

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

IGOR TEDESCHI FRANCO

MANUTENÇÃO PREDITIVA UTILIZANDO TÉCNICAS
DE MACHINE LEARNING EM UM SISTEMA
EMBARCADO

São Leopoldo, RS
2020

IGOR TEDESCHI FRANCO

**MANUTENÇÃO PREDITIVA UTILIZANDO TÉCNICAS
DE MACHINE LEARNING EM UM SISTEMA
EMBARCADO**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação pelo curso de Engenharia de Controle e Automação da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador:

Prof. Dr. Rodrigo Marques de Figueiredo

São Leopoldo, RS

2020

Dedico este trabalho aos meus pais, por investirem a vida deles em me tornar uma pessoa honesta, prestativa e um excelente profissional. Ao meu irmão, por todo o suporte e debates sobre o assunto. À minha namorada, por todo o incentivo e conforto. Ao meu orientador, por me guiar e ajudar no desenvolvimento de todo este trabalho. À Altus, por permitir a utilização do espaço para execução do projeto. Enfim, a todos que acreditaram em meu potencial.

“Everybody is a genius. But if you judge a fish by its ability to climb a tree, it will live its whole life believing that it is stupid.”
(Albert Einstein)

LISTA DE FIGURAS

Figura 1 – Manutenção Corretiva - Diagrama dos Tempos	17
Figura 2 – Manutenção Preventiva - Custos Associados	18
Figura 3 – Diagrama Geral da Utilização de um Sensor	22
Figura 4 – Diagrama da Medição de um Sensor	22
Figura 5 – Processo de Descoberta no KDD	25
Figura 6 – Base de Dados com Convenções	26
Figura 7 – Diagrama de Aprendizagem	30
Figura 8 – Resultados obtidos por Yan <i>et al.</i> (2017) - Comparação das redes neurais com e sem o Modelo de Energia (múltiplas fontes de dados)	33
Figura 9 – Resultados obtidos por Mathew, Luo e Pang (2017) - Comparação do método de predição usando SVR (regressão <i>kernel</i>) com o sinal real	36
Figura 10 – Resultados obtidos por Susto <i>et al.</i> (2012) - Comparação do método de predição <i>Elastic Net</i> (PdM) com a média e mediana (PvM)	37
Figura 11 – Resultados obtidos por Susto <i>et al.</i> (2013) - Comparação do método de predição SVM (PdM), linear e RBF, com a média e mediana (PvM)	38
Figura 12 – Resultados obtidos por Susto <i>et al.</i> (2015) - Comparação dos métodos de predição KNN e SVM - linear, RBF e com múltiplos classificadores - (PdM), com a média e mediana (PvM)	39
Figura 13 – Resultados obtidos por Susto <i>et al.</i> (2014) - Comparação do método de predição para diferentes valores de k (PdM) com a média e mediana (PvM)	41
Figura 14 – Resultados obtidos por Butte, R e Patil (2018) - Comparação do número de <i>wafers</i> processados até a execução da manutenção para diferentes métodos de predição	44
Figura 15 – Fluxograma do Projeto	48
Figura 16 – Diagrama do Sistema Proposto	50
Figura 17 – Fluxograma de Teste	51
Figura 18 – Fluxograma de Validação	53
Figura 19 – Fluxograma do Primeiro Cenário	55
Figura 20 – Fluxograma do Segundo Cenário	56
Figura 21 – Gráficos das Matrizes de Confusão - Comparação entre SVM, <i>Random Forest</i> e Regressão Logística	60
Figura 22 – Fluxograma do Projeto Final	62
Figura 23 – Fluxograma da Aplicação de Validação	64
Figura 24 – Sistema Supervisório Node-RED [®]	66
Figura 25 – Sistema Supervisório em Falha	67
Figura 26 – Sistema Supervisório Predição de Verdadeiro Negativo	68

Figura 27 – Sistema Supervisório Predição de Verdadeiro Positivo	69
Figura 28 – Relação entre Pares de Características	70
Figura 29 – Matriz de Confusão - Validação	71

LISTA DE TABELAS

Tabela 1 – Resultados obtidos por Xayyasith, Promwungkwa e Ngamsanroj (2018) - Comparação entre vários algoritmos de <i>Machine Learning</i> na predição de falhas no sistema de resfriamento de uma hidroelétrica	40
Tabela 2 – Resultados obtidos por Mathew <i>et al.</i> (2017) - Comparação entre vários algoritmos de <i>Machine Learning</i> na predição da RUL de motores <i>Turbofan</i> .	43
Tabela 3 – Trabalhos Correlatos	46
Tabela 4 – Requisitos do Projeto	49
Tabela 5 – Resultados dos Testes - Computador	58
Tabela 6 – Resultados dos Testes - Sistema Embarcado	58

LISTA DE ABREVIATURAS E SIGLAS

AdaBoost	<i>Adaptive Boosting</i>
AGV	<i>Automated Guided Vehicle</i>
AMLS	<i>Azure Machine Learning Studio</i> [®]
ANN	<i>Artificial Neural Network</i>
Anova	<i>Analysis of Variance</i>
CLP	Controlador Lógico-Programável
CNC	Comando Numérico Computadorizado
CPS	<i>Cyber Physical Systems</i>
CRISP-DM	<i>Cross-Industry Standard Process for Data Mining</i>
DA	<i>Discriminant Analysis</i>
DL	<i>Deep Learning</i>
DNN	<i>Deep Neural Network</i>
DRF	<i>Distributed Random Forest</i>
DT	<i>Decision Trees</i>
EC	<i>Ensemble Classifiers</i>
EUA	Estados Unidos da América
FN	Falso-Negativo
FP	Falso-Positivo
GA	<i>Genetic Algorithm</i>
GBM	<i>Gradient Boosting Method</i>
GLM	<i>Generalized Linear Model</i>
GRU	<i>Gated Recurrent Unit</i>
HMM	<i>Hidden Markov Model</i>
IA	Inteligência Artificial

IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IoT	<i>Internet of Things</i>
KDD	<i>Knowledge Discovery in Databases</i>
KMA	<i>K-Means Algorithm</i>
KNN	<i>K-Nearest Neighbors</i>
LASSO	<i>Least Absolute Shrinkage and Selection Operator</i>
LDR	<i>Light Dependent Resistor</i>
LSTM	<i>Long Short-Term Memory</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
ML	<i>Machine Learning</i>
MSE	<i>Mean Square Error</i>
NASA	<i>National Aeronautics and Space Administration</i>
NoSQL	<i>Not Only Structured Query Language</i>
OSA-CBM	<i>Open System Architecture for Condition-Based Maintenance</i>
PCA	<i>Principal Component Analysis</i>
RAM	<i>Random Access Memory</i>
RBF	<i>Radial Basis Function</i>
RDN	<i>Rede Neural Densa</i>
RF	<i>Random Forest</i>
RL	<i>Regressão Logística</i>
RMSE	<i>Root Mean Square Error</i>
RNN	<i>Recurrent Neural Network</i>
RUL	<i>Remaining Useful Life</i>
RUS	<i>Random Undersampling</i>
SL	<i>Super Learning</i>
SVM	<i>Support Vector Machine</i>

SVR	<i>Support Vector Regression</i>
SW	<i>Sliding Window</i>
TMF	Tempo Médio para Falha
UNISINOS	Universidade do Vale do Rio dos Sinos
VN	Verdadeiro-Negativo
VP	Verdadeiro-Positivo

LISTA DE SÍMBOLOS

<i>sigla</i>	Descrição
<i>mm</i>	Milímetros
<i>obs./s</i>	Observações por Segundo
<i>s</i>	Segundos
<i>GHz</i>	Gigahertz
<i>GB</i>	Gigabytes
<i>V</i>	Volts
<i>mA</i>	Miliampères
Ω	Ohms

SUMÁRIO

1	INTRODUÇÃO	12
2	REFERENCIAL TEÓRICO	14
2.1	Conceitos Básicos	14
2.1.1	Manutenção	14
2.1.1.1	<i>Corretiva</i>	15
2.1.1.2	<i>Preventiva</i>	17
2.1.1.3	<i>Preditiva</i>	19
2.1.2	Sensores	21
2.1.3	Métodos Computacionais	23
2.1.3.1	<i>Mineração de Dados</i>	24
2.1.3.2	<i>Big Data</i>	27
2.1.3.3	<i>Técnicas Inteligentes</i>	28
2.2	Trabalhos Relacionados	31
2.2.1	Estado da Arte	32
3	METODOLOGIA	48
4	ANÁLISE DOS RESULTADOS	54
4.1	Testes	54
4.2	Projeto	61
4.3	Validação	66
5	CONCLUSÃO	72
	REFERÊNCIAS BIBLIOGRÁFICAS	74

1 INTRODUÇÃO

Os avanços tecnológicos trazem diferentes benefícios tanto para a vida cotidiana doméstica quanto industrial. Celulares e computadores disponibilizam um montante significativo de informações com apenas um toque. Assim, dando origem a novos conceitos como Internet das Coisas, que consiste em conectar diversos equipamentos comuns, como lâmpadas, janelas, sistemas de ar condicionado, *etc.*, em uma única rede distribuída. Em um ambiente industrial, esta evolução vai além, trazendo o conceito de Indústria 4.0, onde diversas ferramentas modernas surgem para auxiliar nos diferentes processos de manufatura: sistemas supervisórios, realidade aumentada para verificação do estado de uma máquina, entre outros. Contudo, um tópico que vem ganhando destaque na indústria é a gestão de manutenção, que são geralmente baseadas em ações reativas (corretivas), preventivas e preditivas, sendo esta última um conceito em ascensão e o mais avançado dentre as três.

A escolha correta do modelo de gestão de manutenção que uma indústria adota afeta diretamente nos custos, uma vez que um planejamento bem feito contribui para a redução dos mesmos e melhora o aproveitamento das máquinas. Para tal, sistemas baseados em predição tornam-se mais eficientes, pois são capazes de contornar as falhas presentes nos dois demais métodos de manutenção. Um sistema de predição é baseado em algoritmos computacionais inteligentes capazes de prever, com base em dados históricos e reais de um processo ou equipamento, a vida útil do objeto em análise, possibilitando uma tomada de decisão mais precisa e eficiente quanto às ações que serão realizadas para evitar problemas e/ou potencializar a produção.

O presente trabalho busca apresentar um sistema de predição baseado em um dispositivo embarcado. A intenção é analisar se é possível utilizar os algoritmos de aprendizagem para realizar o processamento dos dados sem a necessidade de um computador, possibilitando a distribuição de sistemas independentes capazes de diagnosticar máquinas em campo. Diferentes informações devem ser entregues ao usuário, possibilitando uma melhoria na tomada de decisões.

Especificamente, o sistema proposto ao longo deste trabalho será validado com base em um ambiente capaz de simular uma atuação real, sendo ele composto por sensores para a aquisição dos dados e variáveis contínuas, que serão armazenadas e processadas pelo sistema embarcado. Devido ao ambiente de validação controlado, falhas serão simuladas, com o intuito de certificar as capacidades de previsão do sistema proposto. Os algoritmos serão estudados e selecionado por meio de testes, os quais irão considerar uma base de dados pública. Os resultados obtidos na etapa de teste devem atender requisitos mínimos de desempenho, visto que este é um ponto crucial na utilização de qualquer dispositivo embarcado.

Desta forma, com os testes corretamente analisados, e a etapa de validação realizada, os resultados serão discutidos. O texto é constituído de uma introdução (o presente capítulo),

seguido pela apresentação do referencial teórico, onde os conceitos básicos e o Estado da Arte são descritos. Após, tem-se o desenvolvimento da metodologia proposta, a apresentação dos testes e resultados de validação provenientes da execução do estudo de caso, concluindo-se o trabalho com uma análise geral do desenvolvimento e diferentes pontos que podem ser melhorados em futuros trabalhos.

2 REFERENCIAL TEÓRICO

Neste capítulo, procura-se introduzir os assuntos abordados neste trabalho. Para tal, alguns conceitos básicos são apresentados como: o que é manutenção, quais os dispositivos utilizados para aquisição dos dados de um sistema e as técnicas computacionais existentes para alcançar o objetivo principal: manutenção preditiva. O Estado da Arte será exposto com artigos que utilizaram de uma manutenção preditiva como solução para algum processo, seja na detecção de falhas ou na estimação da vida útil restante de um sistema. Os trabalhos serão correlacionados de acordo com suas diferentes aplicações, evidenciando os métodos utilizados e seus desempenhos.

2.1 Conceitos Básicos

Na indústria atual, com a grande evolução dos meios de produção, onde robôs são empregados e as fábricas estão cada vez mais inteligentes e conectadas, os produtos devem ser competitivos para garantir o sucesso da empresa. A fim de garantir essa competitividade, a redução de custos torna-se um dos objetivos principais em um sistema de manufatura. Dito isso, um dos pontos mais críticos durante uma produção é a parada de manutenção, que podem ocorrer por uma falha no equipamento, desgaste de uma ferramenta, dentre outras possibilidades. Se for considerada uma máquina-chave para um determinado processo de manufatura, a parada desta máquina de forma não planejada geraria uma perda de produção, uma vez que todo o sistema pararia, gerando um custo alto para a empresa. Na seção a seguir, define-se o que é manutenção e os principais tipos empregados em seu planejamento.

2.1.1 Manutenção

A palavra "manutenção" é derivada do latim *manus tenere* (manter o que se tem), ou seja, "o conjunto de cuidados e procedimentos técnicos necessários ao bom funcionamento e também ao reparo de máquinas", segundo Almeida (2015). Um outro ponto interessante levantado pelo autor é o fato da manutenção não interferir apenas na operação de uma máquina (ou equipamento), mas também em seu projeto, pois se deve considerar a melhor forma de realizar a manutenção: acessibilidade aos conjuntos mecânicos e disposição e dimensão de peças que facilitem sua troca, sendo normatizado por diversos órgãos certificadores (e.g. Norma Brasileira Regulamentadora 5.462/1994). O emprego da manutenção garante alguns benefícios como: uma maior confiabilidade, devido a diminuição dos tempos e do número de paradas; uma melhoria de segurança, pois minimiza os riscos de acidentes e falhas, uma vez que o equipamento torna-se mais previsível; aumento na qualidade, considerando que máquinas mantidas inadequadamente têm maior probabilidade de apresentar um desempenho abaixo do padrão; redução dos custos de

operação; maior tempo de vida útil; e valorização, visto que o equipamento é bem mantido e pode ser vendido facilmente como segunda mão (SELEME, 2015).

Portanto, a manutenção deve ser devidamente planejada ao longo do desenvolvimento e operação de um equipamento. Para tal, o planejamento do processo de manutenção torna-se subjetivo à empresa que o desenvolve, podendo ser centralizada (profissionais dedicados a executar a manutenção), descentralizada (cada área ou unidade realiza os reparos) e híbrida (uma mistura entre a centralizada e descentralizada, também chamada de sistema em cascata), além de ser possível optar por terceirizar a manutenção, onde uma empresa é contratada para efetuar as trocas e ajustes necessários de forma adequada (SELEME, 2015). Visto que, por exemplo, uma indústria siderúrgica apresenta 30% do custo de produção voltado para manutenção, o emprego destes procedimentos e seu planejamento deixaram de ser "um mal necessário" (LOBO, 2010), ganhando foco para auxiliar na redução dos custos de operação e tornar o produto (ou serviço) mais competitivo no mercado.

Basicamente, existem três principais abordagens que podem ser utilizadas no plano de manutenção de uma organização: corretiva, preventiva e preditiva. As duas primeiras são as mais utilizadas pelas indústrias atualmente (LOBO, 2010), enquanto esta última encontra-se em ascensão. Segundo Seleme (2015), cada um destes modelos permitem que certas políticas sejam adotadas, a fim de assegurar que o plano de manutenção seja efetivo e cause pouco impacto na produção, como: redundância (reserva) de equipamentos; o treinamento dos operadores para lidar com as possíveis falhas; um maior número de equipamentos no processo; um projeto robusto, capaz de suportar a sobrecarga de trabalho sem esforço adicional; uma equipe de manutenção capacitada e suficiente para atender ocorrências simultâneas; manutenibilidade, que consiste na compra de equipamentos que permitam uma manutenção fácil de seus componentes; e um estoque suficiente de peças sobressalentes para facilitar a troca imediata de peças danificadas. Estas políticas devem ser estudadas e discutidas durante o planejamento e gerenciamento da manutenção, onde a melhor abordagem deve ser escolhida segundo a complexidade e criticidade da manutenção da máquina no processo a que pertence. As três abordagens citadas serão detalhadas nas próximas seções.

2.1.1.1 Corretiva

Segundo Almeida (2015), a manutenção corretiva destina-se a efetuar o reparo após a ocorrência de uma pane, trocando-se o item defeituoso por um outro em condições de fazer a máquina operar normalmente. Assim, é evidente que este tipo de manutenção só é aplicada durante a ocorrência de uma falha, gerando uma parada imprevista do sistema de produção. Geralmente, este tipo de abordagem é dita como não-programada, pois é realizada de forma imprevisível, aleatória e considerada emergencial (SELEME, 2015). Mesmo que esta ação de manutenção apresente menores custos quanto ao seu planejamento, os prejuízos gerados pela parada da produção e muitas vezes pelo reparo em si tornam este método pouco atraente do

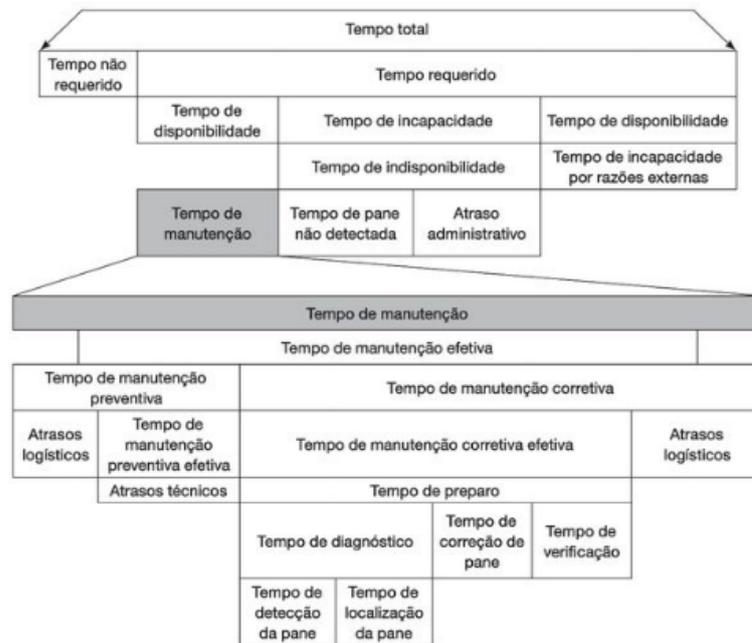
ponto de vista econômico.

Apesar de ser uma técnica de gerência reativa, a manutenção corretiva pode apresentar algumas tarefas básicas para prevenção das falhas, como lubrificação e ajustes da máquina, além de ser necessário um estoque de peças sobressalentes, gerando mais custos para a operação (LOBO, 2010). Para a atuação efetiva da equipe de manutenção corretiva, algumas etapas devem ser seguidas, conforme Seleme (2015):

- **Identificação da falha:** consiste em identificar que o equipamento está com um comportamento anormal, indicando o efeito causado de uma falha (*e.g.* em uma máquina de corte, a mesma para de cortar).
- **Localização da falha:** é a análise do sistema, onde se deve verificar o local da mesma, considerando o desempenho, assim como o seu acesso (com base no exemplo anterior, verificar que a falha ocorreu na área onde a peça é cortada).
- **Diagnóstico:** realiza-se a análise da extensão e o tipo de falha, assim como o modo de correção que será empregado (considerando o mesmo exemplo da máquina de corte, constatou-se que a falha ocorreu devido ao desgaste da ferramenta de corte, sendo necessária sua substituição).
- **Correção:** é a ação efetiva de reparo, onde a equipe de manutenção aplica o método identificado na etapa anterior (a ferramenta de corte é substituída por uma nova).
- **Verificação:** identifica se a manutenção foi eficaz, analisando se o sistema está operando corretamente após a ação corretiva (a máquina de corte voltou a cortar as peças).

Somente após estas etapas, o equipamento em manutenção deve ser devolvido para sua operação normal. Enquanto a equipe responsável pelos reparos é realocada para serviços de usinagem, solda ou operação de máquinas, devendo abandonar estes postos tão logo uma nova necessidade de reparo ocorra (ALMEIDA, 2015). Conforme Lobo (2010), a manutenção corretiva apresenta um custo cerca de três vezes maior que os demais métodos, onde o reparo (ou troca) é feito de forma programada ou preventiva. Além disto, é importante ressaltar que no método corretivo, o tempo é crucial na minimização dos custos devido às paradas imprevistas. A Figura 1 mostra um diagrama envolvendo estes tempos.

