PCS 3850 - Projeto de Formatura II

Classificador de falhas em máquinas de chaves em operações ferroviárias - Manutenção Preditiva

Artur Vieira Ribeiro

Orientador: João Batista Camargo Júnior

Co-orientadores: Lúcio Flavio Vismari

Jamil Kalil Naufal Júnior

Pedro Pinheiro Garcia

Agradecimentos

Primeiramente, agradeço muito a minha família por todo o suporte, e que, mesmo de longe em Vitória, se fizeram presentes ao longo de toda minha graduação.

Ao GAS-PCS e FDTE, por possibilitar meu primeiro contato com iniciação científica e pela oportunidade de realizar este trabalho com seu apoio.

Por fim, agradecimentos especiais aos meus orientadores, João, Pedro, Lúcio e Jamil por todo o aprendizado e mentoria ao longo deste projeto.

Resumo

Ao longo deste trabalho, foi realizado um estudo descritivo e qualitativo de defeitos encontrados ao longo da operação de Máquinas de Chaves em malhas ferroviárias. Este tipo de máquina é responsável por realizar a mudança de via entre trilhos. Testamos a hipótese de que redes neurais convolucionais seriam capazes de diferenciar e rotular defeitos encontrados em uma base de validação. Em sistemas críticos, que necessitam de alta confiabilidade e segurança, como transportes, é essencial se adaptar às técnicas mais modernas de se otimizar a manutenção de seu maquinário. Esse sistema utiliza estudos já realizados pelo GAS-PCS em pesquisas anteriores e se dispõe de forma a complementar estudos de degradação temporal. O projeto abrangeu tratamento de dados, classificação e visualização, tendo como objetivo auxiliar iniciativas de manutenção preditiva nestas operações. Utilizando técnicas de manipulação de dados de forma a resumir leituras completas em imagens e detectando os padrões característicos a cada tipo de defeito. Este estudo treinou e comparou diferentes modelos, e o resultado final englobou 95.5% dos defeitos esperados em testes de validação.

Palavras-Chave - Manutenção Preditiva, Detecção de Defeito/Falha, Redes Neurais Convolucionais, Sistemas Ferroviários

Abstract

Throughout this work, a descriptive and qualitative study of the faults found during the daily operation of Switch Machines in Railway Systems was realized. This type of machine is responsible for changing tracks between rails. We tested the hypothesis that convolutional neural networks are capable of differentiating and labeling faults present on a validation dataset. In critical systems, which rely on high reliability and safety, such as transport, it is crucial to adapt to the most modern techniques to optimize the maintenance of its machinery. This system takes reference in previous GAS-PCS researches and aims to complement further studies of temporal deterioration. The project covered data processing, classification and visualization, with the objective of assisting predictive maintenance initiatives in these operations. Using data manipulation techniques in order to summarize complete measurements in images while also detecting the distinct patterns that belongs to each type of fault. This study trained and compared different models, and the final result encompassed 95.5 of the expected faults in validation datasets.

Keywords - Predictive Maintenance, Fault/Failure Detection, Convolutional Neural Networks, Railway Systems

Sumário

1. Introdução

1.1 Motivação

A Vale S.A opera cerca de 2 mil quilômetros de malha ferroviária no Brasil e possui acordos para utilizar linhas até em outros continentes. Há ainda em operação dois trechos de longa distância importantes operados com trens de passageiros. Com isso, há uma grande preocupação com a manutenção destas ferrovias.

Entre as principais técnicas de manutenção existentes estão manutenções corretivas e manutenções preditivas. O custo da manutenção preditiva, tanto direto quanto indireto, é, normalmente, menor do que o da manutenção corretiva. Assim, o desenvolvimento de técnicas para identificação de falhas se tornam úteis para identificar o melhor momento para uma vistoria técnica, considerando a gravidade das falhas e defeitos previstos.

1.2 Objetivo

Este projeto visa atuar de forma complementar aos atuais esforços do GAS-PCS de predição de falhas nestes sistemas eletromecânicos.

O objetivo principal é desenvolver um classificador para identificar os diferentes tipos de falha de acordo com leituras de campo indicativas do comportamento defeituoso atual. Os objetivos secundários deste trabalho são: complementar futuros trabalhos de observação da evolução de tipos mais específicos de falha com o passar do tempo, comparar a acurácia/precisão/recall do algoritmo quando utilizado diferentes bases de dados.

1.3 Justificativa

O GAS-PCS já possui parceria de longa data com a Vale neste âmbito de analisar diferentes sistemas da empresa em busca de otimizar o processo de manutenção preditiva e testar novas técnicas que contribuam para o desenvolvimento desta área. Neste caso, há um projeto em andamento que avalia dados de funcionamento e ocorrências das máquinas de mudança de via, as máquinas de chave AMV, processados pela própria Vale. A justificativa é que ao saber o tipo de falha mais recorrente, a manutenção preditiva pode ser otimizada para satisfazer as condições específicas deste tipo.

O problema em questão, é realizado atualmente por inspeção visual destes dados de corrente explicados anteriormente por analistas com conhecimento técnico. Assim, é de boa suposição que a análise imagética/matricial destes dados de corrente possa fornecer boa distinção entre os tipos de falhas que pretendemos observar neste projeto. Por isso, será utilizado uma abordagem em Redes Neurais Convolucionais (CNN em inglês) em nosso modelo.

1.4 Organização do trabalho

Este trabalho está dividido em 6 partes indicadas pela contribuição da professora Selma Melnikoff, com o modelo de TCC esperado. São realizadas reuniões semanais com o time completo de pesquisa do GAS-PCS bem como uma específica para o acompanhamento deste TCC. A seção de aspectos conceituais (parte 2) busca introduzir algumas das definições utilizadas ao longo do projeto e a forma como se apresentam. A etapa de metodologia (parte 3) do trabalho indica de qual maneira e com

que métodos serão realizados os desafios propostos. A especificação de requisitos (parte 4) adentra um pouco mais sobre as linhas técnicas do que será desenvolvido, declarando algumas entradas e saídas necessárias e o fluxo operacional do classificador.

2. Aspectos Conceituais

Nesta seção serão discutidos os principais conceitos teóricos necessários para o desenvolvimento do projeto.

2.1 Máquinas de Chave (MCH)

O objeto central desta pesquisa é a segurança do equipamento ferroviário “máquina de chave”, este é maquinário responsável por realizar a mudança de via.

As MCHs analisadas por este trabalho (modelo KY) têm em sua composição: um motor de corrente contínua, para realizar a movimentação das agulhas do aparelho de mudança de via (AMV), uma unidade controladora do motor (CNAP), que é utilizada para supervisionar e comandar o funcionamento deste motor, determinando o seu sentido de rotação e encerrando seu funcionamento, e por fim, um redutor que é composto de diversas engrenagens, incluindo em especial uma embreagem eletromagnética que é encarregada de realizar o acoplamento final entre as engrenagens primárias e secundárias com o pinhão da embreagem **(Da Silva Ferreira, 2021).**

2.1.1 Funcionamento da Máquina de Chave

A primeira fase (Fase I) do acionamento de uma máquina de chave, ou seja, da mudança de via, é encarregada por vencer o atrito estático em que o braço para esta mudança de via comece a se movimentar, traduzida por um pico de corrente curto, a segunda fase (Fase II) é encarregada de vencer o atrito dinâmico do braço internamente antes de chegar a barra que efetivamente ligue a parte do trilho a outra ponta, traduzido por uma corrente relativamente estável. A terceira fase (Fase III) é encarregada de efetivamente transladar essa barra do sentido de uma via para outra, se traduz em uma corrente relativamente estável, porém, em um patamar mais alto que a da fase 2. A fase 4 (Fim do evento) seria apenas a conclusão da tarefa, onde a corrente retorna ao nível 0 de corrente necessária. A **figura 1** apresenta a curva de corrente relativa a cada fase e a **figura 2** um exemplo real de uma MCH.

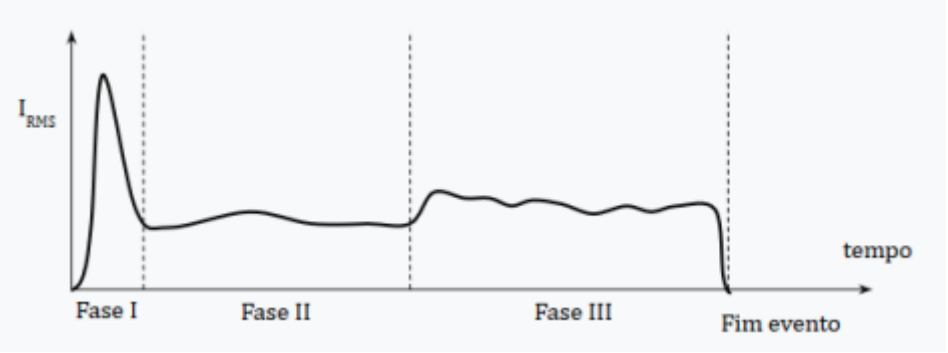


Figura 1. Curva de corrente do acionamento de uma MCH. Fonte:(Endo, 2020)

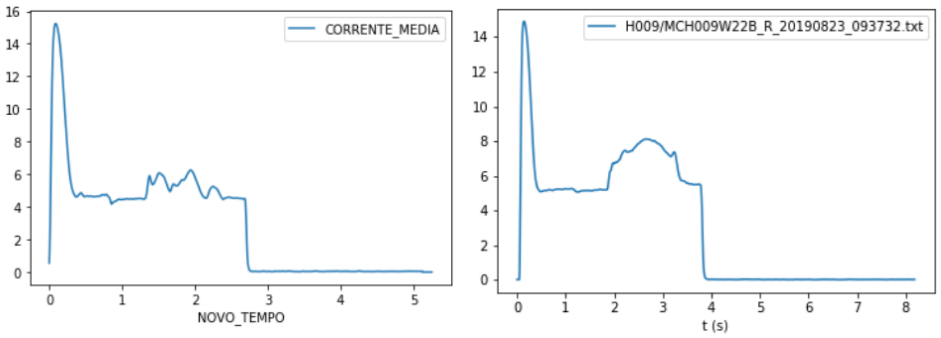
Figura 2. Máquina de chaves KY e sua aplicação na ferrovia.
Fonte: Pedro Pinheiro Garcia

Estas máquinas se encontram dispostas em Houses de funcionamento, que são literalmente pequenas casas ao longo de uma ferrovia que abriga vários circuitos. Dentro de cada House tem a eletrônica responsável por 4 MCH's (W21A, W21B, W22A, W22B).

2.1.2 Classificações através da leitura da curva de corrente

O funcionamento de uma máquina de chave é rotulado como Normal, Defeito e Falha. **Normal** sendo um evento sem oscilações estranhas ao comportamento padrão e que é completa e operacional; **Defeito** sendo um evento que apresenta certos comportamentos e oscilações incomuns porém completa seu funcionamento (alternar a via); e **Falha** sendo um evento não operacional, que não consegue realizar a mudança de via.

Esses padrões de funcionamento podem ser percebidos pelas leituras de corrente de um evento registrado por essa máquina de chave, conforme apresentado nas **figuras 3 e 4**.



Figuras 3 e 4. Leituras da corrente do funcionamento normal(1) e de defeito (alto atrito)(2)
Fonte: (Autor)

2.2 Dados fornecidos pela VALE

A qualidade dos dados é um fator primordial para uma boa análise e a construção de sentido para uma manutenção preditiva, é fundamental levar em consideração o espaçamento das amostras com o passar do tempo e desconsiderar possíveis variações que fujam da atuação mecânica do sistema, que podem ser atribuídas ao sistema de medição por exemplo.

Existem três bases de dados principais disponibilizadas atualmente pela VALE: base de manutenção (corretivas e preventivas); base de movimentações (classificadas e não classificadas). Neste trabalho serão utilizadas as bases de movimentações classificadas e não classificadas e a base de manutenção corretiva.

2.3 Redes Neurais

As redes neurais foram idealizadas para representar o funcionamento dos neurônios humanos e suas interações no intuito de reproduzir sua capacidade de aprender (tarefas, dados, reconhecimento) com reforços. Essas redes neurais computacionais se baseiam em uma simplificação de um neurônio biológico, que realiza uma função simples sobre uma ou mais

entradas para gerar uma saída, e, seu erro determina a variação de seus parâmetros internos (o que biologicamente seria a grossura da bainha de mielina). **(Peter Norvig, Stuart Russel)**

2.3.1 Técnicas de treinamento supervisionado

Existem diversas técnicas para treinar uma rede neural. Uma das mais comuns é conhecida como *backpropagation*, onde os dados são inseridos na entrada da rede e, ao se obter o resultado dessa função realizada na saída, é feita a comparação com os dados esperados e uma função de perda é aplicada, de forma que a rede entenda o quanto deva punir/bonificar o funcionamento de cada neurônio envolto nesta rede para que seu desempenho geral se aproxime mais do objetivo. Esse processo de melhora da rede é chamado de aprendizado e é repetido diversas vezes, sobre conjuntos de dados variados, que recebem o nome de lote. O conjunto de lotes que se utilizam de todos os dados disponíveis é chamado de época. Um treinamento usual passa por uma sequência de épocas.

2.3.2 Redes Neurais Convolucionais (Convolutional Neural Networks)

Dentro da família de métodos de reprodução de redes neurais, uma das técnicas de IA mais indicadas para problemas de inspeção visual ou dados matriciais em geral é a de Redes Neurais Convolucionais(CNN).

Apesar de não haver um entendimento claro do porquê desse bom desempenho desta técnica para reconhecimento de imagens, muitos estudos investigaram o funcionamento destas em ação para entender os padrões gerados dentro de cada camada, as sub-imagens, onde visualmente

induzem o pensamento algorítmico dentro do processo de “ver algo” **(Zeiler MD, Fergus R.).**

Este tipo de técnica é utilizada em reconhecimento de padrões em dados de imagem por se valer de várias camadas convolucionais para associar pontuações ao identificar características comuns entre objetos e as agrupar com características cada vez mais complexas com o avançar das camadas. Ao final do processo temos a camada de resposta que indica a maior probabilidade de rotulação final de qualquer sistema.

Podemos entender visualmente o funcionamento de uma rede Neural acompanhando o seguimento da **figura 5,** onde partindo de uma entrada matricial que seja relevante com o que deseja-se analisar e em formato padronizado, o problema é destrinchado em vários subconjuntos, que podemos observar pela **figura 6** dentro deste primário, e então, é analisado sob diferentes funções, cada uma representada por um neurônio.

