Pontifícia Universidade Católica de Minas Gerais

Campus de Poços de Caldas

Curso de Engenharia de Produção

Antonella Vasconcelos; Wiliam Regone

Rede Neural Artificial Aplicada em Dados de Manutenção Preditiva

Projeto de Iniciação Científica apresentado no curso de Engenharia da Produção da Pontifícia Universidade Católica de Minas Gerais, Campus de Poços de Caldas EDITAL FIP Nº 075/2022.
Número do Projeto: 2023/28909

Orientador(a): Wiliam Regone
Orientando(a): Antonella Vasconcelos

Poços de Caldas

2023

Rede Neural Artificial Aplicada em Dados de Manutenção Preditiva

Wiliam Regone; Antonella Vasconcelos

PUC MINAS - campus Poços de Caldas

Curso de Engenharia de Produção

Av. Pe. Cletus Francis Cox, 1.661 – Jardim Country Club

37.714-620 – Poços de Caldas – Minas Gerais

wiliamregone@gmail.com; antonella.vasconcelos@sga.pucminas.br

Resumo: A manutenção preditiva realiza o acompanhamento de variáveis e parâmetros de desempenho de máquinas, visando determinar o momento mais adequado para intervenção. Os sistemas produtivos são capazes de obter dados gerados por sensores, os quais podem ser interpretados e utilizados por algoritmos que empregam técnicas de inteligência artificial, como redes neurais artificiais. Este estudo explora a implementação da manutenção preditiva, abordando o desequilíbrio nos dados de manutenção, obtidos por um conjunto de dados com 10.000 registros e 10 variáveis do processo com características distintas que geram falhas em máquinas. Para isso, emprega métodos avançados de inteligência artificial. A metodologia inovadora inclui o uso da biblioteca 'imblearn' e da técnica 'SVMSMOTE' para equilibrar as classes de dados, seguida pela construção de modelos de redes neurais. Os resultados, ilustrados por meio de histogramas, gráficos de dispersão, comparações estatísticas e 'heatmaps' de correlação, demonstram a eficácia dos modelos na previsão de falhas e no funcionamento normal das máquinas. O estudo destaca variações significativas na performance dos modelos, especialmente no modelo tipo de falhas, ressaltando a complexidade da manutenção preditiva na era da Indústria 4.0. Assim, os resultados obtidos melhoram o processo ao fornecer avaliações que podem ser usadas na tomada de decisões para prever quais tipos de falha podem causar paradas nas máquinas.

Palavras-Chave: Manutenção Preditiva, Rede Neural Artificial, Falha de Máquina.

***Financiado pela PUCMINAS – Campus de Poços de Caldas - FIP - EDITAL Nº 075/2022**

1. Introdução

A Indústria 4.0, marca o advento da quarta revolução industrial, e é um salto evolutivo caracterizado pela fusão de tecnologias digitais avançadas com processos industriais tradicionais. Originada na Alemanha, esta era é definida por uma customização em massa, interconectividade sem precedentes, e um fluxo numeroso de dados, onde dispositivos inovadores e sistemas de produção controlados reduzem a necessidade de inventário extensivo. A característica distintiva da Indústria 4.0 reside na habilidade dos componentes de serem autoconscientes, autopreditivos, e na capacidade das máquinas de autoavaliação. Além disso, sistemas de produção nesta era podem se autoconfigurar e autossustentar, otimizando continuamente os processos com base em dados em tempo real. Esses dados não são apenas cruciais para prevenir falhas, mas também essenciais para facilitar a tomada de decisões preditivas e aprimorar a eficiência operacional [LEE, Jay et al., 2015].

A manutenção preditiva, um conceito-chave na Indústria 4.0, utiliza uma base de dados históricos, modelos avançados e conhecimento do domínio para prever falhas antes que aconteçam, otimizando assim o processo de manutenção e reduzindo significativamente o tempo de inatividade. A adoção da manutenção preditiva impulsiona a transição para uma indústria mais inteligente, onde a complexidade crescente das interações em ecossistemas de manufatura expandidos é gerenciada de forma eficiente. A integração da Internet das Coisas Industrial (IIoT) em ambientes industriais combina a conectividade e automação da Internet das Coisas (IoT) com tecnologias de ponta como aprendizado de máquina e Big Data. Por exemplo, em uma fábrica equipada com IIoT, sensores podem coletar dados em tempo real de máquinas, como temperatura e velocidade, e enviar essas informações para um sistema central. Este sistema, usando aprendizado de máquina, analisa os dados para identificar padrões ou anormalidades, antecipando falhas antes que ocorram. Já o sistema Ciberfísico (CPS) leva essa integração um passo adiante, combinando esses dados e análises com sistemas de controle físico, resultando em uma automação mais sofisticada e adaptativa. A integração da IIoT incorpora tecnologias de IoT em ambientes industriais, unindo-as com aprendizado de máquina e Big Data. Isso resulta em máquinas "inteligentes" que superam as capacidades humanas na gestão de dados. O Sistema CPS, utilizando a IoT, representa a próxima geração de sistemas de engenharia, com uma arquitetura que inclui múltiplas camadas como conexão digital e infraestrutura computacional de alta performance. Assim, a infraestrutura computacional de alta performance e conexões digitais avançadas permitem um controle mais eficaz e eficiente dos processos industriais, como ajustar automaticamente as configurações de uma máquina com base nas análises de desempenho em tempo real. Esta integração de conceitos forma a base para o desenvolvimento avançado da área de manutenção preditiva [LEE, Jay et al. 2015 and SEZER, Erim et al. 2018].

A evolução contínua da manutenção preditiva e da manufatura preditiva na Indústria 4.0 destaca a importância crescente de integrar novas tecnologias, como elemento estratégico no ambiente industrial. Enquanto a manutenção preditiva foca na prevenção de falhas e no aumento da vida útil dos equipamentos através de dados coletados por sensores, a manufatura preditiva vai além, empregando esses dados para aprimorar e otimizar toda a cadeia produtiva. Esta sinergia entre manutenção e produção, fundamentada no gerenciamento da saúde de sistemas e prognósticos, permite não apenas a observação, mas também análises profundas e ações preditivas baseadas em dados coletados em tempo real. Essa abordagem integrada é vital para elevar a produtividade, melhorar a qualidade dos produtos e aumentar a eficiência operacional, minimizando falhas e interrupções nos processos produtivos, contribuindo significativamente para a sustentabilidade e a competitividade no mercado moderno [ZONTA, Tiago et al., 2020. and LEE, Jay et al. 2015].

Na integração técnica entre manutenção preditiva, manufatura preditiva e aprendizado de máquina para a fabricação sustentável, observa-se um alinhamento estratégico com os princípios da Indústria 4.0. A manutenção preditiva, fundamentada em análises preditivas avançadas, emprega algoritmos de aprendizado de máquina para monitoramento contínuo e diagnóstico precoce de falhas

em equipamentos. Esta metodologia não apenas otimiza a eficiência operacional, mas também prolonga a vida útil dos ativos. Paralelamente, a manufatura preditiva, ao sincronizar-se com esses insights, adapta os processos produtivos, resultando em redução de desperdícios e promoção da eficiência energética. Essa sinergia cria uma cadeia de produção mais sustentável e inteligente, refletindo um compromisso com a sustentabilidade ambiental e a eficiência de recursos [ABIDI, Mustufa Haider; MOHAMMED, Muneer Khan; ALKHALEFAH, Hisham, 2022].

A manutenção preditiva sobressai por sua implementação estratégica de dados temporais e análise preditiva, fornecendo alertas antecipados sobre falhas potenciais e possibilitando intervenções proativas. Este método pode ser reforçado pelo emprego de algoritmos de Inteligência Artificial (IA) e ou Inteligência Artificial Explicável (XAI), que proporcionam uma análise de dados mais detalhada e compreensível. Algoritmos de IA são conjuntos de regras e técnicas projetados para simular capacidades cognitivas humanas, como aprendizado, raciocínio e auto-correção, em máquinas. Eles desempenham um papel fundamental em várias aplicações, desde reconhecimento de voz até análise preditiva. Em contrapartida, a XAI foca no desenvolvimento de algoritmos e modelos de IA que são eficazes, mas também transparentes e compreensíveis para os usuários. O objetivo da XAI é clarear os processos e decisões de IA promovendo a confiança e a gestão eficaz destas tecnologias, especialmente em contextos onde a tomada de decisão da IA necessita ser justificada ou auditada [AHMED, Imran; JEON, Gwanggil; PICCIALLI, Francesco, 2022 and FORESTI, Ruben et al. 2020]. Essas técnicas avançadas facilitam uma gestão mais eficiente dos processos de manutenção, variando desde abordagens baseadas em modelos físicos, utilizando modelagem matemática e estatística, até abordagens baseadas em conhecimento, que simplificam esses modelos, e métodos baseados em dados, que aplicam reconhecimento de padrões e técnicas avançadas de IA, como Redes Neurais Artificiais (RNAs), para a previsão precisa de falhas.

Os métodos que empregam RNAs e técnicas de aprendizado profundo são destacados na literatura científica devido à sua habilidade em modelar relações complexas e não-lineares entre múltiplas variáveis de entrada e saída e estabelecendo padrões de reconhecimento. Assim, tornam-se apropriadas e úteis para análise de dados de manutenção. Neste estudo, empregou-se um conjunto de dados extenso, contendo 10.000 observações e 10 variáveis distintas, sendo estas, parâmetros como 'Air temperature', 'Process temperature', Rotational speed', 'Torque', 'Tool wear' e 'Failure Type'. O objetivo é prever cinco tipos distintos de falhas de máquinas utilizando métodos avançados de IA para análise preditiva. Este trabalho ilustra a aplicação prática de IA na manutenção preditiva, com foco em otimizar a vida útil das máquinas e minimizar os riscos de falhas operacionais, refletindo um avanço significativo no campo da manutenção assistida por IA.

2. Metodologia

No contexto da manutenção de equipamentos ou processos produtivos, é comum que os dados de funcionamento correto superem em quantidade os dados de falhas, o que representa um desafio para a manutenção preditiva devido ao desequilíbrio nos dados de entrada da rede neural. Para obter um modelo preditivo de RNAs eficaz, é crucial equilibrar o conjunto de dados, garantindo uma representação adequada tanto de instâncias de falha quanto de operação normal.

