

# Emprego de Inteligência Artificial no Reconhecimento e Extrapolação de Dados de Placa de Motores Industriais/Comerciais

Ricardo Ataide de Oliveira Junior, Ricardo de Morais Rainha

**Resumo**—Nem sempre a identificação de um motor de indução é um processo intuitivo, uma vez que a placa de identificação do equipamento a qual carrega consigo valiosas informações a respeito de especificações técnicas do motor é passível de avarias a depender de diversos fatores, tais como o material de sua construção e as condições a qual a mesma fora exposta. Buscando auxiliar no processo de identificação de tais equipamentos, este trabalho propõe um aplicativo para dispositivos móveis que, a partir de uma imagem/fotografia de uma placa de identificação de um motor de indução realiza a leitura de informações relevantes a sua identificação, e, busca estimar as informações não legíveis, para, enfim, baseando-se nos k-vizinhos mais próximos das informações coletadas e estimadas a respeito do equipamento, exiba uma lista com os equipamentos registrados em nosso dataset, os quais mais se assemelham ao equipamento analisado. **Palavras-chave:** Motores elétricos. OCR. Dados ausentes. Placa de identificação.

**Abstract**—Identifying an induction motor is not always an intuitive process, as the equipment nameplate, which provides valuable information about the technical specifications of the motor, is subject to damage due to several factors, such as construction material and conditions to which it was exposed. . Seeking to assist in the identification process of such equipment, this work proposes an application for mobile devices that, from an image/photograph of the nameplate of an induction motor, reads information relevant to its identification, and seeks to estimate the unreadable information, and finally, based on the nearest k-neighbors of the collected and estimated information about the equipment, display a list with the equipment registered in our dataset, which are more similar to the analyzed equipment. **Keywords:** Electric motors. OCR. Missing data. Nameplate.

## I. INTRODUÇÃO

Apesar de serem reconhecidos pela sua longa longevidade, motores de indução não estão à salvo de problemas, quer sejam mecânicos, elétricos, ou mesmo a necessidade de substituição para proveito de tecnologias mais recentes. Por conta de tais fatores, é essencial que um componente de tamanha importância para a continuidade de processos industriais, ou mesmo, o funcionamento de outros equipamentos, seja fácil de se identificar para tornar o processo de manutenção o mais eficiente possível. Apesar de possuírem uma placa de identificação com informações valiosas a respeito de especificações técnicas, mecânicas, e, identificação de modelo fornecida pela fabricante, nem sempre é fácil realizar a leitura deste componente uma vez que, devido a condições ambientais a qual o equipamento fora exposto é possível que a placa seja danificada ou contenha ruído, o que torna inviável a identificação de certos dados a respeito do equipamento a partir da simples ação de leitura da placa. Tal dificuldade

caracteriza-se como um atraso no processo de manutenção, pois a identificação de tais informações é essencial para substituição ou manutenção do próprio equipamento.

Com o objetivo de diminuir o atraso neste processo de manutenção, este trabalho propõe uma maneira de auxiliar na etapa de identificação do equipamento, através de um aplicativo para dispositivos móveis, (inicialmente desenvolvido para aparelhos *android*), o qual utilizando-se da imagem da placa de identificação do motor de indução, e, com o auxílio de ferramentas OCR (do inglês *Optical Character Recognition*), juntamente com algoritmos de *machine learning* busca identificar os dados presentes na placa do motor elétrico, estimar os dados cujo a leitura não foi possível, para por fim listar dentre os equipamentos catalogados em nossa base de dados os que mais se assemelham às informações extraídas da imagem fornecida.

## II. TRABALHOS SEMELHANTES E METODOLOGIA GERAL

Ao procurarmos por trabalhos com objetivos similares aos nossos, foi identificado uma escassez de aplicações similares a de extração de informações em placas de identificação de equipamentos. Boa parte das publicações relacionadas placas de identificação são voltadas para a identificação de placas de automóveis [1] [2], no entanto estes trabalhos tem por objetivo principal a identificação do objeto e reconhecimento do conteúdo descrito neste, o que não contempla toda as nossas necessidades, uma vez que desejamos realizar a classificação da informação extraída da imagem, processo este, essencial para nossos objetivos, uma vez que além de extrair os dados da placa do motor de indução, é necessário identificar a qual parâmetro do motor tal informação se refere.

Uma das poucas propostas em que identificamos objetivos que se assemelham em muito aos nossos, é a solução desenvolvida por Chen et.al [3] onde é proposta uma arquitetura dividida em 02 etapas, identificação e reconhecimento do texto, no entanto na proposta feita por Chen et.al fora utilizado além de um *dataset* gerado genericamente (possuindo 410 mil amostras), um outro *dataset* com imagens reais de placas de identificação de transformadores, o qual consiste de em um total de 2500 imagens, o que torna a reprodução do trabalho inviável uma vez que não possuímos acesso à um acervo tão extenso de imagens de qualidade para placas de identificação de motores elétricos. Um outro ponto é que a proposta não aborda uma etapa de classificação da cadeia de caracteres reconhecida na entrada, sendo esta uma atividade essencial

para a utilização de um modelo que estima os dados não legíveis na placa de identificação.

Buscando por alternativas para o problema de classificação das cadeias de caracteres presentes na imagem, nos deparamos com algumas propostas apresentadas por Bensch, Popa e Spille [4] para a realização de tal atividade. Dentre as propostas apresentadas por Bensch, Popa e Spille uma das mais recentes é o CUTIE [5] (*Convolutional Universal Text Information Extractor*).

Em poucas palavras o CUTIE funciona da seguinte forma: primeiramente é realizada a extração do texto presente no documento através de uma ferramenta OCR, com este texto em mãos é então realizada a construção de um *grid* o qual, é basicamente uma matriz contendo as cadeias de caracteres extraídas da imagem, as quais são separadas por um espaço em branco. Por se tratar de uma matriz bidimensional, o *grid* é capaz de representar informações com respeito às posições relativas de cada cadeia de caracteres no documento original. Desta maneira a proposta utiliza-se deste *grid* juntamente com a imagem do documento para comporem a entrada de uma rede convolucional na qual se obtém como saída um *grid* classificado.

A solução é bastante promissora, no entanto possui um custo elevado para se rotular o *dataset* (cada cadeia/classe da imagem deve ser rotulado), além da própria necessidade de um *dataset* suficientemente grande para garantir a generalização da rede neural, ainda mais considerando a grande variabilidade de modelos de placa de motores elétricos, o que implica em uma variação no posicionamento dos parâmetros do equipamento, assim como evidenciado pela figura 1. Uma outra limitação da metodologia está na dependência da ferramenta de OCR, como estamos utilizando uma ferramenta de OCR externa, é necessário uma qualidade mínima da imagem para que a ferramenta seja capaz de extrair corretamente os caracteres presentes na imagem, no entanto, como não estamos lidando com imagens de documentos estruturados (fundo originalmente branco, texto preto) as ferramentas de OCR variam muito quanto ao desempenho no processo de extração dos caracteres, o que pode levar a um maior custo temporal para a convergência da rede, além da necessidade de uma maior quantidade de dados disponíveis para o treinamento de modo a garantir uma boa generalização da solução. No entanto, obtivemos acesso à uma pequena quantidade de imagens, as quais possuem uma qualidade que varia muito, o que leva a resultados variáveis nas ferramentas de OCR, tornando a reprodução da solução bastante propensa à falhas.

Esta limitação em nosso *dataset* de imagens de placas de identificação de motores elétricos levou-nos a procurar por uma solução que não dependa do treinamento de redes profundas (de autoria própria) para a extração de tais informações, porém que fosse capaz de extrair um conjunto de dados legíveis e, posteriormente, estimar as variáveis deste conjunto que não puderam ser extraídas na primeira etapa. Visando isso, propomos a arquitetura apresentada na figura 2.

De modo geral, nossa solução se divide em 03 etapas principais, as quais são: leitura dos dados presentes na placa de identificação do equipamento através da utilização de uma ferramenta de OCR, extração/estimação dos dados que

por algum motivo não puderam ser extraídos a partir da imagem da placa de identificação, e a execução de um modelo treinado do algoritmo KNN qual recebe como entrada um vetor completo (com os dados originalmente lidos e estimados da placa de identificação do motor elétrico) para então retornar uma lista dos equipamentos dentro de nosso *dataset*, que mais se assemelham a amostra utilizada como entrada. Cada uma das etapas da solução serão descritas das seções IV - VI.

### III. CRIAÇÃO DO *dataset*

#### A. *Imagens*

Para realização de testes com a ferramenta de OCR assim como testes com o processo de classificação das partes do texto extraído, foi gerado um pequeno *dataset*, com imagens de placas de motores elétricos extraídas dos sites de venda Mercado Livre e Olx além de utilizar-se também do agregador de imagens do Google. A partir destas fontes, formamos um *dataset* com aproximadamente 207 imagens, com dimensões que variam de 267 X 189 - 2808 X 2106, das fabricantes Weg, Hércules, Nova Motores, Herbele e outras. Ao gerar o *dataset*, notamos que equipamentos trifásicos, em geral, possuem mais informações em sua placa de identificação, em especial, equipamentos monofásicos da Weg (os quais conseguimos mais amostras) não apresentam informações a respeito do rendimento e ocasionalmente do fator de potência e índice de proteção do equipamento (salvo algumas exceções). Por conta disto, separamos o *dataset* de imagens em equipamentos em que era possível observar o fator de potência, rendimento e proteção do equipamento, e imagens de placas em que esta tarefa não era possível ou ocorria raramente.