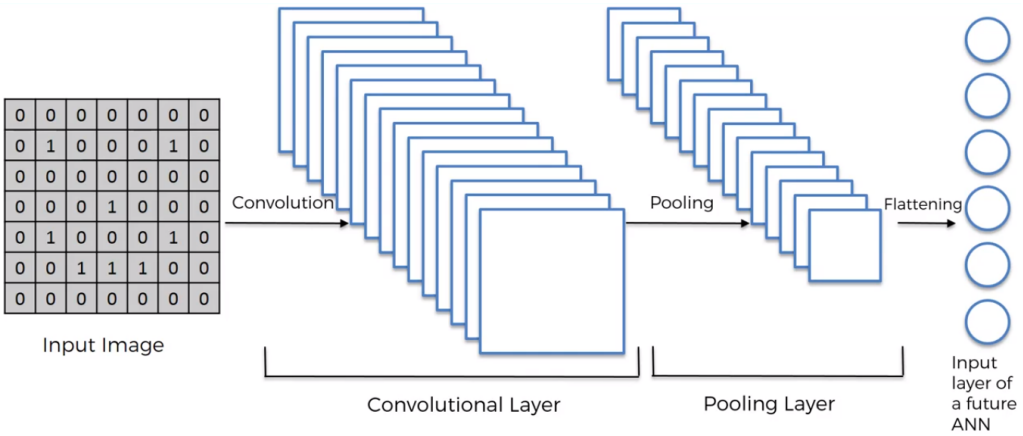


Figura 5. Representação de rede neural convolucional
Fonte: Panadda Kongsilp

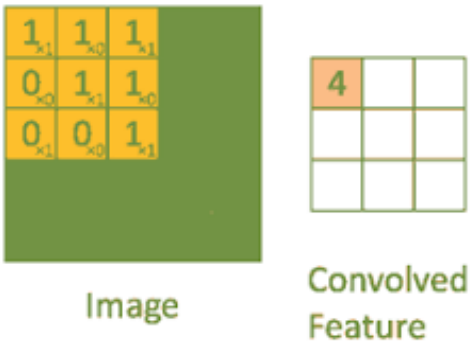


Figura 6. Exemplo de convolução a partir de um subconjunto
Fonte: DeepLearning.Stanford

Esse processo de criar novas camadas a partir dos subconjuntos é chamado de *pooling*. Temos como definir a quantidade de camadas e formato desses subconjuntos de forma a definir a complexidade esperada de cada análise sucessiva e assim até identificar sub padrões que podem ser relevantes para a análise final e guiar o desenvolvimento de uma nova testagem.

2.3.2.1 Acuracia

Formalmente, a acurácia em modelos de classificação e em estatística geral é a quantidade de acertos por tentativas, neste caso, leituras rotuladas com o mesmo rótulo entregue pela Vale pelo número total de leituras analisadas.

2.3.2.2 Loss

Loss ou Perda é a penalidade imposta aos modelos de classificação para a previsão feita. Idealmente, se busca minimizar essa perda, pois um nível alto deste atributo pode indicar *overfitting*. O treinamento do modelo avalia as alterações de perda e indica o algoritmo a encontrar os menores valores.

2.3.2.2 Precisão e Recall

Precisão e *Recall* são métricas que nos ajudam a avaliar modelos de previsão conforme a criticidade das suas aplicações **(Peter Norvig, Stuart**

Russell). Precisão é a proporção entre positivos verdadeiros e a soma das **classificações** positivas. ***Recall*** é a proporção entre positivos verdadeiros e a quantidade real de positivos (positivos verdadeiros e falsos negativos).

No contexto de manutenção preditiva, entendemos que a maior criticidade seria um equipamento que não indique uma manutenção quando for preciso, ou seja, se busca diminuir o ***recall***.

2.4 Manutenção baseada em condições

Esta linha de pesquisa solicitada pela Vale busca atingir um modelo de manutenção baseada em condições, cuja finalidade é intervir no funcionamento corriqueiro de um sistema conforme avaliações de estranhamento neste, de forma a manter sua produtividade e gerar menos indisponibilidade no futuro. “Esta é a técnica de manutenção mais investigada tanto pela indústria quanto pela comunidade científica atualmente” **(BAUR, 2020).**

Este tipo de manutenção pesquisada é chamada de baseada em condições justamente por necessitar de diagnósticos de padrões que causem estranhamento para um ajuste mais específico.

2.5 Manutenção preditiva

Essa metodologia visa aplicar devido suporte, manutenção ou mesmo "reinícios" de forma anterior a falha ocorrer, aumentando o *Mean Time Between Failures* (MBTF) de um sistema. Geralmente utilizado quando a criticidade da operação requer disponibilidade, segurança e confiabilidade elevadas, como é o caso de sistemas em que o prejuízo de não se ter um *uptime* confiável e robusto pode ser muito custoso material ou imaterialmente.

3. Metodologia

Neste tópico será feito a declaração dos métodos e dos motivos e *insights* utilizados para justificar as escolhas feitas para as etapas do desenvolvimento do projeto do classificador.

A organização deste projeto pode ser separada em algumas etapas. A etapa inicial do projeto será a chamada *Feature Engineering*, isto é, a preparação dos dados fornecidos pela Vale de forma a ajustá-los a formatos mais específicos que atendam aos métodos utilizados de *Machine Learning*.

A segunda etapa é o desenvolvimento deste módulo classificador, considerando todo o processo de treinamento e organização da sua aplicação.

A etapa final é a avaliação dos resultados dos dados classificados, seus parâmetros estatísticos como sua assertividade. Também será feito contestação/confirmação com técnicos a fim de comprovar o sucesso desta avaliação.

A **figura 7** ilustra as principais etapas deste processo.

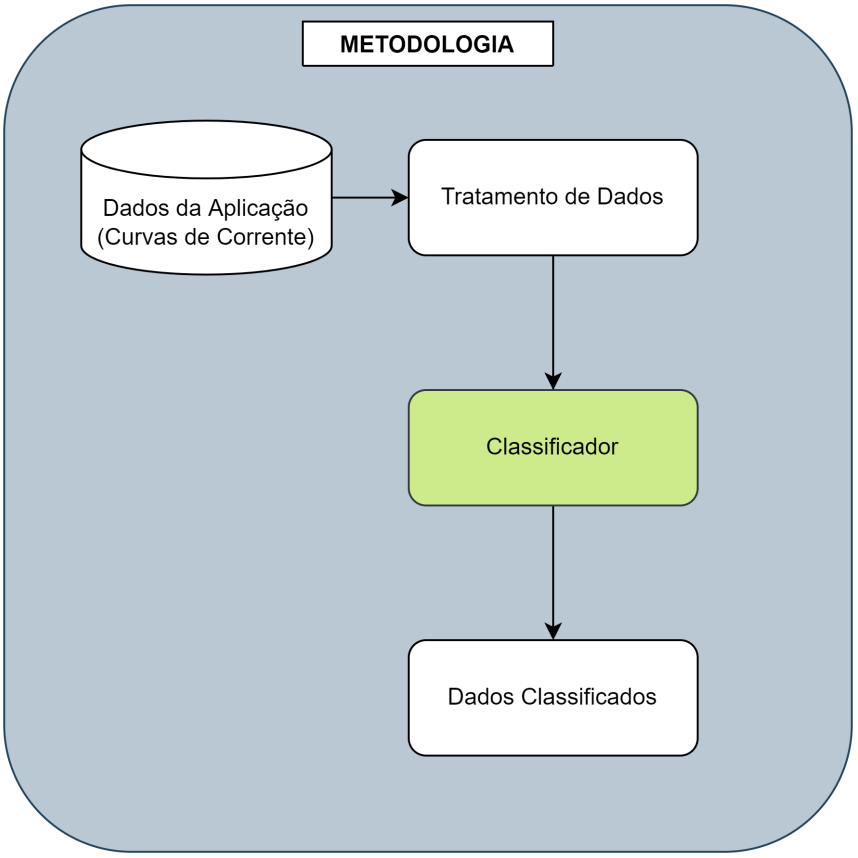


Figura 7. Fluxograma da metodologia do projeto.
Fonte: Autor

3.1 Tratamento de Dados

Nesta etapa será realizado o pré-processamento dos dados para transformar os dados da aplicação (curvas de corrente) em informações úteis para o aprendizado das técnicas relacionadas a este projeto.

3.1.1 Representação em imagem da curva de corrente

Podemos observar na **figura 8** duas representações do registro de corrente de funcionamento de uma máquina de chave, a primeira um *plot* simples, dado a corrente pela quantidade de medições que são feitas de acordo com um *clock* fixo. A segunda representação é o mapa de calor desse vetor de medições, com tons mais quentes representando valores mais altos de corrente e vice-versa.

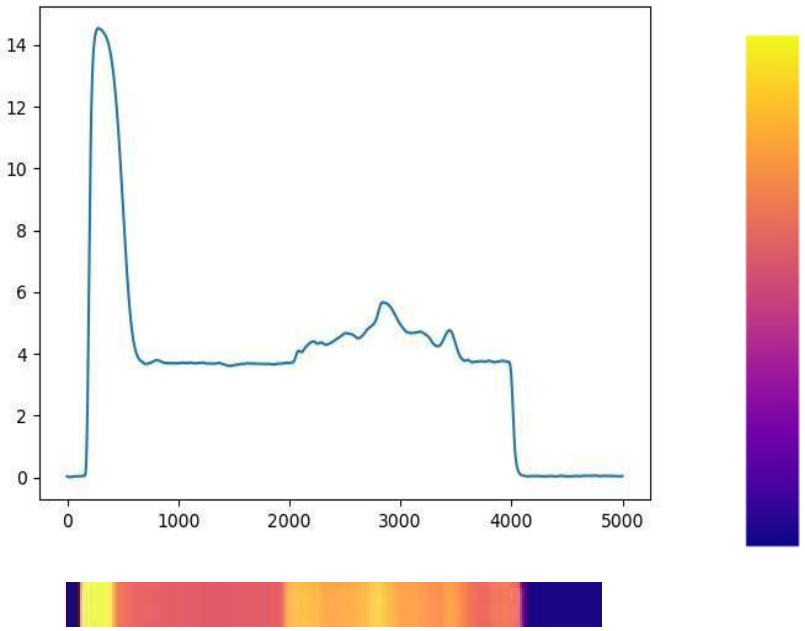


Figura 8. Exemplo de funcionamento e sua representação em "Mapa de calor"

Outra representação que também pode ser utilizada, evidenciada pelo trabalho de Yokowo **(Yokowo, 2021),** é a técnica chamada de *Gramian Angular Fields* (GAF). A **figura 9** traz um exemplo de uma leitura, das fases 2 e 3, transformada em uma imagem através desta técnica.

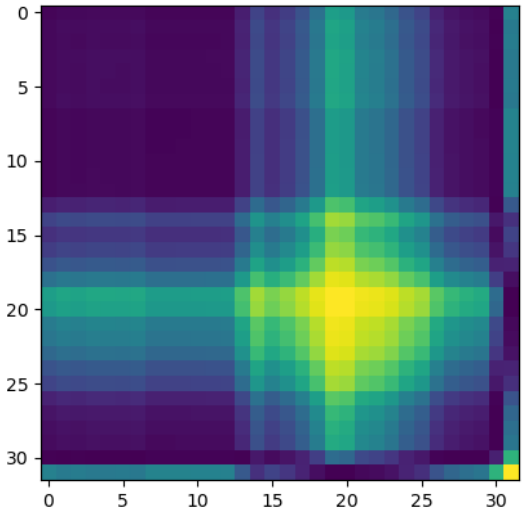


Figura 9. Leitura das fases 2 e 3 de uma curva de corrente - GAF

3.1.2 Representação em features da curva de corrente

A Vale atualmente calcula e nos fornece, além das leituras de corrente, 60 características acerca de propriedades estatísticas dos eventos mensurados pelas máquinas de chave, como médias de corrente por fase, variação, tempo que durou cada fase, dentre outras. Estes dados serão utilizados para explorar métodos de machine learning de forma a descobrir relações entre a evolução da degradação de uma máquina pelo tempo.

A partir destas leituras, a própria Vale refinou essas leituras de forma a nos enviar um conjunto de 60 atributos com estatísticas relevantes acerca do funcionamento no contexto geral e em fases específicas do funcionamento deste sistema. Cabe destacar que esse relatório inclui informações como desvio padrão por fase, média, duração de fases, tempo até o início da movimentação, área debaixo das fases, dentre outros.

3.2 Técnicas de classificação

Técnicas como a CNN podem trabalhar com dados convertidos em imagem para satisfazer as operações matriciais do algoritmo. Visto que o algoritmo CNN tira seu maior proveito reconhecendo padrões provenientes do cruzamento de dados.

Nesse contexto de manutenção baseada em condições (CBM), também corroborou para a escolha deste tipo de rede neural a revisão de literatura disponível em **“(Pinheiro, 2022)”**, de onde foi retirada a **figura 10**. Onde a partir da extração de artigos em CBM que desenvolveram modelos para aplicações ferroviárias, foi realizada uma avaliação dos sistemas mais utilizados.

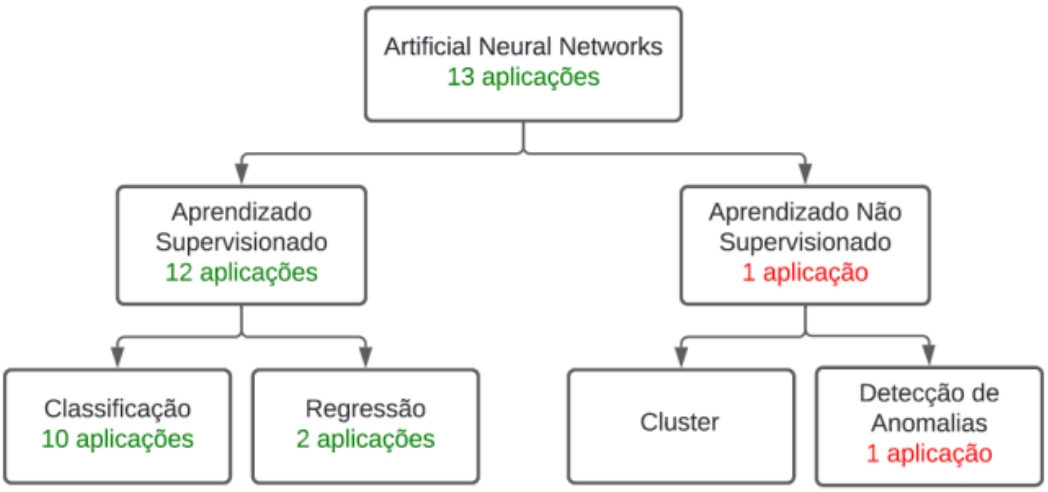


Figura 10. Distribuição da literatura de ANN's

Fonte: Pinheiro, 2022

Dentro da base de dados extraída, as redes neurais artificiais(ANN) estiveram presentes em pouco mais de 50% dos artigos, com destaque para técnicas supervisionadas de classificação, sendo esta alternativa a escolhida para o projeto desta documentação (Pinheiro, 2022).

A segunda etapa do trabalho será a de desenvolvimento do classificador. Na etapa de desenvolvimento do classificador, a técnica de desenvolvimento será baseada no ciclo de prototipagem padrão. Onde a partir da preparação dos requisitos é elaborado um plano rápido de atuação de forma a guiar a modelagem e aplicação rápida do protótipo.

Esta agilidade na etapa de construção fornece mais tempo para avaliar resultados preliminares de forma a indicar o caminho mais promissor a ser desenvolvido na próxima versão do protótipo, aproveitando o feedback dos orientadores e dos técnicos da Vale.