A aplicação de RNAs exige um conjunto de dados reais de manutenção preditiva, os quais são desafiadores de adquirir, dada a relutância das indústrias em compartilhar informações que podem ser vistas como vantagem competitiva estratégica. No entanto, recentemente o artigo MATZKA, Stephan. Explainable artificial intelligence for predictive maintenance applications. In: 2020 third international conference on artificial intelligence for industries (ai4i). IEEE, 2020. p. 69-74 disponibilizou um conjunto de dados de manutenção que espelha os dados reais de manutenção

preditiva encontrados no setor industrial, de acordo com conhecimento e experiência dos autores, o que é extremamente valioso para pesquisas nesse campo. Os dados podem ser acessados em S. Matzka, www.explorate.ai/dataset/predictiveMaintenanceDataset.csv, 2020.

2.1. Conjunto de Dados Para Aplicação nos Modelos de RNA

Este conjunto de dados compreende 10.000 registros com 10 variáveis cada. Os dados foram pré-processados, incluindo limpeza de ruídos e outras formas de tratamento como normalização de dados, para otimizar o desempenho da RNA. Neste projeto de pesquisa, as variáveis de entrada serão 'Air temperature [K]', 'Process temperature [K]', Rotational speed [rpm]', 'Torque [Nm]' e 'Tool wear [min]'. Os dados de saída representarão as falhas da máquina, as quais são classificadas em cinco modos independentes: 'Power Failure', 'Tool Wear Failure', 'Overstrain Failure', 'Random Failures' e 'Heat Dissipation Failure'. O objetivo é prever o tipo específico de falha que causou a interrupção da máquina.

A Figura 1 representa um histograma de barras que mostra a contagem de duas categorias distintas dentro de um conjunto de dados intitulado "Original Data". As categorias são 'No Failure' e 'Failure', representadas pelos números 0 e 1, respectivamente. O eixo y indica a contagem de ocorrências em cada categoria, revelando uma grande disparidade entre elas: a categoria 'No Failure' ocorre muito mais frequentemente, com aproximadamente 9652 ocorrências, enquanto a categoria 'Failure' é significativamente menos comum, com apenas 348 ocorrências. Este desequilíbrio sugere que falhas são eventos raros dentro deste conjunto de dados específico, o que pode ter implicações importantes para a modelagem preditiva e análise de risco.

O gráfico, Figura 2 é um histograma de barras que representa a distribuição de diferentes tipos de falhas do conjunto de dados "Original Data". Ele mostra cinco tipos de falhas: ‘Power Failure’ ocorre 95 vezes, ‘Tool Wear Failure’ cerca de 45 vezes, ‘Overstrain Failure’ 78 vezes, ‘Random Failures’ são menos comuns com cerca de 18 ocorrências, e ‘Heat Dissipation Failure’ é o mais frequente com 112 ocorrências. Isso indica a prevalência de diferentes falhas, sendo as falhas de dissipação de calor as mais comuns e as falhas aleatórias as menos comuns neste conjunto de dados específico.

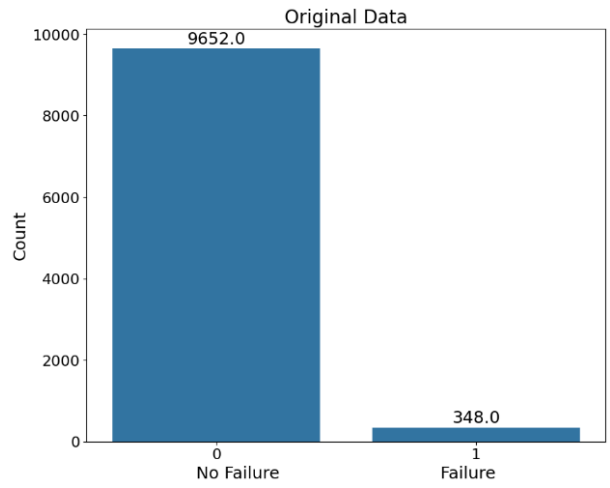

Figura 1 - Histograma de barras com categorias de 'No Failure' e 'Failure', representadas pelos números 0 e 1.

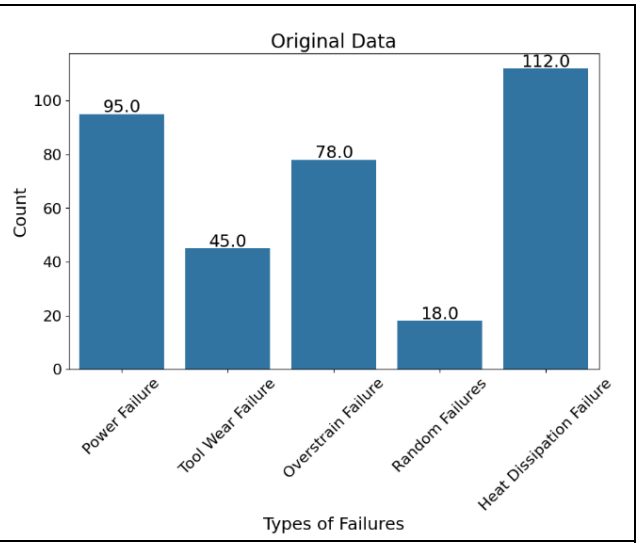

Figura 2 - Histograma de barras que representa a distribuição de diferentes tipos de falhas.

A Figura 3 apresenta uma série de gráficos, que formam uma ferramenta de análise exploratória, ilustrando as relações entre várias variáveis categóricas e numéricas do conjunto de dados, "Original Data". As cores nos gráficos representam diferentes categorias dentro das variáveis 'Target' e 'Failure Type'. Os gráficos ocupam as posições abaixo e acima da diagonal principal de uma matriz.

Os gráficos abaixo da diagonal principal são coloridos de acordo com a variável categórica 'Target', onde o azul representa a categoria 0 (sem falha) e o vermelho representa a categoria 1 (com falha), facilitando a distinção visual entre as duas condições operacionais distintas. Estes gráficos de dispersão são úteis para identificar correlações e anomalias, bem como para observar a concentração e dispersão dos dados, fornecendo insights imediatos sobre como diferentes condições operacionais estão distribuídas em relação às variáveis numéricas. A disposição dos gráficos permite uma análise comparativa entre pares de variáveis, como 'Air temperature', 'Process temperature', 'Rotational speed', 'Torque' e 'Tool wear'.

Acima da diagonal principal, a variável categórica 'Failure Type' é utilizada para colorir os pontos, diferenciando diversos tipos de falhas, como 'Power Failure', 'Tool Wear Failure', 'Overstrain Failure', 'Random Failures' e 'Heat Dissipation Failure'. A coloração variada ajuda a distinguir como as condições de falha se relacionam com as variáveis contínuas, ou seja, isso ajuda a identificar padrões de falha em relação às variáveis medidas, além de, proporcionar uma perspectiva adicional sobre como diferentes falhas estão associadas a combinações específicas das variáveis numéricas. Desta forma, para a metade superior da matriz, os gráficos, das distribuições de densidade para cada variável numérica são representadas, oferecendo uma visualização da frequência dos dados ao longo de um intervalo de valores. Essas distribuições de densidade são úteis para entender a variação intrínseca e a distribuição de cada variável individualmente.

A matriz de gráficos de dispersão, juntamente com as distribuições de densidade e a coloração diferenciada dos pontos, fornece uma base robusta para a análise exploratória de dados. Esta visualização composta é essencial para reconhecer padrões, identificar relações potenciais e compreender como as variáveis interagem entre si sob diferentes condições operacionais e de falha.

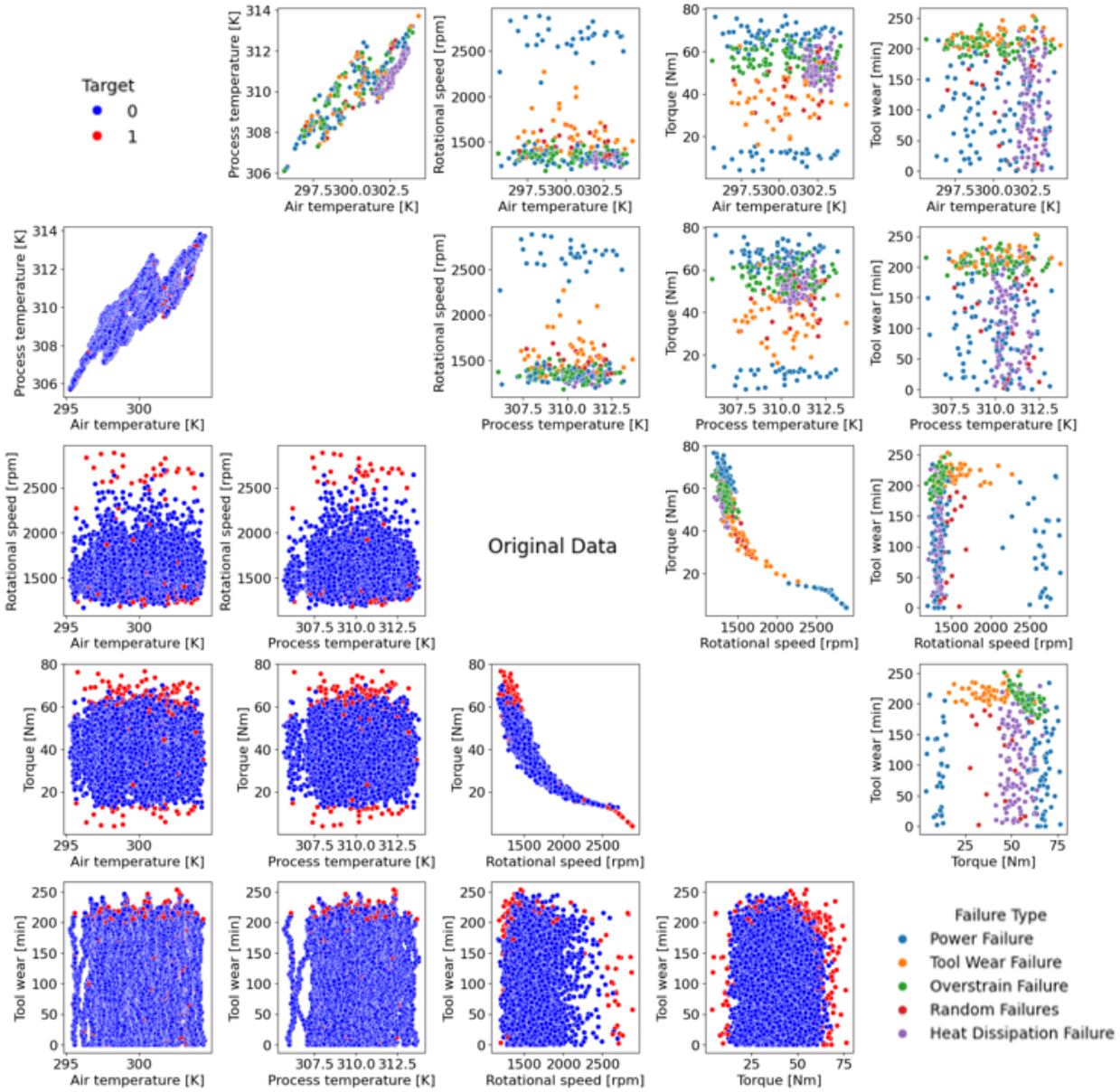

Figura 3 - Gráficos, entre várias variáveis categóricas ('Target') e numéricas de um conjunto de dados ('Failure Type').