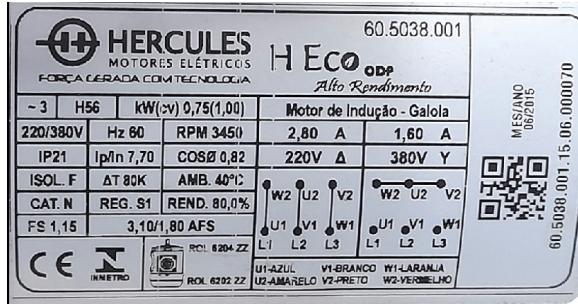
#### B. *Dados de motores elétricos*

Para montar um *dataset* com informações técnicas a respeito de motores de indução, foram exploradas as informações presentes nos catálogos da Weg [6] e Hércules motores elétricos [7] (ambos os catálogos estão disponíveis no próprio site de suas respectivas fabricantes), juntamente com as informações que estão presentes na maioria das placas de motores elétricos (para este passo, utilizamos as imagens do *dataset* que montamos). A partir da intersecção dos campos presentes em todas as fontes citadas, elegemos um conjunto de 10 parâmetros, os quais são utilizados para a construção do nosso *dataset* tabular com as especificações dos motores elétricos. *Dataset* este, que é utilizado para o treinamento dos modelos de imputação das variáveis ausentes.

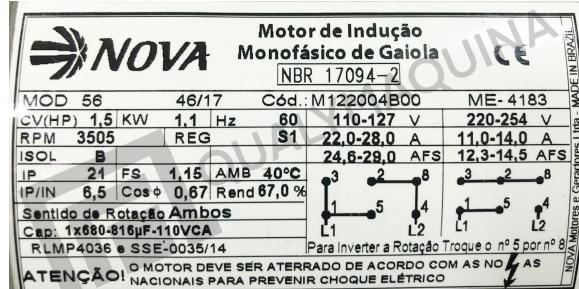
Foram escolhidos 10 parâmetros incluindo o rendimento e fator de potência (informações frequentemente ausentes em placas de motores monofásicos) devido a boa correlação que encontramos entre o fator de potência e rendimento com parâmetros bastante relevantes, tais como a potência e corrente nominal do equipamento (a figura 2 apresenta o mapa destas correlações).

Os campos escolhidos para compor nosso *dataset* foram:

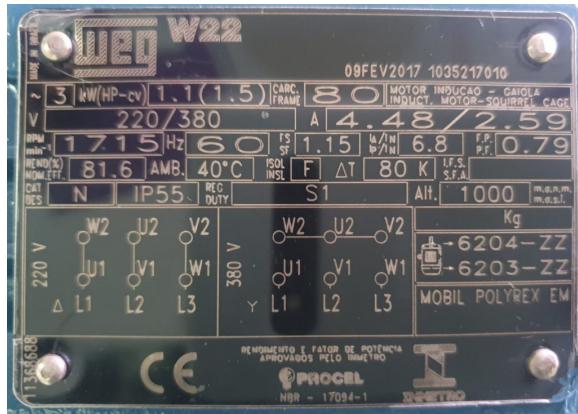
- Tensão nominal
- Corrente nominal
- Potência



(a) Hércules motores



(b) Nova



(c) Weg trifásico



(d) Weg monofásico

Figura 1: Diferenças entre placas de motores de elétricos das fabricantes Hércules motores, Nova, e Weg.

- RPM
- Carcaça
- Relação entre a corrente de pico e corrente nominal (Ip/In)
- Rendimento
- Fator de potência
- Fator de serviço ( $\cos(\phi)$ )
- Índice de proteção.

Por fim, fora construído um *dataset* com especificações de um total de 13634 equipamentos, sendo 13462 da fabricante Weg, e 172 da Hércules Motores.

#### IV. EXTRAÇÃO DOS DADOS

A estratégia adotada para a extração e classificação dos dados presentes na placa do equipamento, baseia-se em utilizar uma solução OCR para a extração dos caracteres, e, criar um conjunto de regras com expressões regulares para realizar uma busca por cadeias-chave de caracteres as quais, apresentam-se sempre nas proximidades do valor do parâmetro que estamos buscando classificar.

Ao buscar por uma solução OCR que melhor atendesse as nossas necessidades, primeiramente foram realizados testes com a ferramenta de OCR gratuita e de código aberto atualmente mantida pelo Google, o pytesseract. No entanto, apesar de pré-processamentos realizados na imagem (recorte, alongamento de contraste, *thresholding*) a ferramenta não identificava os textos presentes na fotografia da placa do motor elétrico. Após descartar a solução, foram realizados testes

com ferramentas *online* que disponibilizam uma API gratuita, sendo que, a melhor ferramenta que encontramos foi o OCR Space [8]. A qual consegue extrair uma parcela considerável do conteúdo de texto presente na imagem, apesar de não classificá-lo.

Após extrair o conteúdo textual da imagem, foi realizada a criação do conjunto de regras com expressões regulares, as quais foram geradas baseadas em diversos testes utilizando a API da ferramenta OCR para o reconhecimento dos caracteres presentes nas imagens presentes no *dataset* que criamos.

#### V. ESTIMANDO OS DADOS AUSENTES

Para estimar os dados ausentes do motor elétrico, ou seja, os campos que não puderam ser identificados a partir da etapa de extração de dados com a ferramenta de OCR, à priori, escolhemos a abordagem que pareceu ser a mais intuitiva para realização de estimativas de parâmetros, ou seja, modelos de regressão. No entanto, modelos de regressão em sua forma de aplicação padrão são treinados para um conjunto bem definido de variáveis de entrada, e para a solução que desejamos propor, o conjunto de variáveis de entrada é incerto, pois apesar de termos bem definidas as variáveis a serem lidas pelo OCR, a priori não conhecemos quais serão as variáveis que a ferramenta de OCR será capaz de extrair da imagem para compor nosso conjunto de entrada, o que torna esta utilização padrão de modelos de regressão inapropriada para a nossa aplicação.

Ao buscar na literatura propostas de soluções para problemas deste escopo, encontramos maior similaridade em tra-

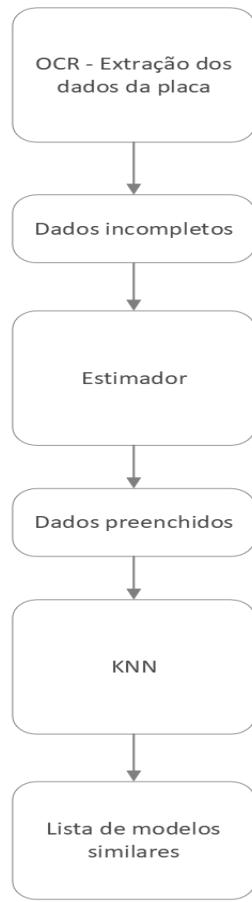


Figura 2: Arquitetura geral

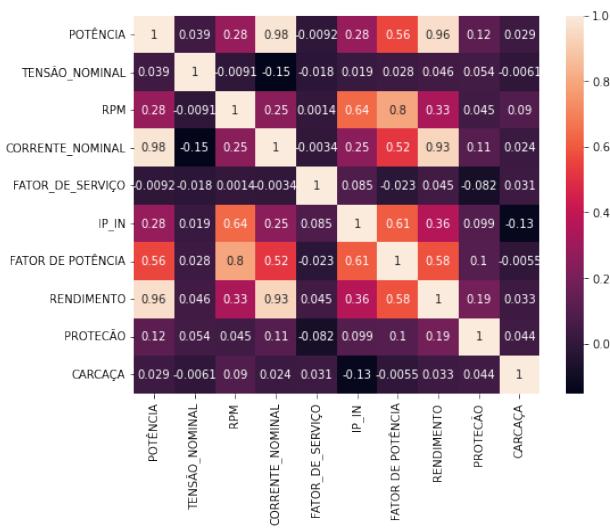


Figura 3: Matriz de correlações de Spearman dos dados selecionados para compor o *dataset* com as especificações dos equipamentos

lhos que lidam com entradas ausentes, ou *missing imputations* [9] [10]. Apesar dos trabalhos relacionados a entradas ausentes lidarem efetivamente com o problema de preenchimento de *datasets* incompletos, tais propostas muitas vezes possuem como objetivo a estimativa de dados não observados a partir da informação observada, uma vez que nosso objetivo é justamente realizar esta tarefa, no entanto para apenas uma amostra e, possuindo como referência um *dataset* completo (com todas as variáveis presentes para cada amostra), optamos por adotar tais modelos como referência para nossa solução, logo, buscamos inspiração nos modelos de *multiple imputations* para propor uma solução capaz de ser executada em um *smartphone* através de um aplicativo.

Antes de escolher um modelo de imputação, é importante ter conhecimento de que tais modelos fazem certas considerações a respeito do mecanismo para a ausência dos dados, ou em outras palavras, a probabilidade de ausência dos parâmetros, classificando os dados de acordo com as variáveis as que influenciam na ausência de um certo parâmetro. Esta classificação, resulta em 3 tipos de dados, MCAR (*Missing Completely at Random*), MAR (*Missing at Random*) e MNAR (*Missing not at Random*). De maneira simplificada, podemos definir cada uma delas conforme se segue.

