelagem de novos ativos, mas limita os “ajustes finos” para cada modelo e cenário específico. Em caso de necessidade de ajustes mais finos dos modelos, a ferramenta também possui outras técnicas parametrizáveis, como Neural Networks e OPTiCS.

A partir da criação do profile de operação é possível implantar o modelo em operação. Esta etapa envolve questões como o tempo de amostragem do modelo, os níveis de alarme e janela de tempo para ativação dos alarmes, o profile de operação desejado, dentre outros. Uma vez em operação, o monitoramento do modelo passa a ser feito na versão web da aplicação. A aplicação possui dashboards para gestão dos modelos e ferramentas para detalhar as análises de cada modelo, agilizando o processo de diagnóstico de falhas e anomalias.

A Figura 7 exemplifica as saídas geradas pelo sistema. A parte superior apresenta um gráfico no tempo chamado de *Overall Model Residual* (ORM), que apresenta a

soma percentual do resíduo do modelo para todas as variáveis. De forma resumida, pode ser interpretada como o desvio da normalidade do ativo. Quanto maior esse nível, mais distante da normalidade o ativo se encontra. Logo abaixo o sistema possui uma estatística das variáveis que mais estão contribuindo para o desvio de normalidade do ativo. O próximo quadro apresenta os modos de falhas mapeados durante o processo de modelagem. Quando o padrão desses modos de falha mapeados se instala no equipamento, ele aparece neste quadro de forma destaca. Esses modos de falha podem ser criados com a descrição de ações ou próximos passos a serem realizados, ajudando as equipes de inspeção e monitoramento a tomarem as ações necessárias para resolver um possível problema no ativo. Abaixo desses quadros é possível visualizar todas as variáveis mapeadas no modelo e a diferença entre o valor real e o valor previsto pelo modelo.



Figura 7. Exemplo de resultados gerados pela interface web da aplicação.

Mais importante que monitorar é gerar ações para resolver os possíveis problemas. Esta etapa envolve a estruturação das equipes para que exista um processo bem definido e pessoas treinadas para gerar ações em cima desta ferramenta. Muitas vezes um alarme gerado na ferramenta pode ser apenas uma questão de retreino do modelo para um novo modo de operação. Portanto, a simplicidade da ferramenta apoia tanto especialistas do processo quanto equipes de manutenção preditiva a manter o sistema atualizado e assertivo.

2.3 EXPERIMENTOS E RESULTADOS

Nesta seção são apresentados elementos e casos que demonstram o valor da ferramenta para a rotina das equipes de confiabilidade e gestão de ativos em plantas industriais.

O projeto teve uma duração de 9 meses, passando por toda estruturação de infra, mapeamento e modelagem dos ativos até sua fase de operação e ajuste dos modelos. Nesse período foram criados e implantados 44 modelos. A Tabela 1 apresenta os ativos e seus quantitativos de modelos e variáveis, sendo a linha 3 o protótipo realizado em Python, para comparação.

Tabela 1. Modelos implantados por área e processo

Area	Ativos	Template ativo	Quant. Variáveis	Quant. modelos
Aciaria	Ativo 1.1	Template 1	53	3
Aciaria	Ativo 1.2	Template 1	50	3
Aciaria	Ativo 1.3	Template 1	43	1 (Python)
Aciaria	Ativo 2.1	Template 2	41	3
Laminação	Ativo 3.1	Template 3	68	8
Utilidades	Ativo 4.1	Template 4	55	5
Utilidades	Ativo 4.2	Template 4	55	5
Utilidades	Ativo 4.4	Template 4	49	3
Utilidades	Ativo 4.5	Template 4	79	5
Utilidades	Ativo 4.6	Template 4	49	3
Utilidades	Ativo 5.1	Template 5	81	3
Ambiental	Ativo 6.1	Template 6	29	3
Total	11+1	6	609+43	44+1

Obs.: Em cinza está o protótipo em Python, deixado separada na totalização para comparações.