Figura 1 – Manutenção Corretiva - Diagrama dos Tempos



Fonte – Seleme (2015)

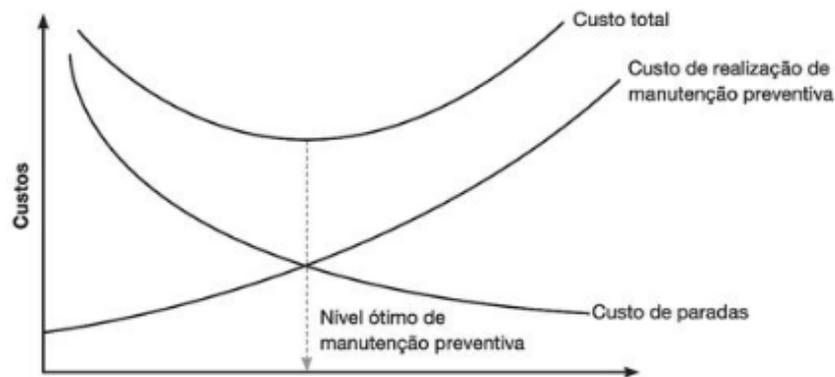
Basicamente, como se pode ver na Figura 1, os principais componentes são: o tempo administrativo e logístico, os atrasos e o tempo de manutenção corretiva efetiva, que consiste no tempo de preparação, localização das falhas, obtenção do item de reposição, correção de falhas, ajustes e calibração, e a liberação da mesma (SELEME, 2015). Em suma, a abordagem de manutenção corretiva apresenta diversas criticidades (tempo, custo, qualificação, disponibilidade) quanto a sua aplicação, sendo necessária uma ação rápida e eficaz para garantir o menor prejuízo possível. Obviamente, esta técnica reativa é ideal para ações não críticas ou de baixa complexidade, visto que a simples troca de uma lâmpada é reparo de atuação reativa (ALMEIDA, 2015).

2.1.1.2 Preventiva

A segunda técnica de manutenção mais utilizada pelas indústrias baseia-se na observação de dados e estatísticas de um equipamento, considerando seu tempo de uso e/ou o número de operações realizadas pelo mesmo. Seleme (2015) afirma que a abordagem preventiva é o cuidado tomado pela equipe responsável pela manutenção, onde as inspeções sistemáticas são executadas para a detecção e correção de falhas, antes de sua ocorrência ou do agravamento das mesmas. Além disto, este tipo de manutenção busca eliminar (ou reduzir) a probabilidade de falhas, utilizando de atividades como limpeza, lubrificação, substituição e verificação das peças que compõem as máquinas. O autor afirma também que esta técnica de manutenção apresenta as seguintes vantagens: aumento da vida útil; redução dos custos; diminuição das

paradas de produção; criação de uma mentalidade preventiva; programação conveniente das ações de manutenção; e melhoria da qualidade dos produtos. Para um plano de manutenção preventiva eficaz, é importante que a ação de reparo seja programada de forma que o seu custo total seja o menor possível, conforme mostra a Figura 2.

Figura 2 – Manutenção Preventiva - Custos Associados



Fonte – Seleme (2015)

Note que o menor custo total ocorre na intersecção entre o custo de realização de manutenção preventiva e o custo de paradas. A manutenção é planejada conforme o Tempo Médio para Falha (TMF), normalmente definido pelo fabricante da maquinaria, apresentando uma alta probabilidade no início da operação de uma máquina nova, pois podem ocorrer falhas devido a problemas na instalação, reduzindo ao decorrer do tempo. Contudo, esta probabilidade aumenta bastante após o período normal de vida da máquina (LOBO, 2010).

Almeida (2015) propõe o uso de formulários contendo o nome da empresa, nome e modelo dos equipamentos, setor onde está instalado, seu fabricante, ano de fabricação e observações de segurança. Este documento seria responsável por conter informações que identificariam as práticas de manutenção programadas, caso ocorra auditorias, troca de funcionários ou até mesmo como evidência do bom estado de conservação do equipamento, quando o mesmo for vendido. Além disto, o autor afirma que uma ficha de controle deve ser preenchida com a data prevista e de execução da manutenção, com a assinatura do responsável pela operação. No entanto, conforme Lobo (2010), a manutenção preventiva não é totalmente eficaz, uma vez que pode levar a reparos desnecessários ou a ocorrência de falhas catastróficas entre os períodos de manutenção.

Entretanto, a aplicação da manutenção preventiva apresenta uma menor probabilidade de uma falha crítica para o sistema de produção, uma vez que diversos cuidados são tomados periodicamente pela equipe responsável por reparos, estendendo a vida útil dos equipamentos. Segundo Seleme (2015), as seguintes ações devem ser realizadas periodicamente:

- **Inspeção:** verificar os itens para determinar a necessidade de manutenção, conforme os

padrões esperados de operação.

- **Manutenção:** limpar e lubrificar os equipamentos, a fim de mitigar possíveis falhas e perda de desempenho.
- **Calibração:** realizar ajustes na máquina, conforme instrumentos com exatidão conhecida e certificada.
- **Testes:** testar componentes com o intuito de identificar problemas ocasionados pela degradação natural do sistema.
- **Alinhamento:** alterar elementos do equipamento, para garantir o desempenho ideal do mesmo.
- **Regulagem:** ajustar os componentes conforme as especificações do equipamento, assegurando o melhor desempenho para o sistema.
- **Instalação:** substituir os itens de vida limitada, desgastados pelo uso da máquina, mantendo o sistema dentro da tolerância exigida.

Para o planejamento de um sistema de manutenção preventiva eficaz, as áreas de interesse devem ser devidamente identificadas, sendo necessário começar por uma ou duas áreas que apresentem um grande número de falhas, a fim de evidenciar o sucesso do plano preventivo. A segunda etapa seria identificar as necessidades da manutenção preventiva, realizando inspeções diárias e ações periódicas para manter o correto funcionamento do equipamento. É importante analisar o histórico do equipamento para estabelecer a frequência das ações de reparo, considerando também as recomendações do fabricante. Além de descrever os procedimentos em detalhes e programar as paradas anualmente. Observado o sucesso do sistema de manutenção preventiva, o mesmo pode ser ampliado para as demais áreas de produção (SELEME, 2015).

2.1.1.3 Preditiva

Por fim, o sistema de manutenção preditiva apresenta uma abordagem mais avançada, utilizando de tecnologias computacionais para estimar a vida útil de equipamentos. Segundo Lobo (2010), a manutenção preditiva deve ser vista como uma filosofia, a qual utiliza da condição operacional em tempo real do equipamento e dos sistemas, para otimizar a operação total da planta industrial. Os principais objetivos da manutenção preditiva são: antecipar a necessidade de serviços de reparo em um determinado componente de uma máquina, aproveitando ao máximo o uso do mesmo; analisar os fenômenos por meio de instrumentos, tornando a desmontagem do equipamento desnecessária; aumentar a disponibilidade do sistema como um todo, visto que a evolução dos defeitos é monitorada e detectada; evitar paradas imprevistas (manutenção corretiva); impedir defeitos graves ocasionados por falhas menores; e reduzir custos e garantir a qualidade do produto (ALMEIDA, 2015).

Esta abordagem utiliza um monitoramento direto das condições mecânicas e/ou elétricas do sistema, analisando seu rendimento e outros indicadores para estimar o TMF do equipamento (LOBO, 2010). Em contrapartida à manutenção preventiva, a preditiva é orientada às condições reais da máquina, não suas estatísticas (SELEME, 2015). Para tal, alguns instrumentos são utilizados com o intuito de identificar as condições normais de funcionamento de uma máquina e caracterizar as falhas, segundo a análise destas condições.

A análise de vibrações é uma técnica bastante empregada, visto que em um ambiente industrial, a maioria dos equipamentos possuem elementos rotativos com eixos (motores), os quais vibram de forma irregular (fora do padrão) quando existe uma falha no sistema. Outra técnica é a análise de lubrificantes, onde a qualidade do óleo lubrificante das máquinas é verificada, uma vez que o desgaste das peças produzem resíduos que contaminam o mesmo, o que altera o comportamento das peças, causando falhas. Além disto, uma análise estrutural pode ser aplicada, com o objetivo de detectar trincas, descontinuidade na solda, alguma avaria por choque mecânico, vazamentos, entre outros. Este tipo de técnica utiliza de ultrassom, ou até mesmo gamagrafia, para realizar um levantamento das condições físicas do equipamento (ALMEIDA, 2015).

Seleme (2015) afirma que muitos dos sistemas de manutenção preditiva empregados não apresentaram benefícios visuais à produção, não por ineficiência dos métodos empregados, mas da falta de capacidade da gestão em propagar o conceito. Segundo o autor, algumas medidas devem ser tomadas para maximizar os benefícios desta abordagem de manutenção, como uma mudança cultural, onde é necessário compreender que as tecnologias de predição são ferramentas de gestão, que devem ser difundidas e presentes em toda a planta da organização; e o uso adequado das tecnologias preditivas, considerando que identificar os problemas individuais de uma máquina não garante a otimização do sistema como um todo, sendo assim, deve-se considerar a planta em sua totalidade para o emprego correto deste sistema de manutenção.

Além disto, Seleme (2015) destaca a necessidade do apoio da alta gerência e a dedicação dos funcionários para garantir a eficácia do programa de manutenção preditiva. Contudo, ele destaca a necessidade de procedimentos de coleta e análise de dados capazes de tratar e interpretar as variáveis que compõem o sistema, que devem ser armazenados em um banco de dados viável para as demandas e sucesso do programa. Se devidamente aplicado, esta abordagem de manutenção traz alguns benefícios ao gerenciamento do processo, como mencionado por Almeida (2015):

- Maior aproveitamento da vida útil do equipamento, considerando que o mesmo irá manter o melhor desempenho pelo maior tempo possível.
- Melhor organização no planejamento das paradas de manutenção, além de um controle melhor sobre a aquisição e estoque das peças.

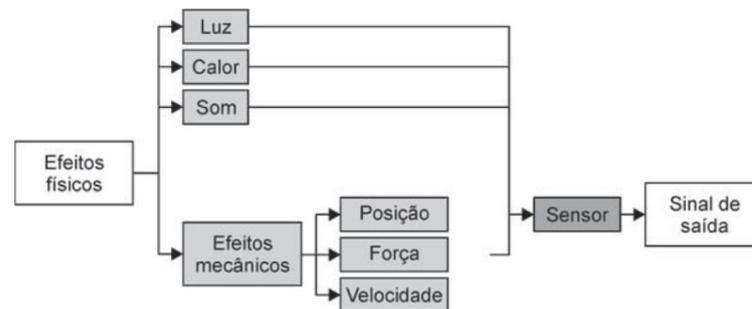
- Os custos de reparos diminuem, uma vez que o local que será reparado será indicado pontualmente (de forma específica), da mesma forma que ocorrem em momentos que não sejam prejudiciais ao fluxo de produção.
- Aumento da eficiência das máquinas, visto que trabalham dentro das melhores condições de desempenho.
- Melhoria nas condições de segurança no trabalho, devido à minimização do número de falhas imprevistas.
- Maior credibilidade ao produto final, uma vez que o sistema permite uma detecção ótima de falhas que poderiam gerar defeitos ao produto.
- Inserção de novas tecnologias, as quais permitem uma melhoria contínua do pessoal de manutenção e da cultura profissional geral da empresa.

Portanto, considerando as vantagens levantadas por Almeida (2015), a manutenção preditiva cobre os problemas remanescentes da abordagem preventiva, garantindo que o equipamento seja aproveitado ao máximo e as falhas, detectadas em tempo real. Com isso, devido aos imensos avanços tecnológicos, diversos dispositivos dedicados a medições físicas são empregados junto aos métodos computacionais para tornar o sistema de predição mais robusto e automatizado. Estes dispositivos são chamados de sensores, e serão tratados em detalhes na próxima seção deste trabalho.

2.1.2 Sensores

Segundo Thomazini e Albuquerque (2011), sensores são dispositivos sensíveis utilizados para realizar a medição de alguma forma de energia da natureza (*e.g.* luminosa, cinética, térmica, elétrica, *etc.*) com base em grandezas físicas (*e.g.* temperatura, velocidade, posição, tensão, corrente, entre outras). Contudo, estes dispositivos nem sempre apresentam as características necessárias para um conexão direta ao sistema de controle, sendo necessário algum circuito que realize o interfaceamento entre o sinal produzido e o controlador. Na Figura 3, tem-se uma ilustração de como o sensor é empregado.

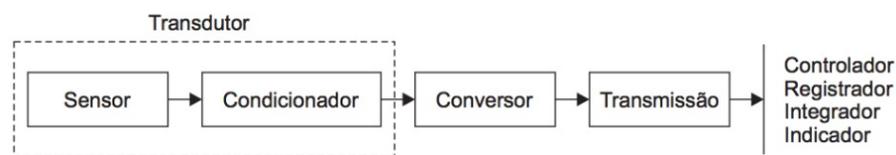
Figura 3 – Diagrama Geral da Utilização de um Sensor



Fonte – Thomazini e Albuquerque (2011)

Alguns destes dispositivos de medição são desenvolvidos de forma a realizar a interface com o controlador, os quais são denominados transdutores. Estes dispositivos apresentam uma solução completa para transformar as grandezas físicas em um sinal de tensão, que é facilmente interpretado pelo sistema de controle. Por esta razão, os termos "sensor" e "transdutor" são usados como sinônimos (THOMAZINI; ALBUQUERQUE, 2011). Na Figura 4, tem-se um diagrama de como o processo de medição é realizado com o uso de sensores, do ponto de vista do sinal lido até a saída, correspondente à entrada do controlador.

Figura 4 – Diagrama da Medição de um Sensor



Fonte – Filho (2014)

Os sensores (ou transdutores) são classificados de acordo com o tipo de variável que é analisada, podendo ser digitais (discretos) ou analógicos (contínuos). Os sinais discretos não são representados por nenhuma grandeza física especificamente (THOMAZINI; ALBUQUERQUE, 2011), mas sim como valores binários ("1" ou "0", ligado ou desligado, presente ou ausente), portanto, segundo Filho (2014), os sensores discretos não deveriam ser denominados sensores, pois se tratam de dispositivos auxiliares automáticos de comando. Estes dispositivos são bastante empregados como instrumentos isolados de medida, devido a sua compatibilidade com sistemas computadorizados digitais. Alguns exemplos deles são os interruptores de proximidade ou dispositivos fotoelétricos, os quais permitem que o controlador interprete estes sinais como eventos, a fim de executar alguma ação no sistema (GROOVER, 2011).

Por outro lado, sensores analógicos são bastante empregados para medir temperatura, pressão, vazão e nível, em indústrias de processos, e distância, velocidade e força, em indústrias de fabricação (FILHO, 2014). Contudo, a medição destas grandezas deve ser processada por meio de um conversor, o qual realizará uma amostragem do sinal para o controlador, utilizando-se de conversores analógico-digital e/ou digital-analógico. Estes dispositivos de conversão são limitados ao seus números de *bits*, que definem a resolução, geralmente em milivolts. O sinal analógico é uma variável contínua, isto é, varia ao longo do tempo, podendo assumir qualquer valor (THOMAZINI; ALBUQUERQUE, 2011).

Algumas características importantes devem ser observadas na seleção do sensor, além do tipo de grandeza que ele é dedicado a medir, como mostrado por Thomazini e Albuquerque (2011):

- **Sensibilidade:** é a capacidade de um sensor em variar frente a uma pequena variação na grandeza medida.
- **Exatidão:** consiste na proximidade entre o valor lido e o real, sendo mais exato à medida que o sinal do sensor se aproxima da resposta verdadeira.
- **Precisão:** representa o grau de repetitividade do valor medido, isto é, uma alta precisão garante que a medição apresenta pouca variação na sua saída, mesmo não sendo exatos.
- **Linearidade:** está relacionado ao comportamento da curva que é obtido pelo sinal de saída, considerando valores de um padrão.
- **Estabilidade:** consiste no quão flutuante a saída do sensor pode ser, o que pode prejudicar a atuação do controlador quando há muitas flutuações.
- **Velocidade de resposta:** é o tempo que o sensor leva para alcançar o valor real do processo, que, idealmente, deve ser instantânea.

Os autores citam algumas outras características, no entanto, as definidas acima são as mais relevantes na seleção de um sensor. Dito isto, é evidente a importância destes dispositivos de medição para a automação de sistemas, pois permitem uma aquisição com maior confiabilidade. Na próxima seção, serão descritos os métodos computacionais empregados para interpretar e elaborar os sistemas de predição.

2.1.3 Métodos Computacionais

Estimar a probabilidade de um sistema falhar até um determinado tempo de interesse faz parte da definição de confiabilidade, sendo definida pela modelagem estatística. Além disto, o tempo até a ocorrência de uma falha está sujeito a variações aleatórias, tornando-o estocástico também, que pode ser obtida discretamente ou continuamente, onde esta última é descrita por uma

função de densidade de probabilidade e uma função de distribuição. Considerando isto, é preciso buscar uma distribuição de probabilidade (*e.g.* exponencial, lognormal, Weibull, *etc.*) que melhor represente o sistema em estudo, que podem apresentar até três parâmetros, classificados em localização (deslocam a distribuição ao longo do eixo do tempo), escala (expandem ou contraem o eixo do tempo) e forma (afetam o formato da função de densidade). Outro ponto importante, é que os estimadores devem ser não-tendenciosos, isto é, não subestimar ou superestimar o valor real do parâmetro; consistentes, devendo convergir rapidamente ao valor real conforme o aumento do número de amostras; eficientes, apresentando uma menor variância dentre os demais estimadores; e suficientes, pois utiliza toda a informação disponível sobre os parâmetros da amostra (FOGLIATTO; DUARTE, 2009).

Como se pôde observar no parágrafo anterior, métodos estatísticos para estimar o tempo até a ocorrência de falhas já são utilizados para predição e tomada de decisões. Contudo, realizar esta análise com uma superabundância de dados incentivou o surgimento de um novo conceito, chamado mineração de dados (CASTRO; FERRARI, 2016). Na próxima seção, a mineração de dados será detalhada e estruturada, identificando o objetivo do processo e suas propriedades.

2.1.3.1 Mineração de Dados

Segundo Castro e Ferrari (2016), o termo mineração de dados (em inglês, *data mining*) é uma alusão ao processo de mineração, onde minerais valiosos são extraídos de uma mina com o uso de ferramentas. Nesta alusão, os autores identificam os minerais como o conhecimento obtido de uma base de dados (mina), utilizando-se de algoritmos computacionais (ferramentas). Com base nisto, é importante que as definições de dado, informação e conhecimento sejam devidamente apresentadas. Silva, Peres e Boscarioli (2016) afirmam que dado é um valor real, algo medido e documentado, enquanto a informação surge apenas quando um significado é atribuído ao dado, sendo obtido o conhecimento quando esta informação é aprendida por algum agente, que realiza a tomada de decisões.

Portanto, a mineração de dados integra um processo mais amplo, denominado *Knowledge Discovery in Databases* (KDD - do português, Descoberta de Conhecimento em Bases de Dados), sendo empregada exclusivamente à etapa de descoberta no KDD, incluindo a seleção e integração das bases de dados, a limpeza das mesmas, a seleção e transformação dos dados, a mineração propriamente e a avaliação dos dados (CASTRO; FERRARI, 2016). A Figura 5 exemplifica este processo de descoberta no KDD.

Figura 5 – Processo de Descoberta no KDD



Fonte – Silva, Peres e Boscaroli (2016)

Note que a mineração de dados ocorre após duas etapas. A obtenção dos dados deve ser organizada em uma base de dados, onde os dados são separados segundo os valores quantitativos e qualitativos de um conjunto de itens, sendo a base para obtenção da informação e, consequentemente, do conhecimento. A etapa de pré-processamento constitui o preparo dos dados para a execução da mineração, onde os dados são filtrados, integrados, selecionados e transformados, garantindo que as informações sejam obtidas da melhor forma possível. Somente feito o pré-processamento, a mineração dos dados pode ser executada com eficácia, onde algoritmos são aplicados para realizar uma análise descritiva, agrupamento, predição, associação e detecção de anomalias nos dados, de forma a extrair o conhecimento das informações processadas. Por fim, a etapa de pós-processamento busca a avaliação e validação dos resultados, identificando o conhecimento útil e não trivial de todo o processo. Vale ressaltar que todas estas etapas estão correlacionadas e não apresentam dependência entre elas, o que explica a horizontalização do processo, devendo todas as etapas serem ajustadas conforme a influência de uma sobre a outra (CASTRO; FERRARI, 2016).

Segundo Silva, Peres e Boscaroli (2016), os dados podem se manifestar como estruturados ou não-estruturados. Os autores afirmam que dados estruturados são tipicamente aqueles advindos de sistemas informatizados convencionais, que são armazenados em estruturas tabulares, onde cada linha armazenada representa um evento com um conjunto de características que descrevem uma instância. Enquanto dados não-estruturados baseiam-se em textos, imagens, vídeos e sons, isto é, fontes não organizadas de dados que devem ser interpretadas de forma a obter-se uma representação mais adequada para a mineração dos dados. Além disto, Silva, Peres e Boscaroli (2016) apresentam algumas convenções, onde a primeira consiste no uso do nome "conjunto de dados" (do inglês, *dataset*) para descrever as bases de dados. Outra convenção é a organização dos dados em linhas com cada atributo descritivo (ou regular) sendo representado em colunas, utilizando de "rótulos" (do inglês, *labels*), que se tratam de atributos específicos com significados especiais, úteis na predição, durante o pré-processamento. Na Figura 6, tem-se um exemplo de base de dados para o processo seletivo de um restaurante, com as convenções

aplicadas, onde y_i representa o rótulo, que define um fator de suma importância na seleção de novos empregados, sua experiência com a área; x_{ij} , os atributos regulares, os quais se referem às descrições dos candidatos; \vec{x}_i , cada evento registrado, ou seja, pessoas que tentaram a vaga de emprego; e ID, um número de identificação, que, no exemplo, trata-se da matrícula.