3.2.1 Tipificação dos Defeitos/Falhas

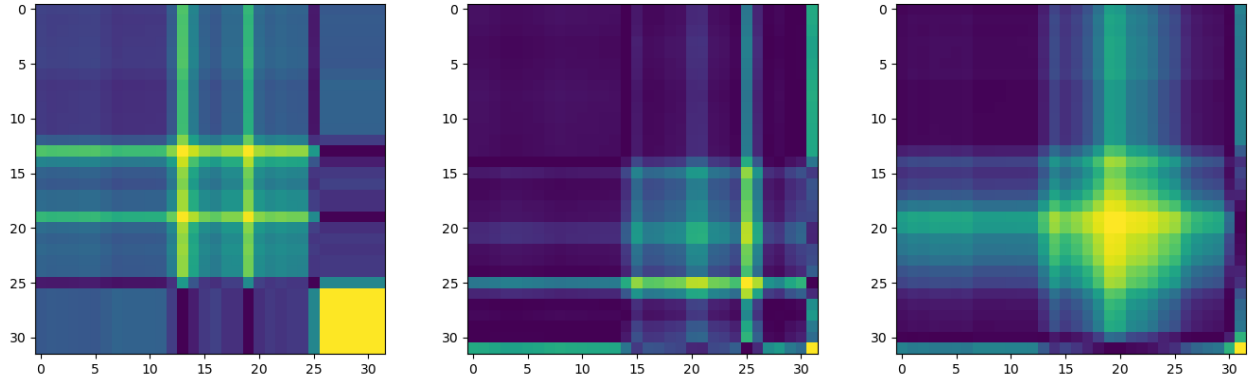
A partir de uma análise sintomática das operações crê-se possível neste caso definir por meio deste classificador os tipos de defeitos e falha atualmente propostos pela Vale, que são 12 tipos de defeito e 6 de falha.

Os tipos de defeitos geralmente estão relacionados com as eventuais tipificações de falha e indicam esta tendência. Temos exemplos como desligamento de motor pode ser classificado como defeito ou como falha, dependendo se a operação da máquina conseguiu se reativar e concluir sua mudança de via ou não, e atrito entre os deslizamentos da agulha serem suficientes para interromper a operação da máquina ou não.

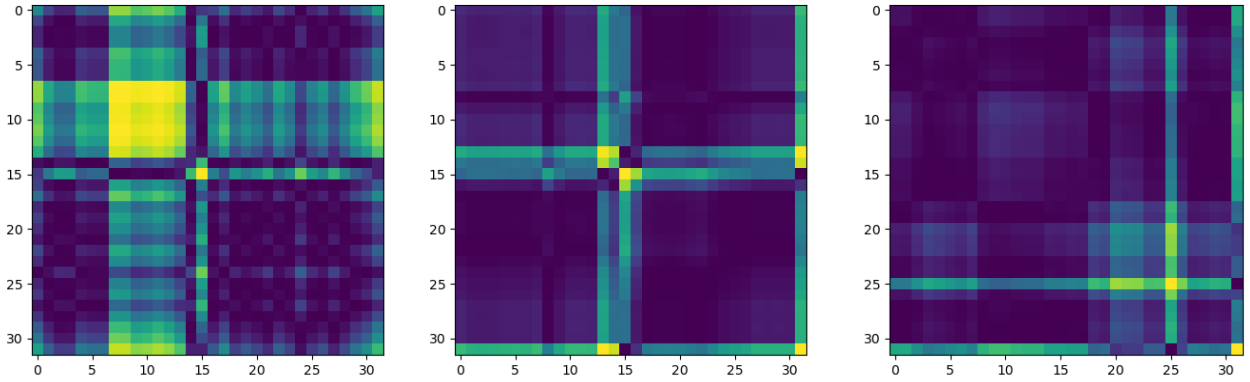
Há a indicação para defeitos e falhas não identificados atualmente, o que também permite uma lacuna para o descobrimento de novos padrões de estranhamento no futuro. Entre os tipos de falha e defeito estão causas como atritos, múltiplos acionamentos, obstruções, oscilações, embreagem danificada, atrasos na partida, etc.

3.2.2 Viabilidade e Distinção

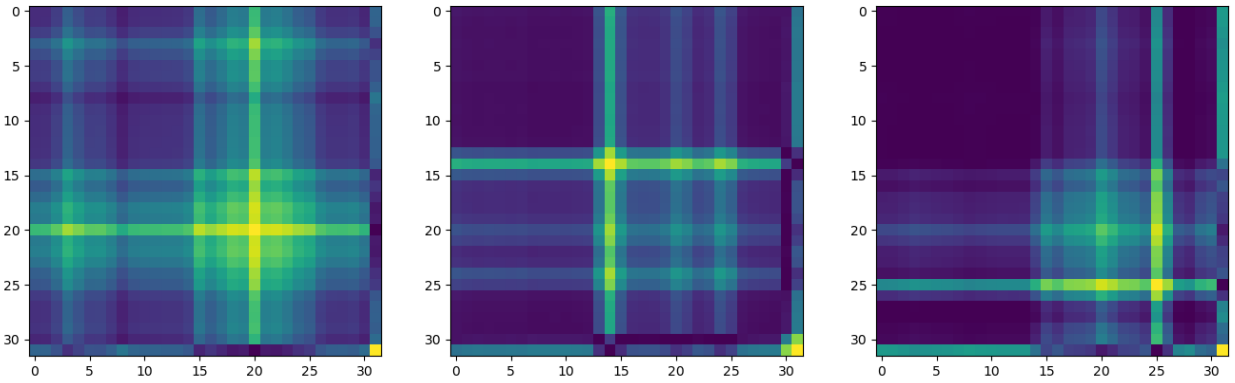
A rotulação das leituras fornecida pela Vale bem como suas leituras correspondentes nos permitiram realizar uma análise prévia deste funcionamento. De acordo com cada tipo de defeito/falha podemos enxergar essas diferenças ao realizar o produto gramiano em cima de seus mapas de calor em cima de suas leituras devidamente cortadas de modo a haver apenas a fase 2 e 3 de seu funcionamento. Alguns exemplos de leituras convertidas em mapas de calor do produto gramiano podem ser observados a seguir, entre as **figuras 11 e 21**.



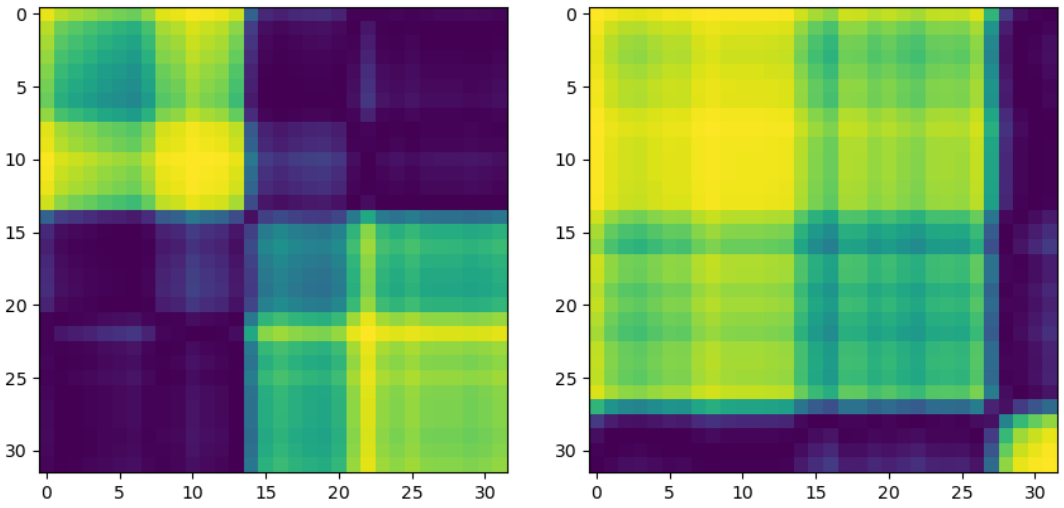
Figuras 11, 12 e 13. Produto Gramiano de leituras rotuladas respectivamente como, D3 Atraso Partida, D7 Médio Atrito, D2 Alto atrito
Fonte: Autor



Figuras 14, 15 e 16. Produto Gramiano de leituras rotuladas respectivamente como, Corrente baixa e oscilando, Desligamento do motor, Erro algoritmo
Fonte: Autor



Figuras 17, 18 e 19. Produto Gramiano de leituras rotuladas respectivamente como D10 Problema Fase 2, D11 Sobrecorrente Destravamento e D8 Múltiplos acionamentos
Fonte: Autor



Figuras 20 e 21. Produto Gramiano de leituras rotulados respectivamente como F4 Excesso de atrito e F5 Obstrução
Fonte: Autor

Estas leituras serviram para exemplificar um pouco do que se pretende extrair aplicando o produto Gramiano às fitas unidimensionais de mapa de calor. Como observado nestas imagens a posição do xadrez de informação é mais facilmente visualizada por um humano, e supõe-se, para um algoritmo de classificação em CNN também. É também notório como as leituras classificadas como Falhas (**figuras 20/21)** produzem uma imagem com manchas bem maiores em valores altos, como representado pela cor amarela, com padrões que fogem bastante de uma Defeituosa por não conseguirem completar seu funcionamento.

A parte final do classificador deve ser composta por uma camada de neurônios ReLU, representada na **figura 22**, definida para rotular a leitura entre os diferentes tipos de defeito e falha. Os valores que chegam a esses neurônios podem ser extraídos para fornecer a certeza da escolha tomada pelo algoritmo, o que pode ser útil para analisar a assertividade e guiar o aprimoramento do classificador.

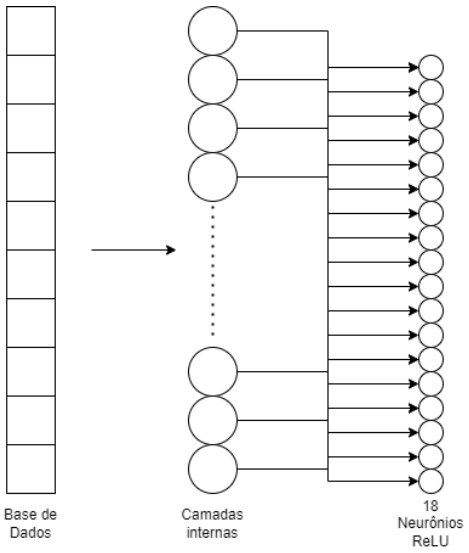


Figura 22. Representação das camadas da rede neural.

3.3 Dados classificados

A classificação fornecida ao final do classificador deve ser em um dos rótulos descritos na seção **3.1**, bem como sua certeza de afirmação e seu DF. Posteriormente esses dados serão armazenados para revisão técnica e avaliação contínua, o que influenciará em outro projeto complementarmente para entender a degradação acontecendo no sistema.

3.4 Classificador

O projeto de um classificador envolve a devida rotulação dos dados fornecidos em diferentes tipos de defeitos e falhas para o correto treinamento. Este deve ser feito assim que observado alguma relação justificável entre as leituras tratadas e seus rótulos. O classificador deve ser capaz de ler novas leituras em formato adequado ao seu treinamento e disponibilizar sua análise.

4. Especificação de Requisitos

Numa avaliação inicial dos requisitos necessários a este projeto, convém compreender exatamente a área de atuação e onde se encaixa em um fluxograma das operações de uma manutenção baseada em condições (**figura 23)**.

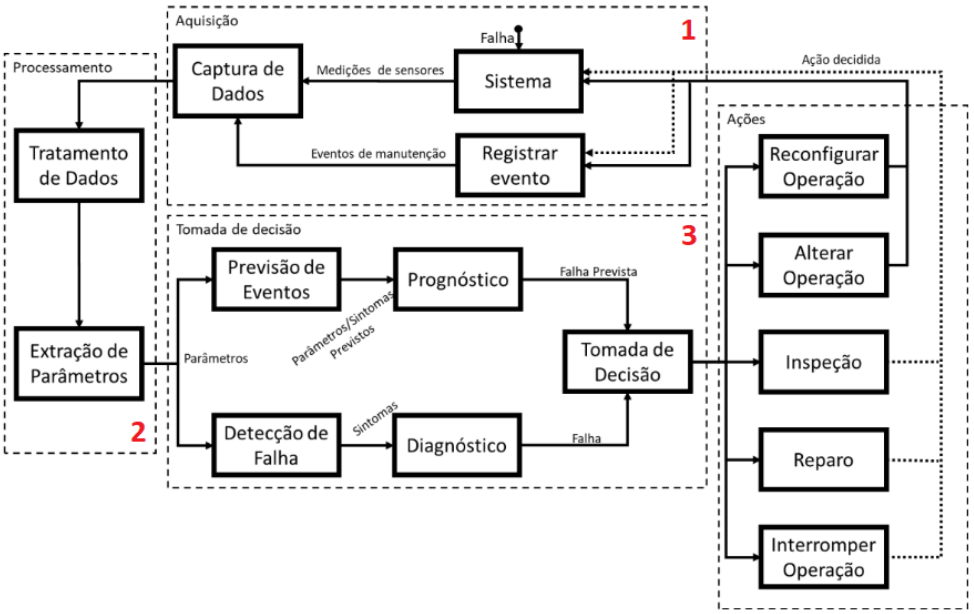


Figura 23. Fluxo de operações de um projeto de manutenção baseada em condições

Fonte: (OLIVEIRA, 2020)

O projeto do classificador proposto encaixa nesta etapa após o devido tratamento de dados e extração de parâmetros, como exemplificado na **figura 24**, etapas que ainda permeiam o projeto mas não fazem parte da ideia central a ser desenvolvida, porém, para um bom resultado é crucial rever e estudar os dados fornecidos. O classificador atua mais precisamente na parte de diagnóstico de sintomas depois das detecções de falhas.

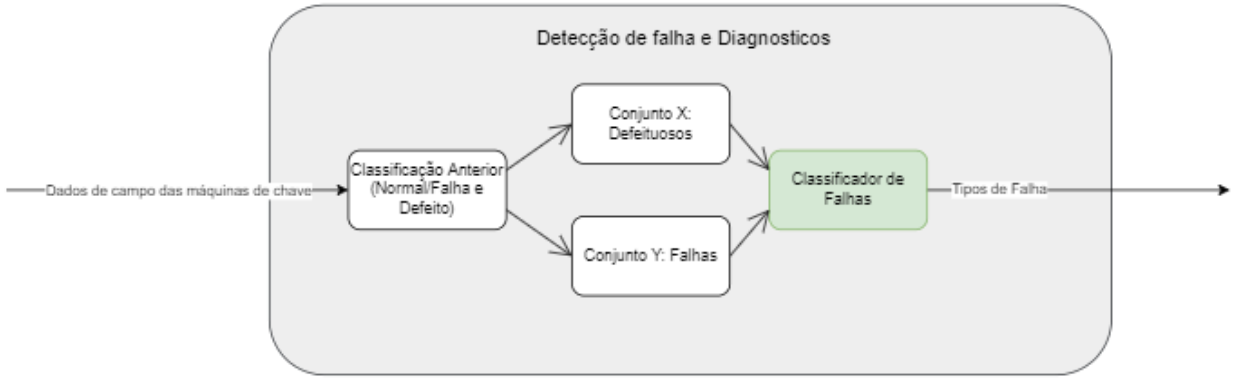


Figura 24 Fluxo detalhado do módulo de detecção de falha e diagnósticos.
Fonte: Autor

A complexidade maior do projeto está no desenvolvimento do classificador de falhas, então esta etapa concentra a maior parte dos requisitos. Para o funcionamento deste, o primeiro requisito é tratar os dados e entregá-los da melhor forma possível ao segundo módulo do classificador. Este segundo módulo é a etapa do treinamento do classificador com a técnica desenvolvida e posteriormente, realizar as comparações com a base de testes e sua assertividade será mostrada junto com as classificações realizadas. A organização desses módulos está disposta na **figura 25.**

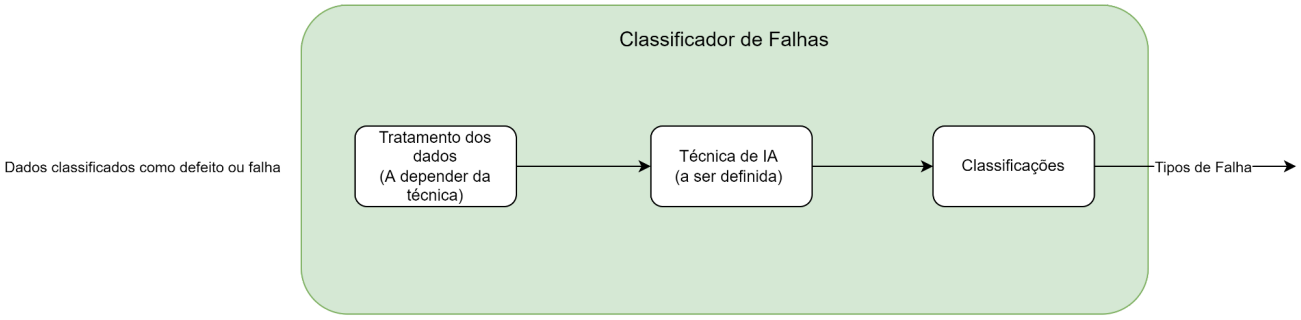


Figura 25. Fluxo detalhado do classificador de falhas.
Fonte: Autor

Para utilização de técnicas robustas como o CNN, temos que organizar esses dados temporais como imagens/matrizes para fazer sua avaliação, por isso o correto tratamento destes dados é importante.