Tendo como  $X_{aus}$ , as variáveis ausentes, e,  $X_{obs}$  como as variáveis observadas da amostra e, dada uma matriz  $A_{ij}$ , com  $ij$  representando a mesma dimensão do nosso *dataset*, se  $A_{ij} = 0$ , temos que a amostra na posição  $ij$  está ausente, caso  $A_{ij} = 1$ , representa que a amostra na posição  $ij$  foi observada [9]. Podemos descrever cada um dos mecanismos de ausências da seguinte forma:

- MCAR - *Missing Completely at Random*: Uma tradução literal poderia ser “Ausências completamente aleatórias”, ou seja, a probabilidade da variável estar ausente é totalmente aleatória, ou:

$$P(A_{ij} = 0) = P(A_{ij} = 0 | X_{obs}, X_{aus}) \quad (1)$$

- MAR - *Missing at Random*: Para este caso, a probabilidade da variável estar ausente, depende totalmente das variáveis observadas, ou seja:

$$P(A_{ij} = 0 | X_{obs}) = P(A_{ij} = 0 | X_{obs}, X_{aus}) \quad (2)$$

- MNAR - *Missing not at random*: Nesta situação, as variáveis não observadas dependem tanto das variáveis observadas, quantos das demais variáveis não observadas:

$$P(A_{ij} = 0 | X_{aus}) = P(A_{ij} = 0 | X_{obs}, X_{aus}) \quad (3)$$

Para a proposta deste trabalho, assumir que os dados são MCAR é suficiente, uma vez que as ausências, não dependem diretamente dos demais parâmetros observáveis do motor, elétrico, mas, das circunstâncias ambientais as quais o equipamento foi submetido.

Após assumir que para o este caso de aplicação os dados de entrada possuem ausências completamente aleatórias, passamos então para a análise de modelos que na literatura, são recomendadas para esta classe de ausências.

### A. Modelos generativos

Dentro do catálogo de propostas para lidar com dados ausentes, uma das principais classes destas soluções são baseadas em *autoencoders* [11] e *generative adversarial networks* [12].

Tais arquiteturas são atrativas por possuírem apenas um único modelo para realizar a estimativa/imputação dos dados não observados. Dentre estes modelos, a proposta de preenchimento de campos ausentes em imagens feita por Allen e Li [13] alcançou resultados bastante promissores, o que nos levou a adaptar seu trabalho para nosso contexto com dados tabulares. No entanto, após a adaptação da arquitetura e, treinamento do modelo utilizando o *dataset* com informações a respeito das especificações dos motores de indução, o modelo infelizmente não convergiu. A ousada empreitada de se treinar um modelo gerativo com pouco mais de 13 mil amostras infelizmente não alcançou sucesso, o mesmo ocorreu com modelos baseados em *autoencoders*.

Com os modelos gerativos alcançando resultados insuficientes para a continuidade do trabalho utilizando-se tal abordagem, identificamos em soluções baseadas na utilização do MICE (*Multiple imputations by chained equations*) [14] grande potencial de ser uma possível solução para a estimativa dos dados não observados do motor elétrico.

### B. MICE

Sendo uma adaptação de modelos de *multiple imputations* [14], o MICE, trata-se de uma metodologia prática para imputar valores ausentes a qual utiliza-se de em um conjunto de modelos de imputação, um para cada variável ausente [15]. Para a execução do MICE padrão [14], primeiramente é escolhida a variável com a menor quantidade de ausências no *dataset*, a qual chamaremos de  $x_1$ . Selecionada a variável, o *dataset* é filtrado para conter somente as amostras em que  $x_1$  fora observado no *dataset*,  $x_1$  é então estimada utilizando-se de um modelo de imputação definido para a mesma (para o nosso exemplo, vamos supor que este modelo é uma regressão linear), tal modelo irá utilizar o *dataset* filtrado para seu treinamento, e caso estes dados que serão utilizados como entrada possuam mais variáveis ausentes (além de  $x_1$ ) tais variáveis serão definidas aleatoriamente.

Após o procedimento, passa-se a estimar a segunda variável, que podemos chamar de  $x_2$ ,  $x_2$  é estimada utilizando-se seu modelo de imputação, o qual irá receber como entrada, os dados das amostras em que  $x_2$  é observado, além das amostras em que  $x_1$  fora estimado no passo anterior.

Desta forma o processo se repete até que todas as variáveis sejam estimadas, gerando-se um novo *dataset*. Este procedimento pode ser executado em 10 a 20 iterações [14]. Para a realização de multiplas imputações, o algoritmo é então executado novamente para garantir que sejam gerados novos resultados devido a imputação aleatória realizada no processo de imputação inicial. Por fim, os *datasets* gerados são combinados utilizando alguma técnica, definindo um novo *dataset* de saída, com as amostras preenchidas.

## VI. ESTIMANDO OS PARÂMETROS COM MICE

Baseando-se no princípio de funcionamento do MICE, o qual imputa valores ausentes baseado em um conjunto de

modelos de imputação, fora inserido um bloco estimador dentro de nossa arquitetura. Dentro deste bloco, é realizada a utilização de regressores neurais não lineares (para variáveis com distribuição contínua) tal qual como feito por Samad, Abrar, e Diawara [9] assim como a utilização de modelos de classificação (para variáveis com distribuição discreta). A tabela 1 lista cada um dos modelos gerados, assim como a pontuação da métrica de avaliação do alcançada pelo modelo (acurácia para classificadores e erro absoluto médio para regressores).

Tabela I: Modelos de regressão

Modelo	Erro absoluto médio
Corrente nominal	0.00920
Potência	0.00956
Ip/In	0.0332
RPM	0.0260
Rendimento	0.0141
Fator de potência	0.0210
Carcaça	0.0398
IP (Índice de proteção)	0.0775

Tabela II: Modelos de classificação

Modelo	Acurácia (%)
Tensão nominal	91
Fator de serviço	70

### A. K-nearest neighbors para listagem de equipamentos similares

No *k-nearest neighbors*, ou em português o k-vizinho mais próximo, cada instância/objeto do *dataset* é representado por um ponto em um espaço [16], ponto este que é definido pelos atributos/variáveis do objeto. Desta maneira o k-vizinho mais próximo, nada mais é que os  $k$  elementos deste espaço, que estão à uma métrica de distância menor que os demais objetos deste mesmo espaço. Esta métrica de distância, pode ser definida por exemplo como a distância euclidiana.

Decidimos inserir uma camada com o KNN pois por melhor que seja o desempenho de um modelo de regressão, em situações reais o mesmo é incapaz de apontar com precisão os valores exatos de determinado parâmetro, e em alguns casos, como o nosso onde devido ao processo de imputação inicial (o qual é essencial para tornar possível a execução dos primeiros modelos de regressão), a presença de ruído torna-se inerente à solução, é extremamente necessário possuir alguma ferramenta para se filtrar a propagação deste ruído, evitando a aparição de estimativas absurdas para o usuário final da solução.

Por esta motivação, para evitar que estes absurdos gerados pelo ruído presente na entrada, quer seja ele na forma de imprecisão na leitura do OCR, ou na forma da própria imputação inicial, a qual é descrita em mais detalhes na subseção que se segue, propomos a inserção de um KNN ao fim do bloco estimador, para que seja listado ao usuário possíveis candidatos (equipamentos com especificações reais), os quais são apontados como equipamentos mais similares ao

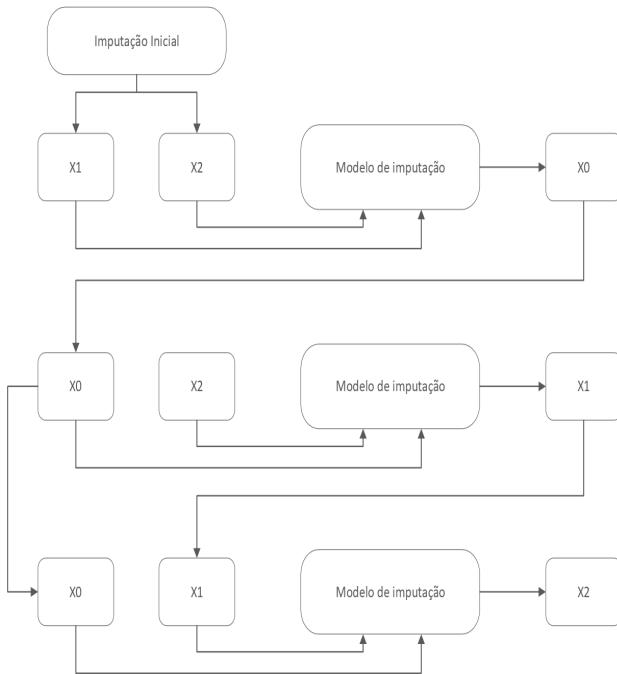


Figura 4: Composição do bloco estimador

equipamento cuja as informações foram repassadas na entrada (ou seja, aqueles cujo as especificações mais se aproximam aquelas lidas e estimadas durante o processo), guiando o mesmo às reais especificações do motor que possuí em mãos.

#### B. Bloco estimador

Baseado no MICE, o bloco estimador é composto por modelos de imputação encadeados conforme a figura 4. Inicialmente, é necessário realizar uma estimativa inicial para os dados ausentes, uma vez que os modelos de imputação foram treinados com todas as demais variáveis. Esta estimativa pode ser feita com valores aleatórios, conforme a descrição do MICE para múltiplas imputações, como também podemos buscar um modelo que procure auxiliar na acurácia das regressões, como por exemplo a média dos parâmetros dos k-vizinhos mais próximos da entrada (neste caso, os k-vizinhos mais próximos são dados preenchendo os dados ausentes pela média).

Para nossa solução, partimos para a busca de uma abordagem que procura uma boa estimativa inicial que auxilie as demais regressões. Para tanto, realizamos testes com 3 tipos de imputação, começando com a imputação com a média geral dos valores da variável ausente, depois realizamos a clusterização do dataset para que a média imputada seja a média do *cluster* da entrada, e por fim utilizamos a média da variável ausente dos 5 vizinhos mais próximos da entrada.

Após a imputação inicial o algoritmo entra na etapa de execução dos modelos de imputação. Nesta etapa, para a escolha da ordem em que os parâmetros ausentes serão estimados, geramos um mapa de correlações com o coeficiente de correlação de Spearman. Este mapa serve como guia, de modo que seja verificada a correlação entre os dados observados e os dados a serem estimados, para que a ordem das estimativas

sigua a ordem decrescente do valor absoluto destas correlações, ou seja, partindo do parâmetro ausente com o maior coeficiente de correlação com algum parâmetro observado.