Figura 6 – Base de Dados com Convenções

	ID	x_{ij}						y_i	
		$i1$	$i2$	$i3$	$i4$	$i5$	$i6$		
\vec{x}_i	Matrícula	Nome	Sexo	Estado civil	Idade	Especialidade	Pretensão salarial	Experiência	
	1	01	João	masculino	viúvo	65	garçom	R\$ 12.000,00	SIM
	2	02	Maria	feminino	solteiro	32	cozinheiro	R\$ 4.200,00	SIM

	n	n	Pedro	masculino	casado	18	entregador	R\$ 1.650,00	NÃO

Fonte – Silva, Peres e Boscaroli (2016)

Como mencionado anteriormente, existem algumas etapas principais pertinentes à mineração de dados, sendo descritas por Castro e Ferrari (2016) como:

- **Análise descritiva dos dados:** consiste em sumarizar e compreender os objetos da base de dados e seus atributos, onde é possível analisar a distribuição de frequência, as medidas de centro e variação, e as medidas de posição relativa e associação dos dados, sem o uso de ferramentas sofisticadas de computação.
- **Predição:** é utilizada para a construção de um modelo capaz de classificar um objeto não rotulado (valores discretos) ou estimar o valor de um ou mais atributos (valores contínuos). Considerando que os rótulos das classes são geralmente conhecidos, este processo recebe o nome de treinamento supervisionado.
- **Análise de grupos:** também chamada de agrupamento, consiste na organização de dados em diferentes grupos, conforme suas características. Este processo busca maximizar a distância intraclasse e minimizar a distância interclasse, tornando os atributos similares mais próximos. Em geral, como os rótulos não são conhecidos, o agrupamento é denominado de treinamento não supervisionado.
- **Associação:** esta análise, também conhecida como mineração de regras de associação, é a descoberta de regras capazes de relacionar diferentes atributos, comumente utilizado em ações de *marketing*, onde regras como "90% dos clientes que compram um *smartphone* assinam um plano de dados" são utilizadas. Com isto, a construção eficiente e quantificação da significância das regras são aspectos importantes para a mineração dos dados.

- **Detecção de anomalias:** mesmo que algumas ferramentas de mineração de dados descartem anomalias (ruídos ou exceções presentes nos dados), dependendo da análise desejada, é necessário que estas anomalias sejam interpretadas pelo processo, mesmo que sua principal característica seja uma baixa frequência frente aos demais conjuntos de dados. Além disto, este tipo de análise deve ser feita por algoritmos capazes de processar estas ocorrências.

Castro e Ferrari (2016) apresentam dicas de como realizar uma análise eficiente e eficaz durante a mineração. Em primeiro lugar, é importante estabelecer a significância da mineração, isto é, se a base de dados apresenta uma boa confiabilidade frente aos resultados obtidos, avaliando se os resultados são coerentes e se os algoritmos apresentam um desempenho desejado. Outro ponto é reconhecer a influência das características dos dados em todos os resultados, além de ser necessário conhecer os dados analisados. O algoritmo utilizado também deve ser escolhido de forma a considerar a complexidade do modelo, visto que cada uma das técnicas podem apresentar resultados distintos. Por fim, verificar possíveis erros em cada uma das etapas (agrupamento, predição, associação e detecção de anomalias) é um passo importante para manter a análise eficiente, assim como o emprego de métodos de validação, para garantir que os resultados fazem sentido e são de boa qualidade.

2.1.3.2 *Big Data*

É evidente que a quantidade de dados armazenados aumentou significativamente, as indústrias estão cada vez mais conectadas, com informações provenientes de diferentes tipos de canais e fontes do mundo todo, alcançando um valor global de um trilhão de *terabytes* por ano. Esta abundância de dados recebe o nome de *Big Data*. Contudo, o conceito de *Big Data* vai além de uma quantidade massiva de dados, estando presente em seu núcleo a capacidade de previsões, com o intuito de obter dados probabilísticos por meio de métodos matemáticos (JR.; CEGIELSKI, 2016).

Segundo Andrade (2015), o ambiente de *Big Data* é formado por diversas fontes de informações, coletadas em tempo real, que vão além de dados de sensores, onde dados vindos da Internet são considerados, como postagens em mídias sociais, vídeos, imagens, entre outros. Basicamente, este ambiente é formado por dados estruturados, não estruturados e semiestruturados, com uma grande variedade de dados, que não se encaixam perfeitamente em bancos de dados tradicionais (JR.; CEGIELSKI, 2016). Andrade (2015) enfatiza que um dos aspectos mais importantes do *Big Data* é seu impacto sobre a tomada de decisões, pois todos os dados obtidos podem ser analisados por meio deste ambiente, em contrapartida aos métodos tradicionais, os quais trabalham com amostragens, onde a decisão era tomada com base na intuição e na experiência pessoal do responsável. O autor ainda cita que três grandes transformações estão ocorrendo: disseminação do uso de dados quantitativos para criação da cultura do ambiente *Big Data*; desenvolvimento de técnicas e algoritmos para o tratamento de *terabytes* de dados, com foco

no negócio; e desenvolvimento da forma como os resultados são apresentados e comunicados, sendo eles transparentes, facilitando assim a tomada de decisão.

O *Big Data* é, conforme Jr. e Cegielski (2016), constituído por três características:

- **Volume:** basicamente, este ambiente conta com uma imensa quantidade de dados, os quais são sempre valiosos, quando analisados de forma eficaz.
- **Velocidade:** a taxa com que os dados são obtidos e tratados pelo *Big Data* permitem uma vantagem competitiva para empresas.
- **Variedade:** como mencionado, este ambiente conta com diversos tipos de variáveis que podem ser utilizadas da melhor forma possível, considerando o objetivo de análise destes dados.

Jr. e Cegielski (2016) ainda apresentam como pode ser feito o gerenciamento deste ambiente, onde banco de dados não relacionais (*Not Only Structured Query Language*, NoSQL) são utilizados para o processamento do *Big Data*, como o Hadoop[®]. Além disto, este ambiente é comumente utilizado por computação em nuvem, onde os dados não estão centralizados em apenas um servidor, permitindo maior capacidade e segurança para os dados (GOLDSCHMIDT; BEZERRA, 2015). Contudo, o processamento de dados pode ainda ser realizado por outros métodos baseados em *Machine Learning* e Inteligência Artificial, que será o tópico da próxima seção.

2.1.3.3 Técnicas Inteligentes

Com os avanços da computação, técnicas inteligentes foram desenvolvidas para realizar tarefas mais complexas, ou simplesmente auxiliar em algumas outras. Segundo Faceli *et al.* (2011), a partir da década de 1970, técnicas baseadas em Inteligência Artificial (IA) foram ganhando força. No entanto, os modelos desenvolvidos eram baseados no conhecimento de especialistas, onde regras lógicas eram obtidas e implementadas como programas de computador. Estas técnicas receberam o nome de Sistemas Especialistas, pois sua aquisição de conhecimento baseava-se exclusivamente em entrevistas realizadas com pessoas experientes no assunto para o qual o sistema estava sendo desenvolvido. No entanto, o alto volume de dados e a crescente complexidade dos problemas fizeram com que os sistemas de IA melhorassem, tornando-se mais sofisticados e capazes frente aos modelos mais antigos.

Dito isto, é importante entender o que é IA e quais as definições e conceitos por trás destes modelos computacionais inteligentes. Primeiramente, Coppin (2017) afirma que uma IA consiste na utilização de métodos computacionais que se baseiam na inteligência humana (ou de outros animais) para solucionar problemas complexos. Sendo que um sistema inteligente apresenta capacidades de adquirir conhecimento, planejar eventos, resolver problemas, representar informações, armazenar este conhecimento adquirido, comunicar-se e aprender (LOPES; SANTOS;

PINHEIRO, 2014). Entretanto, o ramo de estudo sobre IA foi segmentado em duas principais crenças de como este conceito deve seguir, onde de um lado tem-se aqueles que acreditam que os computadores possam ser verdadeiramente inteligentes e conscientes, assemelhando-se à consciência humana, recebendo o nome de IA forte. Enquanto isto, um outro grupo defende que apenas um comportamento inteligente pode ser modelado para a solução de problemas complexos, mas a ação inteligente de um computador não comprova que ele apresente a mesma capacidade intelectual de um ser humano, sendo esta a visão de IA fraca (COPPIN, 2017).

Coppin (2017) apresenta ainda o conceito de métodos fracos e fortes em uma Inteligência Artificial, que não deve ser confundida com os conceitos de IA fraca e IA forte apresentados no parágrafo anterior. Um método fraco utiliza de sistemas de lógicas, raciocínio automatizado e outras estruturas para a solução de problemas, enquanto o método forte considera um conhecimento sólido, sendo este dependente do primeiro, visto que é necessário que uma metodologia seja empregada para lidar com o conhecimento. Quando ambos os métodos são empregados, obtém-se um dos mais importantes segmentos da Inteligência Artificial, o *Machinel Learning* (ML - do português, Aprendizado de Máquina).

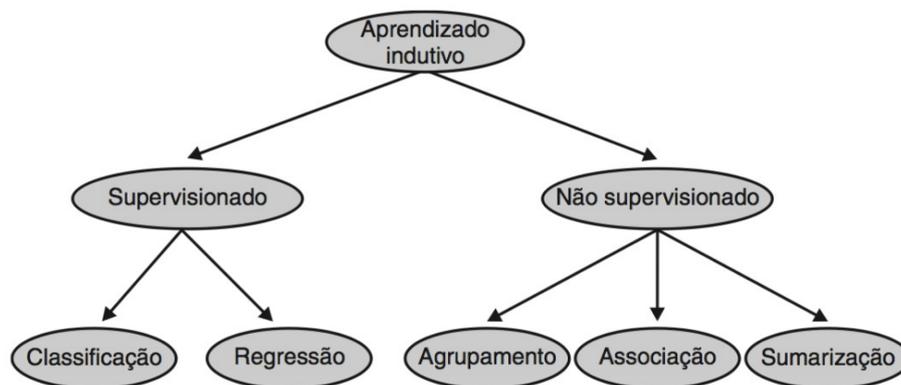
A linha de pesquisa focada em *Machine Learning* busca obter sistemas capazes de adquirir conhecimento de forma automática, onde os algoritmos têm por objetivo descobrir a relação entre as variáveis de um sistema, com base em dados amostrados (LOPES; SANTOS; PINHEIRO, 2014). Aprender é essencial para um comportamento inteligente, onde memorizar, observar e explorar situações é fundamental para aprender fatos, melhorar habilidades e organizar o conhecimento novo. Este processo de aprendizagem é programado em computadores com base em experiências passadas, ou seja, os algoritmos de ML aprendem a induzir uma função, ou hipótese, que é utilizada com o intuito de resolver um problema, baseado em um conjunto de dados (FACELI *et al.*, 2011).

O método de aprendizado indutivo busca encontrar uma função para mapear um conjunto de dados utilizados para treinamento, a fim de elaborar uma hipótese capaz de identificar comportamentos não observados anteriormente. Este treinamento é efetuado com o objetivo de se obter uma generalização para o conjunto de dados, onde exemplos positivos (hipóteses corretas) e negativos (hipóteses falsas) compõem esta característica geral (COPPIN, 2017). Contudo, como mostrado por Faceli *et al.* (2011), uma hipótese deve ser ajustada corretamente, onde uma baixa generalização é justificada como a memorização dos dados de treinamento, isto é, o sistema está superadaptado ao conjunto de dados, enquanto uma baixa taxa de acerto por parte da hipótese indica que os exemplos são pouco representativos, configurando uma condição de subajustamento, devendo o ML ser capaz de lidar com dados imperfeitos (ruídos, inconsistências, ausências e redundância dos dados).

Os algoritmos de ML são divididos em dois principais métodos de aprendizagem: o supervisionado e não supervisionado, como mostra a Figura 7. O aprendizado supervisionado aprende por meio de dados de treinamento pré-classificados (COPPIN, 2017), isto é, por meio

de um "supervisor externo", que conhece o valor desejado para as saídas, a hipótese induzida é avaliada, a fim de garantir que novos exemplos sejam previstos, sendo assim formada por uma etapa de classificação, que utilizam de dados discretos, e regressão, onde seus dados são contínuos (FACELI *et al.*, 2011). Por outro lado, a aprendizagem não-supervisionada busca aprender sem a intervenção humana, sendo útil para dados que necessitem ser classificados ou agrupados em conjuntos, contudo sem conhecimento de classificações prévias (COPPIN, 2017). Basicamente, esta forma de aprendizagem baseia-se em tarefas descritivas com foco em agrupar os dados de acordo com sua similaridade, sumarizar de acordo com uma descrição simples compacta do conjunto de dados ou associar os padrões frequentes entre os atributos destes dados (FACELI *et al.*, 2011).

Figura 7 – Diagrama de Aprendizagem

Fonte – Faceli *et al.* (2011)

Faceli *et al.* (2011) mostram alguns modelos tanto preditivos (supervisionados) quanto descritivos (não-supervisionados). Alguns modelos preditivos utilizam de métodos baseados em distância (*e.g.* *K-Nearest Neighbors* - KNN), probabilísticos (*e.g.* redes Bayesianas), baseados em procura (*e.g.* árvores de decisão) e baseados em otimização (*e.g.* *Support Vector Machine* - SVM), além de modelos múltiplos preditivos. Entretanto, um outro tipo de aprendizagem merece destaque: a aprendizagem por reforço, onde o sistema aprende por meio de reforços positivos e negativos. Por exemplo, um robô que consegue pegar corretamente um objeto recebe um sinal positivo, enquanto um que deixa cair este objeto é reforçado negativamente (COPPIN, 2017). Visto que o objetivo deste trabalho é realizar a predição focada em manutenção, a próxima seção apresenta trabalhos relacionados, mostrando aplicações e exemplos dos diferentes métodos e conceitos básicos explanados ao longo deste capítulo.

2.2 Trabalhos Relacionados

Devido à necessidade de aumento da produção com o melhor aproveitamento das máquinas, o método de manutenção empregado deve ser o mais eficiente possível, a fim de garantir baixos custos, tanto pelo custo relacionado à manutenção em si quanto ao valor perdido com a parada da produção. As ações tomadas pela equipe responsável podem ser divididas em três categorias de manutenção: corretiva, que consiste em realizar o conserto tão logo o equipamento falhe, sendo pouco eficiente e acarretando em um período indesejado de parada na produção; preventiva, a qual é baseada em dados estatísticos que possam indicar o tempo de vida útil de um equipamento (tempo ou operações até a ocorrência de uma falha), podendo ser realizada sem necessidade ou ainda não prevenindo a falha; e preditiva, baseada em um modelo de Inteligência Artificial (IA) e/ou *Machine Learning* (ML) capaz de prever a ocorrência de uma falha com base no estado atual da máquina e de seu histórico. Este último método será o tema abordado neste trabalho, como mencionado nos capítulos anteriores.

Ao longo dos anos, diversos métodos e usos foram dados para os sistemas de predição, a maioria deles para estimar a *Remaining Useful Life* (RUL - do português, Vida Útil Restante) ou o tempo até a ocorrência de uma falha em diferentes aplicações. A fim de abranger estas diferentes aplicações, será visto que tanto o uso de sistemas de *Artificial Neural Network* (ANN - do português, Redes Neurais Artificiais) podem ser usados para estimar a vida útil em máquinas industriais (YAN *et al.*, 2017) como para prever o tempo até a ocorrência de falhas em ferramentas (SPENDLA *et al.*, 2017). Contudo, as redes neurais vão um pouco mais além do que sua forma convencional, como nos é apresentado por Cachada *et al.* (2018), onde foi empregado um sistema de predição utilizando uma *Recurrent Neural Network* (RNN - do português, Rede Neural Recorrente), sendo uma variante deste método. Entretanto, alguns outros métodos matemáticos também se destacam em sistemas de predição, como a regressão linear em aeronaves (WEI, 2012) ou regressão *Kernel*, utilizando *Support Vector Regression* (SVR) para prever a RUL de máquinas (MATHEW; LUO; PANG, 2017).

Métodos de regressão são úteis para a predição de falhas, como mostrado pelo uso de *Elastic Net* na estimação do tempo até o rompimento do filamento de uma máquina de implantação de íons em uma indústria de semicondutores (SUSTO *et al.*, 2012). Em Susto *et al.* (2013), vê-se outro método empregado para o mesmo objeto de estudo (filamento), onde empregou-se o método de regressão *Support Vector Machine* (SVM). Um outro estudo dos mesmos autores propôs ir além de um simples SVM, onde foram utilizados múltiplos classificadores para SVM, sendo comparado com *K-Nearest Neighbors* (KNN) (SUSTO *et al.*, 2015). Assim como Xayyasith, Promwungkwa e Ngamsanroj (2018), que realizaram uma comparação dos métodos SVM, KNN, *Decision Tree* (DT - do português, Árvore de Decisões), *Discriminant Analysis* (DA - do português, Análise do Discriminante) e *Ensemble Classification* (EC - do português, Conjunto de Classificações) na predição de falhas no sistema de resfriamento de uma hidroeletétrica. Além disto, Susto *et al.* (2014) também propõem um método de regressão matemático, onde os dados

são tratados por *Principal Component Analysis* (PCA - do português, Análise do Componente Fundamental) em um sistema de gravação por feixe de íons.

Outros modelos como o *Hidden Markov* (HMM), apresentado por Kinghorst *et al.* (2017), que o utilizou combinado com *Sliding Window* (SW - do português, Janela Deslizante) e *Genetic Algorithm* (GA - do português, Algoritmo Genético) para prever o tempo até a contaminação de uma ferramenta de deposição em semicondutores, também demonstram ser capazes de fornecer um valor confiável. O método *Random Forest* (RF) mostrou ser eficaz, quando aplicado em máquinas de corte (PAOLANTI *et al.*, 2018). Mathew *et al.* (2017) apresentam uma comparação entre diferentes métodos de aprendizado de máquina, como regressão linear, *Decision Tree*, *Support Vector Machine*, *Random Forest*, *K-Nearest Neighbors*, *K-Means Algorithm* (KMA), *Gradient Boosting Method* (GBM), *Adaptive Boosting* (AdaBoost), *Deep Learning* (DL - do português, Aprendizado Profundo) e *Analysis of Variance* (Anova - do português, Análise da Variância), para estimar a vida útil de motores *Turbofan* em aeronaves. Butte, R e Patil (2018) realizam um comparativo entre quatro diferentes métodos (*Generalized Linear Model* - GLM; *Distributed Random Forest* - DRF; *Gradient Boosting Method* - GBM; e *Deep Neural Network* - DNN) para prever o tempo até a ocorrência de uma falha em equipamentos de uma indústria de semicondutores, utilizando o conceito de *Super Learning* (SL - do português, Super Aprendizado) para tornar o sistema de predição mais robusto. Enfim, diferentes métodos podem ser empregados dependendo da abrangência da sua aplicação, por exemplo, a utilização de um sistema avançado de predição que conta com RNN, HMM e alguns outros métodos para prever a RUL e a detecção de falhas de uma infraestrutura domiciliar (KOVALEV *et al.*, 2018).

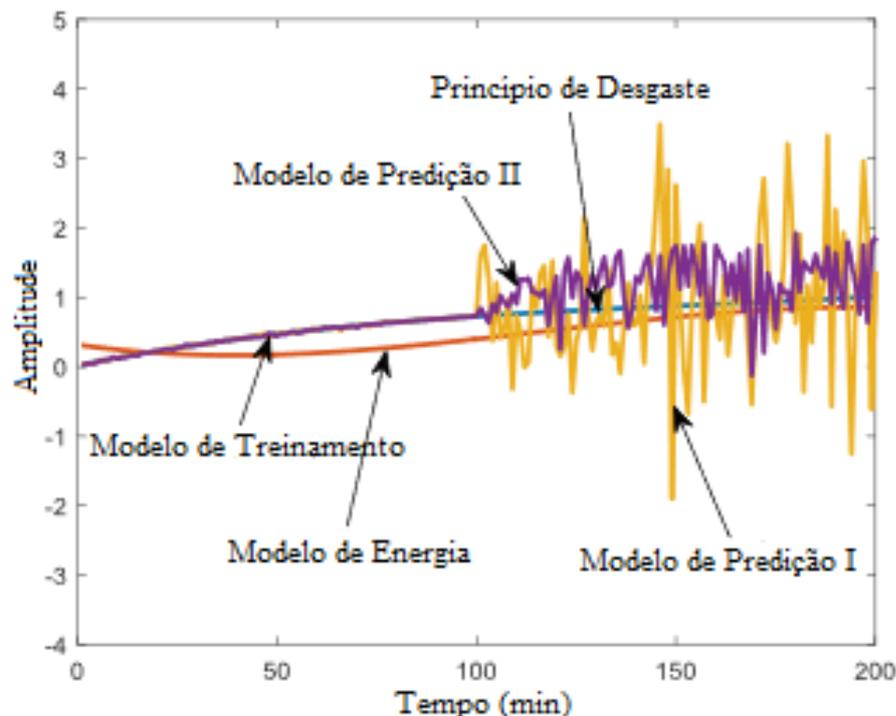
Tendo em vista estas diferentes aplicações, fica evidente o potencial dos sistemas de predição principalmente como ferramenta para estimar a manutenção de máquinas e ferramentas. Na próxima seção, o sistema de predição será detalhado, relacionando os diferentes métodos e aplicações utilizados pelos autores com os resultados e metodologias que os mesmos usaram.