4.1 Requisitos Funcionais e Não-Funcionais

O objetivo final é apresentar um módulo independente, que conclua uma análise das leituras e forneça classificações orientadas por dados, em formato ideal para que possa ser utilizada de maneira a auxiliar a avaliação de degradação temporal. Dessa forma podemos destacar alguns requisitos.

Requisitos Funcionais

- Coletar informações acerca do funcionamento das máquinas de chave
- Classificar leituras
- Organizar evolução temporal de uma máquina
- Coletar, processar e agregar dados de medição a longo prazo

Requisitos Não-Funcionais

- Funcionamento em tempo real
- Alertas em caso de degradação avançada
- Relatórios automáticos da degradação

4.2 Funções

O desenvolvimento por partes do funcionamento geral deste módulo de detecção de falhas pode ser explicado por meio de funções base desta infraestrutura. Convém destacar que essas funções serão definidas na linguagem Python e tem seus alicerces em código legado de outras pesquisas do GAS-PCS e bibliotecas de análise de dados e machine learning.

SeparaDados/Módulo de detecção de Falhas e Diagnósticos

Esta função receberá as leituras provenientes do classificador NDF(Normal, Defeito ou Falha), com sua data de medição, corrente pelo tempo, classificação NDF e separá-las em duas bases, base de Falhas e Defeitos.

ChecaIntegridade/Módulo de detecção de Falhas e Diagnósticos
Esta função deve receber a base de dados e remover leituras muito discrepantes com o contexto e duplicatas e substituir a base de dados original.

CortaDados/Tratamento de Dados
Esta função deve ser capaz de receber uma leitura de corrente e separar apenas a duração da fase 2 e fase 3 e retorná-las em substituição das leituras completas para o banco de dados.

ConverteHeat/Tratamento de Dados
Esta função recebe os valores de corrente pelo tempo de uma leitura ou o produto gramiano de uma leitura, converte na representação em mapa de calor e os armazena em um banco de dados para análise temporal.

ConverteGramiano/Tratamento de Dados
Esta função recebe os valores de corrente pelo tempo de uma leitura e realiza seu produto gramiano, deve fornecer na saída seu resultado em formato de uma matriz de valores.

SeparaTrainTest/Técnica de IA
Esta função recebe uma base de dados dividida em *features* (comumente representada por X) e sua rotulação (representada por y), e a divide entre leituras a servirem de treinamento e leitura a servirem de teste, em duas bases de dados diferentes. A proporção é por padrão 75/25 mas pode ser alterada manualmente indicando mais uma entrada.

TreinaClassificador/Técnica de IA

Esta função será encarregada de receber a base de treinamento e treinar o modelo de forma a definir os parâmetros internos devidamente ajustados. Este deve fornecer o modelo pronto, e com suas funcionalidades de entrada e saída e acesso aos parâmetros internos.

TestaClassificador/Técnica de IA

Esta função, baseada na arquitetura de uma CNN (**figura 26**), será encarregada de receber a base de teste e o classificador treinado de forma a testar, de forma supervisionada, a classificação realizada pelo classificador. Este deve retornar a acurácia do classificador para o grupo de testes.

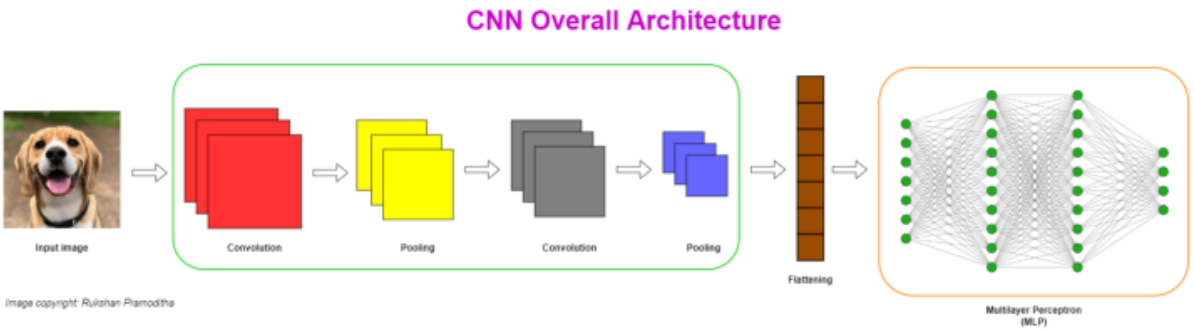


Figura 26. Diagrama esquemático da arquitetura genérica de uma CNN.
Fonte: https://towardsdatascience.com/coding-a-convolutional-neural-network-cnn-using-keras-sequential-api-ec5211126875

5. Desenvolvimento do Trabalho

Esta etapa de desenvolvimento teve especial cuidado em inserir os métodos escolhidos para tratamento de dados e o que seria um breve

exemplo de entrada devidamente tratada cujo próximo passo seria apenas a classificação.

De forma a manter o algoritmo o mais genérico possível para que ele utilize apenas o dado cru de corrente pelo tempo, é preciso definir bem como estas funções serão aplicadas e em que sequência de acontecimentos, dependendo do teste escolhido.

O teste inicial do algoritmo fará uso das funções definidas na especificação de requisitos aplicadas ao exemplo de uma leitura da chave MCH004W21B.

No primeiro exemplo (**figura 27**), a leitura é avaliada inteiramente, com os dados de corrente sem o recorte da fase 2 e 3.

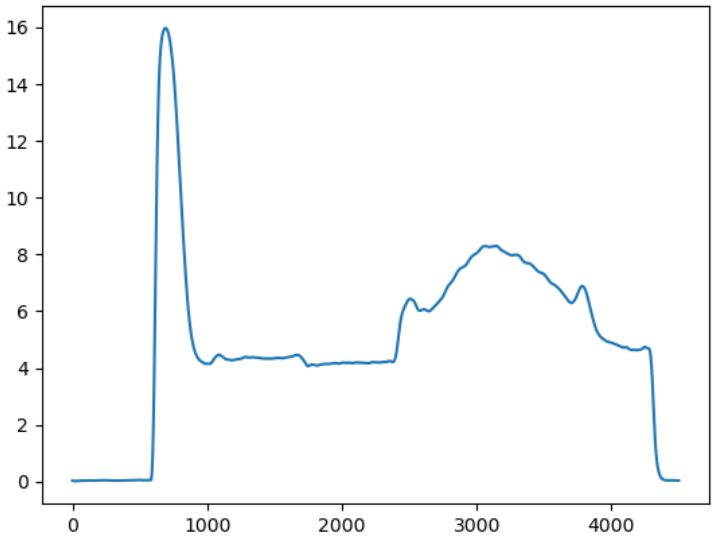


Figura 27. Leitura completa do funcionamento da máquina de chave.
Fonte: Autor

Essa leitura é aplicada às funções **ConverteHeat** e **ConverteGramiano**. Onde podemos observá-las como na **figura 28 e 29**:

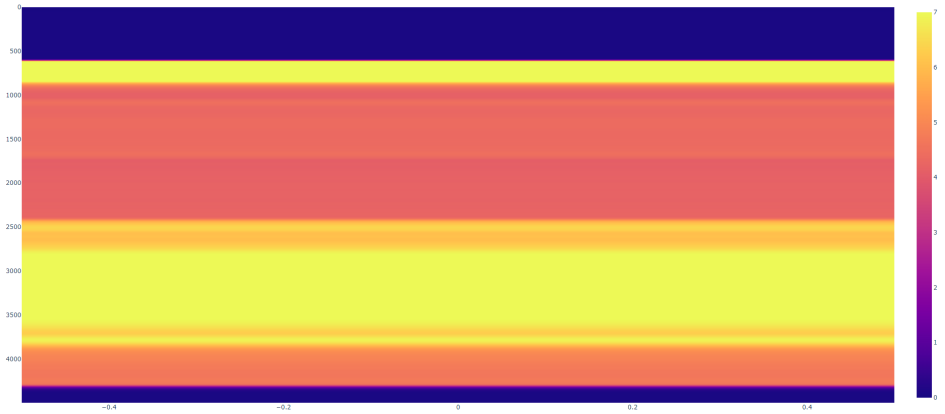


Figura 28. Heatmap do funcionamento da máquina de chave
Fonte: Autor

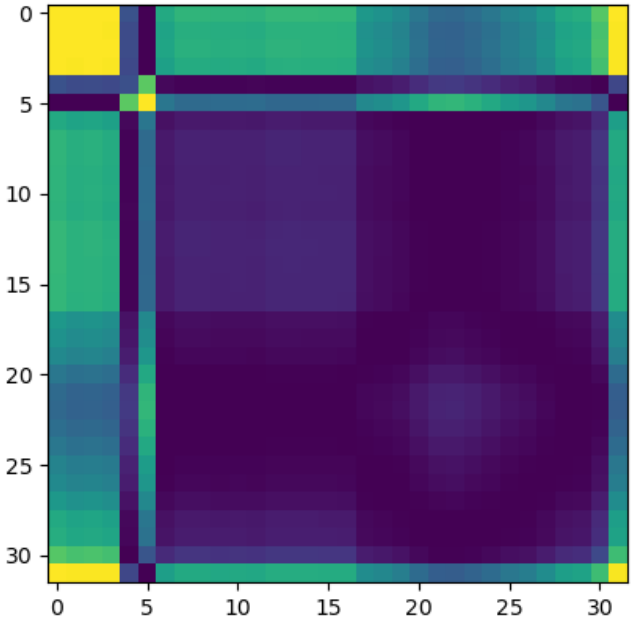


Figura 29. Gramian Angular Field do funcionamento da máquina de chave
Fonte: Autor

Esses dois dados convertidos devem ser então repassados ao classificador pela função **TestaClassificador**, onde o resultado esperado seria uma rotulação de defeito Alto Atrito.

A outra rota de teste planejada seria utilizar os dados apenas da fase 2 e 3 da leitura de corrente, como na **figura 30**.

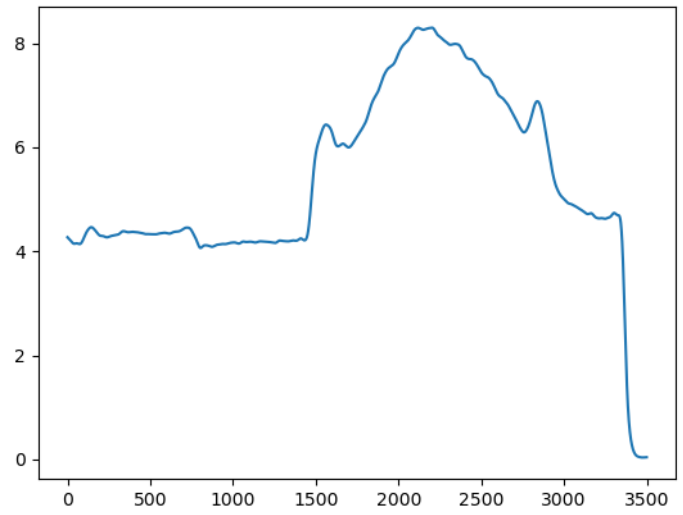


Figura 30. Leitura das fases 2 e 3 do funcionamento da máquina de chave.
Fonte: Autor

Essa leitura será também aplicada às funções **ConverteHeat** e **ConverteGramiano**. Onde podemos observá-las nas **figuras 31 e 32**:

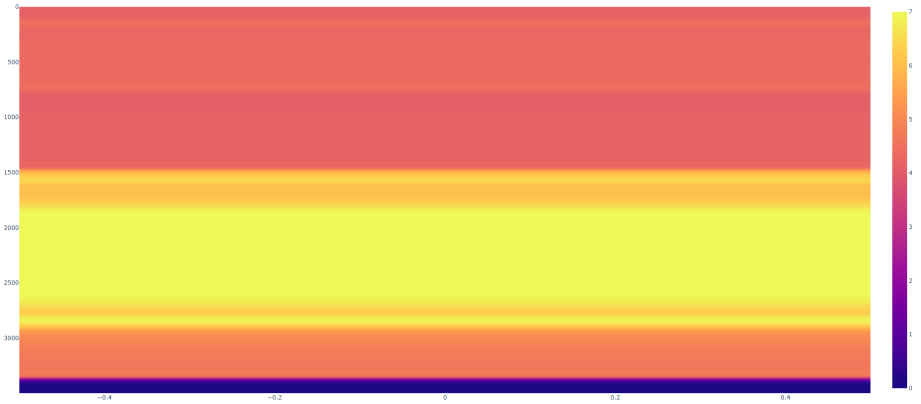


Figura 31. Heatmap do funcionamento da máquina de chave
Fonte: Autor

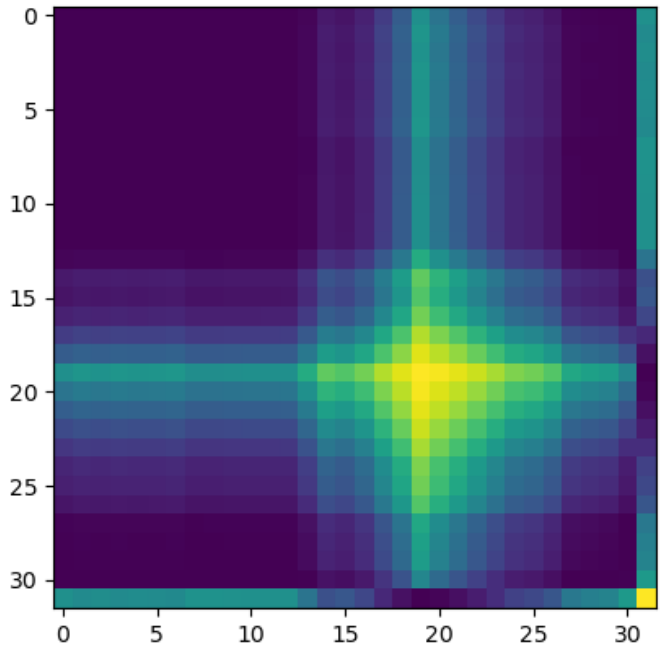


Figura 32. Gramian Angular Field do funcionamento da máquina de chave
Fonte: Autor

Essa representação da leitura também tem como rótulo esperado um defeito Alto Atrito ao final do módulo classificador.