O coeficiente de correlação de Spearman foi escolhido devido ao fato do mesmo possuir uma característica menos restritiva em relação ao de Pearson, pois o coeficiente não depende de distribuições simétricas, assim como não é sensível a presença *outliers* [17], fatores que o tornam mais adequado ao nosso conjunto de dados.

Após estimar todos os parâmetros o processo ocorre novamente, no entanto sem utilizar a imputação inicial, porém os dados descobertos na execução anterior do algoritmo. O processo se repete até que um número de iterações seja alcançado. Segundo as recomendações de dispostas em [14] definimos este número como 10 iterações.

Após a execução do bloco estimador, obtemos como saída um vetor completo contendo todos os dados do motor elétrico (observados e estimados) e, utilizamos este resultado como entrada de um modelo treinado de KNN, que busca os 5 vizinhos mais próximos à saída e os apresenta como possíveis candidatos ao modelo do equipamento. Tais candidatos são apenas uma referência de especificações que o equipamento possa vir a realmente possuir, uma vez que o KNN conhece somente a base de dados com a qual foi treinado, ou seja, o nosso *dataset*, o qual infelizmente, não contempla todos os equipamentos do mercado.

## VII. APlicativo desenvolvido

### A. Estrutura do aplicativo

O aplicativo desenvolvido como produto final deste projeto tem por missão unir ambos os modelos já descritos anteriormente. Para tanto, definimos o fluxo de execução apresentado na figura 5.

Para possibilitar a utilização dos algoritmos desenvolvidos em linguagem Python durante o processo de desenvolvimento da solução, optamos por uma alternativa que fosse capaz de aproveitar tais algoritmos, diminuindo o custo de desenvolvimento. Por conta disso, propomos uma aplicação com esquema cliente-servidor, onde a aplicação local é desenvolvida na linguagem Java, a linguagem de programação padrão para dispositivos *android*, enquanto o lado do servidor foi escrito em Python.

Conforme descrito na figura 5 após o encaminhamento da imagem fornecida pelo usuário, são realizados os procedimentos de alongamento de contraste como tentativa de aumentar a distinção entre os elementos da imagem, além da realização da conversão do esquema de cores da imagem para tons cinza e posterior redimensionamento da imagem para 1024x768, os quais são necessários para atender ao requisito de tamanho máximo da imagem a ser enviada através da opção gratuita da API do OCR Space. Todos os processos são executados na etapa de pré-processamentos.

Foi inserida também a possibilidade do usuário de editar as informações extraídas pelo analisador, pois, conforme é apresentado nas próximas seções tal *feedback* é essencial para a melhoria dos resultados.

### B. Fluxo de telas do usuário e funcionalidades implementadas

Para maior esclarecimento quanto a experiência do usuário durante o processo de utilização do aplicativo, são apresentadas as figuras 6 e 7.

O fluxo se inicia quando o usuário abre o aplicativo pela primeira vez, neste momento o usuário será direcionado para a tela de *login* (figura 7a), na qual terá a opção de se registrar na parte inferior da tela. Ao clicar no botão “*registrar*”, o usuário é então encaminhado à tela de criação de conta (figura 7b). Após o registro, o aplicativo passa para a sua tela principal na qual são exibidos dois botões flutuantes no lado direito inferior do *display* (figura 7c). O botão mais acima é responsável por acessar a galeria, na qual será possível selecionar e enviar uma imagem de alguma placa de motor elétrico. O botão mais abaixo dá acesso à câmera, permitindo que o usuário realize a captura de uma nova imagem para envio.

Ainda na tela principal (figura 7c) existe um ícone no lado superior direito, que abre um *Navigation Drawer* (barra de navegação) dando acesso a duas opções: “Enviar Dados” e “Sair”. A primeira opção levará o usuário para a tela apresentada na figura 7e na qual o é possível realizar o preenchimento dos dados de uma placa do motor elétrico sem a necessidade do envio de uma imagem. O resultado deste processo é o mesmo de quando se envia uma imagem, e é apresentado na tela exibida pela figura 7f. Já na segunda opção o usuário terá a possibilidade de encerrar a sessão de sua conta.

O usuário também poderá durante o processo de leitura da placa de identificação, consultar e alterar as informações extraídas da imagem (figura 7d), para posteriormente enviá-las. Esses dados são reenviados para o processamento no servidor, o qual retorna uma lista contendo os cinco equipamentos que mais se assemelham às características do equipamento analisado (figura 7g). Após isto o usuário poderá retornar à tela principal.

## VIII. TESTES

### A. Leitura dos parâmetros OCR + Analisador

Conforme descrito na seção IV, foram realizados testes na ferramenta de OCR escolhida (OCR Space) [8], para a extração dos caracteres presentes nas imagens de nosso *dataset*, e ao avaliar os resultados da ferramenta OCR, fora realizada a construção de um conjunto de expressões regulares na tentativa de realizar a extração das informações referentes aos parâmetros definidos na seção III-B.

Após a criação destas regras, realizamos testes de maneira separada para cada conjunto de imagens que levantamos, ou seja, equipamentos trifásicos com os 10 parâmetros que elencamos somando um total de 164 imagens e, monofásicos com somente 8 destes 10 parâmetros somando um total de 43 imagens.

Os testes foram executados aplicando-se inicialmente às imagens, os pré-processamentos apresentados na seção VII. Após o pré-processamento é então realizado o envio das imagens para a API que realiza a extração do texto e retorna um arquivo *.json* com estas informações. De posse deste arquivo são realizadas as análises do texto contendo todas as cadeias de caracteres extraídas da imagem através das

expressões regulares geradas em busca de cada um dos 10 parâmetros que levantamos como presentes na maioria das placas de identificação dos motores elétricos.

Os resultados desta etapa são apresentados nas tabelas III e IV. É importante ressaltar que não se trata da acurácia, ou seja a capacidade de identificar corretamente o parâmetro, mas somente a capacidade das expressões regulares extraírem algum candidato que possa ser rotulado como o valor do parâmetro citado. Os resultados que apresentamos nesta seção serão discutidos posteriormente.

Tabela III: Extração de informações sobre os parâmetros da ferramenta OCR para placas de motores trifásicos

Parâmetro	Parâmetros extraídos	(%)
Corrente nominal	149	91
Potência	132	80
Ip/In	86	52
RPM	80	49
Rendimento	75	46
Fator de potência	89	54
Carcaça	6	4
IP (índice de proteção)	56	34
Tensão nominal	153	93
Fator de serviço	121	74

Tabela IV: Extração de informações sobre os parâmetros da ferramenta OCR para placas de monofásicos

Parâmetro	Parâmetros extraídos	(%)
Corrente nominal	37	86
Potência	36	84
Ip/In	18	42
RPM	43	100
Rendimento	5	12
Fator de potência	13	30
Carcaça	3	7
IP (índice de proteção)	3	7
Tensão nominal	42	98
Fator de serviço	35	81

### B. Estimativa dos parâmetros

Para realização de testes com modelos de imputação, responsáveis por realizar as estimativas dos parâmetros ausentes, fora utilizada a parcela do *dataset* não vista na etapa de treinamento das redes neurais as quais são utilizadas para a estimativa de cada parâmetro.

Para simular as ausências dos parâmetros realizamos a análise combinatória de todas as combinações possíveis de configurações de entrada para o modelo. As quais configuram um total de 1022 possibilidades conforme a equação 4.

$$\sum_{i=1}^{10} C_{10,i} = 1022 \quad (4)$$

Para a avaliação dos resultados, definimos 02 métricas de análises as quais são: o erro médio dos modelos de regressão