2.2.1 Estado da Arte

Yan *et al.* (2017) apresentam seu trabalho com conceitos que abrangem tecnologias avançadas como *Big Data*, IA, *Internet of Things* (IoT - do português, Internet das Coisas), entre outros. O caso de estudo abordado pelos autores visa o desenvolvimento de um método de manutenção preditiva em uma fábrica inteligente, constituída de máquinas, linhas de montagem flexíveis, uma impressora 3D, manipuladores, *Automated Guided Vehicles* (AGVs), plataformas de computação na nuvem, entre outros. Diversas fontes diferentes de dados compõem a estrutura de *Big Data* utilizada para prever o tempo de vida restante (RUL) de componentes chaves de uma máquina de Comando Numérico Computadorizado (CNC): fresadora. Os dados foram levantados durante a usinagem de um ferro tipo 45 com dimensões $100 \times 100 \times 60 \text{ mm}^3$, onde foram coletados sinais de vibração, imagens (com um escâner 3D à laser), sinais acústicos e dados de consumo de energia. Com isto, os autores focaram no uso dos sinais de vibração da

máquina CNC para avaliar o desempenho atual e prever a vida útil da mesma. O sistema de predição foi desenvolvido com base em uma rede neural, constituída por 30 neurônios em suas camadas ocultas, com 100 amostras utilizadas para treinamento e outras 100 amostras para teste (validação). Dois modelos foram propostos pelos autores, um considerando o método convencional de redes neurais com 10 variáveis como entrada (Modelo de Predição I), enquanto o outro modelo considerava as múltiplas fontes de dados, com a inserção de um Modelo de Energia como décima primeira variável de entrada (Modelo de Predição II). Os resultados obtidos pelos dois modelos podem ser vistos na Figura 8, onde o Modelo de Predição I apresentou um *Mean Square Error* (MSE - do português, Erro Quadrático Médio) de 0,1823 e um desvio padrão de 0,0482, enquanto o Modelo de Predição II obteve um MSE de 0,1504 com o mesmo desvio padrão do primeiro modelo. Yan *et al.* (2017) concluem que o modelo proposto de múltiplas fontes de dados apresenta melhor desempenho quando comparado ao modelo tradicional de predição. É notável, pela Figura 8, que o segundo modelo (linha roxa) obteve maior precisão nos resultados, em comparação ao primeiro (linha amarela), lembrando que a variável chave utilizada no trabalho foi o sinal de vibração da máquina de fresa CNC.

Figura 8 – Resultados obtidos por Yan *et al.* (2017) - Comparação das redes neurais com e sem o Modelo de Energia (múltiplas fontes de dados)



Fonte – Adaptado de Yan *et al.* (2017)

Spendla *et al.* (2017) utilizam uma abordagem diferente, onde o sistema de manutenção preditiva foi desenvolvido utilizando descoberta de conhecimento, baseada na metodologia

Cross-Industry Standard Process for Data Mining (CRISP-DM), a qual consiste em seis estágios: compreensão do empreendimento, compreensão dos dados, preparação dos dados, modelagem, avaliação e desenvolvimento. Os dados utilizados para a plataforma de descoberta de conhecimento são integrados em um *Data Lake* (um repositório onde são armazenados dados brutos) por meio da interface HiveTM, que são tratados utilizando-se a ferramenta Hadoop[®], o qual garante alto desempenho quando utilizada com massas de dados grandes (*Big Data*). Os dados foram coletados em uma indústria anônima, onde se adquiriu dados referentes aos processos de manufatura, materiais, máquinas, equipamentos, ferramentas, entre outros, por 10 meses de produção, o qual totalizou 148.124 aquisições com 340 atributos regulares. Estes dados foram processados pelo modelo de descoberta de conhecimento desenvolvido pelos autores. No Hadoop[®], os dados carregados tiveram seus atributos dispensáveis para o modelo removidos. Em seguida os dados com falhas foram separados dos dados de operação normal, removendo todos os valores não utilizados em ambos os conjuntos de dados, a fim de evitar distorção nos resultados. O modelo de descoberta de conhecimento proposto pelos autores utiliza uma rede neural para classificar as falhas, onde os dados de um mesmo tipo de ferramentas foram utilizados, considerando o mesmo material e a mesma (ou equivalente) operação em um processo de manufatura. O modelo apresentou que a ferramenta analisada deve ser trocada a cada 750 operações, devido a desgastes pelo uso. A intenção de Spendla *et al.* (2017) foi demonstrar o uso do seu modelo de descoberta de conhecimento, com rede neural, para a obtenção do número ótimo de operações até a inutilização da ferramenta analisada.

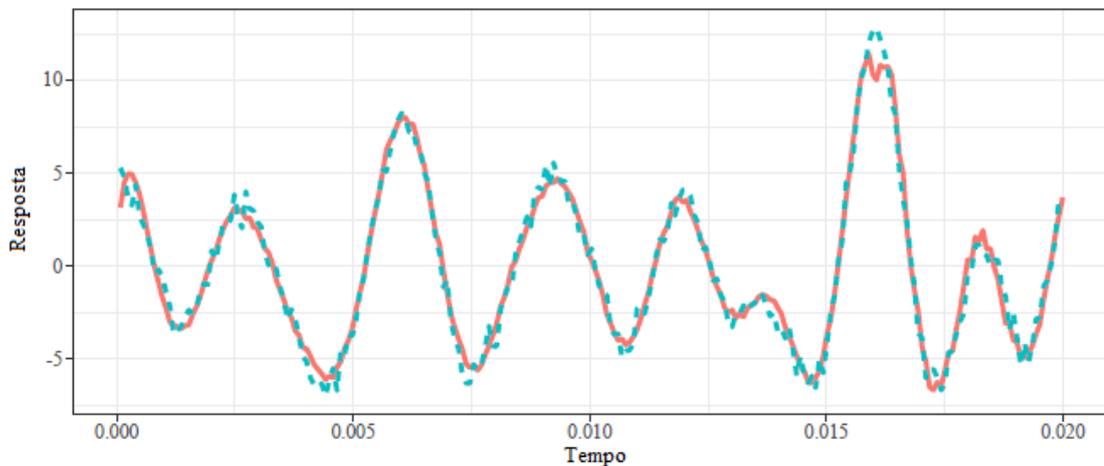
Cachada *et al.* (2018) propõem um sistema de predição baseado em uma arquitetura *Open System Architecture for Condition-Based Maintenance* (OSA-CBM), constituída por seis blocos funcionais: aquisição de dados, manipulação de dados, detecção do estado, avaliação da saúde, avaliação dos prognósticos e geração de sugestão. Os dados foram coletados pelos sensores e pela máquina de estampagem de metal produzida para a indústria automotiva de forma automática, enquanto os dados referentes ao operador foram inseridos manualmente por meio de uma *Human-Machine Interface* (HMI - do português, Interface Homem-Máquina) com acesso direto à base de dados. Concluída esta etapa, os autores testaram uma rede neural recorrente (RNN) com duas variantes: *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU). Os dados de entrada da RNN consistem em mais de 43.000 eventos anteriores, com um total de 147 funcionalidades representadas em um vetor para cada evento. Estes dados foram separados para treinamento (70%) e validação (30%), onde o treinamento foi completado utilizando-se a variante LSTM com 200 células. Cachada *et al.* (2018) apresentam também um módulo de monitoramento dinâmico e de suporte à tomada de decisão. O primeiro consideraria regras criadas pela análise de dados feita pela RNN/LSTM para indicar a necessidade de manutenção devido a alguma falha detectada no instante. O segundo módulo seria responsável por sugerir o procedimento de manutenção mais adequado para a falha identificada. Além disto, o uso de realidade aumentada para auxiliar na manutenção das máquinas é proposto. O sistema de predição apresentado pelos autores é apenas demonstrativo, não havendo resultados concretos

sobre o emprego do sistema, com o objetivo de introduzir uma arquitetura inteligente de predição alinhado ao conceito de Indústria 4.0 e os blocos funcionais OSA-CBM.

Wei (2012) aborda um problema relacionado a aeronaves, buscando um sistema de predição para o número de manutenções efetuadas, devido a falhas nos aviões. O trabalho foi realizado com base em dados levantados em uma filial de uma companhia aérea internacional chinesa, em Guiyang. O conjunto de dados é composto por 8 anos (2004 à 2011) de análise, com dados mensais sobre a manutenção e o número total de aterrissagens de diversos tipos de aeronaves. Foi identificada uma relação linear entre os dados mensais de manutenção e o número de aterrissagens (uma correlação de 0,984). Com isto, o autor propôs um modelo matemático de regressão linear para realizar a predição, baseado no método dos mínimos quadrados para obter os parâmetros do modelo de regressão. Feito isto, o autor apresentou um exemplo de aplicação do modelo proposto, onde foi capaz de prever, a partir de 2010, o número de manutenções efetuadas em diversos modelos de aeronaves pelos três meses seguintes, durante dois anos. Os resultados do sistema de predição indicaram 46, 22 e 34 sortidas de manutenção em Março, Abril e Maio de 2010, respectivamente, para aeronaves modelo *Boeing 737-300*[®]. Os resultados reais são de 46, 21 e 35 manutenções efetuadas para Março, Abril e Maio, respectivamente. Dito isto, é evidente a eficiência do modelo de regressão linear em estimar o número de manutenções devido a falhas, permitindo uma melhor alocação de recursos para as aeronaves encontradas em solo.

Mathew, Luo e Pang (2017) apresentam um método diferente de regressão linear, onde é proposto um modelo de SVR adaptado com funções *kernel* de regressão. Este modelo foi comparado com o SVR tradicional. Os dados utilizados pelos autores foram obtidos por meio de um modelo de combinação linear de múltiplos sinais senoidais, que representam os ciclos de diferentes operações ou máquinas. Dez ciclos diferentes foram gerados utilizando o modelo relatado acima. Mathew, Luo e Pang (2017) buscam demonstrar que, com o modelo proposto de SVR utilizando regressão *kernel*, é possível prever a vida útil restante das máquinas simuladas em seus dados. Dos dados simulados, 70% foram utilizados para o treinamento de ambos os modelos considerados no trabalho. Considerando o problema teórico proposto pelos autores, obteve-se um valor *Root Mean Square Error* (RMSE) de 0,732 para o SVR tradicional, enquanto o SVR com regressão *kernel* foi de 0,552. Na Figura 9, tem-se a comparação do método SVR com regressão *kernel* (linha vermelha), proposto pelos autores, com o valor real do sinal (pontilhado azul). O trabalho ainda ressalta que o estudo apresentado trata-se de um modelo preliminar (em progresso) para ser usado em aplicações de prognóstico.

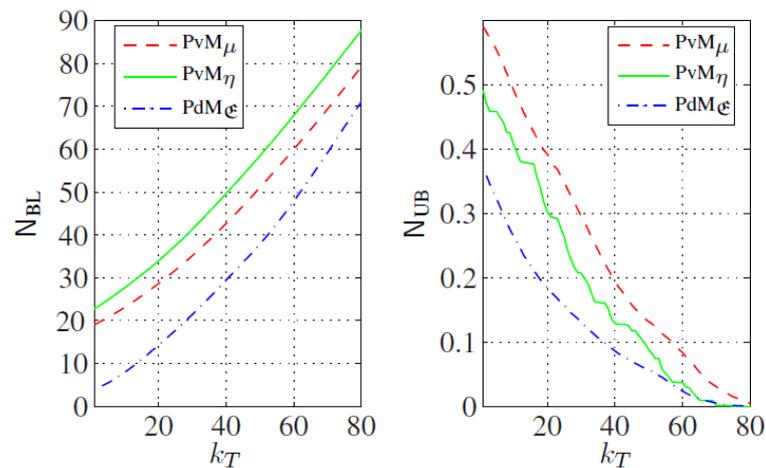
Figura 9 – Resultados obtidos por Mathew, Luo e Pang (2017) - Comparação do método de predição usando SVR (regressão *kernel*) com o sinal real



Fonte – Adaptado de Mathew, Luo e Pang (2017)

Susto *et al.* (2012) contemplam um método diferente de regressão linear. Em seu trabalho, eles desejam elaborar um sistema de predição para estimar a vida útil restante do filamento de uma máquina de implantação de íons, em uma indústria de semicondutores. Esta máquina é um gargalo para o sistema de produção dos *wafers*, o que motivou a pesquisa dos autores. Para o desenvolvimento do sistema, considerou-se uma base de dados com manutenções corretivas (troca dos filamentos), pois dados vindos da manutenção preventiva poderiam mascarar informações, uma vez que não é certo que o objeto analisado realizou um ciclo de vida completo. Susto *et al.* (2012) propõem a utilização de *Elastic Net*, baseada no algoritmo *glmnet* (métodos de menores coordenadas cíclicas), para a predição do rompimento do filamento da máquina de implantação de íons. A massa de dados disponível para os autores consistia em 33 ciclos de manutenções corretivas de um total de 3.671 lotes produzidos, com 125 variáveis físicas. O teste do método de predição baseado em *Elastic Net* ocorreu por meio de uma validação cruzada de Monte Carlo, onde os resultados mostram uma predição pessimista (pior caso), quando comparado ao método tradicional de validação. Efetuou-se 10.000 simulações de Monte Carlo, onde buscou-se prever o número médio de manutenções indesejadas, representado por N_{UB} , e de lotes que ainda poderiam ser processados com o mesmo filamento antes de uma troca preventiva, N_{BL} . Os resultados obtidos podem ser observados na Figura 10, a qual é possível observar que a predição foi capaz de gerar um número menor para N_{UB} e N_{BL} , quando comparado a média (μ) e mediana (η) do tempo de troca do filamento, utilizados como base para sistemas de manutenção preventiva, onde o filamento é trocado a cada $\mu - k_T$ (ou $\eta - k_T$) lotes. Estes resultados indicam que o sistema de predição proposto é superior ao preventivo em relação ao número de manutenções indesejadas, que acarretam em um custo maior, devido à parada da máquina, e ao desperdício do filamento no processamento dos *wafers*.

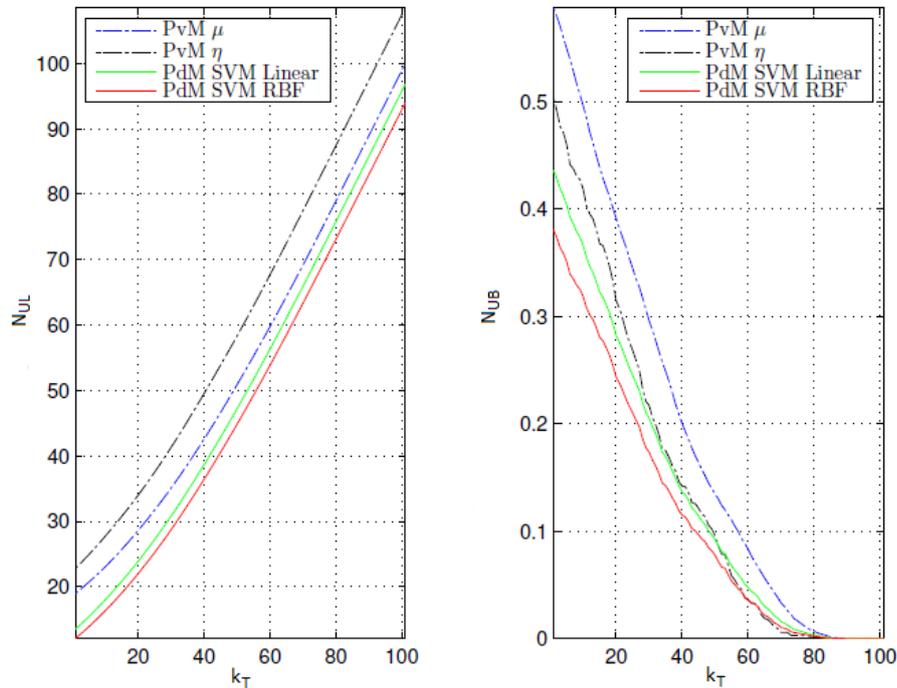
Figura 10 – Resultados obtidos por Susto *et al.* (2012) - Comparação do método de predição *Elastic Net* (PdM) com a média e mediana (PvM)



Fonte – Susto *et al.* (2012)

Em Susto *et al.* (2013), tem-se uma continuação do trabalho exposto no parágrafo anterior. Aqui, os autores utilizam do mesmo objeto de estudo (rompimento do filamento de uma máquina de implantação de íons) exposto no trabalho anterior. Contudo, eles propõem o emprego do método SVM, na sua forma tradicional (linear) e a variante *Radial Basis Function* (RBF). Como o caso estudado por Susto *et al.* (2013) é o mesmo de Susto *et al.* (2012), a mesma massa de dados foi utilizada, onde foram considerados 3.671 lotes, com 33 ciclos de manutenção corretiva e 125 variáveis físicas, apenas remodelando o sistema de predição para o uso de SVM. O método de Monte Carlo também foi utilizado para a validação, com 1.000 simulações, e as variáveis de interesse continuaram sendo o número médio de manutenções indesejadas (N_{UB}) e de lotes que ainda poderiam ser processados pelo filamento anterior a troca (N_{UL}). Os resultados obtidos pelos autores foram comparados com a média e a mediana, como realizado no trabalho anterior. Na Figura 11, tem-se um gráfico semelhante ao mostrado pela Figura 10, onde também é possível verificar que o método de predição apresentou melhor desempenho em relação ao sistema de manutenção preventiva, com destaque para a variante RBF, pois garantiu melhor aproveitamento do filamento com um risco baixo.

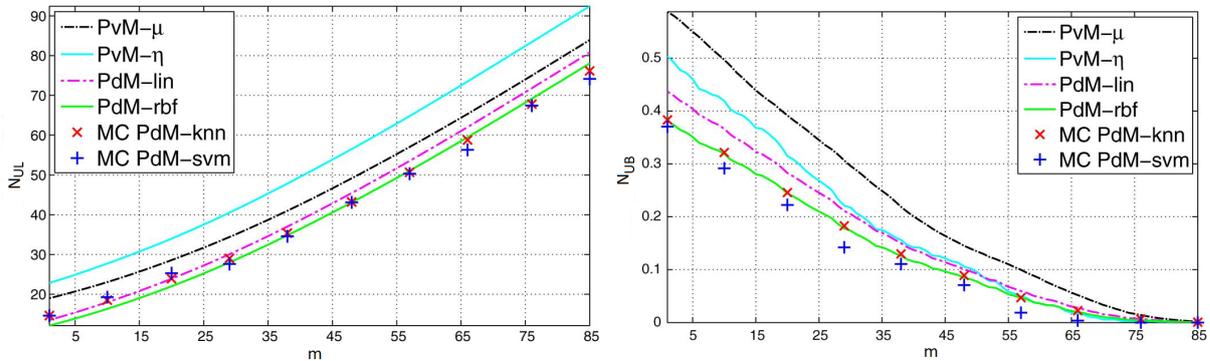
Figura 11 – Resultados obtidos por Susto *et al.* (2013) - Comparação do método de predição SVM (PdM), linear e RBF, com a média e mediana (PvM)



Fonte – Adaptado de Susto *et al.* (2013)

Não obstante, Susto *et al.* (2015) realizam uma nova abordagem para o caso do filamento da máquina de implantação de íons, indo mais além das simples técnicas de regressão. Neste trabalho, os autores propõem um modelo de múltiplos classificadores para SVM e KNN, utilizando os mesmos dados levantados pelos trabalhos anteriores (3.671 lotes, 33 ciclos de manutenção corretiva e 125 variáveis físicas). Métodos de predição baseados em múltiplos classificadores apresentam um melhor desempenho quanto maior for o número de classificadores k , pois estes fornecem mais informações referentes ao processo analisado. Considerando as mesmas variáveis N_{UB} e N_{UL} descritos anteriormente, os resultados obtidos foram comparados com os resultados de Susto *et al.* (2013), mostrados na Figura 12. Onde m representa o horizonte de falhas, com um k igual a 10 e considerando uma validação de Monte Carlo de 1.000 simulações. É possível verificar que tanto os métodos de predição tradicionais (SVM linear e RBF) quanto os de múltiplos classificadores (SVM e KNN) apresentam resultados similares. Além disto, a precisão dos métodos de múltiplos classificadores diminui à medida que o horizonte de falhas aumenta. Apesar das similaridades, o método SVM com múltiplos classificadores mostrou melhor desempenho em relação aos demais modelos.