A diferença entre estas representações é clara ao aplicar o **ConverteGramiano**. Avaliações prévias sugeriram uma melhora na precisão de algumas avaliações utilizando as leituras completas mesmo havendo parecer técnico para utilizar-se do corte das fases 2 e 3. Para este projeto a avaliação conjunta destas será de maior utilidade para comprovar o bom funcionamento e ampliar a generalização deste algoritmo.

Vale a pena ressaltar que, com os blocos de funções bem definidos, o fluxo de análise pode ser reformulado e incluir mais tipos de teste.

Em uma primeira etapa de desenvolvimento, foi escolhida a House 4 da base de dados fornecida pela vale desta está distribuída da seguinte forma no próximo tópico.

5.1 Testes dos modelos

5.1.1 Base de dados

O primeiro modelo treinado, presente nos arquivos como **Modelo1-Original**, foi treinado com os 8 tipos de defeitos rotulados pela Vale. Existem mais do que 8 tipos de defeitos conhecidos mas esses foram os encontrados ao longo do período de 3 anos de avaliação deste tipo de máquina em operação.

A base com imagens 32x32 original está distribuída como apresentado na **figura 33** e a conclusão do treinamento como mostrado na **figura 34**.

	Alto Atrito	Atraso partida	Defeito Não Identificado	Desligamento Motor	Médio Atrito	Múltiplos acionamentos simultâneos	Problema Medição	Problema na fase 2
Treinamento	359	82	27	53	7325	32	5178	1456
Validação	95	20	7	12	1820	8	1254	360

Figura 33. Distribuição das rotulações disponíveis pela Vale

Fonte: Autor

```
Epoch 99/100
113/113 [==============================] - 20s 179ms/step - loss: 0.1412 - acc: 0.9417 - val_loss: 0.3534 - val_acc: 0.8932
Epoch 100/100
113/113 [==============================] - 20s 178ms/step - loss: 0.1398 - acc: 0.9410 - val_loss: 0.3811 - val_acc: 0.8863
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (
```

Figura 34. Logs de conclusão de treinamento do **Modelo1-Original** e métricas.

Fonte: Autor

O modelo treinado teve uma acurácia elevada, porém, ao analisar a distribuição dos rótulos essa porcentagem poderia indicar *overfitting*, ou seja, o modelo aprendeu a chutar rótulos majoritários visto que pontuou bem assim, mas não demonstrou um “aprendizado” que é o objetivo.

Por seguinte, foi utilizado uma base reduzida, com apenas as 5 leituras que indicam degradação do aparelho. Foram removidas leituras classificadas como Múltiplos Acionamentos, Problema Medição e Defeito não-Identificado e a distribuição se encontra na **figura 36**. Este modelo foi salvo nos arquivos como **Modelo2-Reduzido**.

Este apresentou boa capacidade de generalização, medida pela distância entre as perdas entre treinamento e validação (**figura 35)**, seu log de conclusão de treinamento está disponível na **figura 37**.

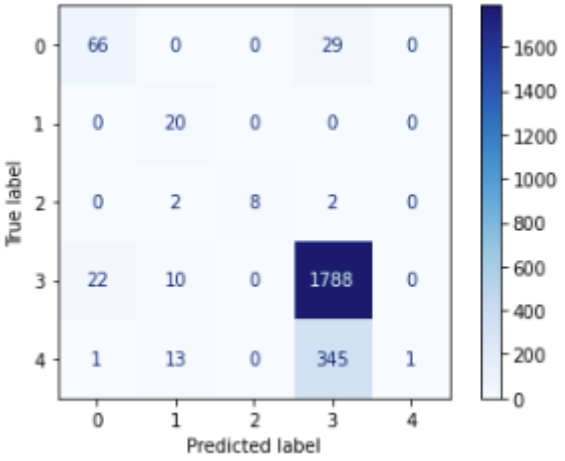


Figura 35. Matriz de confusão da performance do **Modelo2-Reduzido**
Fonte: Autor

Apresentou uma dificuldade perceptível em discernir médio atrito de problema na fase 2, mas apresentou separabilidade considerável entre o resto dos rótulos.

	Alto Atrito-1	Atraso partida-2	Desligamento Motor-3	Médio Atrito-4	Problema na fase 2-5
Treinamento	359	82	53	7325	1456
Validação	95	20	12	1820	360

Figura 36. Distribuição dos 5 rótulos que indicam degradação disponíveis pela Vale
Fonte: Autor

```
Epoch 99/100
72/72 [==============================] - 13s 178ms/step - loss: 0.1967 - acc: 0.9163 - val_loss: 0.5052 - val_acc: 0.8407
Epoch 100/100
72/72 [==============================] - 13s 178ms/step - loss: 0.1922 - acc: 0.9163 - val_loss: 0.5640 - val_acc: 0.8273
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _jit_compiled_convolutior
```

Figura 37. Logs de conclusão de treinamento do **Modelo2-Reduzido** e métricas.

Fonte: Autor

Ao final do treinamento, nos deparamos com um modelo com menor acurácia que o anterior, porém com detecções mais concisas, o que é preferido pela nossa avaliação.

Por último, foram realizados testes em uma base com equiparação na quantidade de leituras por cada rótulo, utilizando uma técnica de *oversampling*. De forma a manter uma base balanceada, repetimos leituras de cada rótulo na etapa de treinamento.

Este foi salvo nos arquivos como **Modelo3-Oversampling.**

Treinado com batch_size = 128, epochs = 98. Seu resultado e logs de treinamento estão dispostos na **figura 38 e 39.**

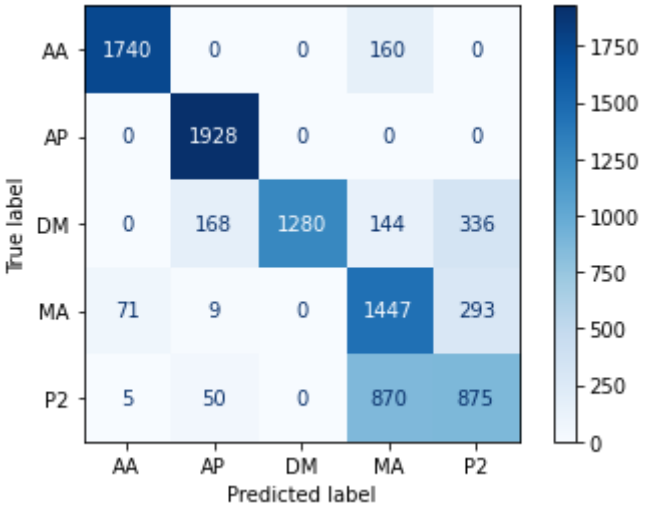


Figura 38. Matriz de confusão da performance do **Modelo3-Oversampling.**

Fonte: Autor

```
Epoch 98/98
294/294 [==============================] - 51s 173ms/step - loss: 0.1744 - acc: 0.9289 - val_loss: 3.1383 - val_acc: 0.7768
```

Figura 39. Logs de conclusão de treinamento do **Modelo3-Oversampling** e métricas.

A conclusão dessa primeira etapa testes foi que a técnica de oversampling melhorou bastante a generalização do modelo, que agora

acerta razoavelmente uma quantidade acima de 48% para cada rótulo. A maior dificuldade foi na discernibilidade entre **médio atrito** e **problema na fase 2**. Apesar da acurácia geral ainda rodar no patamar de 77%, definindo esta como a melhor base para atuar podemos testar o algoritmo para gerar modelos alternativos em numeros de camadas, *pool-size*, e objetivo de otimização.

5.2 Parâmetros internos

O modelo padrão utilizou 45413 parâmetros treináveis e 15 camadas compostas por convoluções, ativações, *poolings*, achatamentos, *dropouts* e camadas densas. Distribuídas da seguinte forma na **figura 40**.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
activation (Activation)	(None, 30, 30, 32)	0
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	9248
activation_1 (Activation)	(None, 13, 13, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 32)	0
conv2d_2 (Conv2D)	(None, 4, 4, 64)	18496
activation_2 (Activation)	(None, 4, 4, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 64)	16448
activation_3 (Activation)	(None, 64)	0
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 5)	325
activation_4 (Activation)	(None, 5)	0

Total params: 45,413
Trainable params: 45,413
Non-trainable params: 0

Figura 40. Resumo do **Modelo3-Oversampling** compilado.

A primeira forma de criar novos modelos baseados neste original foi alterar a quantidade de camadas, inicialmente, reduzindo esta a 12. Essa redução pode ser muito útil para se ajustar ao tamanho das imagens

utilizadas na base e encontrar seu número ideal pode diminuir o *overfitting* do modelo.

Este modelo foi salvo nos arquivos como **Modelo4-12Layers**. Apresentou 167237 parâmetros treináveis e 12 camadas.

O resumo de compilação do modelo e sua matriz de confusão dos resultados obtidos estão disponíveis na **figura 41 e 42**.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 activation (Activation)     (None, 30, 30, 32)        0

 max_pooling2d (MaxPooling2D  (None, 15, 15, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 13, 13, 64)        18496

 activation_1 (Activation)   (None, 13, 13, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 flatten (Flatten)           (None, 2304)              0

 dense (Dense)               (None, 64)                147520

 activation_2 (Activation)   (None, 64)                0

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 5)                 325

 activation_3 (Activation)   (None, 5)                 0

=================================================================
Total params: 167,237
Trainable params: 167,237
Non-trainable params: 0
```

Figura 41. Resumo do **Modelo4-12Layers** compilado.

Fonte: Autor

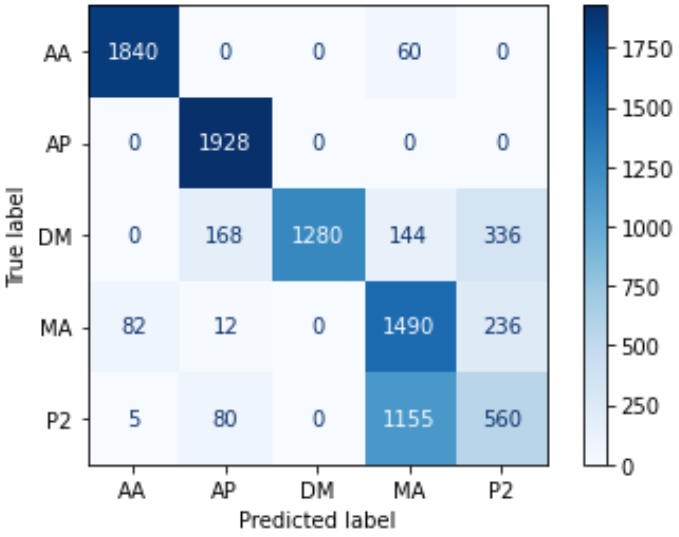


Figura 42. Matriz de confusão da performance do **Modelo4-12Layers.**

Fonte: Autor

Pela matriz de confusão apresentada, pode-se perceber que o modelo teve uma maior confusão nas leituras de desligamento do motor e problema na fase 2.

Essa redução no número de camadas, ao remover uma camada convolucional, uma de pooling e uma de ativação, não apresentou as melhorias esperadas para diferenciar os rótulos mais críticos, logo, voltamos a analisar modelos com 15 camadas.

A próxima alteração foi aumentar a quantidade de neurônios na camada densa, responsável por reduzir os resultados das camadas convolucionais para condensar os resultados nos rótulos esperados. Este modelo foi salvo como **Modelo5-Dense**, com 62181 parâmetros treináveis, 15 camadas e 128 neurônios densos na 11ªcamada.

O resumo de compilação do modelo e sua matriz de confusão dos resultados obtidos estão disponíveis na **figura 43 e 44**.

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_32 (Conv2D)          (None, 30, 30, 32)        896

 activation_32 (Activation)  (None, 30, 30, 32)        0

 max_pooling2d_26 (MaxPoolin  (None, 15, 15, 32)       0
 g2D)

 conv2d_33 (Conv2D)          (None, 13, 13, 32)        9248

 activation_33 (Activation)  (None, 13, 13, 32)        0

 max_pooling2d_27 (MaxPoolin  (None, 6, 6, 32)         0
 g2D)

 conv2d_34 (Conv2D)          (None, 4, 4, 64)          18496

 activation_34 (Activation)  (None, 4, 4, 64)          0

 max_pooling2d_28 (MaxPoolin  (None, 2, 2, 64)         0
 g2D)

 flatten_3 (Flatten)         (None, 256)               0

 dense_6 (Dense)             (None, 128)               32896

 activation_35 (Activation)  (None, 128)               0

 dropout_3 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 5)                 645

 activation_36 (Activation)  (None, 5)                 0

=================================================================
Total params: 62,181
Trainable params: 62,181
Non-trainable params: 0
```

Figura 43. Resumo do **Modelo5-Dense** compilado

Fonte: Autor

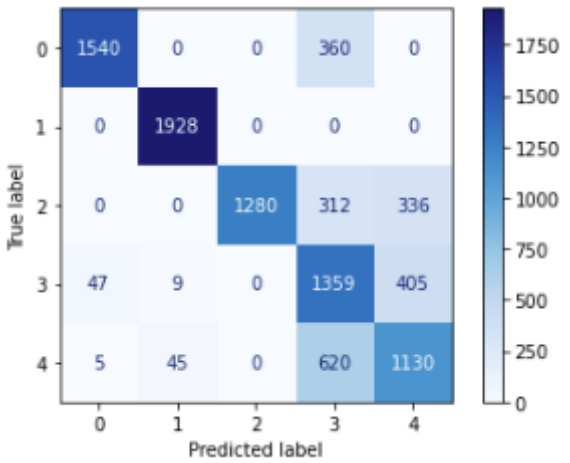


Figura 44. Matriz de confusão da performance do **Modelo5-Dense.**

Fonte: Autor

Apresentou uma melhor diferenciação entre problema na fase 2 e médio atrito, rótulos bem complicados de fazer essa separação. Além de reduzir a quantidade de rótulos confundidos com outros.

O próximo teste foi realizado com as mesmas características do anterior, porém, com menos épocas e batch size ajustado. Este modelo foi salvo como **Modelo6-Dense-ajustado** com 25 epochs e batch_size de 64.(**Figura 45**)

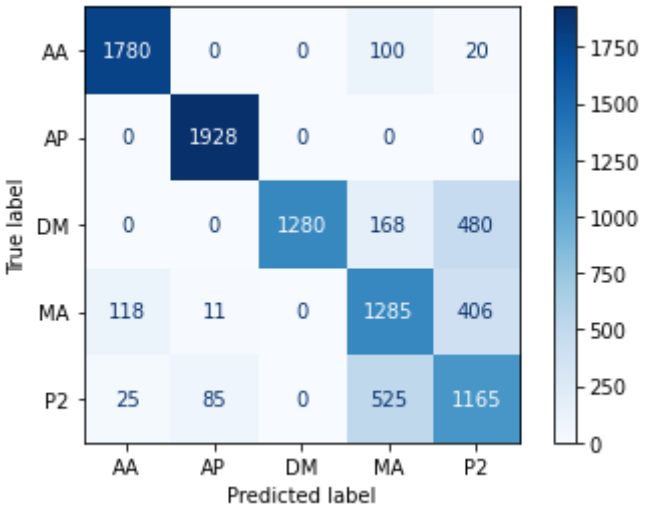


Figura 45. Matriz de confusão da performance do **Modelo6-Dense-ajustado.**

Fonte: Autor

Foi verificado uma distribuição similar ao modelo anterior mas com mudanças não necessariamente positivas para a natureza da aplicação. Um maior acerto de leituras corretamente rotuladas como alto atrito mas confundindo uma quantidade como problema na fase 2 não é necessariamente melhor que menos acertos em alto atrito, mas com todas as leituras alto atrito rotuladas corretamente ou como médio atrito.