Continuando a observação (Figura 3) das relações entre as variáveis, nota-se uma relação quase linear entre 'Air temperature' e 'Process temperature', refletindo uma correlação positiva substancial, sugerindo que ambas as variáveis podem ser influenciadas por fatores comuns ou que uma pode ser um indicador direto da outra. Em contraste, 'Torque' e 'Rotational speed' parecem seguir uma relação inversa, onde valores elevados de um implicam em valores reduzidos do outro, sinalizando uma potencial correlação negativa, que pode ser fundamental para o controle de processos e a detecção de condições anormais de operação.

A distribuição dos dados revela padrões igualmente significativos. 'Air temperature' e 'Process temperature' apresentam uma dispersão de dados mais concentrada, o que pode indicar consistência e controle efetivo nessas medidas de processo. Em oposição, a ampla dispersão observada em 'Torque' e 'Tool wear' pode ser interpretada como uma variabilidade maior, possivelmente devido à influência de variáveis externas ou não controladas.

Adicionalmente, a densidade de pontos e o overlap (sobreposição) evidentes em alguns gráficos, como entre 'Rotational speed' e 'Torque', destacam regiões onde as falhas são mais frequentes. O overlap de diferentes cores nestas áreas de alta densidade pode sugerir a co-ocorrência de múltiplos tipos de falhas sob condições operacionais similares, ou pode indicar condições que são particularmente propensas a falhas.

Padrões com anomalias emergem com clareza, particularmente onde a distinção entre as cores é mais definida. Por exemplo, no gráfico que correlaciona 'Tool wear' e 'Torque', a predominância de pontos vermelhos em valores altos de torque pode insinuar uma relação direta entre o desgaste excessivo da ferramenta e a incidência de falhas. Além disso, áreas com baixa densidade de pontos podem sinalizar anomalias ou condições operacionais atípicas, servindo como pontos de partida para investigações futuras com o intuito de mitigar falhas inesperadas.

2.2. Técnicas Para Geração, Criação, Visualização e Análise nos Modelos de RNA

Na manutenção preditiva, é comum que as ocorrências de falhas sejam muito menos frequentes que o funcionamento normal. Isso pode resultar em modelos que não performam adequadamente na detecção de classes críticas, tais como 'Target' (indicando falha ou não) e 'Failure Type' (tipos específicos de falha). A predominância de dados de operação normal pode levar a uma positiva negligência das falhas, ou desbalanceamento dos dados, resultando em predições imprecisas e potencialmente custosas.

Em ambientes onde dados desbalanceados são uma preocupação central, como na manutenção preditiva, a biblioteca 'imblearn' (Imbalanced-learn) se apresenta como uma solução essencial, fornecendo métodos para abordar a desigualdade na representação de classes nos dados. Paralelamente, as Redes Neurais Artificiais (RNAs), que são compostas por neurônios artificiais interconectados, requerem técnicas avançadas de análise para avaliar seu desempenho, particularmente em conjuntos de dados desbalanceados. Isso envolve a utilização de gráficos para visualizar métricas como perda (‘loss’) e acurácia (‘accuracy’), essenciais para ajustar e otimizar os modelos. A junção destas técnicas de balanceamento de dados com RNAs e métodos rigorosos de avaliação forma uma abordagem robusta para lidar com desafios em áreas críticas que dependem de dados equilibrados e modelos de predição precisos. A seguir descreve-se a metodologia empregadas para a determinação dos resultados encontrados.

2.2.1. Técnica Para Geração De Dados Sintéticos

Para superar os desafios associados a conjuntos de dados desbalanceados, particularmente em domínios críticos como a manutenção preditiva, a biblioteca ‘imblearn’ (Imbalanced-learn) se torna uma ferramenta essencial. Esta biblioteca fornece uma gama de técnicas eficazes para reequilibrar as classes dentro de um conjunto de dados, assegurando que os modelos de aprendizado de máquina não sejam enviesados e possam identificar corretamente eventos raros, como falhas de máquinas.

Para aliviar isso, a técnica ‘SVMSMOTE’ da ‘imblearn’ é particularmente útil. Esta abordagem de superamostragem (‘oversampling’- no contexto de aprendizado de máquina, é uma técnica usada para corrigir desequilíbrios em conjuntos de dados), que incorpora elementos do SVM (Support Vector Machine) para melhorar a seleção dos exemplos minoritários, permite a criação de novos exemplos sintéticos da classe minoritária. Isso ajuda a equilibrar o conjunto de dados, facilitando o treinamento de modelos que são mais generalizáveis e confiáveis na previsão de falhas. Além de aumentar a quantidade de exemplos da classe minoritária, ‘SVMSMOTE’ considera a qualidade dos exemplos gerados, utilizando a mecânica do SVM para identificar e replicar os casos mais desafiadores e, por conseguinte, mais informativos.

O uso de ‘SVMSMOTE’ e outras técnicas de reamostragem disponíveis na biblioteca ‘imblearn’ são passos fundamentais no pré-processamento de dados para a manutenção preditiva, permitindo a construção de algoritmos robustos que podem prever com precisão as falhas e, assim, contribuir para a eficiência operacional e redução de tempo de inatividade nas indústrias.

O código desenvolvido implementa um procedimento iterativo para ajustar a estratégia de superamostragem (dados novos gerados, ou seja, sintéticos) usando a técnica SVMSMOTE da biblioteca ‘imblearn’. A Tabela 1 mostra os parâmetros utilizados no código com o objetivo de equilibrar um conjunto de dados desbalanceados para um problema de manutenção preditiva. Ele define uma função ‘get_statistics_difference’ para medir a diferença estatística entre os dados originais e superamostrados, excluindo a coluna 'Failure Type' para focar em comparações numéricas. O algoritmo inicia com um ‘sampling_strategy’ de 0.1 e aumenta esse valor incrementalmente até que a média e o desvio padrão dos dados originais e superamostrados estejam dentro de uma tolerância pré-estabelecida, ou até que um número máximo de iterações seja atingido. Se a condição de tolerância for atendida, ele imprime o ‘sampling_strategy’ ajustado; caso contrário, notifica que o ajuste ideal não foi encontrado após o número máximo de iterações.

Tabela 1. Parâmetros utilizados no código e suas respectivas descrições.

Parâmetro	Valor	Descrição
‘sampling_strategy_init’	0.1	Proporção inicial da classe minoritária após o superamostramento.
‘increment_rate’	0.05	Taxa de incremento para ajustar o ‘sampling_strategy’ a cada iteração.
‘tolerance’	0.08	Limite de tolerância para as diferenças de médias e desvios padrão entre os dados originais e sintéticos.
‘max_iterations’	100	Número máximo de iterações para encontrar um ‘sampling_strategy’ adequado.
‘random_state’	42	Semente para reprodutibilidade dos resultados.
‘k_neighbors’	1	Número de vizinhos mais próximos para construir exemplos sintéticos.
‘m_neighbors’	1	Número de vizinhos mais próximos usados na técnica SVM do ‘SVMSMOTE’.
‘factor_ampliador’	0.12	Fator de ampliação para ajuste final do ‘sampling_strategy’ em conjuntos de dados muito dispersos.

Estes parâmetros são configurados para otimizar o processo de superamostramento com o objetivo de tornar os dados de treinamento mais equilibrados, o que é crucial para o desempenho do modelo de aprendizado de máquina em tarefas de classificação com conjuntos de dados desbalanceado (https://www.tensorflow.org; https://keras.io; https://numpy.org).

2.2.2. Técnica Criação de Rede Neural Artificial

Uma Rede Neural Artificial (RNA) é um paradigma de processamento de informação que se inspira na maneira como os sistemas nervosos biológicos, como o cérebro, processam a informação.

Consiste em um conjunto de unidades de processamento - frequentemente denominadas neurônios artificiais - organizadas em múltiplas camadas. Estas unidades funcionam de maneira paralela e são estruturadas de acordo com um modelo matemático projetado para emular as propriedades de neurônios biológicos. As RNAs são caracterizadas pela presença de conexões ponderadas entre os neurônios artificiais, que são ajustadas durante o processo de aprendizado. Este ajuste é feito através de um mecanismo conhecido como algoritmo de aprendizado, que modifica os pesos sinápticos da rede com base nos dados de entrada e na técnica de aprendizado utilizada, como o aprendizado supervisionado ou não supervisionado.

Uma arquitetura comum dentro das RNAs é a rede neural de tipo sequencial, que é composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada neurônio nas camadas está conectado aos neurônios da camada subsequente por meio de conexões ponderadas. Essas conexões não possuem realimentação nem conexões entre as mesmas camadas. A informação se propaga através da rede de maneira sequencial, da entrada para a saída. Em cada neurônio, ocorre um cálculo que envolve a soma ponderada dos valores de entrada, seguida pela aplicação de uma função de ativação, geralmente não linear, como a função sigmoide ou ReLU (Unidade Linear Retificada). O objetivo do treinamento de uma RNA é otimizar os pesos sinápticos de tal forma que a rede possa realizar tarefas específicas, como classificação ou regressão, com alta precisão. Este processo é realizado através de métodos como o ‘backpropagation’, que ajusta os pesos de maneira iterativa com base no erro entre a saída prevista pela rede e a saída desejada. (CHENG, Yiwei et al. , 2018.)

No código criado tem-se dois modelos de Redes Neurais gerados pela biblioteca TensorFlow: o 'Modelo Target', com quatro camadas, e o 'Modelo Failure Type', com seis camadas. Ambos processam 5 características de entrada, utilizam regularização L2 e ‘Dropout’ para prevenir overfitting, e são treinados com diferentes números de épocas e tamanhos de lote. A performance é medida por acurácia. A seguir na Tabela 2 uma visão geral dos principais parâmetros e configurações usadas para treinar os dois modelos de redes neurais, destacando as principais características e diferenças entre eles (https://www.tensorflow.org; https://keras.io; https://numpy.org; https://scikit-learn.org).