Figura 12 – Resultados obtidos por Susto *et al.* (2015) - Comparação dos métodos de predição KNN e SVM - linear, RBF e com múltiplos classificadores - (PdM), com a média e mediana (PvM)



Fonte – Adaptado de Susto *et al.* (2015)

Xayyasith, Promwungkwa e Ngamsanroj (2018) propõem um modelo de predição para a manutenção do sistema de resfriamento da planta hidroelétrica *Nam Ngum-1*, situada em Laos. Os dados utilizados datam desde 1972, quando a planta entrou em operação, contudo, dados entre 2016 e 2018 mostram um alto custo de manutenção, que era agendada (preventiva) de acordo com o perfil dos dados medidos, principalmente temperatura. Com isto, os autores apresentam um estudo comparativo entre métodos de *Machine Learning* (ML): árvores de decisão (DT), análise discriminante (DA), SVM, KNN e EC. Para cada um destes métodos, foi utilizado diferentes algoritmos de classificadores: complexa, média e simples (DT); linear e quadrática (DA); linear, quadrática, cúbica, Gaussiana refinada, Gaussiana média e Gaussiana grosseira (SVM); refinada, média, grosseira, cossenoidal, cúbica e por pesos (KNN); e *Boosted Trees*, *Bagged Trees*, discriminante do subespaço, subespaço KNN e *Random Undersampling* (RUS) *Boosted Trees* (EC). O estudo considerou o intervalo de 2016 à 2018, onde dois anos (67%) dos dados levantados foram utilizados para treinamento e um ano (33%), para validação. Os resultados foram observados para todos os diferentes tipos de classificadores, gerando a Tabela 1, a qual compara a precisão (percentual), velocidade de predição (observações por segundo) e tempo de treinamento (segundos). Os algoritmos que se destacaram foram o SVM (linear, quadrático e cúbico) e a árvore de decisões (complexa, média e simples), além de *Bagged Trees* (EC). A árvore de decisão média apresenta o melhor tempo de treinamento, enquanto a árvore de decisão simples apresenta a maior velocidade de predição, dentre os algoritmos destacados pelos autores.

Tabela 1 – Resultados obtidos por Xayyasith, Promwungkwa e Ngamsanroj (2018) - Comparação entre vários algoritmos de *Machine Learning* na predição de falhas no sistema de resfriamento de uma hidroelétrica

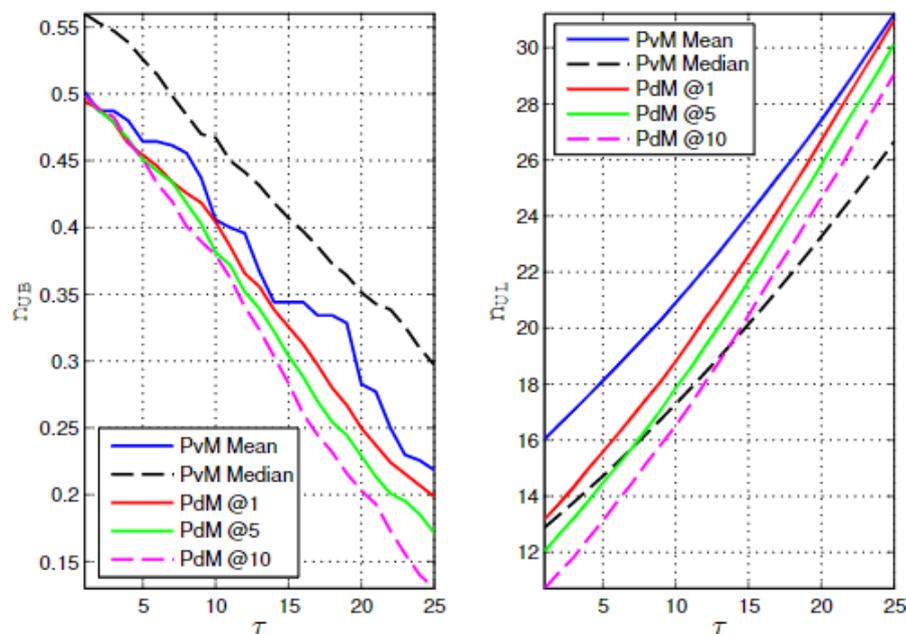
Algoritmo	Precisão	Velocidade de Predição	Tempo de Treinamento
DT: Complexa	100%	52.000 obs./s	15,62s
DT: Média	100%	56.000 obs./s	13,11s
DT: Simples	100%	240.000 obs./s	18,627s
DA: Linear	93,1%	110.000 obs./s	17,778s
DA: Quadrática	99,2%	110.000 obs./s	15,957s
SVM: Linear	100%	56.000 obs/s	37,658s
SVM: Quadrática	100%	52.000 obs/s	93,907s
SVM: Cúbica	100%	120.000 obs/s	68,983s
SVM: Gaussiana Refinada	99,9%	25.000 obs/s	84,301s
SVM: Gaussiana Média	99,7%	34.000 obs/s	95,129s
SVM: Gaussiana Grosseira	99,9%	19.000 obs/s	106,96s
KNN: Refinada	100%	110.000 obs/s	97,517s
KNN: Média	99,7%	72.000 obs/s	97,987s
KNN: Grosseira	99,1%	20.000 obs/s	100,63s
KNN: Cossenoidal	96,6%	5.600 obs/s	108,97s
KNN: Cúbica	99,7%	42.000 obs/s	108,63s
KNN: por Pesos	100%	67.000 obs/s	109,35s
EC: <i>Boosted Trees</i>	82,7%	160.000 obs/s	112,08s
EC: <i>Bagged Trees</i>	100%	31.000 obs/s	117,69s
EC: discriminante do subespaço	93,1%	20.000 obs/s	117,13s
EC: subespaço KNN	85,8%	10.000 obs/s	125,7s
EC: RUS <i>Boosted Trees</i>	82,7%	360.000 obs/s	117,43s

Fonte – Adaptado de Xayyasith, Promwungkwa e Ngamsanroj (2018)

Susto *et al.* (2014) apresentam uma ferramenta de predição de falhas do tipo integral, as quais ocorrem com a degradação gradual de uma máquina. O trabalho busca estimar a vida útil restante (RUL) de um sistema de gravação por feixe de íons em uma indústria de semicondutores, a qual apresenta falhas no módulo de detecção do equipamento devido à degradação. O modelo proposto pelos autores é baseado no método de regularização linear *Least Absolute Shrinkage and Selection Operator* (LASSO). Os dados foram levantados em uma matriz bidimensional, provenientes de uma espectrometria de emissão ótica, que necessita ser reduzida a um subconjunto de dados que contenham apenas as informações necessárias para o sistema de predição, por meio da técnica matemática PCA. O conjunto de dados utilizado para o treinamento do sistema de predição baseado no modelo LASSO consistia em 1.439 *wafers* produzidos, com 22 ciclos de manutenção corretiva e 1.201 variáveis de regressão, que foram reduzidos para 72 variáveis após o uso do PCA, dos quais foram extraídas algumas informações estatísticas: média, variância, máximo e mínimo. O teste do modelo de predição foi realizado com uma validação cruzada de Monte Carlo, onde foi realizado 100 simulações. Os autores buscaram a fração de paradas inesperada (n_{UB}) e a média da vida útil inutilizada (n_{UL}) do sistema estudado,

com base na quantidade k de processamento após uma manutenção e no tempo admissível para manutenção (τ), que deve ser determinada para o sistema de manutenção preventiva. A Figura 13 mostra os resultados obtidos por Susto *et al.* (2014), onde os resultados são comparados entre o sistema preventivo (PvM), usando a média e mediana dos dados, e o sistema de predição (PdM) para diferentes valores de k . É possível verificar que o PdM apresenta um desempenho melhor que os métodos preventivos, principalmente para valores maiores de k , com exceção do n_{UL} , onde, para valores de τ maiores que 15, o PvM considerando a mediana foi melhor em comparação ao PdM. Contudo, se considerar que o custo relacionado a n_{UB} é muito maior que ao n_{UL} , este resultado pode ser desprezado.

Figura 13 – Resultados obtidos por Susto *et al.* (2014) - Comparação do método de predição para diferentes valores de k (PdM) com a média e mediana (PvM)



Fonte – Susto *et al.* (2014)

Kinghorst *et al.* (2017) conduzem seu trabalho com o objetivo de obter um sistema de predição para estimar a vida útil (tempo até contaminação) de ferramentas de deposição, em uma indústria de semicondutores. A bordagem proposta pelos autores busca atender alguns requisitos: encontrar os padrões dos dados de uma forma não-supervisionada, considerando o histórico dos dados; interpretar dados dinâmicos; suportar taxas assíncronas de amostragem; variação do intervalo dos valores entre diferentes ciclos de produção; estimar a degradação em longo prazo da ferramenta sem comparar dados subsequentes; e selecionar os dados usados para a predição de forma automática. Para cumprir os requisitos apresentados, eles propõem o uso de *Sliding Window* (SW), para lidar com taxa assíncronas de dados, à medida que o método busca padrões no conjunto de dados. Além disto, utilizam de algoritmo genético (GA) com o

intuito de selecionar os dados de forma automática e otimizada. Considerando que o sistema de predição deve ser não-supervisionado, capaz de detectar a degradação em longo prazo e variáveis dinâmicas, os autores baseiam sua abordagem em *Hidden Markov Model* (HMM). Portanto, a abordagem utilizada pelo trabalho de Kinghorst *et al.* (2017) consiste em um sistema com SW, GA e HMM, a fim de garantir os requisitos levantados. O conjunto de dados foi composto por oito ciclos de produção (cada um com 25 à 30 dias de dados) contendo até 4.000 *wafers*, para duas ferramentas de deposição. As técnicas foram aplicadas junto a análises estatísticas e diferentes subconjuntos de dados foram utilizados para treinamento, resultando em um sistema capaz de estimar a probabilidade de contaminação das ferramentas, em relação aos dias de produção. Estes resultados mostraram que seis dos oito ciclos de produção analisados no trabalho têm estimativas similares, indicando que o sistema elaborado pelos autores teve sucesso em apresentar uma distribuição confiável das probabilidades de contaminação das ferramentas estudadas.

Paolanti *et al.* (2018) apresentam uma abordagem diferente para o uso de *Machine Learning*, onde o método de *Random Forest* (RF) foi utilizado com *Azure Machine Learning Studio*[®] (AMLS), uma plataforma *online*, fornecida pela *Microsoft Corporation*[®], focada no desenvolvimento de sistemas de inteligência artificial. Os autores propõem o desenvolvimento de um sistema de predição baseado na plataforma *online*, visando prever falhas em uma máquina de corte de uma indústria de carpintaria. A massa de dados consiste em 530.731 leituras dentre 15 características diferentes da máquina analisada, considerando o estado do rotor como a variável que indica a condição da máquina. Os dados foram coletados em tempo real e processados no AMLS, onde 30% destes compuseram o conjunto de treinamento do RF, sendo o restante utilizado para a validação da predição. Um classificador de *Decision Forest* foi utilizado para obter melhor desempenho durante a classificação, que consiste em uma decisão baseada em voto por maioria (classe de saída mais comum dentre diferentes árvores de decisão). Além disto, os dados processados na plataforma foram divididos em três categorias: dados da máquina CNC, relacionados ao sinais de entrada e saída advindos de um Controlador Lógico-Programável (CLP), e de sensores de vibração. Os resultados apresentados por Paolanti *et al.* (2018) indicaram uma precisão de 95% na predição de diferentes estados da máquina, significando que o sistema foi capaz de prever o estado do rotor com eficiência.

Mathew *et al.* (2017) baseiam seu trabalho na estimação da vida útil restante (RUL) de um motor *Turbofan* em aeronaves. Para tal, os autores utilizaram um repositório de dados fornecido pela *National Aeronautics and Space Administration* (NASA), incluindo medidas corretivas de sensores de um motor *Turbofan* em estado de degradação. Cada um dos motores analisados apresenta 21 sensores coletando diferentes tipos de dados referentes ao motor. Estes dados foram divididos em conjuntos para a realização do treinamento, para teste e para verificar a precisão do sistema de predição. Os autores compararam a eficácia de diferentes métodos de *Machine Learning*: regressão linear, árvore de decisões (DT), SVM, *Random Forest*, KNN, KMA, GBM, AdaBoost, *Deep Learning* e Anova. Estes métodos foram empregados utilizando os mesmos conjuntos de dados para prever a RUL dos motores. Os resultados obtidos pelos

autores podem ser observados na Tabela 2, onde os valores RMSE são comparados entre os diferentes métodos para diferentes conjuntos de dados (*Dataset*). É possível verificar que o *Random Forest* apresentou o menor erro dentre os demais métodos, além de apresentar o maior número de observações que fazem parte da predição, enquanto o KMA mostrou ser o pior dentre os dez métodos apresentados.

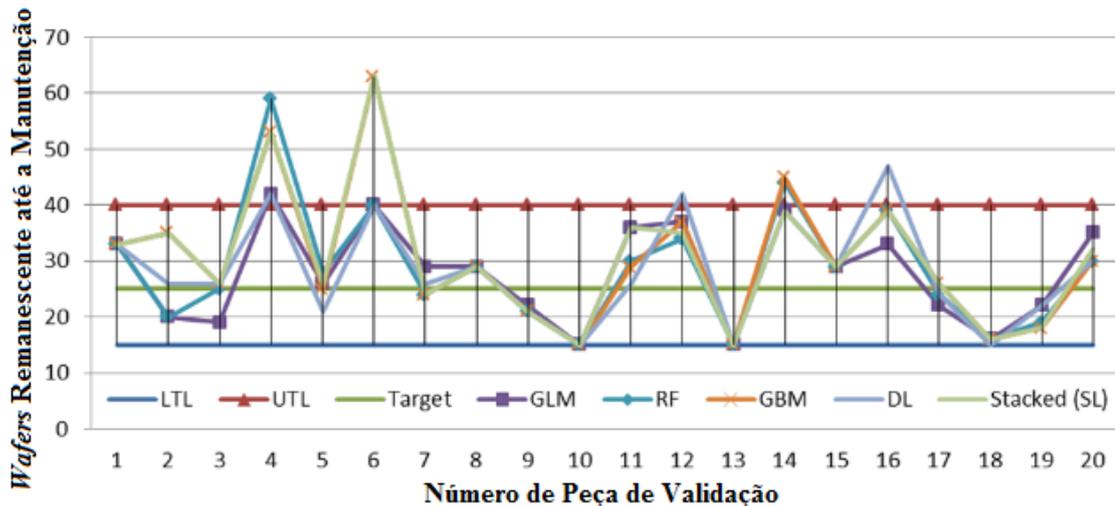
Tabela 2 – Resultados obtidos por Mathew *et al.* (2017) - Comparação entre vários algoritmos de *Machine Learning* na predição da RUL de motores *Turbofan*

Método	<i>Dataset 1</i>	<i>Dataset 2</i>	<i>Dataset 3</i>	<i>Dataset 4</i>	Média
Regressão Linear	29,91	31,49	45,64	39,81	36,71
Árvore de Decisões	28,48	34,52	27,74	45,91	34,17
SVM	48,17	31,12	61,53	34,65	43,86
<i>Random Forest</i>	24,95	29,64	30,55	33,79	29,73
KNN	30,79	34,79	34,44	44,70	36,18
KMA	78,30	90,19	72,92	95,45	84,21
GBM	27,45	33,35	31,78	39,30	32,97
AdaBoost	28,82	33,84	30,91	39,16	33,18
<i>Deep Learning</i>	29,62	42,41	46,82	38,11	39,24
Anova	33,50	41,14	35,46	51,44	40,38

Fonte – Adaptado de Mathew *et al.* (2017)

Butte, R e Patil (2018) propõem a elaboração de um sistema de predição para gerenciar a troca de um componente crítico de um equipamento em uma indústria de semicondutores. Os dados são baseados em 21 sensores, 3 parâmetros de receita do processo e um contador de *wafers*. Os autores utilizaram de diferentes métodos (GLM, DRF, GBM e DNN) para a elaboração do sistema, com o intuito de comparar os resultados e obter o melhor desempenho na predição. Além disto, uma das principais propostas do trabalho é identificar a eficiência do *Super Learning* (SL) na validação dos métodos de predição. Com base nisto, foi considerado 80% dos dados para o treinamento do sistema, enquanto os demais dados foram utilizados para validação. O método utilizado para gerenciar a manutenção do equipamento estudado era preventivo, onde as estimativas indicaram que os componentes possuem uma capacidade próxima de 25 *wafers*. Uma análise inicial mostrou que o DNN apresenta melhor desempenho que os demais métodos. Com a intenção de tornar o modelo de predição mais robusto, o SL foi adotado, onde se buscou uma combinação ótima das bases de aprendizado e, baseando-se em *Random Forest*, GBM e DNN, obteve-se um modelo de meta-aprendizado, o qual foi treinado com os mesmos dados dos demais métodos. Os resultados destas diferentes aplicações são mostrados na Figura 14, onde o *Target* é a estimativa da manutenção preventiva (25 *wafers*), enquanto LTL e UTL representam a tolerância mínima (15 *wafers*) e máxima (40 *wafers*), respectivamente. O método de *Super Learning* é representado pelo *Stacked*, pois envolve três diferentes métodos empilhados. Segundo os autores, a abordagem por SL vai reduzir o risco de alta variabilidade e imprecisões, provenientes do uso de um simples método, o que evidencia a robustez da abordagem na predição.

Figura 14 – Resultados obtidos por Butte, R e Patil (2018) - Comparação do número de *wafers* processados até a execução da manutenção para diferentes métodos de predição



Fonte – Adaptado de Butte, R e Patil (2018)

Por fim, Kovalev *et al.* (2018) apresentam uma proposta mais complexa para o uso de sistemas de predição: manutenção de infraestrutura domiciliar e serviços públicos. Vários sensores foram instalados para realizar a medição dos seguintes equipamentos: transformadores de energia, bombas centrífugas, aquecedores, sistemas de ventilação e de ar condicionado, caldeiras à gás e lâmpadas. Desta forma, temperatura, pressão, corrente, tensão, nível de ruído, de água e de gás, velocidade do ar, e ocorrência de vazamento de água e gás compuseram o conjunto de dados utilizados. Controladores foram conectados a estes sensores, que coletavam os dados a cada 30s, enviando-os para um módulo de *Data Harvesting* baseado em Apache NiFiTM. O sistema de predição apresentado no trabalho busca prever falhas e realizar o monitoramento em tempo real dos equipamentos mencionados. Para atingir este objetivo, diversos métodos estatísticos e de *Machine Learning* foram empregados. Os dados foram pré-processados utilizando SparkTM, enquanto que a extração das características dos dados pré-processados deu-se por: *Short-Time Fourier Transform*, *Wavelet Transform*, *Seasonal Autoregressive Integrated Moving Average*, *Principal Component Analysis*, *Independent Component Analysis* e *Partial Least Squares Regression*. Para estimar a vida útil e estimar a degradação dos equipamentos, o método RNN com LSTM foi utilizado, enquanto HMM foi empregado para monitorar as condições das máquinas. Para a detecção das falhas, o método de regressão SVM foi proposto, com o intuito de realizar uma classificação linear. Kovalev *et al.* (2018) não trazem um resultado prático para o sistema de predição modelado, mas uma comparação entre os diferentes métodos propostos no trabalho, evidenciando as vantagens, desvantagens e aplicações de cada um dentro do objeto de estudo proposto (infraestrutura domiciliar e serviços públicos).

Os trabalhos apresentados até então apresentam diversas aplicações, em que diferentes métodos foram propostos. Na Tabela 3, tem-se um resumo para cada artigo avaliado. É importante ressaltar que as diferentes abordagens realizadas pelos autores compõem o Estado da Arte deste trabalho, pois é evidente a variedade de soluções possíveis utilizando-se um sistema de predição, além dos ganhos de desempenho na sua utilização.

É possível observar nos trabalhos apresentados nesta seção que os sistemas de predição podem ser empregados em diversas aplicações, desde a indústria até o ambiente doméstico. Observa-se que alguns métodos apresentam destaque em muitas aplicações, como o SVM, utilizado por Susto *et al.* (2013), Susto *et al.* (2015) e Xayyasith, Promwungkwa e Ngamsanroj (2018) e como *Random Forest* (MATHEW *et al.*, 2017) e (PAOLANTI *et al.*, 2018). Sistemas baseados em ANN também apresentaram destaque, visto em Yan *et al.* (2017), Spendla *et al.* (2017), Kovalev *et al.* (2018) e Cachada *et al.* (2018).

Além disto, Mathew, Luo e Pang (2017), Wei (2012), Susto *et al.* (2014), Susto *et al.* (2012) apresentam diferentes métodos de regressão linear para atingir seus objetivos. Em Butte, R e Patil (2018), o comparativo dos métodos de *Machine Learning* propostos e a apresentação do conceito de *Super Learning* mostraram combinações interessantes para aumentar a robustez do sistema de predição. Enquanto a combinação de algoritmo genético para tratamento de dados com HMM mostrou-se eficaz em estimar a vida útil de ferramentas (KINGHORST *et al.*, 2017).