As próximas alterações visam alterar o número de camada, mas dessa vez adicionando mais uma camada de convolução, *pooling* e ativação, além de alterar o tamanho do *pooling*. Para estes testes foram

utilizados 25 epochs, batch_size = 64 e *pooling* de (1,1) na 12ª camada. **Modelo7-Pool1. (Figura 46)**

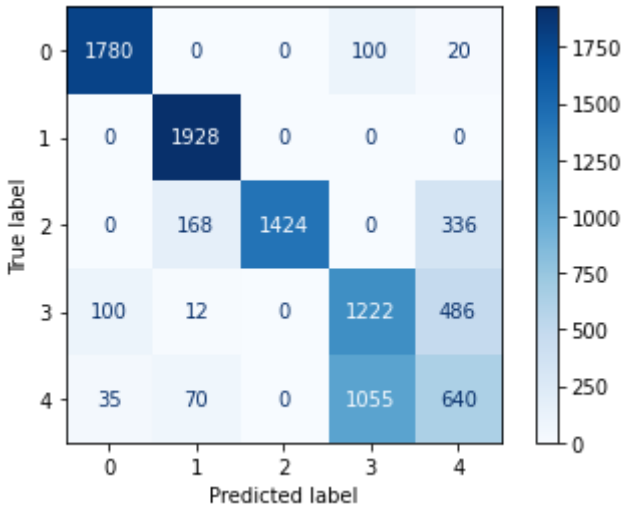


Figura 46. Matriz de confusão da performance do **Modelo7-Pool1.**

Fonte: Autor

Percebe-se que a alteração no tamanho do pooling realizado não apresentou melhorias, na realidade, ajudou a confundir mais as leituras de médio atrito e problema na fase 2.

O próximo teste alterou a primeira camada de *pooling* para utilizar um tamanho de (3,3). **Modelo8-Pool3.**

O resumo de compilação do modelo e sua matriz de confusão dos resultados obtidos estão disponíveis na **figura 47 e 48**.

Layer (type)	Output Shape	Param #
conv2d_20 (Conv2D)	(None, 30, 30, 32)	896
activation_19 (Activation)	(None, 30, 30, 32)	0
max_pooling2d_17 (MaxPooling2D)	(None, 10, 10, 32)	0
conv2d_21 (Conv2D)	(None, 9, 9, 32)	4128
activation_20 (Activation)	(None, 9, 9, 32)	0
max_pooling2d_18 (MaxPooling2D)	(None, 4, 4, 32)	0
conv2d_22 (Conv2D)	(None, 2, 2, 32)	9248
activation_21 (Activation)	(None, 2, 2, 32)	0
max_pooling2d_19 (MaxPooling2D)	(None, 2, 2, 32)	0
conv2d_23 (Conv2D)	(None, 1, 1, 64)	8256
activation_22 (Activation)	(None, 1, 1, 64)	0
max_pooling2d_20 (MaxPooling2D)	(None, 1, 1, 64)	0
flatten_1 (Flatten)	(None, 64)	0
dense_2 (Dense)	(None, 128)	8320
activation_23 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 5)	645
activation_24 (Activation)	(None, 5)	0

Total params: 31,493

Figura 47. Resumo do **Modelo8-Pool3** compilado.

Fonte: Autor

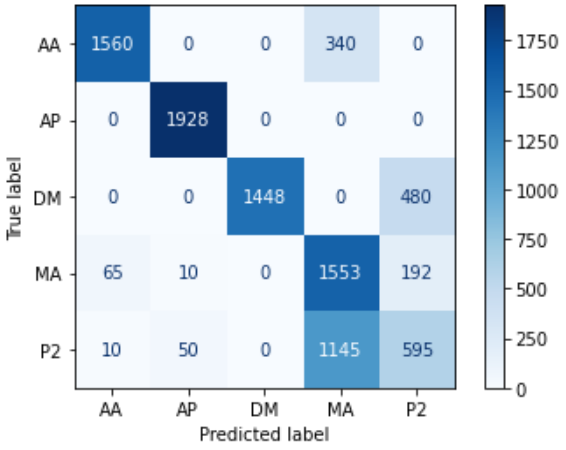


Figura 48. Matriz de confusão da performance do **Modelo8-Pool3.**

Fonte: Autor

Apesar de ter a melhor avaliação de médio atrito entre os modelos testados, não é interessante pois piorou a generalização do algoritmo no quesito separação entre médio atrito e problema na fase 2.

A próxima teve os mesmos parâmetros do modelo anterior, mas alterou o algoritmo de *pooling,* utilizando AveragePooling ao invés de MaxPooling utilizado até então. **Modelo9-AveragePooling**. Sua matriz de confusão dos resultados obtidos estão disponíveis na **figura 49**.

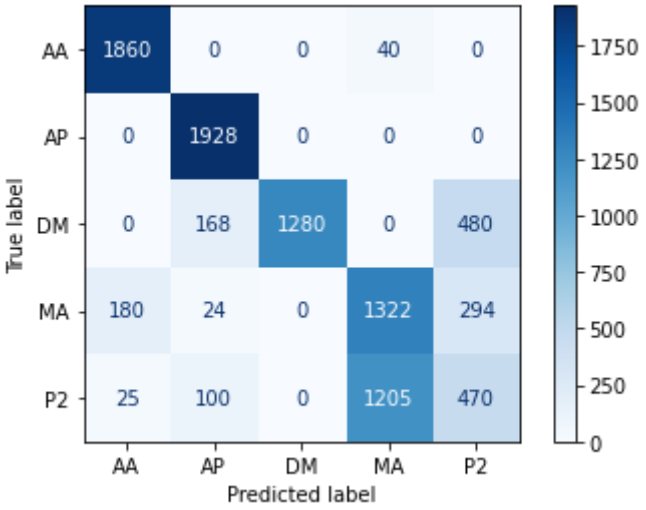


Figura 49. Matriz de confusão da performance do **Modelo9-AveragePooling.**

Fonte: Autor

A utilização do AveragePooling melhorou a classificação das leituras como alto atrito, mas novamente, piorou a confusão entre médio atrito e problema na fase 2.

A próxima alteração foi utilizar um *pooling* na primeira camada inicial com tamanho (4,4), 100 epochs e batch_size = 128. **Modelo10-Pool4**.

O resumo de compilação do modelo e sua matriz de confusão dos resultados obtidos estão disponíveis na **figura 50 e 51**.

Layer (type)	Output Shape	Param #
conv2d_31 (Conv2D)	(None, 30, 30, 32)	896
activation_30 (Activation)	(None, 30, 30, 32)	0
max_pooling2d_26 (MaxPooling2D)	(None, 7, 7, 32)	0
conv2d_32 (Conv2D)	(None, 6, 6, 32)	4128
activation_31 (Activation)	(None, 6, 6, 32)	0
max_pooling2d_27 (MaxPooling2D)	(None, 3, 3, 32)	0
conv2d_33 (Conv2D)	(None, 1, 1, 32)	9248
activation_32 (Activation)	(None, 1, 1, 32)	0
max_pooling2d_28 (MaxPooling2D)	(None, 1, 1, 32)	0
conv2d_34 (Conv2D)	(None, 1, 1, 64)	2112
activation_33 (Activation)	(None, 1, 1, 64)	0
max_pooling2d_29 (MaxPooling2D)	(None, 1, 1, 64)	0
flatten_2 (Flatten)	(None, 64)	0
dense_4 (Dense)	(None, 128)	8320
activation_34 (Activation)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 5)	645
activation_35 (Activation)	(None, 5)	0

Total params: 25,349
Trainable params: 25,349
Non-trainable params: 0

Figura 50. Resumo do **Modelo10-Pool4** compilado.

Fonte: Autor

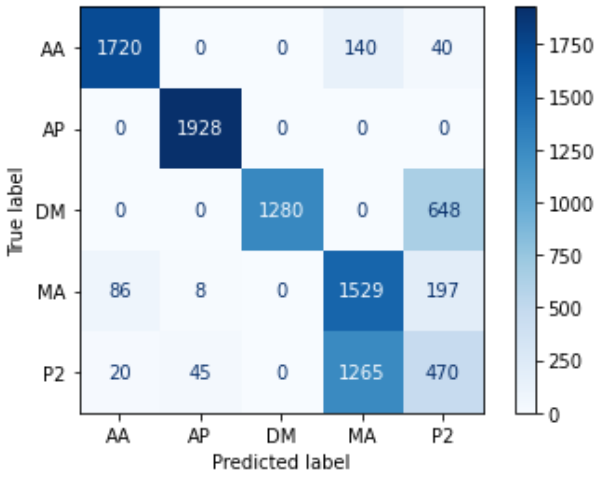


Figura 51. Matriz de confusão da performance do **Modelo10-Pool4.**

Fonte: Autor

Esta tentativa não apresentou melhoras significativas em nenhuma das avaliações.

Como a utilização do AveragePooling apresentou uma melhora, decidimos testar a utilização desse método junto ao melhor modelo definido até agora, o padrão com 15 camadas e 128 neurônios densos na penúltima camada. Este foi salvo nos arquivos como **Modelo11-AveragePooling2**.

O resumo de compilação do modelo e sua matriz de confusão dos resultados obtidos estão disponíveis na **figura 52 e 53**.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 activation (Activation)     (None, 30, 30, 32)        0

 average_pooling2d (AverageP  (None, 15, 15, 32)       0
 ooling2D)

 conv2d_1 (Conv2D)           (None, 13, 13, 32)        9248

 activation_1 (Activation)   (None, 13, 13, 32)        0

 average_pooling2d_1 (Averag  (None, 6, 6, 32)         0
 ePooling2D)

 conv2d_2 (Conv2D)           (None, 4, 4, 64)          18496

 activation_2 (Activation)   (None, 4, 4, 64)          0

 average_pooling2d_2 (Averag  (None, 2, 2, 64)         0
 ePooling2D)