Tabela 2 - Parâmetros usados para treinar os dois modelos de redes neurais.

Parâmetro/Modelo	Modelo 'Target'	Modelo 'Failure Type'
Dimensão de Entrada	5 (features)	5 (features)
Dimensão de Saída	2 (classes)	6 (classes)
Camadas e Neurônios	4 camadas ocultas (128, 64, 32 neurônios)	6 camadas ocultas (512, 256, 128, 64, 32 neurônios)
Regularização (‘L2 Strength’)	0.001	0.002
‘Dropout	0.4 em cada camada oculta	0.3 em cada camada oculta
Função de Ativação	ReLU (camadas ocultas), ‘Softmax’ (saída)	ReLU (camadas ocultas), ‘Softmax’ (saída)
Distribuição dos Dados (Treino/Validação)	65% / 35%	65% / 35%
Número de Épocas de Treinamento	240	360

Tamanho do ‘Batch’	32	64
Taxa de Aprendizado	0.001	0.001
‘Early Stopping’	Não aplicado	Não aplicado
Métrica de Avaliação	Acurácia	Acurácia
Desempenho no Conjunto de Validação	0.95	0.85

2.2.3. Técnicas de Visualização Para os Modelos de RNA

Foram criados códigos que empregam uma variedade de técnicas avançadas para avaliar e verificar o desempenho de modelos de Redes Neurais Artificiais (RNAs). Primeiramente, são utilizados gráficos para visualizar a perda (‘loss’) e a acurácia (‘accuracy’) ao longo das épocas de treinamento e validação para os modelos 'Target' e 'Failure Type'. Esses gráficos são cruciais para compreender o aprendizado dos modelos ao longo do tempo, identificar overfitting (através da comparação entre perda de treinamento e validação) e avaliar a generalização (através da comparação entre acurácias de treinamento e validação).

A matriz de confusão é uma ferramenta essencial no campo do aprendizado de máquina, usada para avaliar de maneira detalhada o desempenho de algoritmos de classificação. Funcionando como uma matriz, ela permite uma visualização clara e estruturada da performance de um modelo supervisionado, sendo particularmente valiosa em contextos de classificação multiclasse. Essa matriz facilita a comparação entre os valores reais, ou observados, e os valores previstos pelo modelo, permitindo assim uma análise tanto das previsões corretas quanto dos diferentes tipos de erros que o modelo pode estar cometendo. Através dela, é possível determinar os verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos para cada classe.

Em uma matriz de confusão para uma classificação, as colunas representam as previsões do modelo, enquanto as linhas representam as classes reais. A seguir o Quadro 1 mostrando um esquema didático de uma matriz de confusão para um problema de classificação.

Quadro 1. Dados de uma matriz de confusão.

	Previsão Positiva	Previsão Negativa
Classe Real Positiva	Verdadeiros Positivos (VP)	Falsos Negativos (FN)
Classe Real Negativa	Falsos Positivos (FP)	Verdadeiros Negativos (VN)

- Verdadeiros Positivos (VP): Casos em que o modelo previu corretamente a classe positiva.
- Falsos Positivos (FP): Casos em que o modelo previu incorretamente a classe negativa como positiva.
- Verdadeiros Negativos (VN): Casos em que o modelo previu corretamente a classe negativa.
- Falsos Negativos (FN): Casos em que o modelo previu incorretamente a classe positiva como negativa.

Para problemas que envolvem múltiplas classes, a matriz se expande, incluindo todas as classes ao longo da diagonal principal, onde são registradas as previsões corretas, enquanto as células fora da diagonal representam as previsões incorretas. Essa expansão facilita a visualização de como

o modelo se comporta em relação a cada classe específica, revelando padrões entre classes e potenciais vieses. Adicionalmente, outras métricas como precisão, recall e F1-score podem ser derivadas da matriz de confusão que serão descritas abaixo.

Assim, o código inclui o uso do 'classification_report' da biblioteca 'scikit-learn', que fornece um relatório detalhado das métricas de classificação, dadas pelos itens a seguir 1, 2 e 3 que são fundamentais para avaliar os modelos, assim essas métricas são dadas por:

1. Precisão ('Precision'):
    - Definição: A precisão é a proporção de previsões corretas de uma classe em relação a todas as previsões feitas para essa classe.
    - Fórmula: $Precisão = \frac{Verdadeiros\ Positivos\ (VP)}{Verdadeiros\ Positivos\ (VP) + Falsos\ Positivos\ (FP)}$
    - Interpretação: Uma precisão alta indica que a maioria das instâncias classificadas como pertencentes a uma classe realmente pertencem a essa classe.
2. Recall (Sensibilidade ou Taxa de Verdadeiros Positivos):
    - Definição: O recall é a proporção de previsões corretas de uma classe em relação ao total de instâncias que realmente pertencem a essa classe.
    - Fórmula: $Recall = \frac{Verdadeiros\ Positivos\ (VP)}{Verdadeiros\ Positivos\ (VP) + Falsos\ Negativos\ (FN)}$
    - Interpretação: Um recall alto indica que o modelo é capaz de identificar a maioria das instâncias dessa classe.
3. F1-score:
    - Definição: O F1-score é a média harmônica entre precisão e recall. É uma medida que busca um equilíbrio entre precisão e recall.
    - Fórmula: $\mathrm{F1_{score}} = 2\mathrm{x}\ \frac{(Precisão \times Recall)}{(Precisão + Recall)}$
    - Interpretação: O F1-score é particularmente útil em situações onde a precisão e o recall são igualmente importantes, ou em casos de desequilíbrio de classe, onde a precisão ou o recall isoladamente podem não refletir adequadamente o desempenho do modelo.

Quando avaliamos essas métricas para cada classe, estamos interessados em saber como o modelo de classificação performa especificamente para cada categoria. Isso é crucial em problemas de classificação multiclasse, especialmente quando algumas classes podem ser mais representativas (comuns) do que outras. A análise dessas métricas por classe ajuda a identificar se o modelo está tendencioso para certas classes ou se está performando de maneira equilibrada em todas as classes.

Essas técnicas, em conjunto, proporcionam uma avaliação abrangente dos modelos de RNAs, permitindo identificar pontos fortes e oportunidades de melhoria. São métodos indispensáveis no processo de desenvolvimento e validação de modelos em aprendizado de máquina, garantindo uma análise profunda e detalhada do desempenho dos modelos (https://scikit-learn.org; https://pandas.pydata.org; https://seaborn.pydata.org; https://matplotlib.org).

2.2.4. Bibliografia Das Técnicas Metodológicas Utilizadas.

As técnicas descritas para geração, criação, visualização e análise/avaliação do modelo são encontradas nos seguintes locais: https://www.tensorflow.org. Keras, uma API de alto nível para redes neurais, pode ser acessada em https://keras.io. NumPy, essencial para computação científica em Python, está disponível em https://numpy.org. Scikit-learn, uma ferramenta para análise de dados e aprendizado de máquina, pode ser encontrada em https://scikit-learn.org. Pandas, ideal para manipulação e análise de dados, está disponível em https://pandas.pydata.org. Seaborn, uma biblioteca de visualização de dados estatísticos, pode ser acessada em https://seaborn.pydata.org. Por fim, Matplotlib, uma biblioteca de plotagem 2D, está disponível em https://matplotlib.org. Esses sites são recursos para documentação, tutoriais e guias de uso dessas ferramentas.

3. Resultados e Discussão

Os resultados são mostrados a seguir onde a aplicação da técnica de superamostragem (‘oversampling’), da biblioteca ‘imblearn’ foi fundamental para balancear os conjuntos de dados desbalanceados. Essa técnica gerou exemplos sintéticos da classe minoritária, permitindo que os modelos de rede neural 'Target' e 'Failure Type' fossem treinados de forma mais eficaz. Gráficos de validação, matrizes de confusão, foram utilizados para visualizar e compreender os modelos. Adicionalmente, a avaliação dos modelos incluiu a análise de métricas chaves como acurácia, precisão e recall, fornecendo insights cruciais sobre o desempenho dos modelos.

3.1. Análise do Conjunto de Dados Sintéticos

Os dados gerados pelo ‘SVMSMOTE’ com foco no balanceamento entre conjuntos de dados originais e sintéticos são demostrados a seguir. As figuras representam histogramas que comparam a distribuição de eventos de falha e não falha entre conjuntos de dados originais e sintéticos, com o propósito de ilustrar as diferenças entre as frequências de falhas. No primeiro gráfico, Figura 4, os dados originais e sintéticos apresentam uma correspondência numérica para a categoria 'No Failure', ambas com 9652 instâncias. Por outro lado, na categoria 'Failure', observa-se um aumento no conjunto sintético para 1105 ocorrências, em comparação com 348 do conjunto original, evidenciando a geração de exemplos adicionais de falhas para atingir um balanceamento de classes. O segundo histograma, Figura 5, detalha a frequência dos diferentes 'Types of Failures'. Nos dados originais, 'Power Failure' e 'Heat Dissipation Failure' são as categorias mais prevalentes, enquanto o conjunto sintético destaca um aumento em todas as categorias de falhas, com uma sobre-representação de 'Overstrain Failure' e 'Heat Dissipation Failure'. Portanto, o aumento proporcional em cada categoria de falha no conjunto sintético visa aprimorar a robustez dos modelos de machine learning ao proporcionar uma distribuição de classe mais equilibrada, essencial para a eficácia do treinamento e da capacidade em cenários de manutenção preditiva.

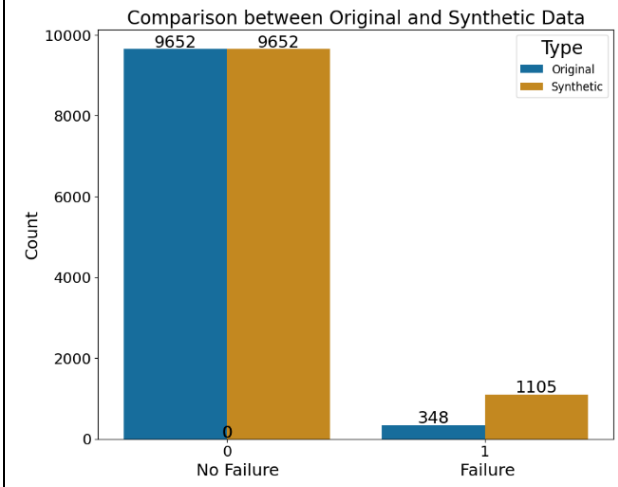

Figura 4 - Histograma que compara a distribuição, onde 0 representa 'No Failure' e 1 representa 'Failure', entre conjuntos de dados originais e sintéticos.