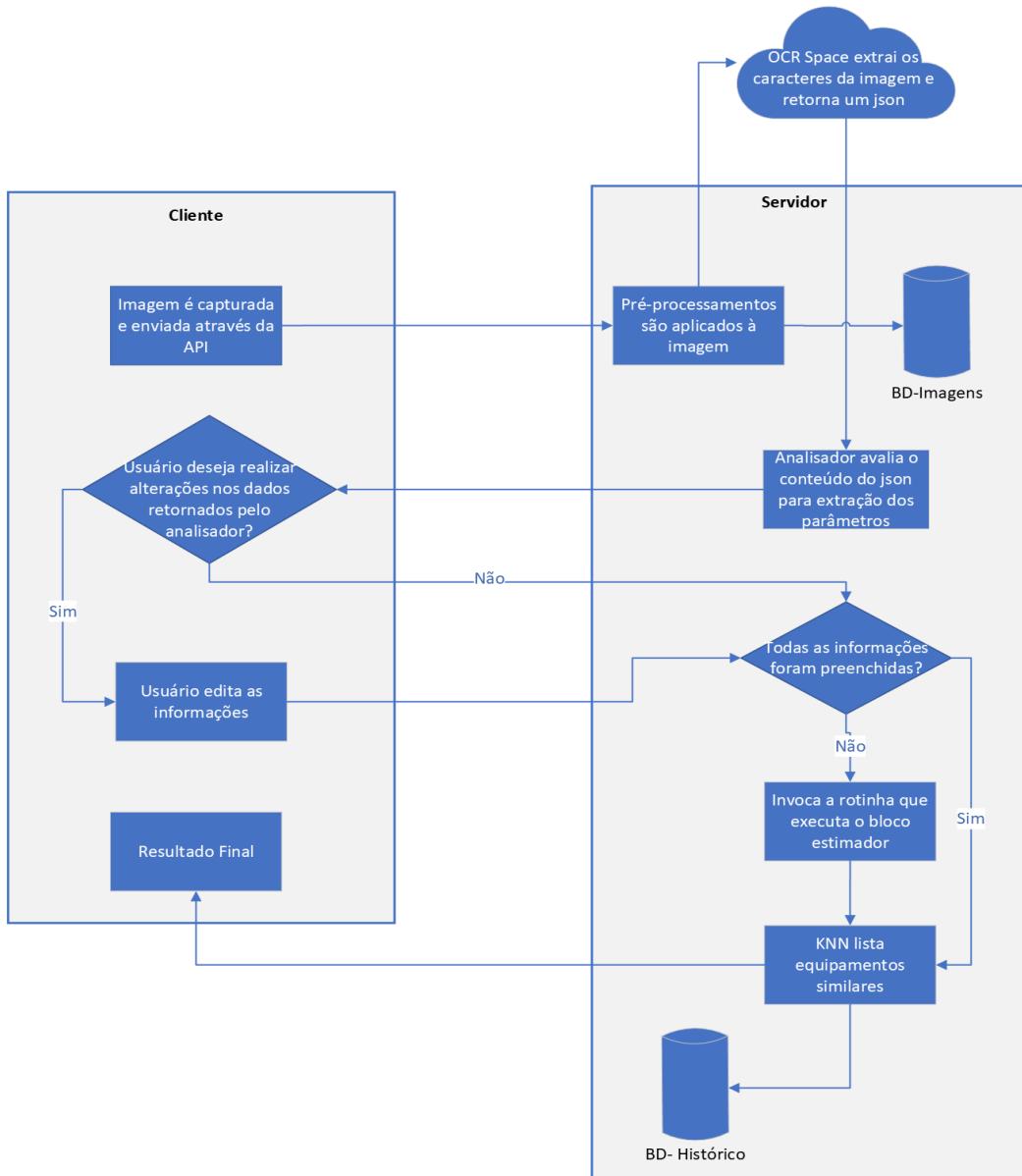


Figura 5: Ciclo de execução do aplicativo

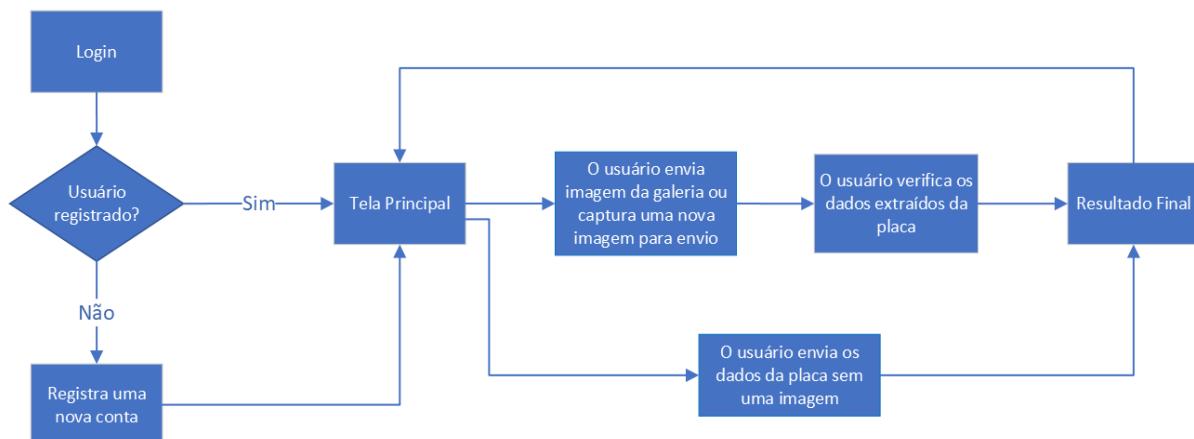


Figura 6: Fluxo de telas de usuário do aplicativo

The figure consists of nine panels arranged in a grid:

- (a) Tela de login: Shows fields for E-mail and Senha, a Login button, and a link to create an account.
- (b) Tela de registro de usuário: Shows fields for Nome, Sobrenome, E-mail, Senha, and checkboxes for Mostrar senha and REGISTRAR.
- (c) Tela principal: Shows a message about historical data and two circular icons.
- (d) Tela de edição dos dados: Shows fields for Potência, Tensão Nominal, RPM, Corrente Nominal, Fator de Serviço, Ip/In, and Fator de Potência, with buttons for ENVIAR and Fazer login.
- (e) Tela para envio direto dos dados: Shows fields for the same parameters as (d), with an ENVIAR button.
- (f) Exibição dos resultados após envio dos dados sem a imagem: Shows a summary table for W22 Category D WEG with values: Potência: 1.10, Tensão Nominal: 220.0, RPM: 1770.0, Corrente Nominal: 4.80, Frequência: 60.0, Fator de Serviço: 1.00, Ip/In: 6.0, and Fator de Potência: 0.78.
- (g) Tela de exibição dos resultados após envio dos dados com a imagem: Shows the same summary table as (f) but includes a small image of a motor label at the bottom.

Figura 7: Telas do processo de utilização do aplicativo

na estimativa dos parâmetros e a acurácia do KNN, ou seja a capacidade do algoritmo de, dentre os 5 itens mais semelhantes listados para o usuário, apresentar aquele que corresponde ao equipamento realmente especificado na entrada.

Para os resultados dos modelos de regressão apresentamos na tabela V o erro médio total dos modelos de regressão ramificado pela evolução da quantidade de parâmetros ausentes. Também apresentamos nas figuras 8 e 9 como esse erro evolui para cada iteração do bloco estimador.

Para os resultados apresentados pelo KNN, também os organizamos por quantidade de parâmetros ausentes, onde a tabela VI apresenta a acurácia final do KNN, e na figura 10 apresentamos a evolução da acurácia do KNN ao longo de cada iteração completa do bloco estimador, o qual também

está subdividindo por quantidade de parâmetros ausentes.

## IX. ANÁLISE DOS RESULTADOS E LIMITAÇÕES DA SOLUÇÃO

### A. Resultados da etapa de extração dos parâmetros

Ao avaliar os resultados apresentados na seção VIII com respeito ao desempenho da ferramenta de OCR juntamente com o analisador de realizarem a extração das informações contidas na placa de identificação do motor elétrico, vemos que para cada parâmetro que desejamos identificar, os resultados variam bastante.

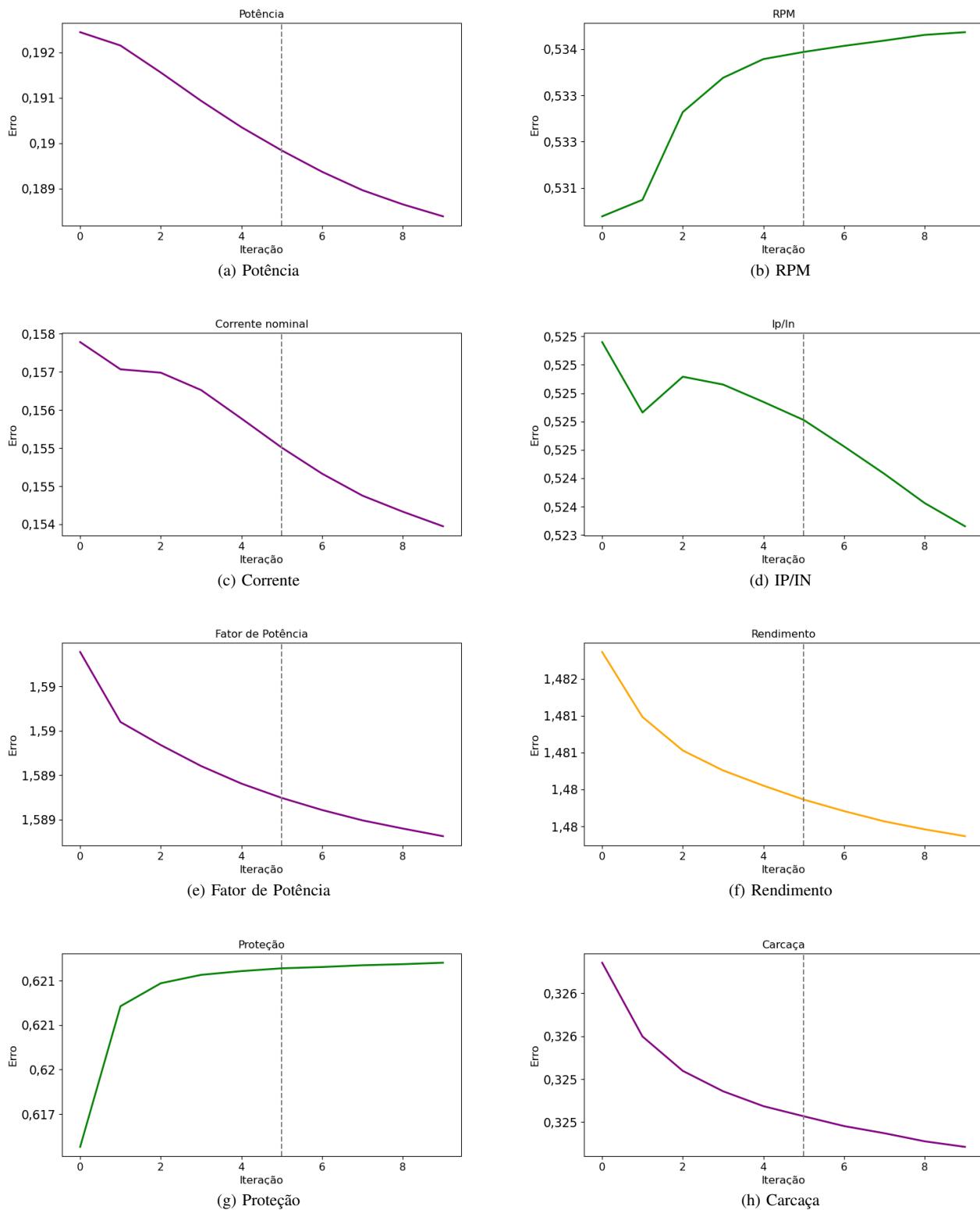


Figura 8: Evolução média (para todas as configurações de entrada) dos erros de cada modelo de imputação ao longo de cada iteração do bloco estimador