 flatten (Flatten)           (None, 256)               0

 dense (Dense)               (None, 128)               32896

 activation_3 (Activation)   (None, 128)               0

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 5)                 645

 activation_4 (Activation)   (None, 5)                 0

=================================================================
Total params: 62,181
Trainable params: 62,181
Non-trainable params: 0
_________________________________________________________________
```

Figura 52. Resumo do **Modelo11-AveragePooling2** compilado.

Fonte: Autor

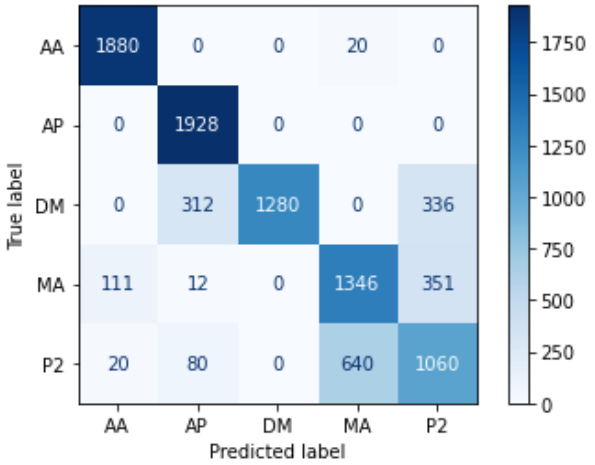


Figura 53. Matriz de confusão da performance do **Modelo11-AveragePooling2.**

Fonte: Autor

Este modelo aumentou a acurácia do modelo, que chegou a 80% na base de validação, mas aumentou a quantidade de leituras distribuídas em mais de 2 rótulos por classificação.

Por fim, testamos a alteração dos neurônios da primeira camada densa, de 128 para 256 no modelo padrão. Mantendo o batch_size = 64 e 50 epochs. Salvo como **Modelo12-Dense256** e seu resultado na **figura 54.**

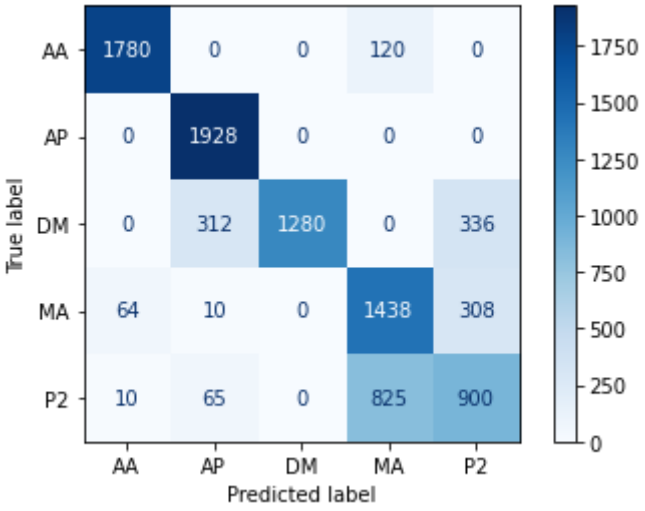


Figura 54. Matriz de confusão da performance do **Modelo12-Dense256.**

Fonte: Autor

Este teste também mostrou uma piora em relação ao modelo padrão.

O objetivo da testagem intensiva de modelos com poucas variações estratégicas entre os modelos é ter uma noção mais clara do desempenho

deste método de predição e definir o componente final, além de estudar como se comportam os parâmetros estudados.

A partir destes objetos, a alteração de 64 para 128 neurônios densos na última camada foi a que mais melhorou a performance do algoritmo no quesito principal desta predição, que é a separabilidade entre **médio atrito** e **problema na fase 2**.

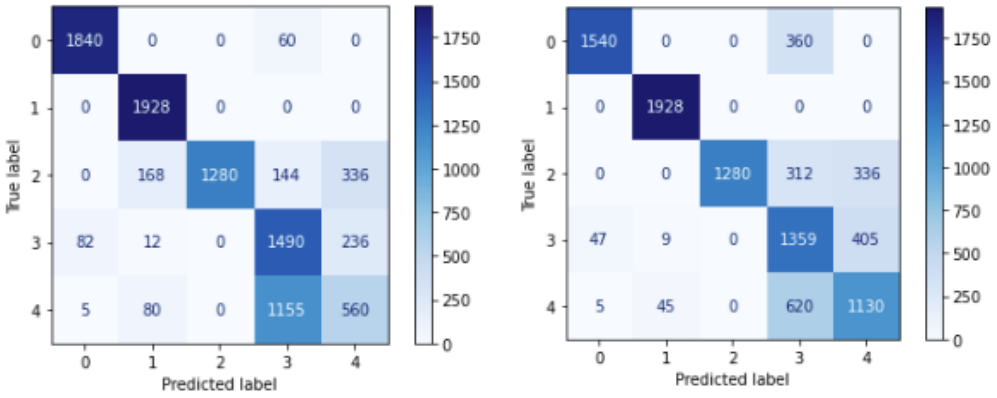


Figura 55 e 56. Matrizes de confusão da performance do **Modelo4-12Layers e Modelo5-Dense.**

Fonte: Autor

Além desta análise disponível nas **figuras 55 e 56**, é correto perceber que o modelo pode perceber algumas leituras de atraso de partida como **médio atrito** e **problema na fase**. Isso pode ser justificado pelo defeito atraso partida se sobressair na classificação da própria Vale, enquanto o modelo treinado não tem ordem de prioridade. Uma leitura teve um comportamento de atraso mas também um pouco de ruído direcionado ao médio atrito, ela será rotulada como atraso partida pela Vale.

5.2 Definição do modelo

A partir dos resultados obtidos, o modelo que obteve a melhor performance para rotular as leituras obtidas obteve 79% de acurácia. Este modelo utilizou 62181 parâmetros treináveis e 15 camadas, além de 128

neurônios densos na 11ª camada. Seu resultado está disponível na **figura 57.**

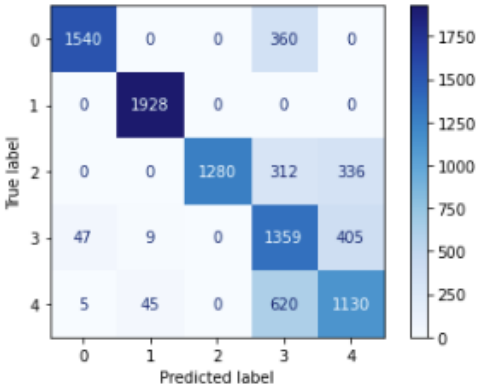


Figura 57. Matriz de confusão da performance do **Modelo5-Dense**, definido como padrão**.**

Fonte: Autor

Este modelo foi treinado com menos épocas do que os outros, o que se mostrou eficiente para reduzir o *overfitting* causado pelo treinamento extensivo.

5.3 Visualização

A fim de resolver o problema do corpo técnico da Vale, entendemos que dar a eles a oportunidade de visualizar direito os dados com as ferramentas auxiliares é a parte mais importante, afinal, qualquer marcação e indicação de manutenção pode ser verificado adicionalmente pelo técnico antes de sua correção.

Para este objetivo, foi proposto uma visualização composta pelo gráfico da corrente pelo tempo, seu mapa de calor proveniente do produto gramiano e a distribuição das probabilidades de rótulos (**figuras 58 a 72**). Apresentar não só o rótulo mais provável mas também a indicação de quais rótulos o algoritmo pode estar se confundindo pode ajudar na avaliação temporal das leituras quando combinadas, além de gerar um relatório mais robusto sobre o funcionamento de cada aparelho.

Convém ressaltar que a última camada do modelo faz uma ativação em função sigmóide, o que ajuda a normalizar as probabilidades de cada rótulo.

Exemplos de Visualização

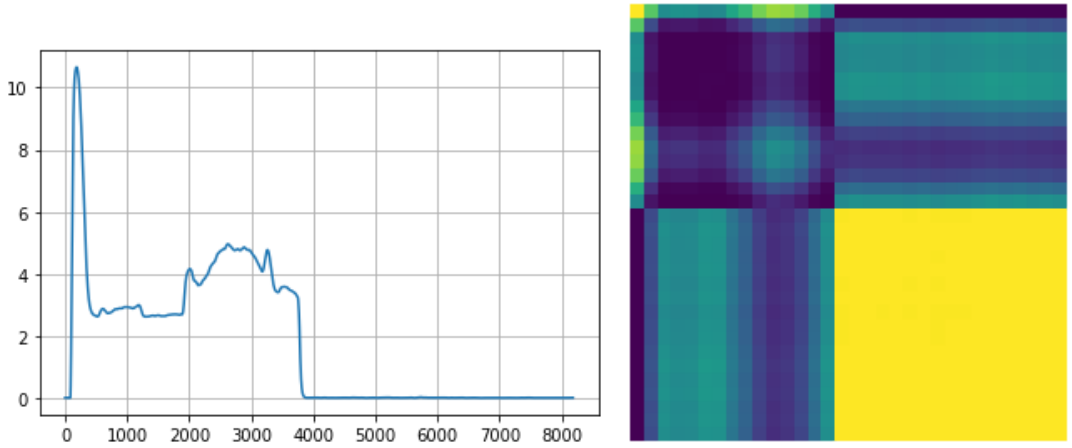


Figura 58 e 59. Gráfico de medição de corrente e produto gramiano da Leitura MCH004W21B_N_20220219_145728, rotulada como Alto Atrito.

Fonte: Autor

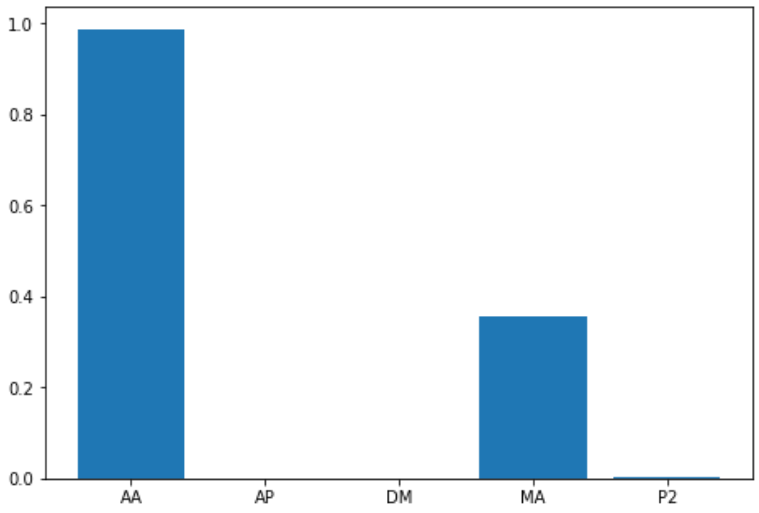


Figura 60. Gráfico das probabilidades de classificação da leitura MCH004W21B_N_20220219_145728 pelo modelo.

Fonte: Autor

A saída do classificador apresenta uma normalização. Neste exemplo apontado pela Vale como Alto Atrito, nosso classificador afirma ser um caso de alto atrito com uma certeza alta, mas também indica uma

tendência a rotular como médio atrito, o que é esperado pela natureza parecida dos dois rótulos.

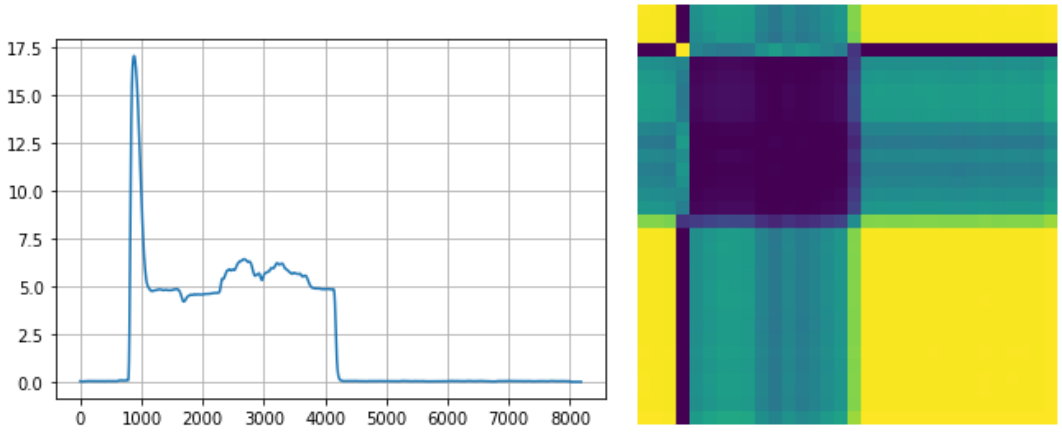


Figura 61 e 62. Gráfico de medição de corrente e produto gramiano da Leitura MCH004W22A_N_20191121_211054, rotulada como Atraso Partida.

Fonte: Autor

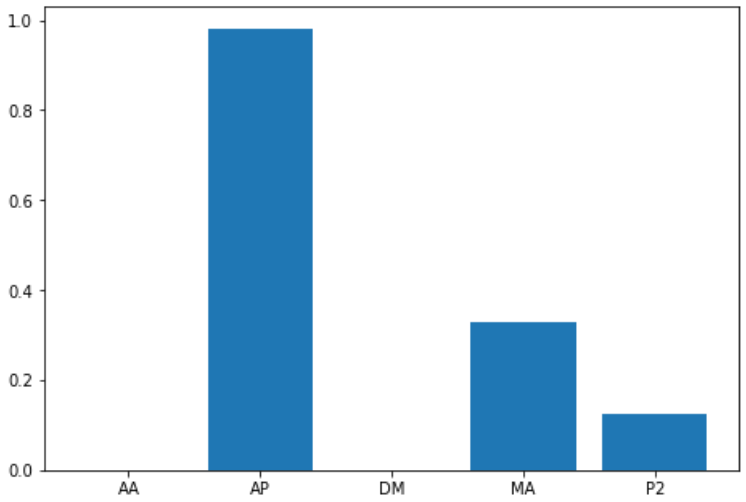


Figura 63. Gráfico das probabilidades de classificação da leitura MCH004W22A_N_20191121_211054 pelo modelo.

Fonte: Autor

Nessa leitura, classificada pela Vale como atraso partida, pudemos avaliar que há o atraso no começo do acionamento, caracterizado pelo bloco amarelo antes da primeira “piscina” no mapa de calor. Mas também, ao avaliar o resto da leitura, podemos perceber que há tendências de médio atrito na fase 3, o que é apontado pela classificação disposta, sendo sua segunda alternativa mais provável.

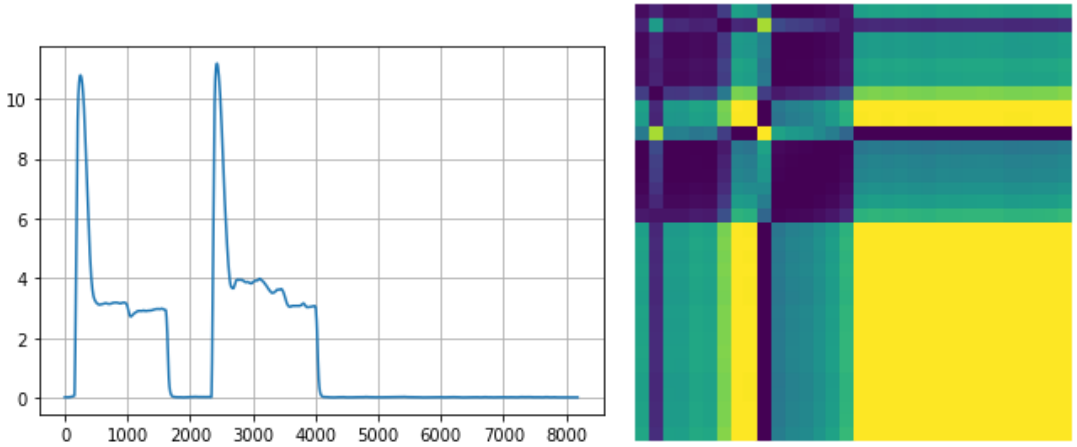


Figura 64 e 65. Gráfico de medição de corrente e produto gramiano da Leitura MCH004W22A_N_20190824_212135, rotulada como Desligamento de Motor.

Fonte: Autor

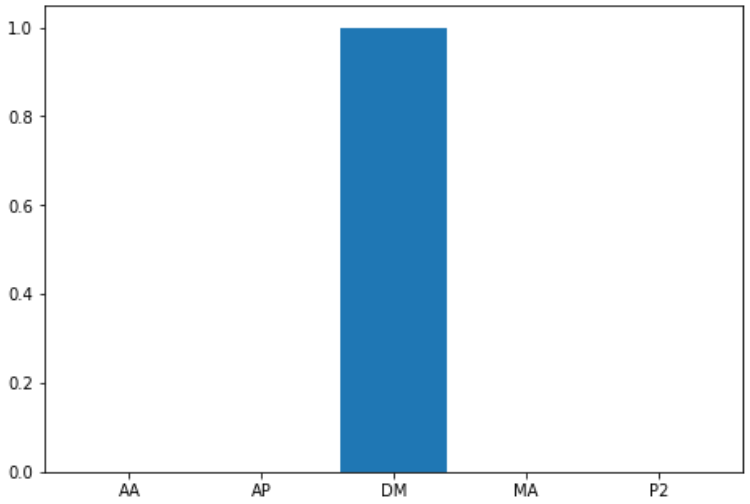


Figura 66. Gráfico das probabilidades de classificação da leitura MCH004W22A_N_20190824_212135 pelo modelo.

Fonte: Autor

Esta leitura, classificada como desligamento de motor pela Vale, também foi classificada como pelo nosso algoritmo como desligamento de motor, percebe-se claramente que foi necessário re-acionar o motor desde o início e continuar o deslizamento da agulha de mudança de via para completar seu funcionamento. Ao mostrar tanto o gráfico da corrente e seu mapa de calor, a classificação fica melhor justificada.

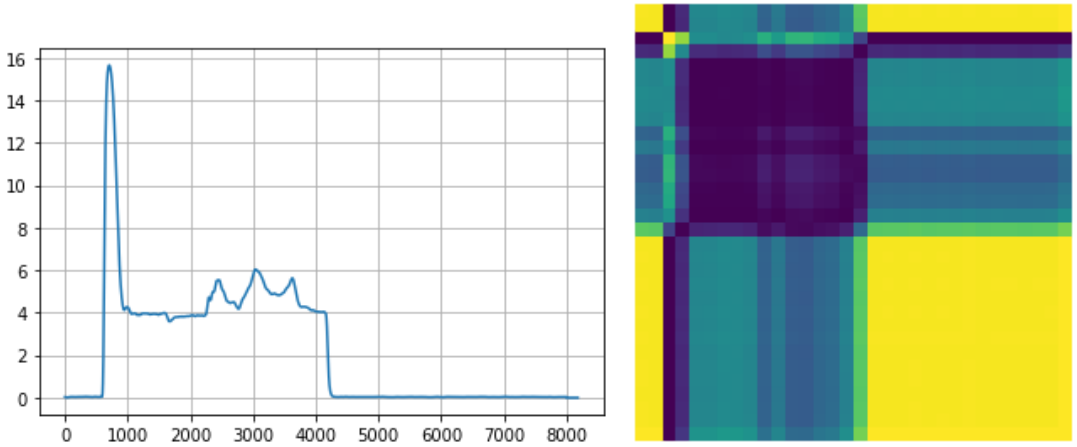


Figura 67 e 68. Gráfico de medição de corrente e produto gramiano da Leitura MCH004W21A_N_20191121_221943, rotulada como Médio Atrito.

Fonte: Autor

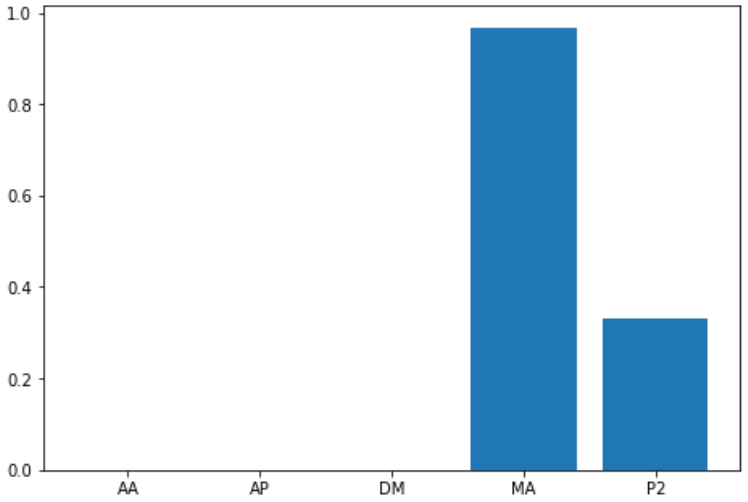


Figura 69. Gráfico das probabilidades de classificação da leitura MCH004W21A_N_20191121_221943 pelo modelo.

Fonte: Autor

Essa leitura foi classificada pela Vale como médio atrito. nota que essa separação entre médio atrito e problema na fase 2 foi a mais confusa na avaliação geral desse algoritmo, realmente há muitas características em comum e uma avaliação parcial de problema na fase 2 ainda está bem presente na visualização, mesmo o médio atrito sendo o rótulo com maior grau de certeza.

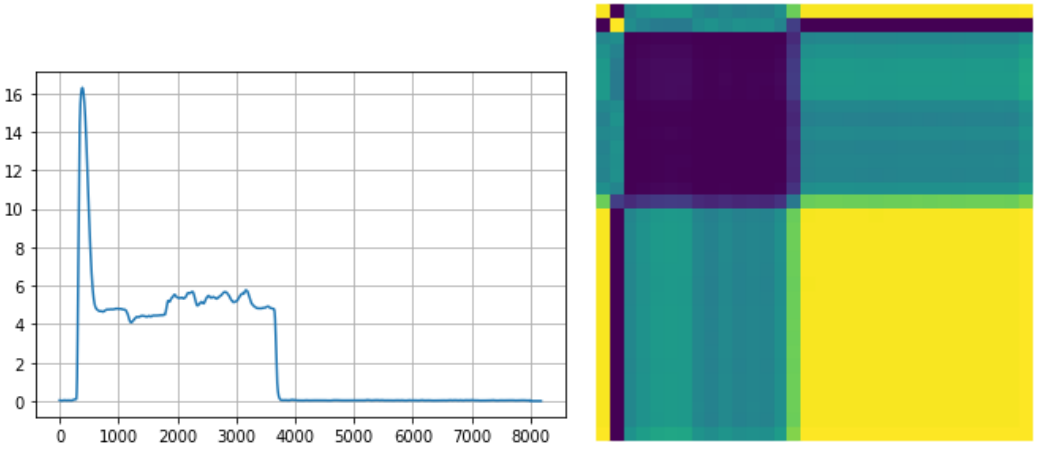


Figura 70 e 71. Gráfico de medição de corrente e produto gramiano da Leitura MCH004W22A_N_20191004_162851, rotulada como Problema na Fase 2.

Fonte: Autor

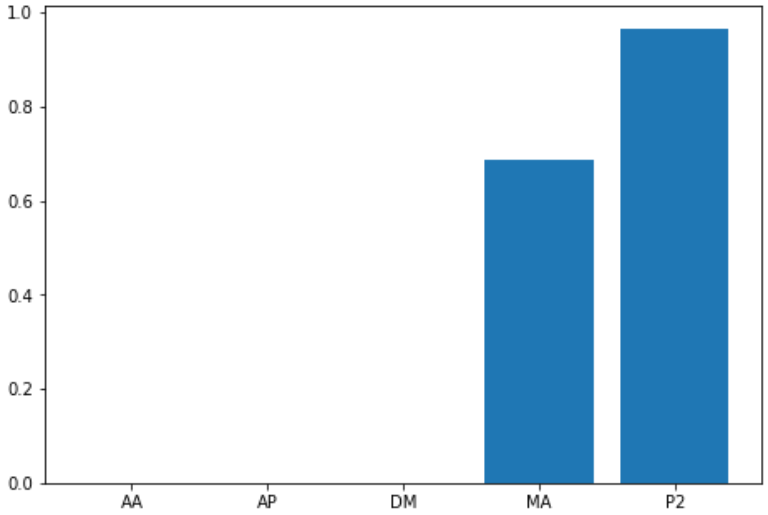


Figura 72. Gráfico das probabilidades de classificação da leitura MCH004W22A_N_20191004_162851 pelo modelo.

Fonte: Autor