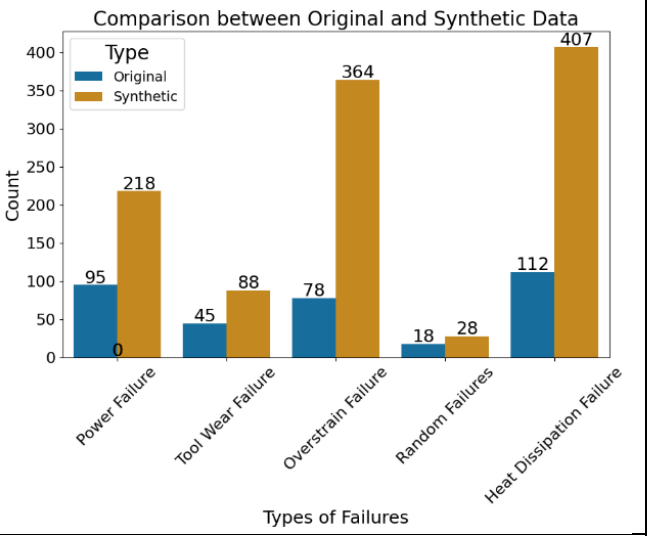

Figura 5- Histograma comparativo entre a frequência de diferentes tipos de falhas em conjuntos de dados originais e sintéticos.

No código gerado na biblioteca ‘imblearn’ com a técnica ‘SVMSMOTE’ a introdução de um 'fator de ampliador' abordou desafios em conjuntos de dados complexos. O ajuste do parâmetro de superamostragem é um processo iterativo inovador para manter a integridade dos dados. Multiplicar o 'sampling_strategy' em conjuntos de dados com alta dispersão é uma estratégia que melhora a representatividade das áreas menos densas, criando mais exemplos sintéticos e potencialmente otimizando a performance geral do modelo de machine learning. Isso é particularmente útil em cenários onde o desbalanceamento de classe pode levar a um viés no desempenho do modelo, favorecendo classes majoritárias. Além disso, um 'sampling_strategy' aumentado pode permitir uma exploração mais abrangente do espaço de recursos, cobrindo uma gama mais ampla de possíveis ocorrências, ao mesmo tempo que reduz o risco de ‘overfitting’ nas classes minoritárias e compensa a perda de informações que podem ocorrer devido à variação nos dados. Cada um desses pontos contribui para uma justificativa plausível e sólida para ajustar o 'sampling_strategy' em resposta à dispersão dos dados.

A funcionalidade do código para comparar estatísticas e assegurar que os dados sintéticos gerados pelo ‘SVMSMOTE’ fossem comparáveis aos originais, mas não podem ser iguais é fundamental. Desta forma, os gráficos, Figura 6, apresentam uma série de comparações entre as distribuições das variáveis de um conjunto de dados original e um conjunto sintético, separados por tipos de falha: ‘Power Failure’, ‘Tool Wear Failure’, ‘Overstrain Failure’, ‘Random Failures’ e ‘Heat Dissipation Failure’. Para cada tipo de falha, são exibidas comparações das distribuições de 'Air temperature', 'Process temperature', 'Rotational speed', 'Torque' e 'Tool wear'. As curvas de densidade azuis representam os dados originais, enquanto as curvas vermelhas representam os dados sintéticos. Estas curvas são úteis para visualizar a distribuição e a frequência de diferentes métricas de manutenção. A sobreposição das curvas entre os dados originais e sintéticos sugere que os dados sintéticos foram gerados com sucesso para imitar a distribuição estatística das variáveis originais, o que é importante para treinamento e validação de modelos preditivos em situações em que dados reais podem ser escassos ou para melhorar o equilíbrio de classe em conjuntos de dados desequilibrados. A análise destas curvas permite aos analistas e engenheiros identificarem discrepâncias e avaliar a qualidade da ‘superamostragem’ realizada, bem como entender melhor as condições de funcionamento e os possíveis pontos de falha do sistema estudado.

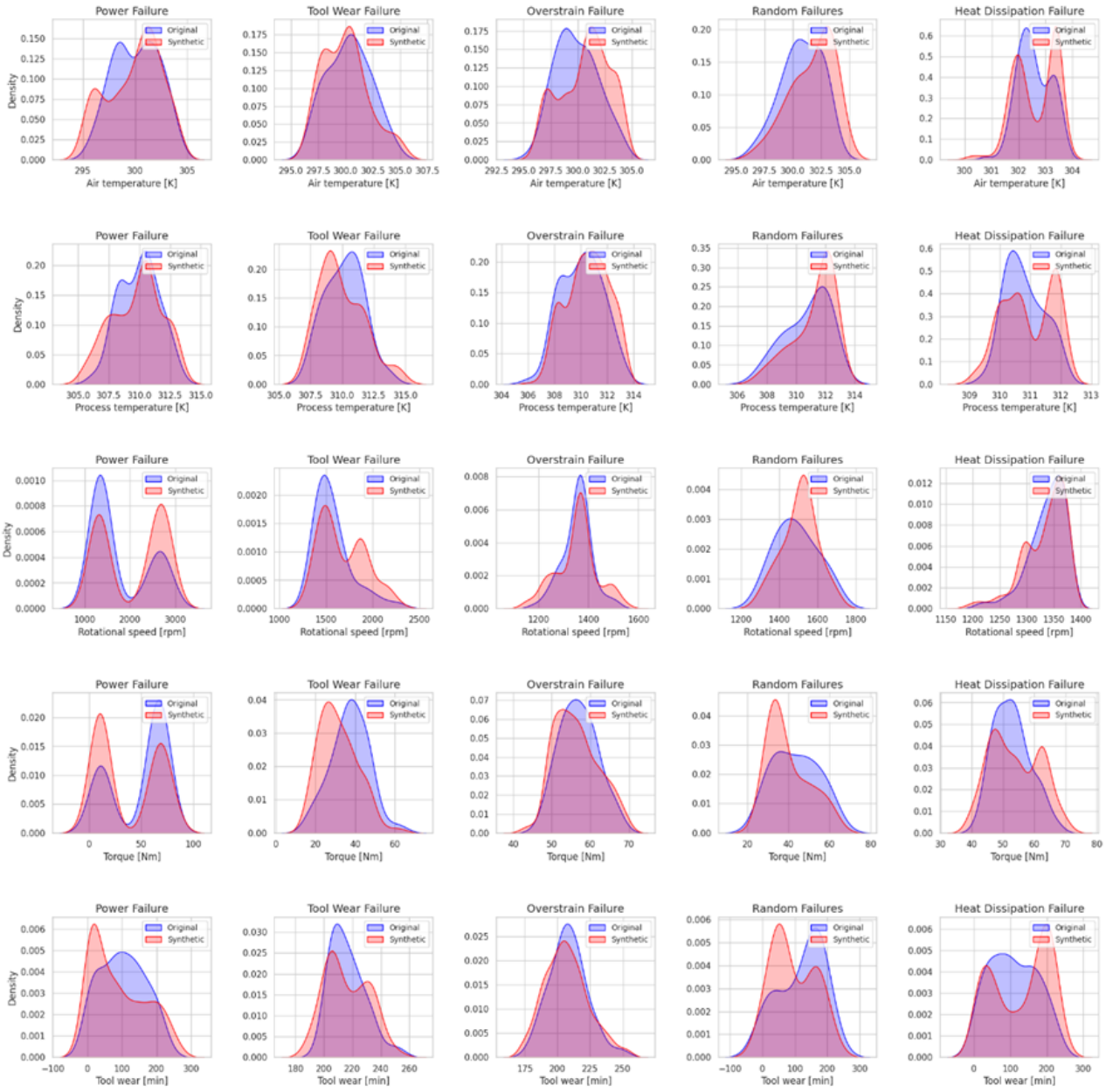

Figura 6- Gráfico que apresentam uma série de comparações entre as distribuições das variáveis de um conjunto de dados original e um conjunto sintético, separados por tipos de falha.

Os ‘heatmaps’ Figura 7 apresentados são instrumentos analíticos que exibem as diferenças nas medidas de correlação de ‘Spearman’ e ‘Kendall’ entre conjuntos de dados originais e sintéticos, possibilitando uma avaliação detalhada da fidelidade com que as relações estatísticas intrínsecas são mantidas após a geração de dados sintéticos. No ‘heatmap’ da correlação de ‘Spearman’, observa-se que a maioria das diferenças é marginal, indicando que a estrutura de dependência linear ou monotônica entre as variáveis é predominantemente preservada no conjunto sintetizado. Similarmente, o ‘heatmap’ da correlação de ‘Kendall’ revela que as discrepâncias são mínimas, reforçando a eficácia da metodologia de ‘superamostragem’ em replicar a hierarquia e a concordância dos rankings entre as variáveis.

Diferenças mais acentuadas em algumas células dos ‘heatmaps’ podem sinalizar alterações específicas na relação entre determinadas variáveis, talvez devido a peculiaridades do processo de

‘superamostragem’ dos dados originais. Tais variações, mesmo que pequenas, são cruciais, pois podem influenciar a performance e interpretação de modelos em aplicações como na manutenção preditiva e diagnóstico de falhas. Assim, essas análises não somente validam a aplicação da técnica de geração de dados sintéticos usada, mas também orientam uma avaliação onde ajustes podem ser necessários para otimizar a integridade dos dados usados para treinar modelos de machine learning.