A seguir é apresentado um caso em que os níveis de vibração do Ativo 4.5 atingiram valores fora do normal, mas dentro dos limites de alarme de outros sistemas de monitoramento. A Figura 8 apresenta o gráfico visual do alarme gerado para o indicador *ORM* e direcionando para a variável alarmada. Na mesma imagem é possível verificar esta mesma variável em outro sistema especialista de análise de vibração, mostrando que a variável não atingiu os limites pré-estabelecidos de alarme. Este evento em particular foi causado pelo deslocamento da escova de aterramento do eixo do gerador, sendo prontamente resolvido pela equipe de area.

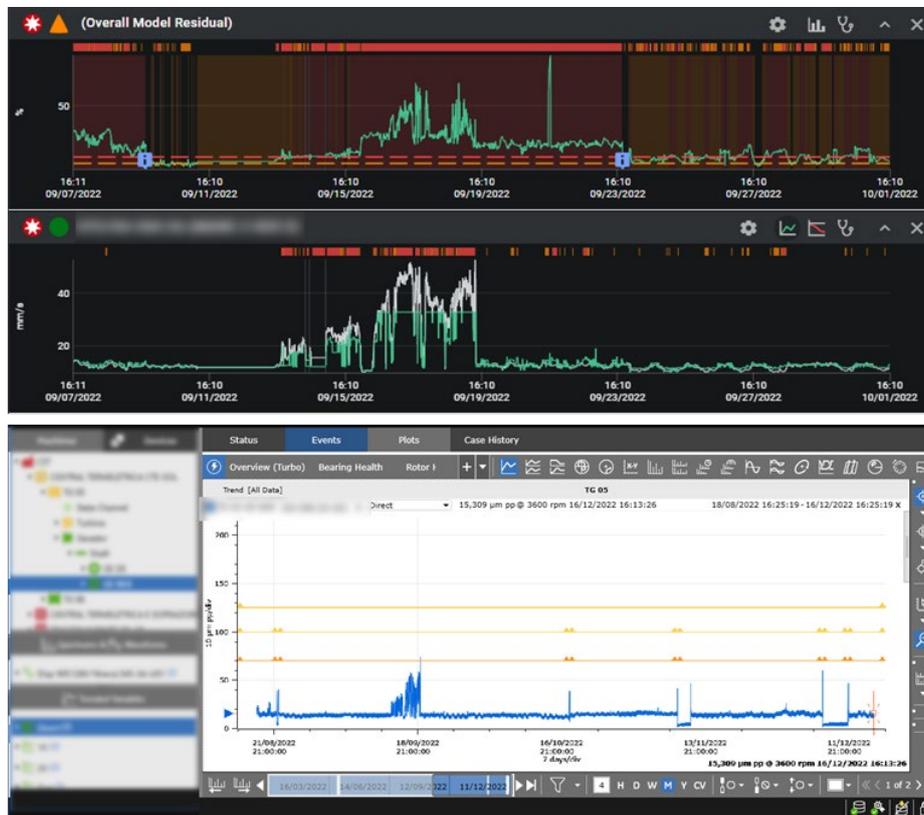


Figura 8. Caso de anomalia identificada no Ativo 4.5 monitorado durante o projeto.

O caso seguinte foi identificado em um dos ativos da Aciaria. O ORM do modelo gerou um alarme com uma tendência de aumento, indicando as variáveis que mais estavam contribuindo para esse desvio de padrão (apresentado na Figura 9). Nesse ponto foi verificado um aumento de vazão de óleo nos mancais do exaustor e uma redução no nível de óleo no tanque. Utilizando a ferramenta de RULE (do inglês, *Remaning Useful Life Estimation*), foi possível estimar quando o nível de óleo no tanque atingiria um valor preocupante para a operação do equipamento em 1,38 dias. Com base nesses dados a equipe de campo foi acionada para inspeção do ativo, onde se constatou um vazamento de óleo pelos mancais (Figura 10). As válvulas de controle do óleo lubrificante foram ajustadas e o nível do tanque de óleo completado. Este caso foi um típico evento de mudança no processo do equipamento que poderia resultar em mudanças na sua condição, normalmente percebidas pelo aumento de ruído, vibração ou temperatura. Um destaque para este caso é que após o ocorrido, foi criada uma regra no sistema para monitorar este modo de falha no futuro, gerando conhecimento de forma sistêmica.