Tabela 3 – Trabalhos Correlatos

Autor(es)	Título	Relação com o Trabalho
Paolanti <i>et al.</i> (2018)	<i>Machine Learning approach for Predictive Maintenance in Industry 4.0</i>	Algoritmo de <i>Machine Learning</i> para previsão de falhas.
Mathew, Luo e Pang (2017)	<i>Regression kernel for prognostics with support vector machines</i>	Comparação entre métodos de regressão linear para estimar a RUL.
Yan <i>et al.</i> (2017)	<i>Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance</i>	Método de IA para estimar a RUL.
Spendla <i>et al.</i> (2017)	<i>Concept of predictive maintenance of production systems in accordance with industry 4.0</i>	Método de IA para predição de falhas.
Kovalev <i>et al.</i> (2018)	<i>Data Mining Methods and Techniques for Fault Detection and Predictive Maintenance in Housing and Utility Infrastructure</i>	Aplicações de diversos métodos matemáticos e de <i>Machine Learning</i> para estimar a RUL e predizer falhas.
Kinghorst <i>et al.</i> (2017)	<i>Hidden Markov model-based predictive maintenance in semiconductor manufacturing: A genetic algorithm approach</i>	Método de <i>Machine Learning</i> e tratamento de dados para predição da RUL.
Wei (2012)	<i>Prediction of the aircraft fault maintenance sorties based on least squares of linear regression</i>	Método de regressão linear para predizer o número de manutenções.
Susto <i>et al.</i> (2013)	<i>A predictive maintenance system for integral type faults based on support vector machines: An application to ion implantation</i>	Método de regressão linear para predição de falhas.
Susto <i>et al.</i> (2014)	<i>An adaptive machine learning decision system for flexible predictive maintenance</i>	Modelo de regressão linear com tratamento de dados para previsão de falhas.
Susto <i>et al.</i> (2012)	<i>A predictive maintenance system based on regularization methods for ion-implantation</i>	Método de regressão regularizada para predição de falhas.
Cachada <i>et al.</i> (2018)	<i>Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture</i>	Utilização de um método de IA para predição, detecção e suporte à decisão de falhas.
Mathew <i>et al.</i> (2017)	<i>Prediction of Remaining Useful Lifetime (RUL) of Turbofan Engine using Machine Learning</i>	Predição da RUL comparando-se diferentes métodos de <i>Machine Learning</i> .
Xayyasith, Promwungkwa e Ngamsanroaj (2018)	<i>Application of Machine Learning for Predictive Maintenance Cooling System in Nam Ngum-1 Hydropower Plant</i>	Comparação entre diversos classificadores de <i>Machine Learning</i> para previsão de falhas.
Susto <i>et al.</i> (2015)	<i>Machine Learning for Predictive Maintenance: A Multiple Classifier Approach</i>	Comparação entre métodos de <i>Machine Learning</i> , simples e de múltiplos classificadores, para predição de falhas.
Butte, R e Patil (2018)	<i>Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks</i>	Comparação entre métodos de <i>Machine Learning</i> para predição robusta da RUL.

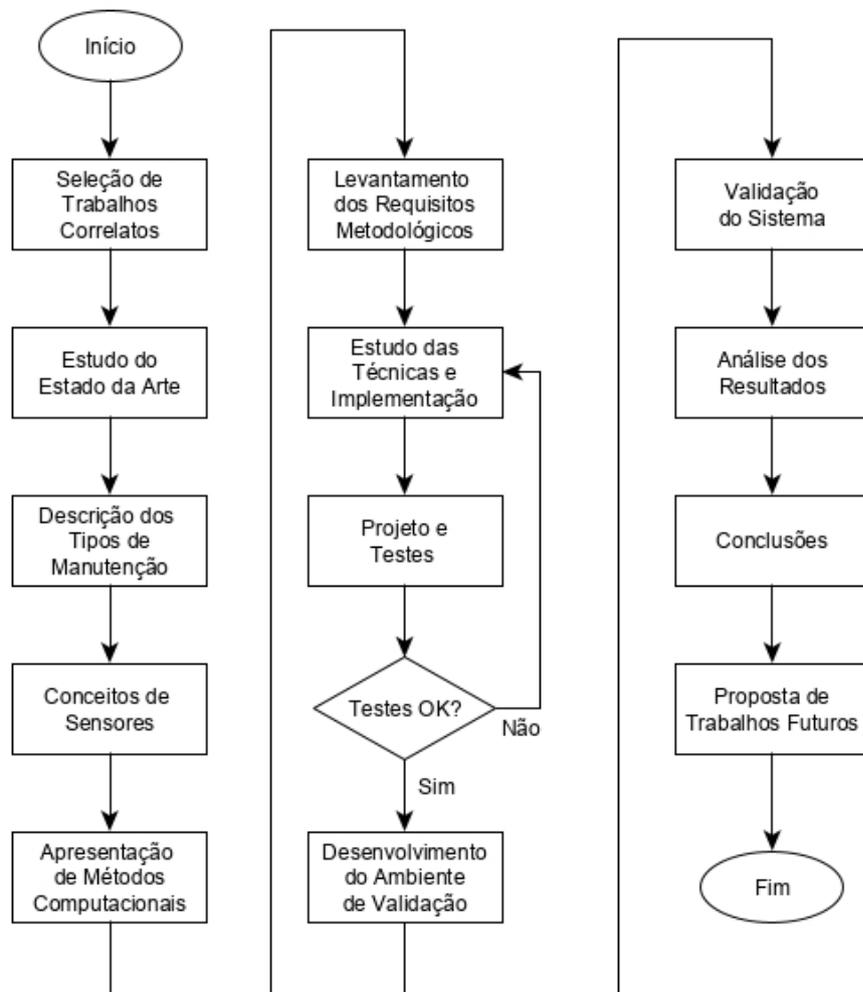
A proposta deste trabalho será elaborada de forma a garantir um ótimo desempenho dos métodos de *Machine Learning*, com foco nos que mais se destacaram dentre os autores citados: ANN, SVM e *Random Forest*. Árvores de decisão mostraram ter uma ótima precisão, uma velocidade de predição razoável e um baixo tempo de treinamento, indicando um excelente desempenho no treinamento (XAYYASITH; PROMWUNGKWA; NGAMSANROAJ, 2018). Também, é importante que o sistema proposto seja capaz de tratar seus dados de forma ótima, seja com algoritmo genético (KINGHORST *et al.*, 2017) ou com algum método estatístico (SUSTO *et al.*, 2014), podendo ser apto a lidar com *Big Data* (YAN *et al.*, 2017) e (SPENDLA *et al.*, 2017), caso a quantidade de dados utilizada represente um cenário próprio para emprego dos conceitos. Assim como devidamente evidenciado em Susto *et al.* (2012), Susto *et al.* (2013), Susto *et al.* (2014) e Susto *et al.* (2015), o sistema de predição será considerado um sucesso quando o mesmo apresentar melhor desempenho frente aos demais sistemas de manutenção (corretiva e preventiva), buscando um melhor aproveitamento das máquinas (ou ferramentas), sem que ocorra uma parada indesejada. É evidente, considerando o conteúdo exposto neste capítulo, que os sistemas de manutenção preditiva estão em ascensão, auxiliando na tomada de decisões e identificação precoce de problemas na produção.

No próximo capítulo, será trabalhada a metodologia proposta neste trabalho, abordando o método de predição empregado, assim como os dados utilizados para teste e um sistema próprio para validação do modelo. É importante ressaltar que o sistema de predição deve respeitar as limitações de desempenho necessárias em um sistema embarcado. Para tal, os métodos propostos nos artigos apresentados nesta seção serão estudados e seus resultados avaliados, de forma a obter-se a melhor escolha para o sistema de predição que será desenvolvido ao longo deste trabalho.

3 METODOLOGIA

Com base no referencial visto até então, neste capítulo são especificados os métodos abordados por este trabalho. Para tal, as ideias foram organizadas desde os conceitos até o projeto desenvolvido. Todas as etapas estão descritas no fluxograma da Figura 15 .

Figura 15 – Fluxograma do Projeto



Fonte – Autor

Como se pode ver na Figura 15, o trabalho teve seu início baseado em uma pesquisa sobre os diferentes trabalhos já existentes, com o objetivo de verificar as diferentes abordagens e aplicações utilizadas para a execução de uma manutenção preditiva. Desta forma, o Estado da Arte deste trabalho foi estudado e desenvolvido, referenciando-se aos trabalhos correlatos, com ênfase para os diferentes métodos e resultados obtidos pelos autores. Com o Estado da Arte

bem definido, alguns conceitos básicos foram descritos, a fim de identificar os conhecimentos necessários para a compreensão deste trabalho e dos correlacionados a ele.

Os conceitos apresentados foram divididos em três tópicos: manutenção, sensores e métodos computacionais. Sobre a manutenção, abordou-se os três principais tipos de gestão utilizadas para as falhas em processos industriais, sendo elas: manutenção corretiva (reparo da máquina quando a mesma apresenta defeito), preventiva (manutenção periódica) e preditiva (aplicada quando há necessidade, com base na possibilidade de falha). Quanto aos sensores, foi explicado brevemente os seus princípios e o seu funcionamento básico, que consiste na medição de grandezas físicas. E, por fim, os conceitos referentes aos métodos computacionais foram apresentados, como ferramentas e métodos de mineração de dados, *Big Data* e técnicas avançadas de *Machine Learning* e Inteligência Artificial. Os requisitos levantados referem-se ao critérios que são esperados pelo projeto. O sistema de predição foi totalmente desenvolvido de forma a atender os requisitos mostrados na Tabela 4.

Tabela 4 – Requisitos do Projeto

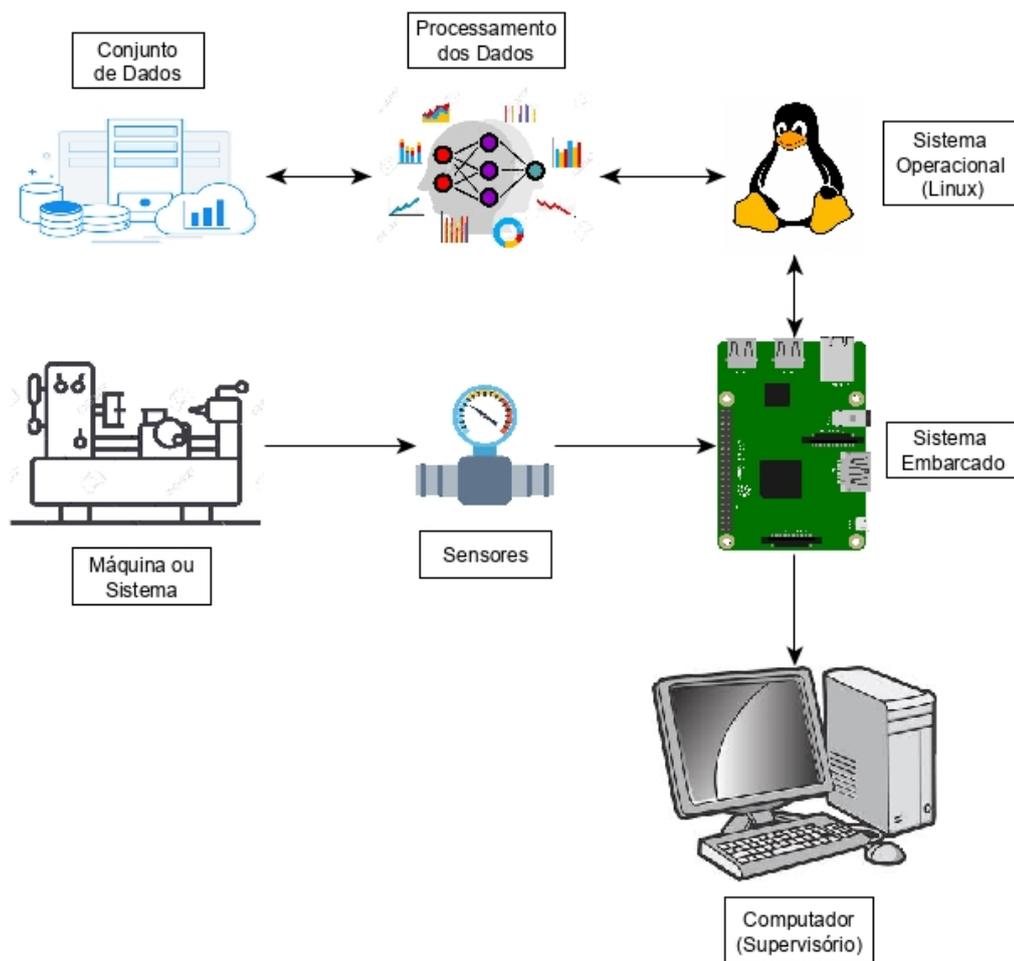
ID	Descrição
R.1	Utilizar um sistema embarcado como meio de aquisição e processamento dos dados
R.2	O sistema embarcado deve ser baseado em Linux
R.3	Os algoritmos serão desenvolvidos preferencialmente em Python
R.4	As variáveis medidas serão analógicas, isto é, contínuas no tempo
R.5	O sistema irá considerar parâmetros de operação normal do equipamento em análise, a fim de identificar as falhas
R.6	É desejável que o sistema seja capaz de aprender sobre o equipamento em análise em tempo real, segundo os dados adquiridos ou um conjunto de dados já existente
R.7	O sistema embarcado deve ser capaz de realizar as suas tarefas sem problemas de desempenho
R.8	O sistema de predição deve identificar falhas em tempo real
R.9	O sistema de predição deve ser capaz de informar quando uma falha deve ocorrer, permitindo ao operador tomar as medidas cabíveis com antecedência (princípio da predição)
R.10	O sistema de predição deve utilizar de métodos <i>Open-Source</i> (do inglês, Código Aberto, corresponde a distribuições para livre utilização, sem nenhuma remuneração envolvida), para tornar o seu custo baixo.

Fonte – Autor

O sistema é composto basicamente de duas *Threads*: uma responsável pela aquisição de dados e outra para o processamento dos dados e treinamento dos algoritmos (R.1), facilitando o uso do processador, de forma a não incorrer em alguma falha por desempenho (R.7). A base de codificação foi formulada em Python (R.3), devido ao fato de se tratar de uma linguagem *Open-Source* (R.10) com suporte a diversos pacotes capazes de aplicar as diferentes técnicas utilizadas, como SVM, Redes Neurais, entre outros. O código foi executado em uma Raspberry Pi 3 Model B[®], rodando uma distribuição Linux (R.2), com base em leituras de sinais analógicos (R.4)

advindas do sistema de validação. O código de aquisição conta com uma variável referente ao estado de operação da máquina simulada, indicando quando a mesma está em um estado de erro ou não (R.5), permitindo a aprendizagem sobre o funcionamento da máquina (ou sistema) que está sendo avaliada (R.6). A aquisição se deu utilizando um protocolo de telemetria assíncrono baseado em publicação/inscrição: o MQTT (*Message Queuing Telemetry Transport*). Tendo em vista o uso deste protocolo, um sistema supervisorio foi desenvolvido na plataforma Node-RED[®], mostrando as variáveis recebidas pelo sistema preditor, bem como a possibilidade de ocorrência de uma falha (R.8 e R.9). O sistema pode ser generalizado conforme a Figura 16, onde é possível ver todos componentes descritos anteriormente e como se relacionam.

Figura 16 – Diagrama do Sistema Proposto

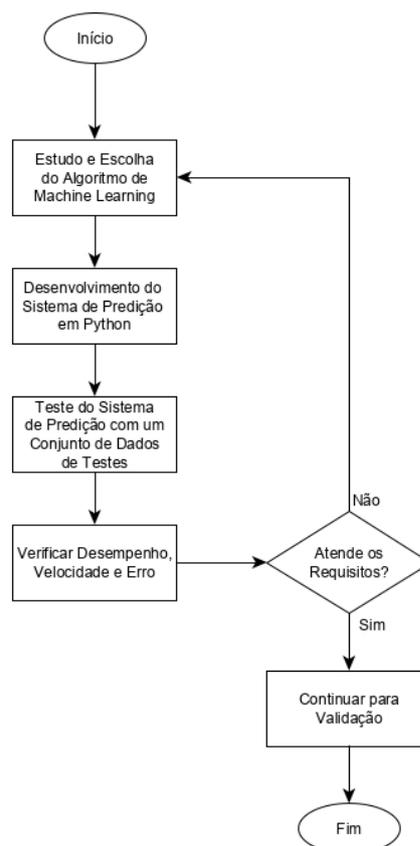


Fonte – Autor

O Conjunto de Dados representa o armazenamento dos dados adquiridos pelo sistema, sendo salvo na memória não-volátil do embarcado. O Processamento dos Dados é a aquisição em si, bem como a mineração e treinamento dos dados para a técnica de *Machine Learning*

empregada. Os dois componentes citados estão codificados e rodam sob o Linux (Sistema Operacional), o qual é a interface entre o nível de usuário e o *hardware* (Sistema Embarcado). Idealmente, a Máquina envia seus dados por meio de Sensores para o embarcado. No entanto, para o sistema de validação que será proposto neste trabalho, o conjunto Máquina e Sensores foi substituído por um Controlador Lógico-Programável (CLP), cujo funcionamento é explicado mais adiante neste capítulo. Por fim, as respostas são dispostas em um sistema supervisor (Computador), o qual permite ao usuário final verificar os dados recebidos do embarcado. Devido à natureza do sistema e por muitas das ferramentas para operar com *Big Data* serem pagas, o que não condiz com o princípio *Open-Source* deste trabalho, este tópico não entrou no escopo final do projeto, embora seja de grande valia os conceitos que o permeiam, visto que o ambiente fabril pode gerar uma aquisição de dados imensamente superior ao avaliado no presente estudo de caso. Dito isto, elaborou-se o fluxograma de testes, conforme exposto pela Figura 17.

Figura 17 – Fluxograma de Teste



Fonte – Autor

A rotina de testes expressa na Figura 17 tem o único objetivo de avaliar dentre os algoritmos propostos qual o melhor em uma aplicação real de um sistema embarcado. Cinco métodos de *Machine Learning* foram selecionados segundo os trabalhos correlatos, sendo eles

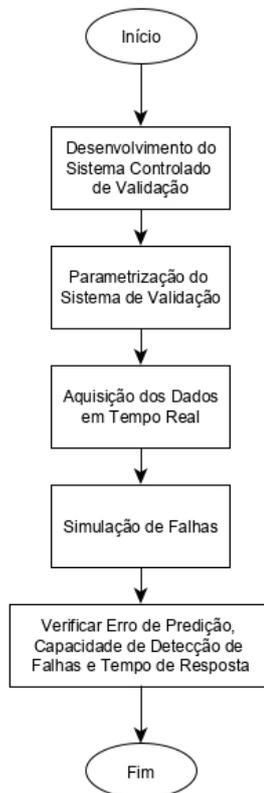
uma Rede Neural [Yan *et al.* (2017); Spendla *et al.* (2017)], SVM [Susto *et al.* (2013); Susto *et al.* (2015); Mathew *et al.* (2017); Xayyasith, Promwungkwa e Ngamsanroj (2018)], *Random Forest* [Paolanti *et al.* (2018); Mathew *et al.* (2017); Butte, R e Patil (2018)] e KNN [Mathew *et al.* (2017); Xayyasith, Promwungkwa e Ngamsanroj (2018); Susto *et al.* (2015)]. Além destes, o quinto algoritmo foi selecionado de acordo com cursos realizados para compreensão das técnicas de *Machine Learning*: a Regressão Logística, que é baseada em métodos de regressão linear, também observados nos trabalhos correlatos. Estas cinco ferramentas matemáticas foram empregadas para realizar a predição de um *dataset* fornecido pela Kaggle[®]: "*Air pressure system failures in Scania trucks*". Este *dataset* consiste em 171 variáveis, onde uma delas representa a classe, negativa ou positiva. Conforme a descrição do problema proposto no Kaggle[®], a classe será positiva quando uma falha em um caminhão da Scania[®] ocorrer devido a problemas no sistema de pressurização de ar do veículo, responsável por muitas funções, como sistema de freios e transmissão, sendo classificada como negativa a falha que não apresenta relação com o sistema de pressurização. Desta forma, essa massa de dados representa o Conjunto de Dados de Testes apresentado no fluxograma da Figura 17. Assim, o sistema de predição para teste foi desenvolvido conforme as seguintes etapas:

1. **Aquisição do dados:** duas *threads* foram codificadas, onde uma realiza a leitura do *dataset* fornecido pelo Kaggle[®] e escreve em um outro *dataset* (simulando a aquisição em tempo real dos dados, realizada a cada 10ms), o qual será utilizado pela segunda *thread*, que verifica se a quantidade ultrapassou uma determinada quantia de dados (10.000 dados para os testes efetuados), realizando o próximo passo assim que o limite é alcançado.
2. **Tratamento dos dados:** nesta etapa, os dados selecionados (10.000 amostras) são pré-processados, onde se verifica a ausência de dados ("NA"), que são substituídos pela média referente ao total de dados adquiridos para a variável específica. Além disto, um método de normalização é empregado, a fim de diminuir a proporção dos valores utilizados, facilitando os cálculos por parte das técnicas de *Machine Learning*.
3. **Treinamento:** os dados processados são separados em dois conjuntos de dados, teste e treinamento (utilizando um parâmetro fixo para a semente aleatória, garantindo sempre o mesmo conjunto de dados para os N testes realizados). Com isso, os cinco algoritmos efetuam o treinamento para o mesmo cenário de dados, para fins de comparação da precisão e do tempo necessário.
4. **Apresentação dos resultados:** para cada método empregado, um relatório contendo os dados de comparação entre a predição e os valores do conjunto de dados de teste é apresentado junto ao tempo que cada um necessitou para efetuar o processamento.