Tabela V: Variação do erro absoluto médio dos modelos de regressão por quantidade de ausências e modelo de imputação inicial

Parâmetros ausentes	Erro absoluto médio dos modelos de regressão por tipo de imputação		
	Média	Média por cluster	Média dos 5-vizinhos mais próximos
1	0,748	0,748	0,748
2	0,747	0,747	0,749
3	0,747	0,746	0,750
4	0,748	0,744	0,751
5	0,749	0,742	0,753
6	0,752	0,742	0,760
7	0,755	0,743	0,760
8	0,759	0,746	0,770
9	0,765	0,752	0,780

Tabela VI: Variação da acurácia do KNN por quantidade de ausências e modelo de imputação inicial

Parâmetros ausentes	Acurácia do KNN (%) por tipo de imputação		
	Média	Média por cluster	Média dos 5-vizinhos mais próximos
1	93,3	93,3	93,3
2	82,1	82,0	82,1
3	67,4	67,3	67,4
4	50,7	50,7	50,8
5	34,3	34,4	34,4
6	20,2	20,3	20,4
7	9,83	9,95	10,1
8	3,72	3,80	3,90
9	0,98	0,97	1,02

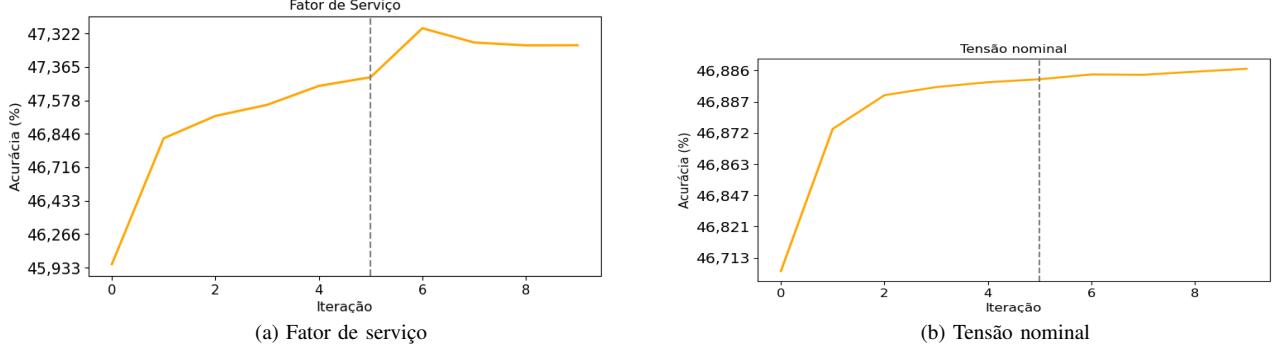


Figura 9: Evolução da acurácia média (de todas as configurações de entrada) dos modelos de imputação classificadores por iteração

Separando os resultados em 02 grupos, temos um primeiro onde a ferramenta conseguiu extrair ao menos 1 candidato para mais de 70% das imagens, e um outro onde a ferramenta obteve resultados inferiores a 60%. Ao olharmos somente para o pequeno subgrupo de placas de equipamentos monofásicos, a quantidade de itens do primeiro grupo aumenta para 05, no entanto a média do segundo grupo cai bastante, com seu maior pontuador alcançando a capacidade de identificação para apenas 42% da amostra.

O que torna complexa a avaliação de tais resultados, é que eles apresentam somente a capacidade do analisador em identificar ao menos um candidato para parâmetro, e não se o candidato é o correto.

Ressalvo os casos em que a falha se deu no processo de

transcrição das informações presentes no objeto analisado, a utilização de expressões regulares tem a limitação de sozinhas, não serem capazes de realizarem considerações a respeito do posicionamento espacial dos objetos na imagem, e como há uma considerável variação no posicionamento dos elementos a depender do modelo de placa, não nos pareceu viável a realização de seu mapeamento.

Um outro fator importante para a análise, é que durante os testes, notamos que conforme varia-se o modelo da placa de identificação, o analisador pode acabar identificando mais de um candidato para cada campo, sendo que, nem sempre o primeiro candidato, acaba sendo o correto. Uma melhoria que poderia ser realizada é a criação de algum sistema de decisão para auxiliar nesta etapa. No entanto, para o estado

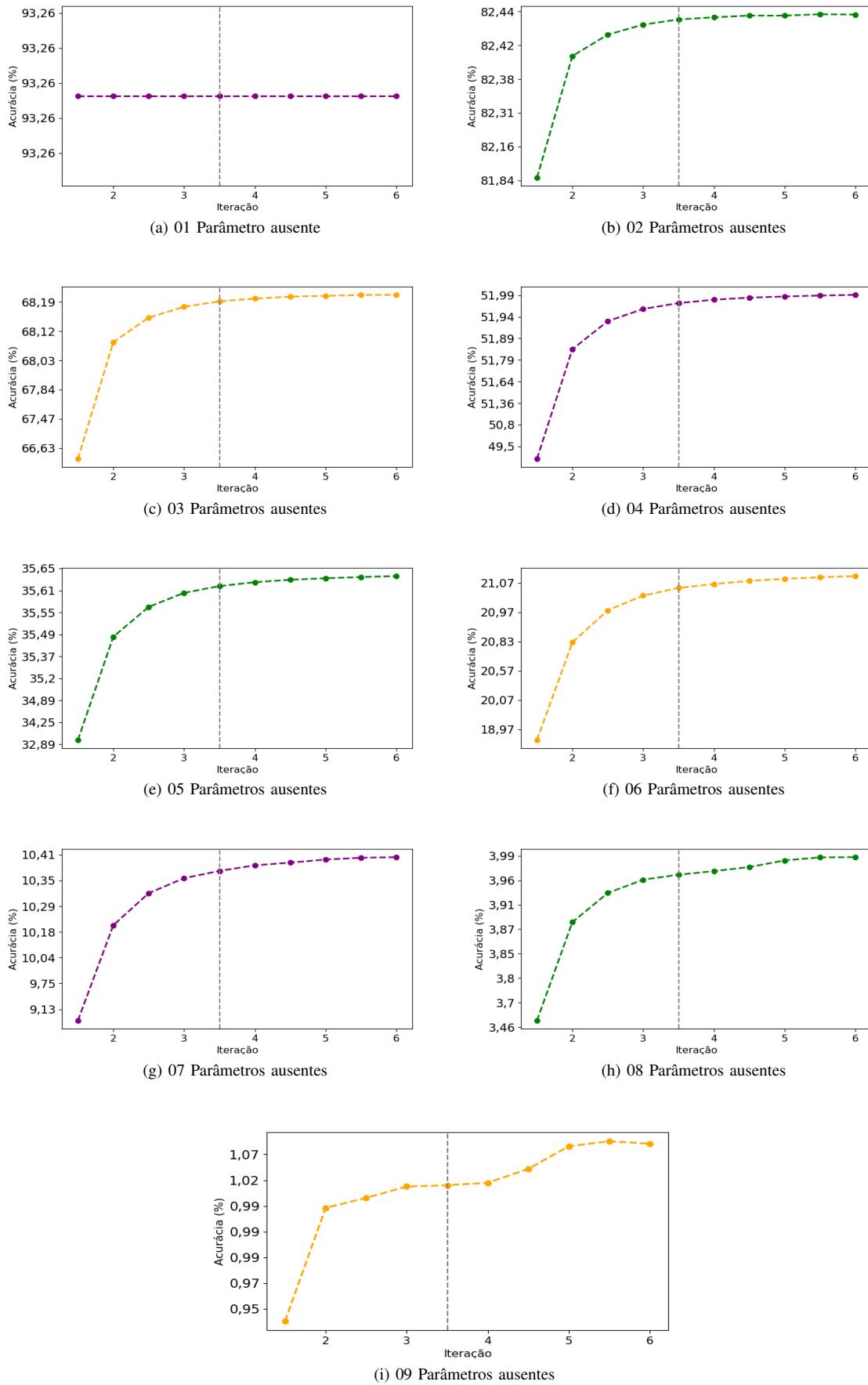


Figura 10: Evolução da acurácia média do KNN ao longo das iterações realizadas pelo bloco estimador subdividido por quantidade de parâmetros ausentes