Figura 9. Caso de anomalia identificada em um dos ativos monitorados durante o projeto.



Figura 10. Inspeção em campo evidenciando vazamento de óleo.

3 CONCLUSÃO

O presente trabalho teve como objetivo a aplicação de ferramentas analíticas para escalar a capacidade de monitoramento de ativos críticos, com a menor quantidade possível de mão de obra especializada. Comparando os 6 (seis) meses do protótipo funcional para um único equipamento com desenvolvimento *open-source* Python, aos 9 (nove) meses para desenvolvimento de 44 modelos em ativos de semelhante porte, a escalabilidade na manutenção preditiva aumentou em ao menos uma ordem de magnitude de 10 vezes. A maior parte dos ganhos se deve pelo uso de interfaces *low-code/no-code* para modelagem e implantação, minimizando a dependência de cientistas de dados para o desenvolvimento.

Outro ganho importante foi a capacidade de monitorar desvios de operação continuamente. Apesar de capacidade dos especialistas em realizarem esta atividade, uma ferramenta pode fazê-la sem descanso, liberando o tempo dos especialistas para atuarem em frentes de melhoria contínua. Além disso, a identificação de anomalias pode alertar os especialistas sobre problemas inéditos, sendo essencial para evitar falhas não previstas em ativos críticos ou fazer diagnóstico de causa raiz.

Como trabalhos futuros, visa-se integração com ambiente de gestão de alarmes centralizado para monitoramento de ativos da ArcelorMittal Tubarão. Esta integração permitirá abertura automática de notas de tratamento e planos de ação. Além disso, espera-se também a expansão dos modelos para mais ativos e processos críticos para a empresa, como também futuras comparações com outras ferramentas analíticas em desenvolvimento e avaliação pelo grupo ArcelorMittal.

REFERÊNCIAS

- 1 ArcelorMittal Tubarão. Quem Somos. 2023 [acesso em 3 de abril 2023]. Disponível em: <https://brasil.arcelormittal.com/>.
- 2 Mobley, R. Keith. An Introduction to Predictive Maintenance. 2ª edição. Netherlands: Elsevier Science, 2002.
- 3 Mobley, R. K., & Higgins, L. R. Maintenance Engineering Handbook, 6ª edição. New York: McGraw-Hill, 2002.
- 4 Kamat P, Sugandhi R. Anomaly detection for predictive maintenance in industry 4.0-A survey. InE3S web of conferences 2020 (Vol. 170, p. 02007). EDP Sciences.

- 5 Cook AA, Mısırlı G, Fan Z. Anomaly detection for IoT time-series data: A survey. IEEE Internet of Things Journal. 2019 Dec 6;7(7):6481-94.
- 6 Han S, Hu X, Huang H, Jiang M, Zhao Y. Adbench: Anomaly detection benchmark. Advances in Neural Information Processing Systems. 2022 Dec 6; 35:32142-59.
- 7 Wang H, Bah MJ, Hammad M. Progress in outlier detection techniques: A survey. IEEE Access. 2019 Aug 5; 7:107964-8000.
- 8 Zhao Y, Nasrullah Z, Li Z. Pyod: A python toolbox for scalable outlier detection. arXiv preprint arXiv:1901.01588. 2019 Jan 6.
- 9 Ahmad N, Hamid A, Ahmed V. Data Science: Hype and Reality. Computer. 2022 Feb 14;55(2):95-101.
- 10 IBM. IBM Global AI Adoption Index 2022.2022 [Acesso em 3 de abril 2023]. Disponível em: <https://www.ibm.com/downloads/cas/GVAGA3JP>
- 11 PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, v. 12, p. 2825-2830, 2011.]