O resultado destas etapas, bem como as diferentes propostas de cenários testadas, estão descritas na Seção 4.1, visto que o objetivo aqui é apenas mostrar a metodologia empregada para

cada etapa do projeto. A seguir, apresenta-se o sistema de validação, como pode ser visto na Figura 18. A validação ocorrerá de uma forma diferente da proposta de utilização do projeto, mas essencialmente com o mesmo comportamento esperado para um uso real.

Figura 18 – Fluxograma de Validação



Fonte – Autor

Como se pode ver pela Figura 18, o sistema de validação será controlado, sendo esta a razão pela qual utilizou-se um CLP para realizar a validação. Desta forma, o controlador simula a aquisição das informações dos sensores por meio das entradas analógicas disponibilizadas pelo controlador. Estes dados são enviados ao sistema embarcado por meio do protocolo MQTT, compondo o conjunto de dados de validação. Falhas são estimuladas após um determinado tempo de execução do sistema, as quais são sinalizadas por um valor binário enviado ao sistema de predição (similar à classe do *dataset* do Kaggle[®]). Por fim, os resultados da predição são publicados também via MQTT, onde é possível avaliar o valor predito. No próximo capítulo, todos os resultados identificados nas etapas de teste e validação são apresentados e detalhados de forma a compreender melhor os conceitos apresentados na metodologia.

4 ANÁLISE DOS RESULTADOS

Os resultados apresentados neste capítulo estarão organizados em três seções: testes, projeto e validação. Na seção sobre os testes, será descrito a linha de desenvolvimento para determinação do algoritmo escolhido para incorporar o sistema de predição proposto. As etapas serão apresentadas conforme os testes elaborados para escolha do método de *Machine Learning*, mencionados pela metodologia: Rede Neural, SVM, *Random Forest*, KNN e Regressão Logística, baseando-se no *dataset* da Kaggle[®] e descrevendo os critérios adotados para eliminação dos algoritmos. A seção de projeto, por outro lado, apresentará o sistema final elaborado para execução do projeto, com todas as partes envolvidas. Encerrando com a seção de validação, a qual mostra os resultados obtidos na execução do estudo de caso.

4.1 Testes

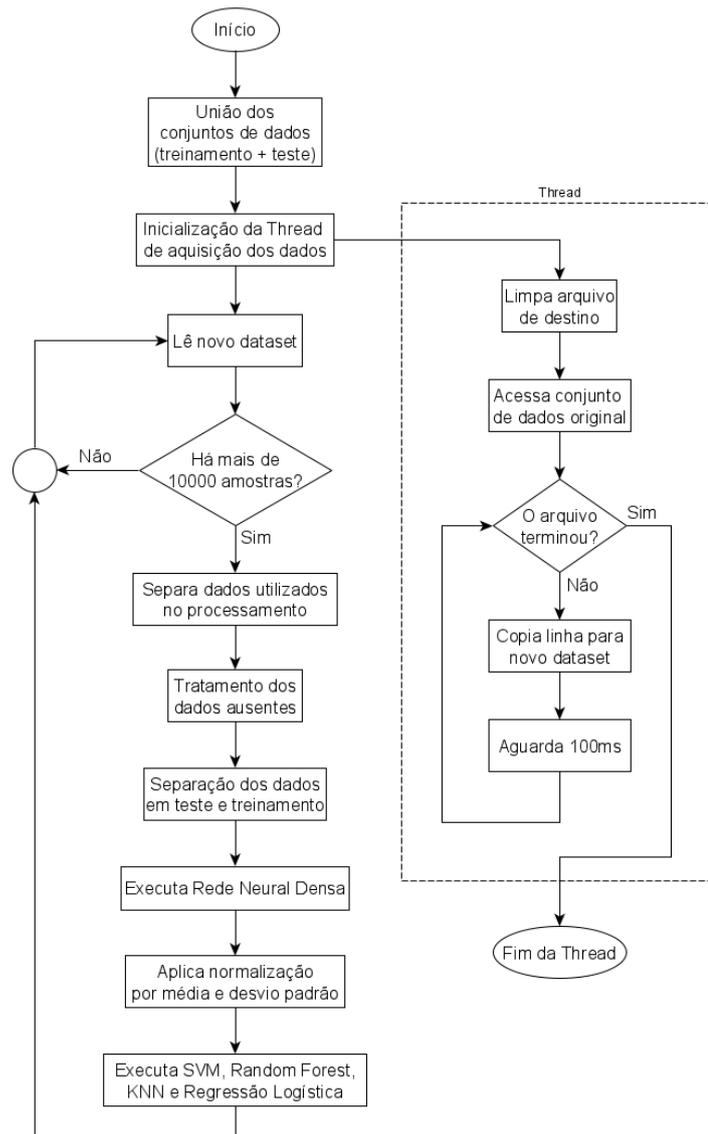
Os testes foram realizados submetendo-se todos os cinco algoritmos a uma mesma base de dados, com uma mesma divisão do *dataset* entre treinamento e teste, sendo que este último é 30% do valor total de dados. O conjunto de dados utilizado (*Air pressure system failures in Scania trucks*) consiste em 170 variáveis que representam características do sistema (*features*) e uma variável de classificação (*label*). Conforme foi visto na metodologia, esta variável de classificação busca identificar se um dado problema que ocorreu em um caminhão da Scania[®] foi por um falha no sistema de pressurização de ar ou não. O *dataset* original encontra-se já dividido em um conjunto de treinamento e outro de teste, contudo, para a realização dos testes que serão descritos nesta seção, os dados de teste foram adicionados ao de treinamento, totalizando 76.000 amostras de cada variável. Vale ressaltar que nem todas as amostras apresentam valores, devendo o sistema ser capaz de tratar a ausência destes dados.

Os testes foram divididos em três cenários principais, com um deles apresentando uma pequena variação em somente um dos algoritmos. Para manter a honestidade no resultado de cada cenário, o código foi ajustado de forma que a divisão entre o conjunto de dados para treinamento e teste fosse sempre igual. O objetivo principal nesta etapa do trabalho é comparar o desempenho de cada algoritmo, além de fazer um paralelo entre um computador executando estas funções e o sistema embarcado proposto.

O fluxograma apresentado na Figura 19 demonstra o comportamento elaborado para o código no primeiro cenário de teste. Note que existem duas ramificações para o código principal, onde há a criação de uma *thread*, representando pela área pontilhada, para simular a aquisição dos dados, visto que os dados são lidos do *dataset* de 76.000 amostras e transferidos para um novo conjunto de dados. A parte responsável pela aquisição de dados não irá se alterar de um cenário para outro, assim como o sistema de separação dos dados utilizados e o tratamento de

dados ausentes. A separação dos dados ocorre pois o sistema de predição irá consumir os dados para realizar os treinamentos dos modelos, desta forma, para evitar que as novas aquisições atrapalhem no treinamentos, considerando 10.000 amostras em um conjunto de dados separado para treinamento. Portanto, ao total são realizados sete treinamentos e testes para cada algoritmo, considerando a entrada de dados pela *thread* de aquisição.

Figura 19 – Fluxograma do Primeiro Cenário

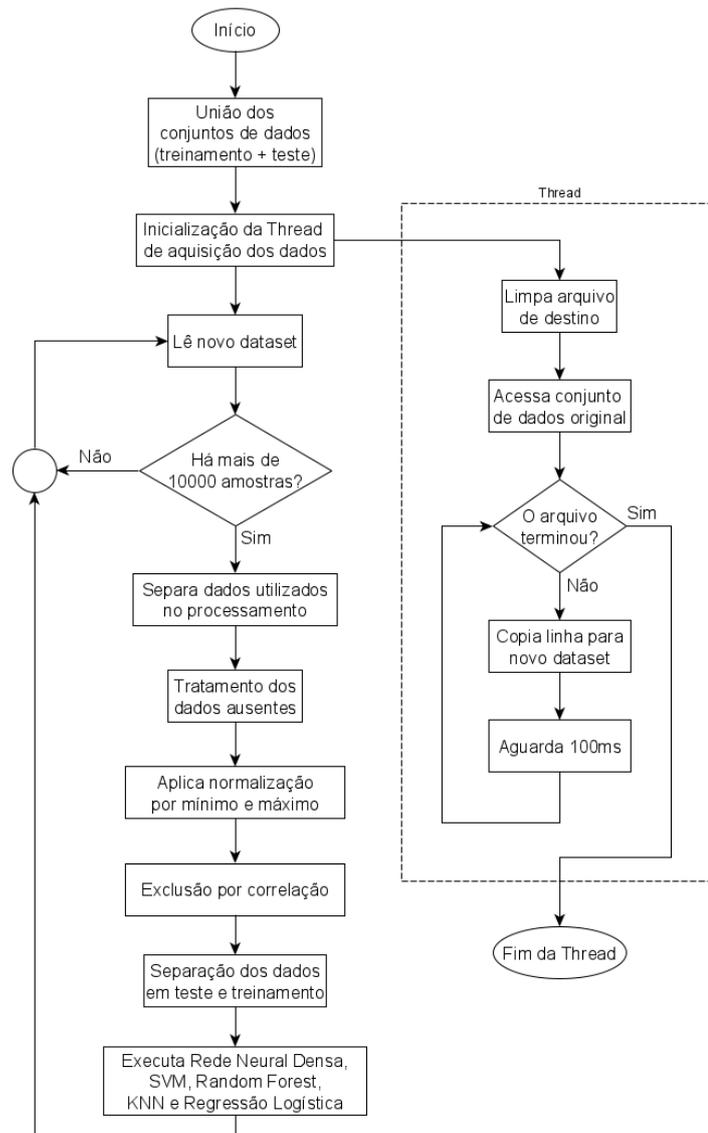


Fonte – Autor

Para o primeiro cenário, a Rede Neural utilizada apresentou duas camadas ocultas com dez neurônios em cada uma, realizando seu processamento para um total de 10.000 épocas. Note que a normalização foi aplicada somente após a execução da Rede Neural. Isto se deve ao fato da Rede Neural não conseguir convergir quando aplicada a normalização, uma vez que esta utiliza

uma equação simples baseada na média e desvio padrão do conjunto, não apresentando uma escala tão amigável para o algoritmo, uma vez que os valores normalizados não apresentam uma escala entre zero e um. Contudo, para execução dos demais métodos, o conjunto de dados foi submetido à normalização, onde o SVM realizou 10.000 iterações; o *Random Forest* utilizou uma profundidade (alcance dos nós) de 1.000 folhas com 100 estimadores; o KNN considerou os cinco vizinhos mais próximos; e a Regressão Logística realizou até 10.000 iterações. Estes valores foram escolhidos conforme os estudos realizados acerca dos algoritmos de *Machine Learning* bem como as recomendações de uso fornecidas pelos próprios distribuidores dos pacotes *Open-Source* utilizados no Python. Algumas pequenas alterações foram realizadas no código, como ilustrado pela Figura 20.

Figura 20 – Fluxograma do Segundo Cenário



No segundo cenário, uma nova implementação foi desenvolvida, com a ocorrência de algumas modificações em outros trechos do código. A novidade deste cenário é um critério desenvolvido para otimização das variáveis que caracterizam o sistema (*features*): exclusão por correlação. Caso um par de *features* seja altamente correlacionado (entende-se como superior a 90%, em módulo), a característica com maior variância permanece no *dataset*, sendo a outra descartada. Uma alta correlação significa, no ponto de vista dos algoritmos de *Machine Learning*, que este par de variáveis apresentam o mesmo efeito no aprendizado do algoritmo, apenas sendo um peso de processamento desnecessário, uma vez que repete as características. Por exemplo, considerando o conjunto de dados da Scania[®] utilizado nesta etapa de testes, a verificação da correlação foi capaz de reduzir as 170 *features* para 125, representando uma redução de mais de 26% do número de variáveis.

A outra modificação realizada para execução da segunda bateria de testes está na modificação do método para normalização, que deixou de operar com a média e desvio padrão para utilizar uma metodologia de mínimos e máximos. Este novo método empregado permite que os valores normalizados estejam presentes numa faixa entre zero e um, permitindo (e facilitando) o emprego do *dataset* normalizado para a Rede Neural Densa. Desta forma, a etapa de execução dos algoritmos foi unificada, realizando o treinamento e teste de todos na mesma etapa do código. Além disto, a estrutura da Rede Neural Densa foi alterada: as duas camadas ocultas foram reduzidas para apenas uma, contudo, empregando duas possibilidades, que serão chamadas de cenários 2A e 2B: a quantidade de neurônios da camada oculta sendo igual ao número de entradas (e.g. 125 neurônios para o *dataset* normalizado) e o dobro deste número (e.g. 250 neurônios). Não alterando o estado dos demais algoritmos.

O último cenário proposto também é representado pelo fluxograma da Figura 20, apenas realizando um processo de otimização (em relação ao tempo de execução) para os algoritmos. Neste cenários, todos os algoritmos tiveram seus parâmetros reduzidos: a Rede Neural Densa permaneceu com a quantidade de neurônios da camada oculta igual ao dobro de número de características (neurônios de entrada), mas sofreu uma redução de 10.000 épocas para 5.000 épocas; o SVM baixou o número de iterações de 10.000 para 5.000; o *Random Forest* teve uma diminuição da sua profundidade de 1.000 para 500, não alterando o número de estimadores (100); o KNN reduziu de 5 para 3 vizinhos mais próximos; e a Regressão Logística teve um decréscimo do número máximo de iteração de 10.000 para 5.000. Os resultados estão dispostos nas Tabelas 5 e 6, onde as sete etapas de treinamento e teste mencionadas estão representadas pela média de tempo e precisão de cada plataforma (computador e sistema embarcado), onde RND é a abreviatura para Rede Neural Densa e RL, para Regressão Logística.

Tabela 5 – Resultados dos Testes - Computador

Algoritmos	Teste 1		Teste 2A		Teste 2B		Teste 3	
	Tempo [s]	Precisão [%]						
RND	763,94608	98,04286	553,44110	99,00000	541,96036	98,97619	319,91484	99,00000
SVM	0,77082	98,28571	0,41141	98,85714	0,42467	98,85714	0,40003	98,85714
RF	2,96933	99,00000	2,27534	99,00000	2,21956	99,00000	2,35588	99,00000
KNN	4,33104	98,71429	2,34916	98,71429	2,22672	98,71429	2,32572	98,85714
RL	0,41897	99,00000	0,06506	98,85714	0,05787	98,85714	0,05858	98,85714

Fonte – Autor

O computador onde foi realizado os testes consiste de um Dell[®] XPS, com 16GB de memória RAM (*Random Access Memory*) e um processador Intel[®] Core[™] i7-4790 @ 3,6GHz, com oito núcleos. Analisando os dados dispostos na Tabela 5, nota-se que a inserção da exclusão por correlação e a normalização por mínimos e máximos (Teste 1 para Teste 2) obteve uma melhoria geral no que tange o tempo de processamento, havendo uma redução em todos os algoritmos. No âmbito da precisão, houve uma melhoria significativa na Rede Neural Densa e SVM, enquanto o *Random Forest* e o KNN permaneceram iguais, sendo a Regressão Logística a única prejudicada neste quesito. Mesmo que o Teste 2A e 2B apresentem diferenças apenas na Rede Neural Densa, o teste foi executado para todos os algoritmos, que apresentaram resultados semelhantes, como esperado. No Teste 3, com a busca da otimização, não houve melhorias significativas nos algoritmos, com exceção da Rede Neural Densa, que teve um novo decréscimo de tempo, mas sem sacrificar a precisão. Estas mesmas verificações foram realizadas no sistema embarcado e estão demonstrados na Tabela 6.

Tabela 6 – Resultados dos Testes - Sistema Embarcado

Algoritmos	Teste 1		Teste 2A		Teste 2B		Teste 3	
	Tempo [s]	Precisão [%]	Tempo [s]	Precisão [%]	Tempo [s]	Precisão [%]	Tempo [s]	Precisão [%]
RND	1.677,21729	98,17619	1.202,88138	98,98095	1.334,63569	98,93333	763,02941	98,99524
SVM	7,00346	98,28571	3,88431	98,85714	3,89497	98,85714	3,88650	98,85714
RF	19,92986	99,00000	15,63011	99,00000	15,92875	99,00000	15,86801	99,00000
KNN	50,55433	98,71429	29,59829	98,71429	29,59952	98,71429	28,70971	98,85714
RL	32,28920	99,00000	3,44306	98,85714	3,50084	98,85714	3,37571	98,85714

Fonte – Autor

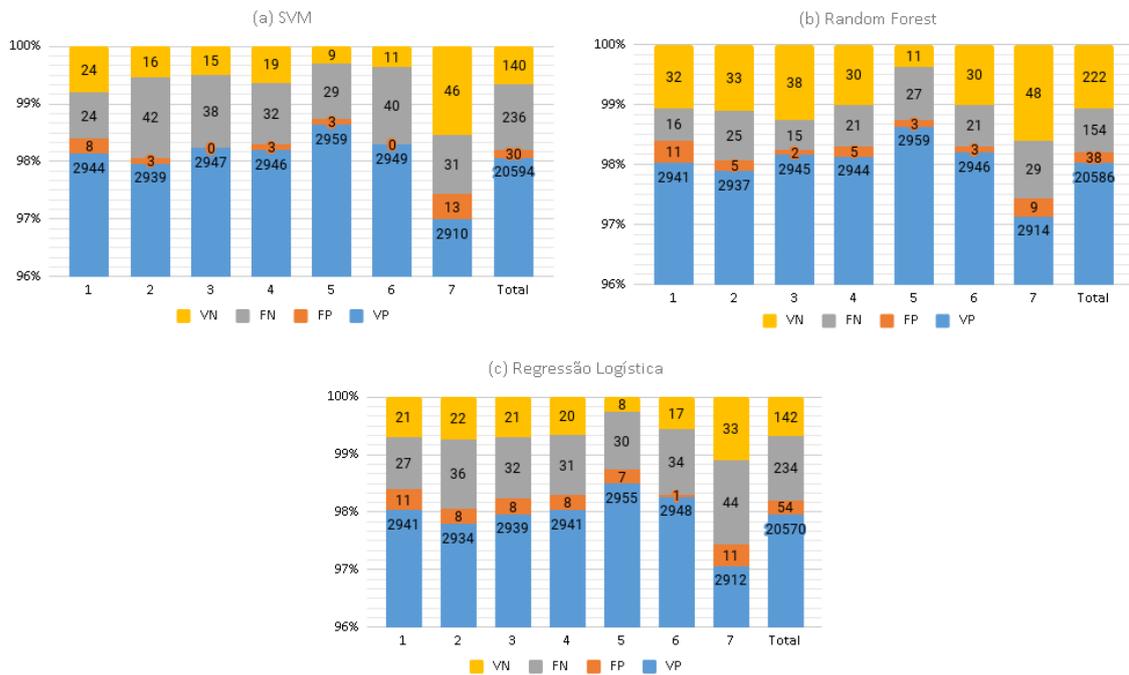
O sistema embarcado, conforme mencionado na metodologia, trata-se de uma Raspberry Pi 3 Model B[®], que possui 1GB de memória RAM e um processador Broadcom BCM2837[™] @ 1,2GHz, com quatro núcleos. Na Tabela 6, como esperado, visto que as especificações do embarcado em relação ao computador utilizados são muito inferiores, os tempos levados pela Raspberry[®] foram muito maiores. Contudo, não houve impacto quanto a precisão dos algoritmos.

Comparando as Tabelas 5 e 6, nota-se que o cenário proposto pelo Teste 1 apresenta as piores condições, ao mesmo tempo que os Testes 2 e 3 são extremamente similares. Também é unânime que a Rede Neural Densa é a que mais ocupa tempo de processamento, mostrando-se

descartável na escolha dos algoritmos. O método com condições mais estáveis foi o *Random Forest*, apresentando uma alta precisão em todos os cenários com um tempo não tão significativo, considerando a quantidade de tempo que leva para o sistema adquirir os dados (nos testes realizados, 10.000 dados com uma taxa de um dado a cada 100ms, resultando em 16 minutos e 40 segundos). O KNN apresentou as menores exatidões em ambas plataformas, enquanto o SVM e a Regressão Logística se destacaram pela alta velocidade de treinamento e teste. Portanto, os algoritmos selecionados até o momento são: SVM, *Random Forest* e Regressão Logística, com as condições propostas no Teste 2, visto seu melhor desempenho sobre o Teste 1 e sua semelhança com o Teste 3, optando-se por manter um valor maior para as configurações.

Com base nisto, considerando os três algoritmos, uma avaliação do desempenho quanto sua precisão foi realizado, onde as matrizes de confusão foram verificadas. Uma matriz de confusão consiste em quatro elementos: Verdadeiro-Positivo (VP), Falso-Positivo (FP), Falso-Negativo (FN) e Verdadeiro-Negativo (VN). Uma predição é dita verdadeira quando a estimativa é correta em relação ao valor de teste (e.g. predição indica 1 e o valor real é 1, VP). Por outro lado, este valor é dito falso quando o valor predito não coincide com o real (e.g. predição indica 0 e o valor real é 1, FN). As matrizes consideradas foram adquiridas junto às diferentes propostas de testes, contudo, para o critério de seleção, apenas foram considerados os resultados obtidos para o Teste 2 para os três métodos selecionados. Os gráficos expostos pelas Figura 21 apresentam os dados da matriz de confusão para cada algoritmo, onde cada uma das sete execuções de cada algoritmo são mostrados (resultado a cada 10.000 dados treinados e testados), com o valor acumulado representado pelo Total. Os percentuais são mostrados acima dos 96% do total avaliado (para cada execução, 3.000 dados, e no Total, 21.000, correspondentes ao percentual de dados para teste de 30%), pois boa parte dos dados apresentados nos testes foram Verdadeiros-Positivos, restando as demais possibilidades um percentual menor.