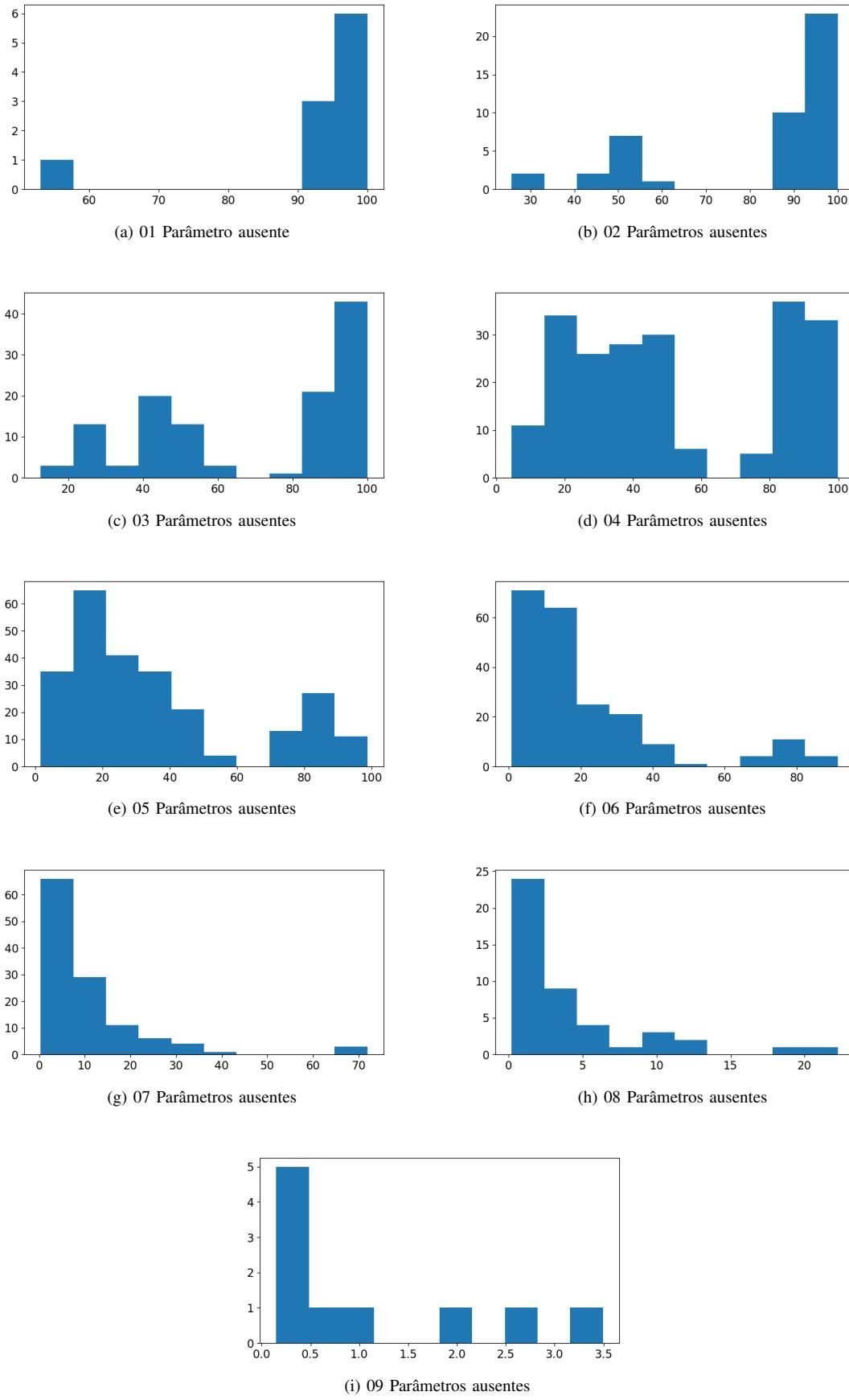


Figura 11: Histogramas com a distribuição das acuráciais do KNN subdivididos pela quantidade de parâmetros ausentes

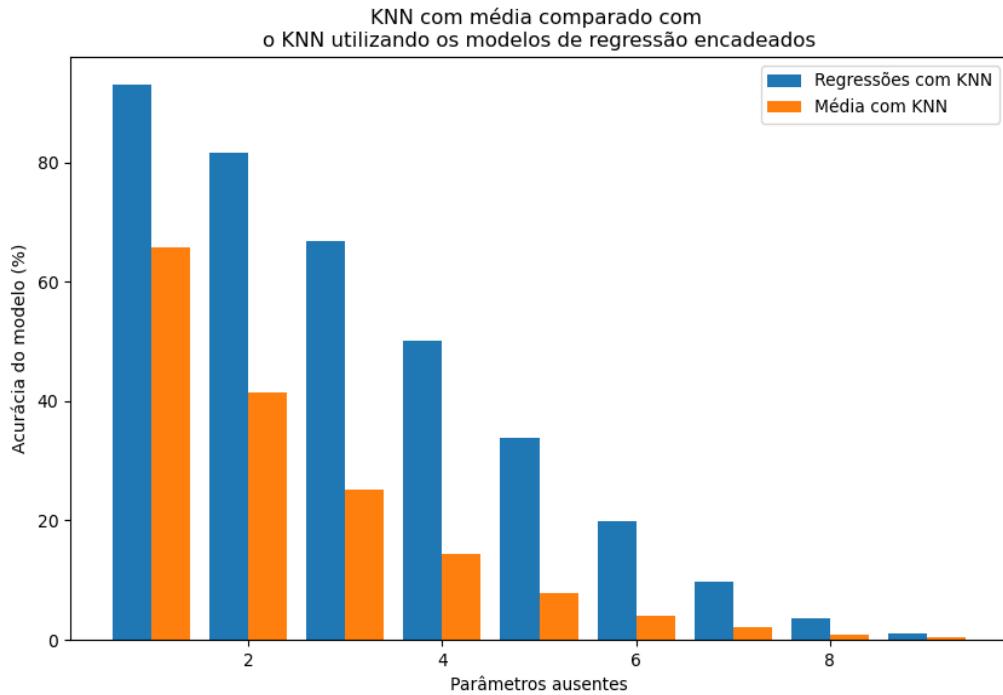


Figura 12: Comparação da acurácia do KNN para uma imputação somente com a média com relação ao KNN com o processo de imputação realizado pelos modelos estimadores encadeados

atual da solução tal característica evidenciou o enviesamento da proposta.

Quanto ao baixíssimo desempenho na capacidade do analisador de identificar algum candidato para a carcaça, a dificuldade está no fato da informação pode, muitas vezes não estar identificada. Ou seja, apenas inserida na placa de identificação sem sua identificação explícita (textual). O que torna desafiadora a tarefa de apenas com expressões regulares conseguir identificar o valor da variável.

Avaliando-se agora, o impacto do OCR Space nos resultados apresentados, vemos que um fator decisivo para o bom desempenho da ferramenta OCR está relacionado à resolução das imagens que utilizamos. Nossa *dataset* possui imagens que variam de 267 X 189 até 2808 X 2106, o que torna o desempenho do OCR Space bastante variável. Além do mais, o tamanho dos caracteres presentes na placa de identificação também variam conforme exposto pela figura 1. Um exemplo de quando isso afeta negativamente a análise pode ser visto na figura 1c, onde o campo "REND (%)" da placa aparece transscrito de forma muito pequena, o que torna necessário uma resolução maior para identificação do campo.

No entanto, apesar do exposto, embora nossos dados possuam uma resolução majoritariamente inferior ao mínimo necessário para que a solução alcance seu desempenho máximo, é esperado que em situações reais, a imagem tenha uma qualidade suficiente para que seja feita uma eficiente extração de caracteres, levando a solução a alcançar um desempenho mais elevado que o exposto.

Um outro ponto que também notamos impactar no desempenho da ferramenta OCR, é que as placas de identificação com

fundo branco, tiveram seus dados extraídos com mais facilidade, porém, apesar deste fato, ao aplicarmos um *thresholding* na etapa de pré-processamento das imagens, mesmo com a imagem sendo enviada com o fundo branco e texto preto (no caso, imagens não nativamente neste formato) a ferramenta passou a extrair uma quantidade menor de informações do que comparado com a imagem somente com os pré-processamentos padrões.

### B. Avaliação dos regressores

Agora, para a realização da análise de desempenho dos modelos responsáveis por realizarem a imputação dos parâmetros ausentes, é importante dividirmos nossa discussão em 02 etapas principais: análise da evolução das métricas de avaliação dos modelos treinados a cada iteração e, análise da variação destas mesmas métricas porém dada a quantidade de parâmetros ausentes.

Começando com a análise de evolução das métricas de avaliação a cada iteração, ao realizar uma avaliação preliminar é nítido apontar, que para a maioria dos modelos treinados a evolução é positiva, com boa parte dos regressores tendo seus erros diminuindo e, os classificadores com suas acurácias aumentando. Com exceção dos modelos para predição do RPM e da proteção do equipamento, os demais modelos obtiveram um comportamento conforme o objetivo da proposta.

No entanto é importante ponderar que estamos analisando somente a média de tais métricas obtidas através da avaliação do desempenho de cada um dos modelos para todas as  $C_{i,10}$

combinações possíveis de entrada para uma quantidade  $i$  de parâmetros ausentes. E, por estarmos avaliando somente a média desses erros, é bem possível que tais erros estejam sendo influenciados por *outliers*, ainda mais levando em consideração que o desempenho dos modelos de regressão são fortemente dependentes de quais são os parâmetros reais que foram extraídos da placa de identificação, os quais irão compor a entrada do modelo.

Agora olhando para como as métricas variam conforme aumentamos a quantidade de ausências, como já era de se esperar, o erro aumenta conforme aumentamos a quantidade de ausências. Quando falamos de modelos de regressão, os mesmos não são treinados para a presença de ruído em sua entrada, o que torna extremamente complicada a tarefa de mantê-los com um bom desempenho conforme o ruído se propaga.

Na tentativa de diminuir o erro propagado pelo ruído inserido na imputação inicial, foram realizadas tentativas de alterar o procedimento, tais como a aplicação do KNN para a realização da imputação com base nos parâmetros dos 5 vizinhos mais próximos da entrada, e a inserção de uma clusterização do *dataset*, para que a média passe a ser a média do cluster da entrada. Para esta proposta, a entrada era classificada com base em um classificador KNN.

No entanto, em ambas as tratativas apresentadas sempre é necessário realizar uma estimativa para tornar possível a execução dos modelos. Para o caso 5 vizinhos mais próximos, para a identificação desta vizinha é necessário preencher os campos ausentes do *dataset*, e, para a classificação do cluster da entrada, o mesmo processo se faz necessário, o que nos levou a resultados muito similares ao da imputação puramente com a média (conforme as tabelas V e VI), porém com um custo computacional mais elevado. Não justificando então, a adoção de tais alternativas.