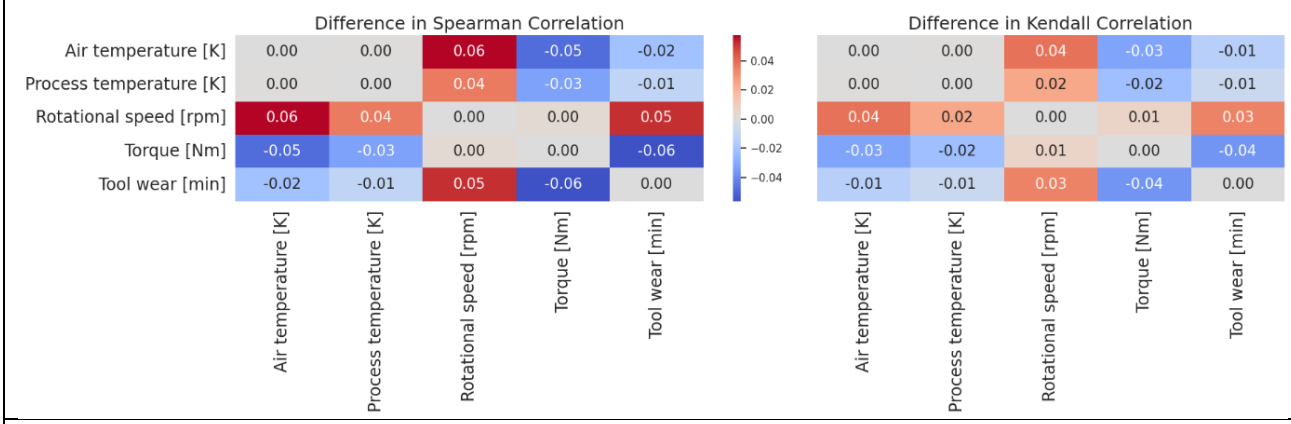

Figura 7 - ‘Heatmap’ mostrando a diferença nas correlações de ‘Spearman’ e ‘Kendall’ entre dois conjuntos de dados: original e sintético.

O gráfico, Figura 8, descreve uma série de gráficos, cada um representando estatísticas descritivas (como média, desvio padrão, mínimos, percentis e máximos) para várias variáveis medidas (como 'Air temperature', 'Process temperature', 'Rotational speed', 'Torque' e 'Tool wear'), com uma comparação lado a lado das métricas para conjuntos de dados originais e sintéticos. Uma análise possível de uma matriz de gráfico como, da Figura 8, é avaliar o quão bem os dados sintéticos correspondem aos dados originais em termos de tendências centrais e dispersão. Se as médias são muito próximas entre os dois conjuntos de dados, isso pode indicar que os dados sintéticos têm uma tendência central semelhante aos dados originais. Diferenças significativas no desvio padrão ou nos valores mínimos e máximos podem sugerir que a variação ou a amplitude dos dados sintéticos não captura completamente a complexidade dos dados originais. Percentis podem dar uma ideia sobre a distribuição dos dados em diferentes pontos, permitindo uma comparação mais detalhada da distribuição de dados.

Para a 'Air temperature', a variação entre as médias é de apenas 0.10 K, com uma ligeira alteração no desvio padrão de 2.00 para 2.05 K. Por exemplo, para a 'Air temperature', a média nos dados originais é 300.00 K e nos sintéticos é 300.10 K, indicando uma ligeira variação. A 'Process temperature' mostra uma mudança ainda menor, com uma variação de 0.01 K na média e uma leve diferença no desvio padrão, indo de 1.48 para 1.50 K. Similarmente, o desvio padrão para 'Rotational speed' aumenta de 179.28 rpm para 209.50 rpm nos dados sintéticos, sugerindo uma dispersão maior nas observações sintéticas, ou seja, refletindo uma variação considerável na simulação. As comparações percentis, como o 50% (mediana), mantêm-se constantes em muitas variáveis, como a 'Process temperature', mas mostram variações em outras, como o aumento do 75% para 'Torque' de 46.80 Nm para 47.80 Nm. O 'Torque' tem diferenças nas estatísticas, uma alta variação no valor mínimo de 3.80 para 3.01 Nm, e variação mínima no 25% e 75% percentil. Para 'Tool wear', a média e mediana tem pequenas variações em ambos os conjuntos de dados, por exemplo há uma ligeira elevação no máximo de 253.00 para 257.29 min nos dados sintéticos para o 75% percentil.

Desta forma, este gráfico (Figura 8) é valioso para validar se os dados sintéticos são uma boa representação dos dados originais e podem ser usados para treinar modelos de machine learning com confiança ou se ajustes adicionais são necessários para melhorar a qualidade dos dados sintéticos.

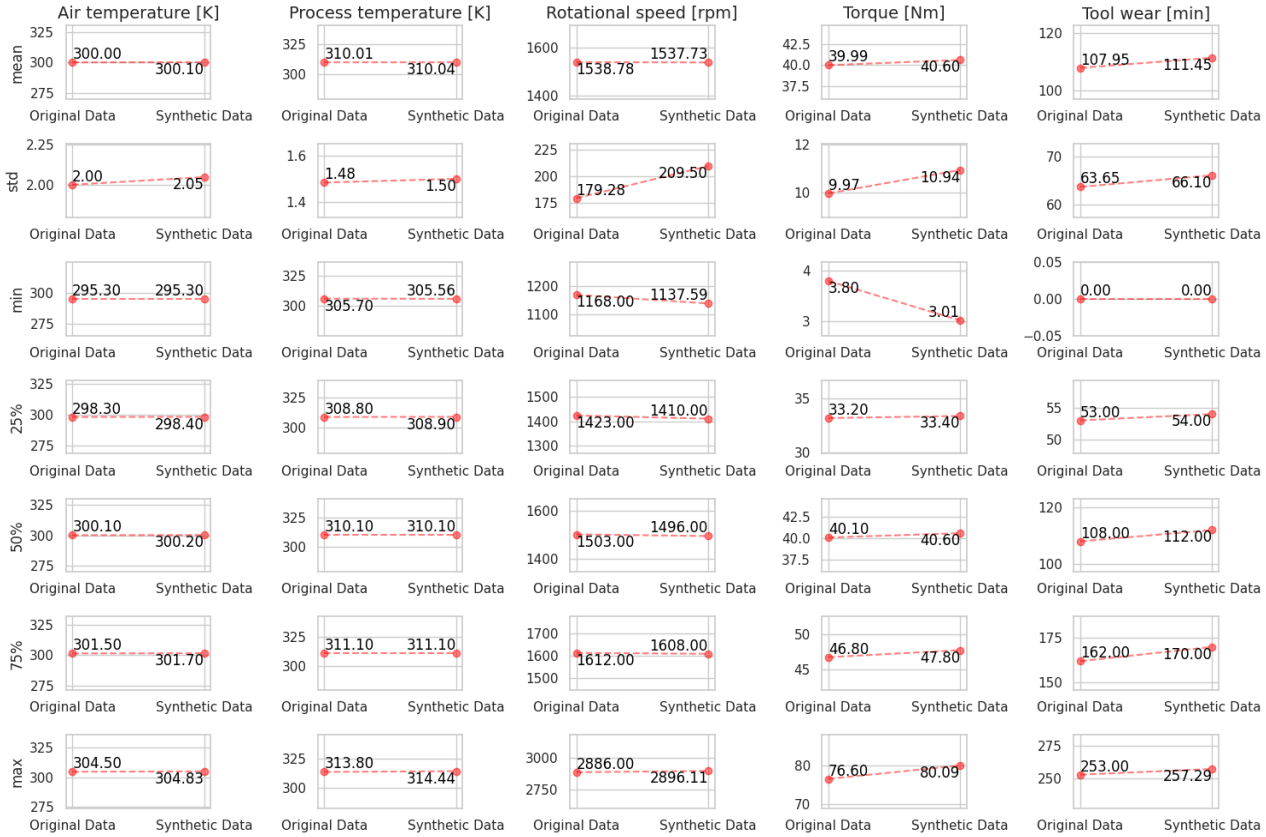

Figura 8 - Uma série de gráficos, cada um representando estatísticas descritivas (média, desvio padrão, mínimos, percentis e máximos) para várias variáveis medidas (como 'Air temperature', 'Process temperature', 'Rotational speed', 'Torque' e 'Tool wear').

Essas diferenças quantitativas podem ser consideradas pequenas, mas são significativas o suficiente para destacar a influência do processo de geração de dados sintéticos na distribuição estatística das variáveis analisadas. Esses exemplos mostram que as estatísticas dos dados sintéticos são bastante próximas às dos dados originais, com exceção do 'Rotational speed', que exibe uma maior dispersão nos dados sintéticos como mostrado no desvio padrão. Isso sugere que, embora o processo de geração de dados sintéticos tenha sido amplamente eficaz em imitar as características estatísticas dos dados originais, algumas discrepâncias podem necessitar atenção adicional, dependendo do contexto de uso desses dados.

3.2. Análise Técnica da Rede Neural Artificial

O código das redes neurais para os modelos 'Target' e 'Failure Type' ilustra uma metodologia sofisticada na construção e avaliação de modelos de aprendizado de máquina. Aspectos como a utilização de pesos de classe para balancear classes desiguais e a implementação de regularização para aprimorar a generalização dos modelos são práticas adotadas. A segmentação dos dados em conjuntos de treinamento e validação, juntamente com a avaliação da acurácia, fornece uma compreensão clara da habilidade dos modelos em prever a ocorrência e o tipo de falhas, tanto em dados reais quanto sintéticos. O código descreve detalhadamente a construção, o treinamento e a avaliação dos modelos de redes neurais utilizando a API 'Keras' do 'TensorFlow'. Estas redes são especificamente projetadas para classificar dados relacionados a falhas em máquinas, com base em características como 'Air temperature', 'Process temperature', 'Rotational speed', 'Torque' e 'Tool wear'.

A acurácia obtida nos conjuntos de validação serve para indicar o desempenho dos modelos e de sua capacidade de generalização. Após o treinamento, o modelo 'Target' alcançou uma acurácia de 0.95 no conjunto de validação, enquanto o modelo 'Failure Type' atingiu uma acurácia de 0.85. Esses valores de acurácia demonstram o quão eficientemente cada modelo conseguiu generalizar a partir dos dados de treinamento para prever as classes correspondentes nos dados de validação. Portanto, conclui-se que os modelos foram adequadamente regularizados e otimizados para lidar com desequilíbrios de classe e minimizar o 'overfitting', beneficiando-se de um número significativo de épocas de treinamento.

As Figuras 9 e 10 apresentam, respectivamente, a perda (‘loss’) e a acurácia (‘accuracy’) de treinamento e validação ao longo das épocas para um modelo preditivo da variável 'Target'. Na Figura 9, observa-se uma rápida diminuição na perda de treinamento nas fases iniciais, estabilizando-se posteriormente e mantendo uma proximidade com a perda de validação, o que indica uma boa generalização sem sinais evidentes de ‘overfitting’. A oscilação na perda de validação é um comportamento comum em dados complexos, refletindo o esforço do modelo em captar nuances dos dados. A Figura 10 mostra que a acurácia de treinamento e validação aumenta rapidamente e se estabiliza, mantendo-se próximas ao longo do treinamento, o que sinaliza um aprendizado consistente e uma capacidade de generalização eficaz do modelo. A falta de divergência significativa entre as acurácias de treinamento e validação ao longo das épocas reforça a ideia de que o modelo não está excessivamente ajustado aos dados de treinamento.