Por último, uma leitura classificada pela Vale como problema na fase 2. Inspecionando visualmente podemos notar a similaridade com a leitura anterior, com a "quebra" característica de uma leitura problema na fase 2 mais acentuada, mas com as oscilações típicas de médio atrito na fase 3. A Probabilidade parcial dos rótulos disposta pelo nosso modelo condiz bem com o que se espera de um modelo que aprendeu a discernir estas características ao longo de toda a curva.

5.4 Desenvolvimento da arquitetura

A organização inicial proposta na especificação foi mantida como planejada, com os 3 módulos de tratamento, classificação e visualização.

Os resultados do ciclo de prototipagem foram muito importantes para a definição da melhor solução para a base de dados ter o menor viés possível e o classificador ter a maior capacidade de generalização.

Apesar de utilizar como entrada as leituras de extensão .txt das máquinas, uma possível alteração nesse formato transmitido pela própria máquina poderia economizar um bom tempo caso transmitisse apenas uma lista com os valores de corrente por tempo.

Também cabe ressaltar que a avaliação temporal da máquina, como o monitoramento de degradação não foi parte de estudo nesse projeto, mas seria um adicional interessante. O algoritmo poderia ter um grau maior de certeza ao aferir não só a leitura atual, mas também sua comparação com as últimas leituras da mesma máquina.

A aplicação desse classificador para funcionamento em tempo real precisa de adaptações à própria máquina *in loco*, mas pode se fazer de parte integral deste projeto, adicionando mais um módulo de transmissão na etapa anterior ao tratamento de dados para conectar ao tratamento de dados. Pode-se incrementar a arquitetura com uma aplicação web na outra ponta do projeto, utilizando das visualizações disponíveis na saída de componente inteiro, como exemplificado na **figura 73**.

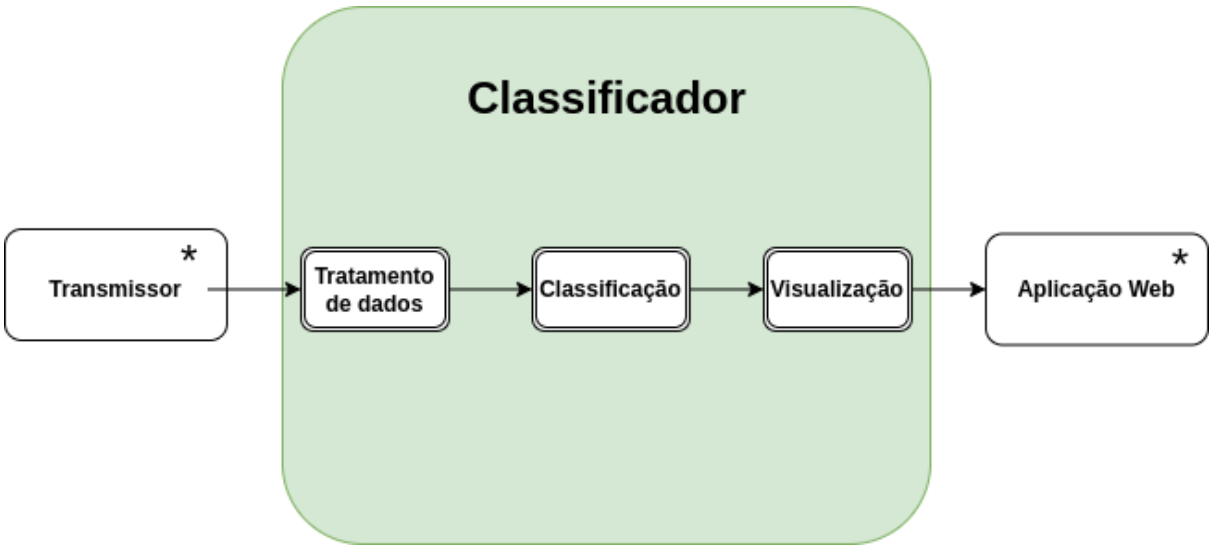


Figura 73. Diagrama da arquitetura com incrementos.
Fonte Autor

6. Conclusão

O maior problema encontrado nesta problemática de diferenciar os tipos de defeitos encontrados nesse tipo de máquina é a necessidade de conhecimento técnico especializado, as leituras utilizadas para treinamento necessitam ter certeza de sua disposição.

Os dados utilizados foram fornecidos pelo atual classificador da Vale que tem uma acurácia estimada de 95%, ou seja, ao tentar reproduzir a classificação desenvolvida, temos um distanciamento do valor ideal mesmo que na verificação empírica nossa classificação esteja melhor treinada.

Apesar disso, o tratamento de dados proposto e a utilização de técnicas diferentes de visualização dessas leituras provocou a identificação de novos padrões, o que reflete em uma melhora na capacidade de *debug* do sistema.

O sistema não precisou de técnicas avançadas de geração de novas imagens, como rotações, distorções, inclinações por se tratar de uma

imagem gerada por nós antes de sua utilização. Esse fator certamente contribuiu para conseguirmos treinar um modelo com poucas imagens e ainda assim atingir uma boa acurácia.

O modelo final atingiu uma acurácia de 77% em sua classificação primária (**figura 74**), mas ao utilizar uma visualização das probabilidades parciais, no caso, apenas as duas maiores probabilidades das leituras analisadas, o modelo englobou 95,5 % dos casos em sua classificação (**figura 75**).

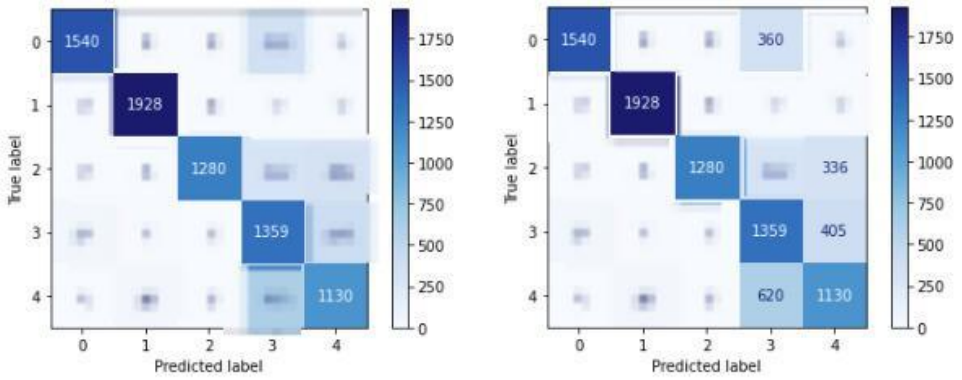


Figura 74 e 75. Matrizes de confusão da performance do modelo padrão, destacando um único rótulo e os dois mais prováveis.
Fonte Autor

O projeto teve que ser avaliado em mais de uma frente, no que faz sentido para sua utilização como ferramenta auxiliar. A partir do compreendimento de que a multi-classificação de uma mesma leitura pode apresentar melhora significativa na segurança do aparelho analisado, o estudo se apresentou muito útil e a forma de visualização dos resultados foi o maior diferencial deste projeto.

A popularização da ciência de dados contribuiu bastante para o rápido desenvolvimento e aplicação deste projeto. O resultado pode ser avaliado como um projeto de ciência de dados, em que é necessário um profundo conhecimento do funcionamento específico do que se quer analisar. Em várias partes do desenvolvimento deste, os comportamentos

mecânicos do aparelho e seus reflexos na classificação apresentaram padrões claros que podem fazer parte de um dossiê mais completo, visto que essa tese busca complementar um trabalho posterior de pesquisa em degradação temporal dos mesmos tipos de máquina.

A próxima abordagem para a resolução deste problema poderia ser uma forma de validar o aprendizado deste modelo com um aprendizado não-supervisionado. A identificação de *clusters* de leituras defeituosas similares entre si pode ser uma forma mais **rápida** de cruzar os dados com uma classificação confiável. Idealmente o aprendizado não supervisionado pode se fazer valer dessas características em comum entre as leituras para até descobrir novos tipos de defeitos não mapeados atualmente pela Vale. Pode-se enxergar melhor as diferenças entre as duas na **figura 76.**

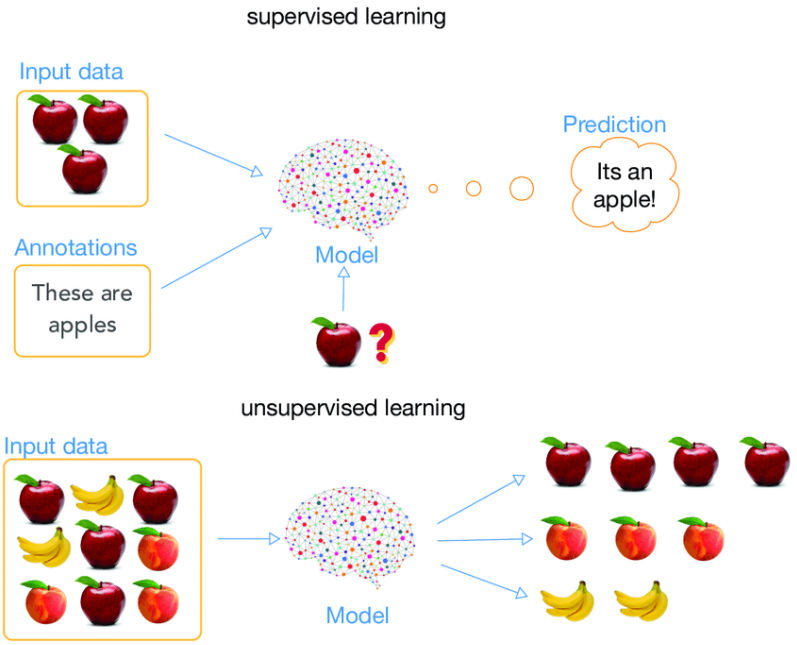


Figura 76. Ilustração do processo de aprendizado supervisionado x não-supervisionado.
Fonte: Background Augmentation Generative Adversarial Networks (BAGANs): Effective Data Generation Based on GAN-Augmented 3D Synthesizing

A alimentação do sistema para treinamento poderia utilizar diferentes conjuntos de classificações não supervisionadas de um

algoritmo auxiliar e testar a assertividade do treinamento deste modelo com esses dados. Com o resultado deste, a comparação com as classificações humanas disponíveis a pesquisa teriam um duplo fator de autenticação e poderíamos descartar ou reciclar leituras com classificações dúbias.

Por último, a importância dessa multi-classificação para a Vale é poder extrair o melhor do aprendizado presente nos neurônios do modelo. Cada defeito tem razões diferentes, como desligamento motor geralmente indica falhas na alimentação e problemas de atrito indicam desgaste entre os trilhos ou sujeiras por exemplo, analisar cada leitura fazendo uso de sua operacionalidade em diversas frentes pode agregar muito valor à manutenção destes componentes.

7. Referências

1. **Artificial Intelligence : a Modern Approach** - Peter Norvig, Stuart Russel, 2010.
2. A review of prognostics and health management of machine tools - Marco Baur, 2020
3. PROFUNDO APLICADOS AO DIAGNÓSTICO E PROGNÓSTICO NÃO SUPERVISIONADO DE FALHAS - David Fernandes Neves Oliveira, 2020.
4. Visualizing and understanding convolutional networks. In: European conference on computer vision - Steven Zeiler, Rob Fergus , 2014.
5. SYSTEMATIC LITERATURE REVIEW: ARTIFICIAL INTELLIGENCE TECHNIQUES APPLIED IN CONDITION-BASED MAINTENANCE MODELS FOR RAILWAYS APPLICATIONS - Pedro Pinheiro, 2021
6. Classificador de Estado Operacional e Preditores baseados em Inteligência Computacional para Aparelhos de Mudança de Via Ferroviária - Edwen Minaguchi Endo, 2020
7. APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA A DETECÇÃO E DIAGNÓSTICO DO ESTADO DE OPERACIONALIDADE DE SISTEMAS: UM ESTUDO DE CASO EM AMVS FERROVIÁRIOS - Macilio da Silva Ferreira, 2021
8. A statistical-based approach to identify the runtime operational state and condition of railway track switches - Rafael Yuji Yokowo, 2021