### C. Avaliação das acurácia

Para a avaliação da acurácia dos modelos, adotaremos a mesma subdivisão adotada na análise anterior, ou seja, primeiramente olhando para a evolução da acurácia a cada iteração, e posteriormente para a análise das acurácia variando-se a quantidade de parâmetros ausentes.

Assim como ocorreu com os modelos de imputação dos parâmetros, a acurácia do KNN de modo geral apresentou o comportamento esperado. A cada nova iteração, a acurácia do modelo aumentou mostrando que o processo iterativo do bloco estimador apresentou um comportamento adequado.

Quando avaliamos os valores das acurácia apresentados na figura 10, é possível perceber que apesar de apresentar um aumento de desempenho, este aumento é sutil quando se comparado a primeira iteração. O que mostra que uma quantidade muito elevada de iterações não contribui tanto para a evolução da solução.

Agora, quando comparamos com a aplicação do KNN sem a realização das estimativas com os modelos de imputação como processo intermediário, torna-se evidente que o ganho na capacidade de se identificar corretamente o equipamento analisado justifica a utilização dos estimadores.

Um detalhe importante desta análise é avaliar até que ponto a solução é capaz de apresentar um possível candidato ao equipamento que está sendo analisado. Notamos que o limite da aplicação é de até 03 parâmetros ausentes, a partir do qual a solução deixa de ter uma acurácia média superior a 60%.

Porém, é válido ressaltar que assim como ocorre para as métricas de avaliação dos modelos estimadores, a acurácia apresentada na figura 10 também trata-se da média das acurácia para todas as  $C_{10,i}$  de cada quantidade  $i$  de ausências. Ou seja, é sensível a *outliers* e pode levar à uma interpretação incompleta do desempenho da solução, sendo necessário além do gráfico apresentado na figura 10, a análise dos histogramas da acurácia do modelo para cada quantidade  $i$  de ausências. Por conta disso, apresentamos também na figura 11 esta distribuição, a qual revela que, em casos com até 04 parâmetros ausentes, uma interessante quantidade de configurações de entrada resultam em uma acurácia superior à 70%.

## X. CONCLUSÃO

Pode-se concluir da seção anterior que cada uma das soluções possuem limitações conhecidas e bem definidas, no entanto a interpretação dos resultados requer que sejam considerados os pontos que iremos discorrer a seguir.

Uma primeira observação que podemos apresentar é que como assumimos que as ausências dos parâmetros são MCAR, ou seja, totalmente aleatórias, ou seja, estamos supondo que a probabilidade de ausência de qualquer parâmetro é a mesma, no entanto, isso não é verdade. Um dos casos em que podemos mostrar isso é com a própria solução OCR, onde é possível identificar a partir dos resultados apresentados, que a mesma detém uma predisposição a identificar com mais facilidade determinados parâmetros como por exemplo casos de modelos de placas já mapeados, ou casos em que a placa possui características que tornam mais fácil a leitura de seus caracteres. Podemos apontar tais circunstâncias como condicionadas ao tipo da placa de identificação que está sendo repassado pelo usuário. Poderíamos supor ser até possível estimar alguma probabilidade do tipo de placa de identificação mais provável para ser repassado como entrada se tivéssemos acesso aos dados relativos à participação de cada fabricante no mercado, o que poderia mudar a maneira de como as análises são realizadas, dando maior foco para os casos reais mais prováveis.

No entanto, tal enviesamento pode ser em certo nível contornado com o auxílio do usuário que manuseia a ferramenta, dada a possibilidade do mesmo realizar as correções de leitura da ferramenta OCR que julgar necessárias, assim como apresentado na seção VII

Quando analisamos os resultados obtidos da proposta desenvolvida para a estimação dos parâmetros ausentes, é possível afirmar que que alcançamos os objetivos almejados, uma vez que, grande parte das estimativas para os casos com até 03 parâmetros ausentes, é capaz de alcançar uma acurácia superior a 60%.

A inserção do KNN foi fundamental para a solução, uma vez que conforme apresentado na figura 8, apesar dos erros dos

estimadores no geral caírem conforme as execuções avançam, quando tais parâmetros são retornados a sua escala natural para a apresentação de resultados ao usuário, este erro cresce exponencialmente (haja vista que os modelos foram treinados com os dados normalizados para a escala logarítmica das entradas), o que torna a apresentação do resultados brutos ao usuário, uma tarefa pouco recomendável (ao menos para os casos mais extremos). Ou seja, a utilização do KNN para a apresentação dos 5 vizinhos mais próximos como retorno da aplicação, alcançou seu objetivo de diminuir o impacto causado pelo ruído repassado aos modelos de regressão pelo processo de imputação inicial.

Por fim, conforme já apresentado na figura 12, a inserção dos modelos de regressão encadeados, baseados no MICE, conseguiram elevar consideravelmente capacidade do KNN em apontar os 5-vizinhos mais próximos, quando comparado com a execução do KNN utilizando-se somente da média para realização da imputação inicial, que justifica a adoção da solução.

Conforme já dito, a inserção do KNN se da também devido ao fato de que somente as regressões, por conta do ruído inerente ao problema podem levar a resultados absurdos, e, o KNN é capaz de, de certa forma abstrair tais absurdos olhando para todo o conjunto de parâmetros e resultando em uma lista de itens reais, os quais se podem vir a ser um bom referencial para os parâmetros do equipamento que está sendo analisado.

Por fim é importante ponderar que quanto mais parâmetros são retirados da entrada mais os registros do *dataset* passam a se tornar indistinguíveis, o que significa que para uma quantidade de  $i$  de ausências, vários equipamentos passam a possuir as mesmas especificações quando consideramos somente os dados observados/mostrados para a entrada. Tal fato torna a solução incapaz de com o KNN apontar o equipamento correto com certa precisão, haja vista que torna-se impossível somente com os dados tabulares da entrada distinguir entre os registros do *dataset*. O que torna o modelo pouco eficaz conforme aumentamos a quantidade de parâmetros ausentes.

## XI. AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brasil, pelo financiamento deste projeto (401842/2021-5) via chamada CNPq/MCTI/SEMPI Nº14/2021.

## REFERÊNCIAS

- [1] C. Zhang, Q. Wang, and X. Li, “V-lpdr: Towards a unified framework for license plate detection, tracking, and recognition in real-world traffic videos,” *Neurocomputing*, vol. 449, pp. 189–206, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231221004951>
- [2] S. Alghalyine, “Real-time jordanian license plate recognition using deep learning,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part A, pp. 2601–2609, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1319157820305152>
- [3] Y. Chen, H. Shu, W. Xu, Z. Yang, Z. Hong, and M. Dong, “Transformer text recognition with deep learning algorithm,” *Computer Communications*, vol. 178, pp. 153–160, 2021.
- [4] O. Bensch, M. Popa, and C. Spille, “Key information extraction from documents: evaluation and generator,” *arXiv preprint arXiv:2106.14624*, 2021.

- [5] X. Zhao, E. Niu, Z. Wu, and X. Wang, “Cutie: Learning to understand documents with convolutional universal text information extractor,” *arXiv preprint arXiv:1903.12363*, 2019.
- [6] WEG, “Electric motor selection,” Disponível em: [https://ecatalog.weg.net/tec\\_cat/tech\\_motor\\_sel\\_web.asp](https://ecatalog.weg.net/tec_cat/tech_motor_sel_web.asp). Acesso em: 06 de fevereiro 2023, 2023.
- [7] Hercules Electric Motors, “Downloads - héracles motores elétricos,” Disponível em: <https://www.herculesmotores.com.br/downloads/en/>. Acesso em: 02 de fevereiro 2023, 2023.
- [8] OCRSpace, “Free ocr api,” Disponível em: <https://ocr.space/ocrapi>. Acesso em: 02 de fevereiro 2023, 2023.
- [9] M. D. Samad, S. Abrar, and N. Diawara, “Missing value estimation using clustering and deep learning within multiple imputation framework,” *Knowledge-based systems*, vol. 249, p. 108968, 2022.
- [10] J. C. Cole, “How to deal with missing data,” *Best practices in quantitative methods*, pp. 214–238, 2008.
- [11] Y. Burda, R. Grosse, and R. Salakhutdinov, “Importance weighted autoencoders,” *arXiv preprint arXiv:1509.00519*, 2015.
- [12] J. Yoon, J. Jordon, and M. Schaar, “Gain: Missing data imputation using generative adversarial nets,” in *International conference on machine learning*. PMLR, 2018, pp. 5689–5698.
- [13] Allen, A and Li, W, “Generative adversarial denoising autoencoder for face completion,” Disponível em: [https://faculty.cc.gatech.edu/~hays/7476/projects/Avery\\_Wenchen](https://faculty.cc.gatech.edu/~hays/7476/projects/Avery_Wenchen). Acesso em: 02 de fevereiro 2023, 2019.
- [14] J. N. Wulff and L. E. Jeppesen, “Multiple imputation by chained equations in praxis: guidelines and review,” *Electronic Journal of Business Research Methods*, vol. 15, no. 1, pp. 41–56, 2017.
- [15] I. R. White, P. Royston, and A. M. Wood, “Multiple imputation using chained equations: issues and guidance for practice,” *Statistics in medicine*, vol. 30, no. 4, pp. 377–399, 2011.
- [16] K. Faceli, A. C. Lorena, J. Gama, and A. C. P. d. L. F. d. Carvalho, *Inteligência artificial: uma abordagem de aprendizado de máquina*, 2011.
- [17] Á. Sousa, “Coeficiente de correlação de pearson e coeficiente de correlação de spearman: o que medem e em que situações devem ser utilizados?” *Correio dos Açores*, pp. 19–19, 2019.