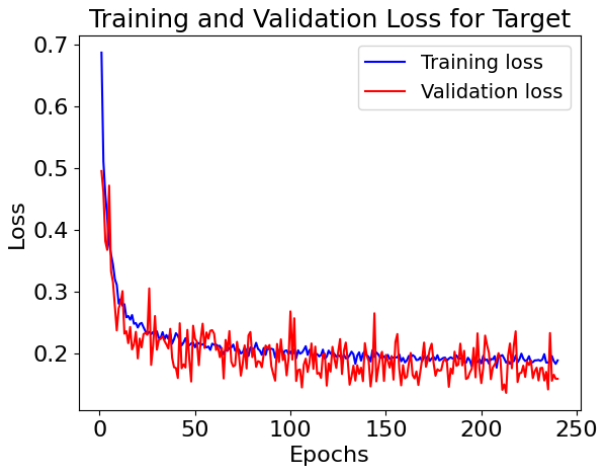

Figura 9 - Gráfico de linha que mostra a perda de treinamento (em azul) e a perda de validação (em vermelho) para variável ‘Target’.

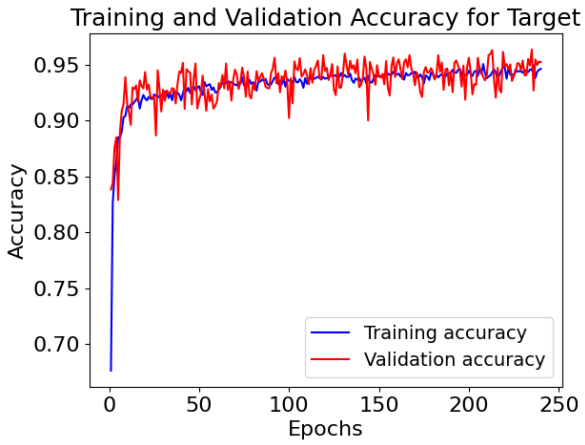

Figura 10 - Gráfico mostrando a acurácia de treinamento (em azul) e a acurácia de validação (em vermelho) para variável ‘Target’.

Na Figura 11, a evolução da perda de treinamento e validação para a variável 'Failure Type' ao longo de 350 épocas é apresentada, mostrando uma rápida diminuição da perda no início, estabilizando-se em seguida, o que sugere um aprendizado eficaz e a falta de grandes ajustes nos pesos com o passar das épocas. A perda de validação, contudo, exibe volatilidade e uma lacuna persistente em relação à perda de treinamento, indicando um potencial ‘overfitting’. Na Figura 12, observa-se a acurácia de treinamento e validação, com melhorias iniciais significativas e uma subsequente estabilização, sugerindo que o modelo generaliza de forma razoável, apesar de haver oscilações na acurácia de validação que podem apontar para a vantagem de se estabilizar ainda mais o modelo.

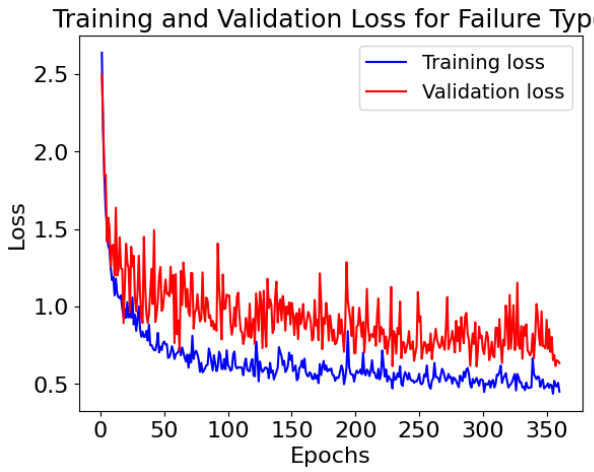

Figura 11 – Gráfico que apresenta a evolução da perda de treinamento (em azul) e da perda de validação (em vermelho) para a variável 'Failure Type'.

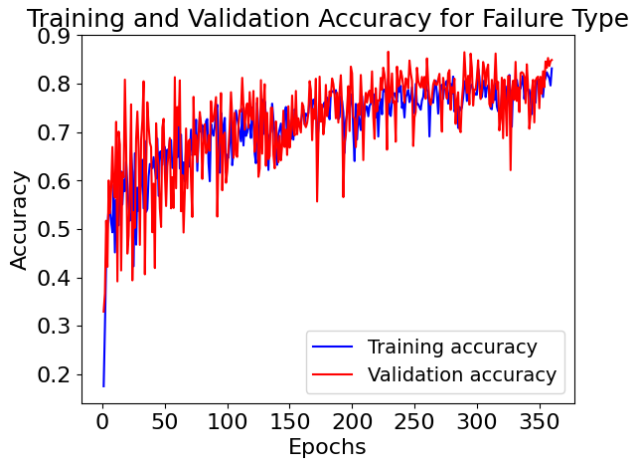

Figura 12 – Gráfico que apresenta a evolução da perda de treinamento (em azul) e da perda de validação (em vermelho) para a variável 'Failure Type'.

A Figura 13 é uma matriz de confusão representando o desempenho de um modelo de classificação para 'Target', com duas classes possíveis: 'No Failure' (classe 0) e 'Failure' (classe 1). A barra de cores à direita serve como uma legenda, indicando a quantidade de instâncias que correspondem a cada quadrante da matriz.

A matriz de confusão, apresenta as previsões de um modelo de classificação para falhas ou não no processo. Os números ao longo da diagonal principal (de canto superior esquerdo para canto inferior direito) representam as quantidades de previsões corretas. Os números fora da diagonal principal indicam as previsões incorretas, onde o modelo confundiu falha com não falha. A matriz apresenta os seguintes valores:

Verdadeiros Positivos para 'No Failure' (3230): O modelo previu corretamente a ausência de falha para 3230 instâncias.

Falsos Negativos para 'No Failure' (168): O modelo previu erroneamente falha quando na verdade era 'No Failure' para 168 instâncias.

Falsos Positivos para 'Failure' (11): O modelo previu 'No Failure' quando na verdade era 'Failure' para 11 instâncias.

Verdadeiros Positivos para 'Failure' (356): O modelo previu corretamente a ocorrência de falha para 356 instâncias.

A matriz da Figura 13 mostra que o modelo tem uma forte capacidade de prever a classe 'No Failure', mas ainda comete alguns erros, classificando 168 instâncias dessa classe como 'Failure'. Por outro lado, o modelo é bastante preciso ao prever a classe 'Failure', com apenas 11 instâncias sendo classificadas erroneamente como 'No Failure'. Isso sugere que o modelo é mais conservador (prioriza a minimização de erros do tipo falsos positivos) ao classificar falhas, com uma tendência a evitar falsos positivos em detrimento de incorrer em mais falsos negativos. A cor mais escura no quadrante superior esquerdo e inferior direito indica uma maior concentração de previsões corretas nessas categorias, o que geralmente é o objetivo em classificação.

A Figura 14 apresenta uma matriz de confusão para um modelo de classificação que prevê diferentes tipos de falhas, ‘Failure Type’. As linhas representam as classes reais e as colunas as classes

previstas pelo modelo. A barra de cores à direita serve como uma legenda para a contagem de instâncias, indicando a frequência de cada resultado na matriz.

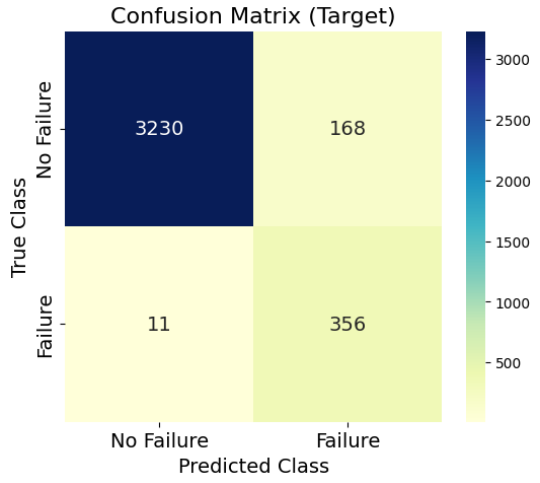

Figura 13 - Matriz de confusão para um modelo de classificação que prediz a ocorrência de falhas ou não no modelo 'Target'.

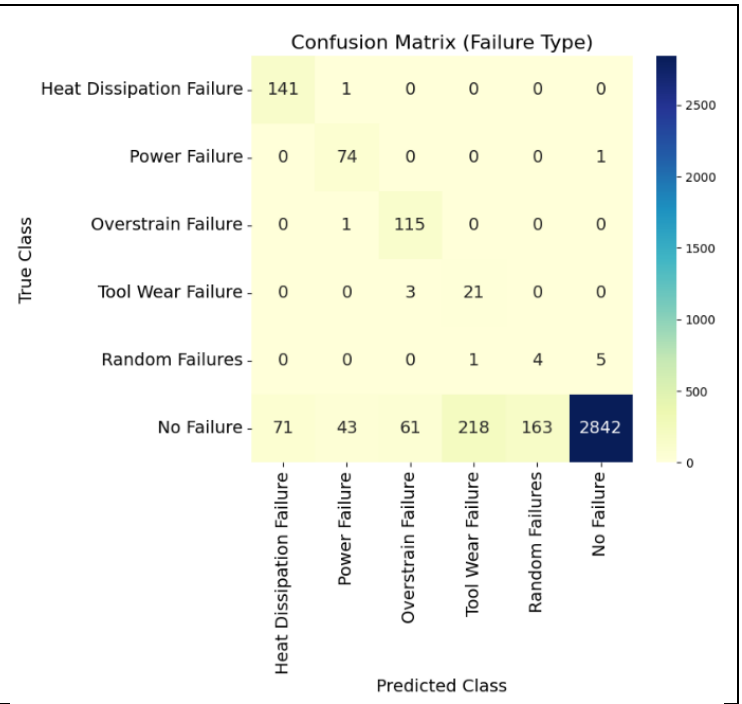

Figura 14 - Matriz de confusão que apresenta as previsões de um modelo de classificação para diferentes tipos de falhas no processo.

Para a Figura 14 a diagonal principal mostra o número de previsões corretas para cada classe, enquanto os outros valores indicam os erros de classificação. Os valores fora da diagonal principal representam os erros de classificação, onde o modelo previu uma classe, mas a verdadeira era outra. Assim, Analisando a matriz temos que:

- ‘Heat Dissipation Failure’ foi corretamente prevista 141 vezes, com apenas uma ocorrência incorretamente prevista como ‘Power Failure’.

- ‘Power Failure’ teve 74 acertos, com um erro classificado como ‘No Failure’ e outro como ‘Heat Dissipation Failure’.

- ‘Overstrain Failure’ foi precisamente identificado 115 vezes, com apenas um caso classificado erroneamente como ‘Power Failure’.

- ‘Tool Wear Failure’ foi corretamente previsto em 21 ocasiões, porém foi confundido com ‘Overstrain Failure’ em 3 casos.

- ‘Random Failures’ teve 4 acertos e 5 ‘No Failure’ e 1 ‘Tool Wear Failure’ que foram erros. Neste item o modelo de classificação teve um desempenho indesejado, possivelmente pela baixa quantidade de dados ‘Random Failures’.

- ‘No Failure’ teve um alto número de previsões corretas (2842), mas também houve significativa confusão com todos os outros tipos de falhas, mostrando a dificuldade do modelo em diferenciar operações normais das diversas falhas.

O número consideravelmente maior na célula inferior direita indica que o modelo é muito eficaz em prever a ausência de falha ("No Failure"), mas ainda confunde essa classe com outros tipos de falha. As outras células com valores baixos fora da diagonal principal sugerem que o modelo tem um bom desempenho na distinção entre os diferentes tipos de falhas.

A precisão é a taxa de acertos do modelo para uma classe, o recall indica quantos casos verdadeiros foram corretamente identificados, e o F1-score é a média harmônica entre precisão e recall. Estas métricas, mostradas na Tabela 3 e 4, evidenciam variações no desempenho do modelo entre diferentes classes de classificação para 'Target' e 'Failure Type'.

Tabela 3 – Resultado para o modelo ‘Target'.

Classe	Precisão	Recall	F1-score	Suporte
‘No Failure’	1.00	0.95	0.97	3398
‘Failure’	0.68	0.97	0.80	367
Acurácia	-	-	0.95	3765
Média Macro	0.84	0.96	0.89	3765
Média Ponderada	0.97	0.95	0.96	3765

Tabela 4 – Resultado para o modelo ‘'Failure Type'.

Classe	Precisão	Recall	F1-score	Suporte
‘Heat Dissipation Failure'	0.67	0.99	0.80	142
‘Power Failure’	0.62	0.99	0.76	75
‘Overstrain Failure’	0.64	1.00	0.78	116
‘Tool Wear Failure’	0.09	0.88	0.16	24
‘Random Failures’	0.02	0.40	0.05	10
‘No Failure’	1.00	0.84	0.91	3398
Acurácia	-	-	0.85	3765
Média Macro	0.51	0.85	0.58	3765
Média Ponderada	0.96	0.85	0.89	3765

Para a Tabela 3, 'Target', a classe ‘No Failure’ tem uma precisão perfeita e um recall alto, resultando em um excelente F1-score, o que significa que o modelo é quase perfeito na identificação desta classe. A classe ‘Failure’, por outro lado, tem uma precisão significativamente menor, mas um recall muito alto, resultando em um F1-score bom, o que indica que, enquanto o modelo é capaz de identificar a maioria das instâncias positivas, ele também faz um número moderado de falsos positivos. A acurácia geral do modelo é de 0.95, o que é bastante alto, e as médias macro e ponderada estão em bom alinhamento com essa acurácia, refletindo um desempenho geral forte do modelo.

Na Tabela 4, 'Failure Type', a distribuição da performance é mais variada entre as classes. As classes ‘Heat Dissipation Failure', ‘Power Failure’ e ‘Overstrain Failure’ têm F1-scores

comparativamente altos, com recall excepcionalmente alto, indicando que o modelo está identificando a maioria das instâncias dessas classes corretamente. No entanto, as classes ‘Tool Wear Failure’ e ‘Random Failures’ têm F1-scores muito baixos devido à sua baixa precisão, apesar de terem recall altos, indicando que o modelo está produzindo um número excessivo de falsos positivos para essas classes. A classe ‘No Failure’ tem uma precisão perfeita, mas o recall é um pouco mais baixo, o que reduz o seu F1-score. A acurácia geral deste modelo é de 0.85, o que é bastante bom, mas as médias macro sugerem uma variabilidade significativa no desempenho entre as classes, enquanto a média ponderada é alta, refletindo a grande influência da classe ‘No Failure’ com maior suporte.

Em ambos os casos, os modelos mostram uma forte capacidade de identificar corretamente as instâncias de cada classe, mas há variações significativas entre as classes, particularmente no modelo 'Failure Type'.

4. Conclusões

a) A aplicação da técnica 'SVMSMOTE' da biblioteca 'imblearn' foi fundamental para o balanceamento dos conjuntos de dados, mostrando eficácia na geração de exemplos sintéticos. Este processo equilibrou as classes dentro do conjunto de dados, onde a categoria 'No Failure' manteve suas 9652 instâncias tanto nos dados originais quanto nos sintéticos. Por outro lado, para a categoria 'Failure', houve um aumento significativo no conjunto sintético, que passou de 348 para 1105 ocorrências, evidenciando a efetiva geração de exemplos adicionais de falhas para atingir um balanceamento de classes.

b) A análise dos tipos de falha nos dados revelou que 'Power Failure' e 'Heat Dissipation Failure' são as categorias mais comuns nos dados originais. No entanto, no conjunto sintético, todas as categorias de falha receberam um aumento na sua representação, com ênfase em 'Overstrain Failure' e 'Heat Dissipation Failure'. Essa sobre-representação em categorias específicas no conjunto sintético visa melhorar a robustez dos modelos de machine learning ao proporcionar uma distribuição de classe mais equilibrada.

c) Comparando variáveis específicas entre os conjuntos de dados originais e sintéticos, observou-se variações estatísticas descritivas (média, mínimos, percentis e máximos) pequenas dos dados de forma geral. Por exemplo, para a 'Air temperature', a média nos dados originais é 300.00 K e nos sintéticos é 300.10 K. Em contrapartida, o desvio padrão para 'Rotational speed' mostrou um aumento de 179.28 rpm para 209.50 rpm nos dados sintéticos, sugerindo uma dispersão maior nas observações sintéticas.

d) O desempenho dos modelos de rede neural foi destacado pela acurácia obtida nos conjuntos de validação. O modelo 'Target' alcançou uma acurácia de 0.95, enquanto o modelo 'Failure Type' atingiu uma acurácia também alta de 0.85. Esses valores indicam que ambos os modelos foram eficientes em generalizar a partir dos dados de treinamento para prever as classes correspondentes nos dados de validação.

e) A matriz de confusão para o modelo 'Target' revelou uma forte capacidade do modelo em prever a classe 'No Failure', com 3230 instâncias previstas corretamente. No entanto, o modelo cometeu alguns erros, classificando incorretamente 168 instâncias dessa classe como 'Failure'.

Para a classe 'Failure', o modelo teve um bom desempenho, com apenas 11 instâncias classificadas erroneamente como 'No Failure', indicando uma tendência do modelo em minimizar erros do tipo falsos positivos.

f) Na análise da matriz de confusão para 'Failure Type', observou-se um desempenho variado do modelo entre diferentes tipos de falhas. 'Heat Dissipation Failure' foi corretamente prevista 141 vezes, e 'Power Failure' teve 74 acertos. 'Overstrain Failure' e 'Tool Wear Failure' também mostraram bons resultados, enquanto 'Random Failures' apresentou um desempenho mais fraco. O modelo foi particularmente eficaz em prever a ausência de falha ('No Failure'), mas ainda confundiu essa classe com outros tipos de falha.

g) As métricas de desempenho para o modelo 'Target' mostraram resultados excelentes para a classe 'No Failure', com uma precisão de 1.00 e um recall de 0.95, resultando em um F1-score de 0.97. A classe 'Failure', embora com uma precisão mais baixa de 0.68, teve um recall alto de 0.97, levando a um F1-score de 0.80. A acurácia geral do modelo foi de 0.95, indicando um desempenho geral forte.

h) Por fim, a análise da Tabela 4 para o modelo 'Failure Type' revelou uma distribuição mais variada de desempenho entre as classes. Enquanto 'Heat Dissipation Failure', 'Power Failure' e 'Overstrain Failure' tiveram F1-scores altos, as classes 'Tool Wear Failure' e 'Random Failures' apresentaram F1-scores muito baixos, devido principalmente à baixa precisão. A classe 'No Failure' teve uma precisão perfeita, mas com um recall um pouco mais baixo.

5. Referências Bibliográficas

https://numpy.org.

ABIDI, Mustufa Haider; MOHAMMED, Muneer Khan; ALKHALEFAH, Hisham. Predictive maintenance planning for industry 4.0 using machine learning for sustainable manufacturing. Sustainability, v. 14, n. 6, p. 3387, 2022.

AHMED, Imran; JEON, Gwanggil; PICCIALLI, Francesco. From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. IEEE Transactions on Industrial Informatics, v. 18, n. 8, p. 5031-5042, 2022.

CHENG, Yiwei et al. Machine health monitoring using adaptive kernel spectral clustering and deep long short-term memory recurrent neural networks. IEEE Transactions on Industrial Informatics, v. 15, n. 2, p. 987-997, 2018.

FORESTI, Ruben et al. Smart society and artificial intelligence: big data scheduling and the global standard method applied to smart maintenance. Engineering, v. 6, n. 7, p. 835-846, 2020.

https://keras.io.

https://matplotlib.org.

https://pandas.pydata.org.

https://scikit-learn.org.

https://seaborn.pydata.org.

https://www.tensorflow.org.

LEE, Jay et al. Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. Procedia cirp, v. 38, p. 3-7, 2015.

Matzka, “AI4I 2020 Predictive Maintenance Dataset”, www.explorate.ai/dataset/predictiveMaintenanceDataset.csv, submitted to UCI Machine Learning Repository, 2020.

MATZKA, Stephan. Explainable artificial intelligence for predictive maintenance applications. In: 2020 third international conference on artificial intelligence for industries (ai4i). IEEE, 2020. p. 69-74.

SEZER, Erim et al. An industry 4.0-enabled low cost predictive maintenance approach for smes. In: 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC). IEEE, 2018. p. 1-8.

ZONTA, Tiago et al. "Predictive maintenance in the Industry 4.0: A systematic literature review". Computers & Industrial Engineering, v. 150, p. 106889, 2020.