Figura 21 – Gráficos das Matrizes de Confusão - Comparação entre SVM, *Random Forest* e Regressão Logística



Fonte – Autor

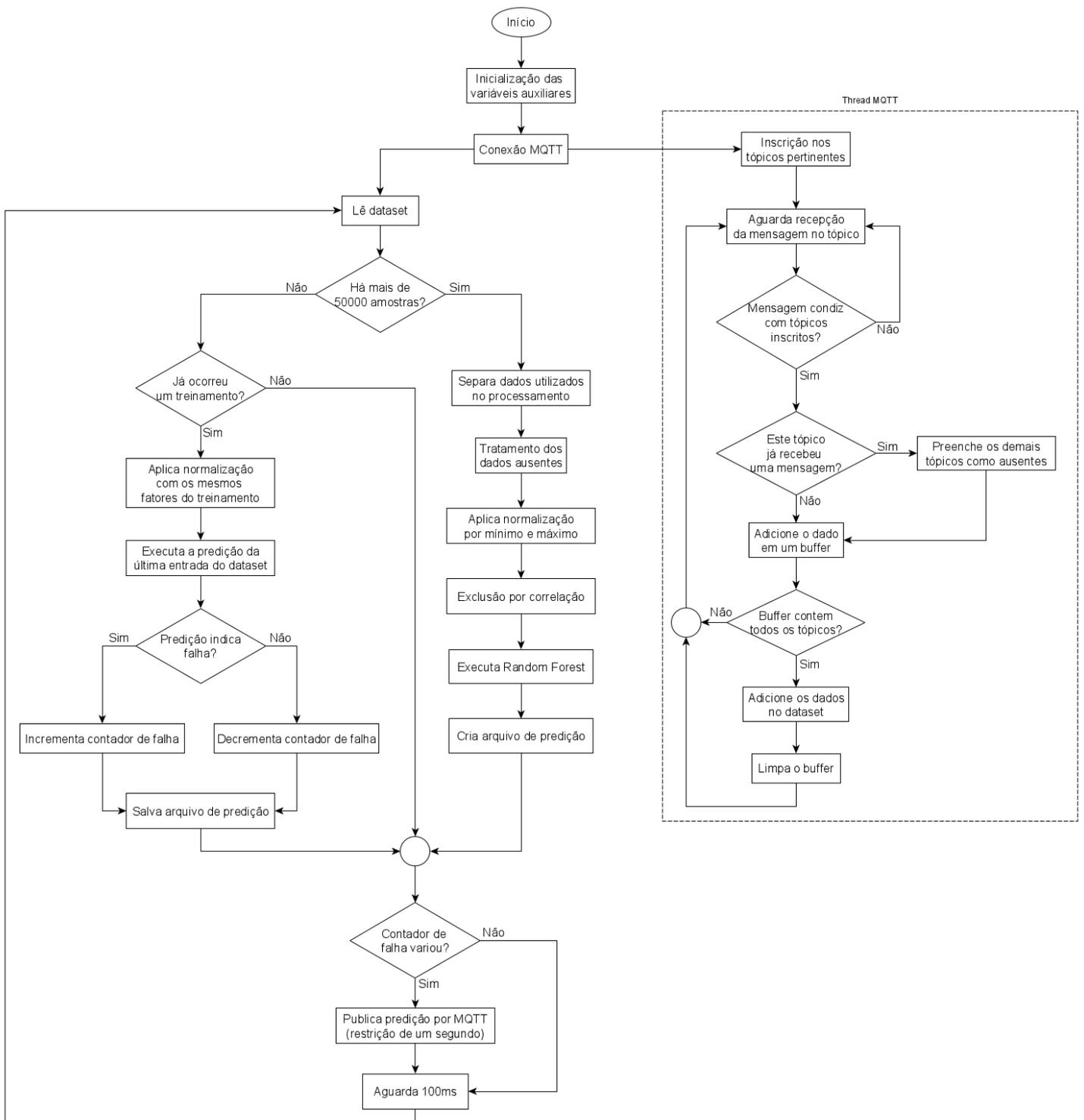
Conforme mostra a Figura 21 e tendo como base que os dados relevantes na predição de qualquer variável são os verdadeiros, constata-se que a Regressão Logística, Figura 21(c), apresenta a pior precisão para os valores preditos, com VP de 97,95238% e VN de 0,67619%, totalizando 98,62857% de acertos. O SVM, Figura 21(a), foi melhor em encontrar os valores verdadeiros quando positivos (VP de 98,06667% e VN de 0,66667%), mas sendo semelhante à Regressão Logística quanto aos negativos, apresentando uma precisão total de 98,73333%. Por outro lado, o *Random Forest*, Figura 21(b), teve desempenho excepcional na detecção de valores negativos, com um resultado semelhante ao SVM para os positivos (VP de 98,02857% e VN de 1,05714%), apresentando uma precisão final de 99,08571%.

Feitas estas análises, o algoritmo selecionado para desempenhar a função de *Machine Learning* no sistema de predição embarcado foi o *Random Forest* com uma profundidade de 1.000 nós e 100 estimadores. Também, considerou-se para o sistema final a exclusão por correlação e a normalização por mínimos e máximos, como proposto no Teste 2, exposto na Figura 20. Assim, finaliza-se as etapas de testes empregadas, servindo como introdução para o sistema de manutenção preditiva. As próximas seções irão evidenciar os resultados obtidos do estudo de caso proposto neste trabalho.

4.2 Projeto

Para a realização do estudo de caso em si, considerando o algoritmo selecionado conforme a seção anterior, algumas alterações foram realizadas no fluxo de operação do código. As modificações empregadas buscam a adaptação do código ao meio de aquisição de dados citado na metodologia deste trabalho, utilizando do MQTT como meio de comunicação entre um CLP e o sistema embarcado responsável pela execução do sistema de predição. Estas alterações estão expressas na Figura 22.

Figura 22 – Fluxograma do Projeto Final



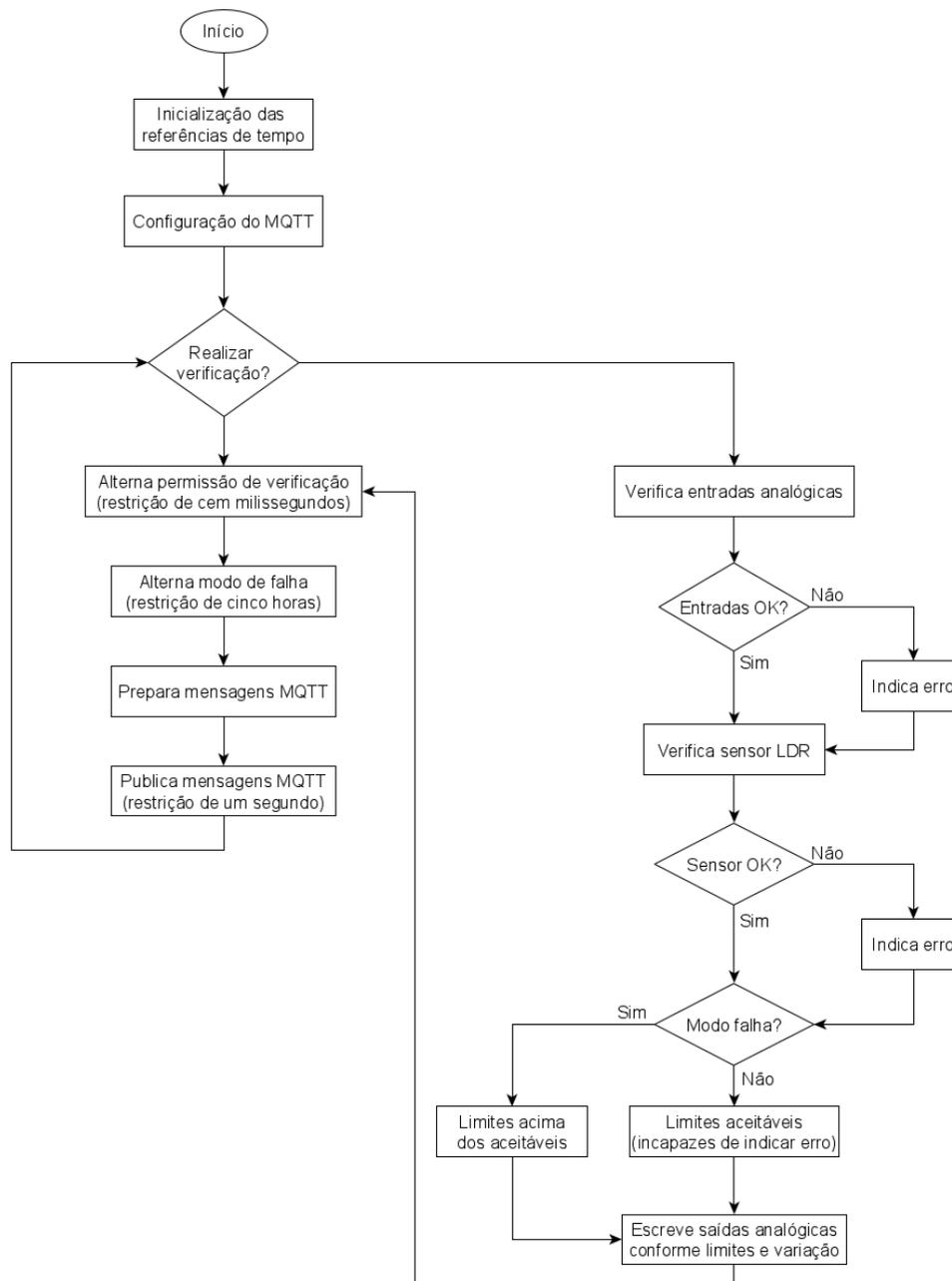
Fonte – Autor

Nota-se que em comparação com os cenários de teste (Figuras 19 e 20), o código proposto para o sistema de predição está muito mais complexo. A primeira observação a ser feita

é quanto ao número de amostras necessárias para a realização do treinamento, que passou de 10.000 para 50.000. A *thread* utilizada para este sistema funciona apenas para verificação da comunicação MQTT, não mais para transmitir dados de um *dataset* para outro. Note também que apenas o algoritmo selecionado está sendo executado (*Random Forest*). Outro ponto é quanto a predição em si, quando um treinamento tiver ocorrido e o número de amostras for inferior aos 50.000: o código irá predizer o valor conforme a última entrada de dados, as características serão normalizadas e um contador de falhas irá variar de acordo com o resultado obtido. Tanto a predição quanto o valor real são salvos em um arquivo, para posterior análise. Após isto, caso haja variação no valor do contador de falhas, o mesmo é publicado via MQTT com uma restrição de tempo de um segundo, visto que várias publicações consecutivas poderiam interferir na comunicação dos demais tópicos MQTT.

Este código foi executado pelo sistema embarcado (Raspberry Pi 3 Model B[®]), enquanto um CLP XP325 da Altus[®] realiza a simulação de uma máquina. O CLP foi escolhido por conter entradas e saídas analógicas integradas, onde as mesmas foram colocadas em *loopback*, isto é, as saídas foram diretamente ligadas as entradas analógicas. Dois destes conjuntos analógicos foram utilizados para simular tensão (0 a 10V), enquanto outros dois conjuntos simularam a corrente (0 a 20mA). Além disto, uma entrada foi configurada para suportar um LDR (*Light Dependent Resistor*), que irá variar seu sinal conforme a intensidade luminosa, com o intuito de ser uma variável constante no sistema. A aplicação desenvolvida para execução da simulação do sistema é ilustrada pela Figura 23.

Figura 23 – Fluxograma da Aplicação de Validação



Fonte – Autor

Como mostra a Figura 23, a aplicação irá configura outro cliente MQTT, que se comunicará com o mesmo *broker* (servidor) do sistema embarcado. Note que há diversas restrições de tempo, que estão ligadas a referência adquirida na inicialização. A primeira restrição diz respeito à verificação das entradas e saídas do controlador, que será realizada a cada cem milissegundos. A etapa de verificação será responsável por verificar os limites impostos para as variáveis e indicar erro caso estes limites não sejam respeitados. Para as variáveis que representam tensão,

o limite é dado entre 8.400 e 16.500 unidades, que correspondem à 2,8V e 5,5V; enquanto a corrente limita-se entre 7.500 e 15.000 unidades (5mA e 10mA, respectivamente). O sensor LDR deve permanecer constante no fundo de escala, indicando 4.200Ω , caso este valor seja inferior a 1.500Ω , a aplicação indicará falha. Vale ressaltar que idealmente estes limites seriam entradas para o sistema de predição, onde o mesmo avaliaria a ocorrência de falhas e indicaria ao operador a presença ou possibilidade do erro. Contudo, para obter maior controle sobre o sistema de validação, as falhas são verificadas no CLP e enviadas ao sistema de predição via MQTT, simplificando a usabilidade do projeto por não dispor de uma tela para configuração dos limites mínimos e máximos aceitos pelos parâmetros.

Os limites citados correspondem aos valores aceitáveis. O modo de falha torna estes limites maiores, extrapolando os valores aceitáveis: tensão entre 6.000 (2V) e 18.000 (6V) unidades; e corrente entre 3.000 (2mA) e 22.500 (15mA) unidades. Este modo é alternado a cada cinco horas, onde as variações das saídas analógicas permitem alcançar estes valores capazes de indicar erro. A variação das saídas ocorre de forma pseudo-aleatória, onde se verifica se o tempo em nanossegundos naquele instante é par (incrementa-se 100 unidades) ou ímpar (decrementa-se 100 unidades). Caso os limites sejam alcançados, o código impede que a variação extrapole os valores impostos de acordo com o modo de falha. Dada a natureza pseudo-aleatória, o sistema é incapaz de saber qual o próximo valor de tensão ou corrente, tornando a validação confiável. Um ponto importante a ser salientado é que a tensão e corrente são relacionadas, cada par (tensão e corrente) foi denominado como um motor (*engine*) A e B com a mesma variação nos seus valores, ou seja, se o valor no instante for par, tanto a tensão quanto a corrente irão ser acrescidas de 100 unidades para o determinado conjunto (motor).

Por fim, um sistema supervisor desenvolvido em Node-RED[®] realiza a última interface proposta pela Figura 16. Esta página foi montada de forma simplificada, onde são apresentados medidores para cada característica utilizada no treinamento, bem como um indicador de falhas e o medidor de contagem de falhas junto a um gráfico de variação destes valores. Estes elementos são mostrados na Figura 24.

Figura 24 – Sistema Supervisório Node-RED®



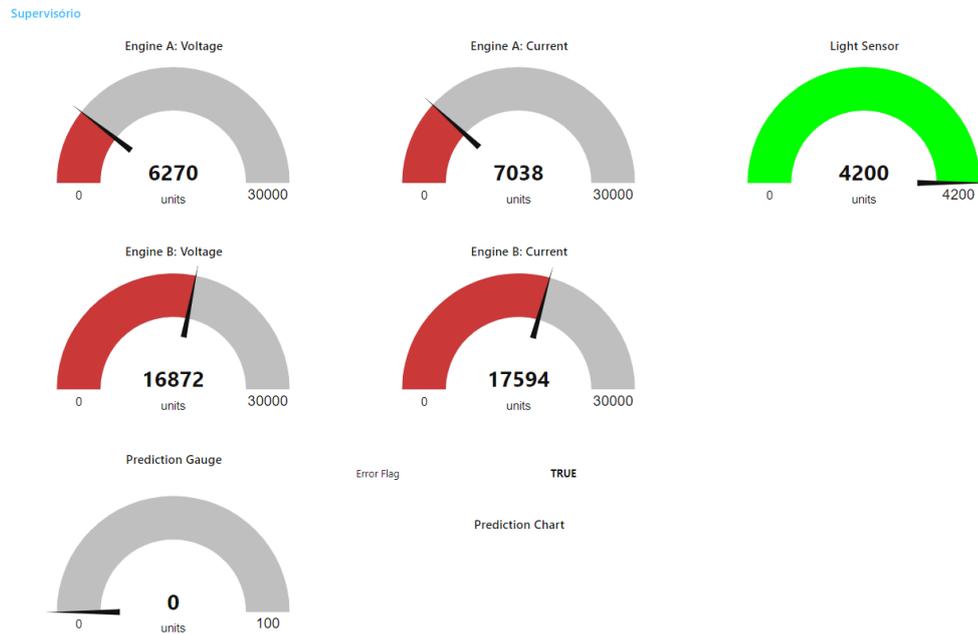
Fonte – Autor

Os dois primeiros medidores da primeira linha (superior) correspondem a tensão e corrente do motor A, sendo o terceiro o sensor LDR. Na segunda linha, tem-se a tensão e corrente do motor B, restando na terceira o medidor de falhas previstas e o indicador de erros. Dito isto, o sistema encontra-se pronto para execução, onde os valores serão gerados e publicados pelo CLP, que será recebidos pela Raspberry Pi 3 Model B®, armazenado e treinado após a aquisição das 50.000 amostras e indicados pelo sistema supervisório do Node-RED®.

4.3 Validação

Desenvolvidas todas as partes propostas pela metodologia, o sistema entrou em execução por mais de quatro dias, realizando a aquisição dos dados e treinamentos. Visto que o número de amostras necessário para cada treinamento é de 50.000 e o CLP realiza uma publicação de todas as variáveis a cada um segundo, estima-se que cada treinamento ocorra a cada 13,8 horas. Sendo assim, para cada nova execução do *Random Forest*, o conjunto de dados irá conter amostras tanto de comportamentos corretos (dentro dos limites aceitáveis), quanto falhas, uma vez que o sistema alterna o modo de falha a cada cinco horas. A Figura 25 mostra como o supervisório indica as falhas para o operador.

Figura 25 – Sistema Supervisório em Falha



Fonte – Autor

O comportamento observado na Figura 25 antecede o primeiro treinamento do *Random Forest*, por isto não há indicação visual do medidor de falhas previstas e o gráfico de predição está em branco. Ao longo da execução do projeto, o sistema foi capaz de indicar as falhas quando a mesma ocorreu (verdadeiro-negativo, VN), como mostrado na Figura 26. Neste caso, o supervisório indica uma *Error Flag* como *TRUE* e o medidor de falhas em seu valor máximo (100% de erros), bem como o gráfico de variação indicando o seu histórico.

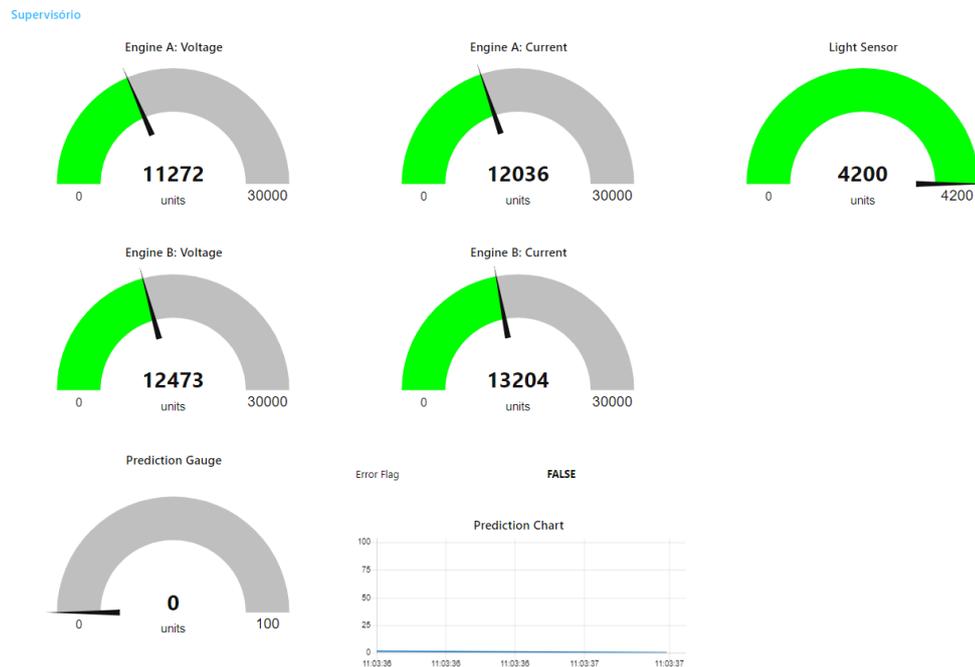
Figura 26 – Sistema Supervisório Predição de Verdadeiro Negativo



Fonte – Autor

Além deste comportamento, que poderia ocorrer dentro do modo de falha, os valores preditos indicaram sucesso quando os valores retornavam para os limites aceitáveis (indicados por verde). Na Figura 27 tem-se um exemplo do momento em que o sistema indicou corretamente que não havia falhas. Note que o *Error Flag* indica *FALSE*, enquanto o medidor de falhas apresenta um valor em zero, junto ao seu gráfico de variação.

Figura 27 – Sistema Supervisório Predição de Verdadeiro Positivo

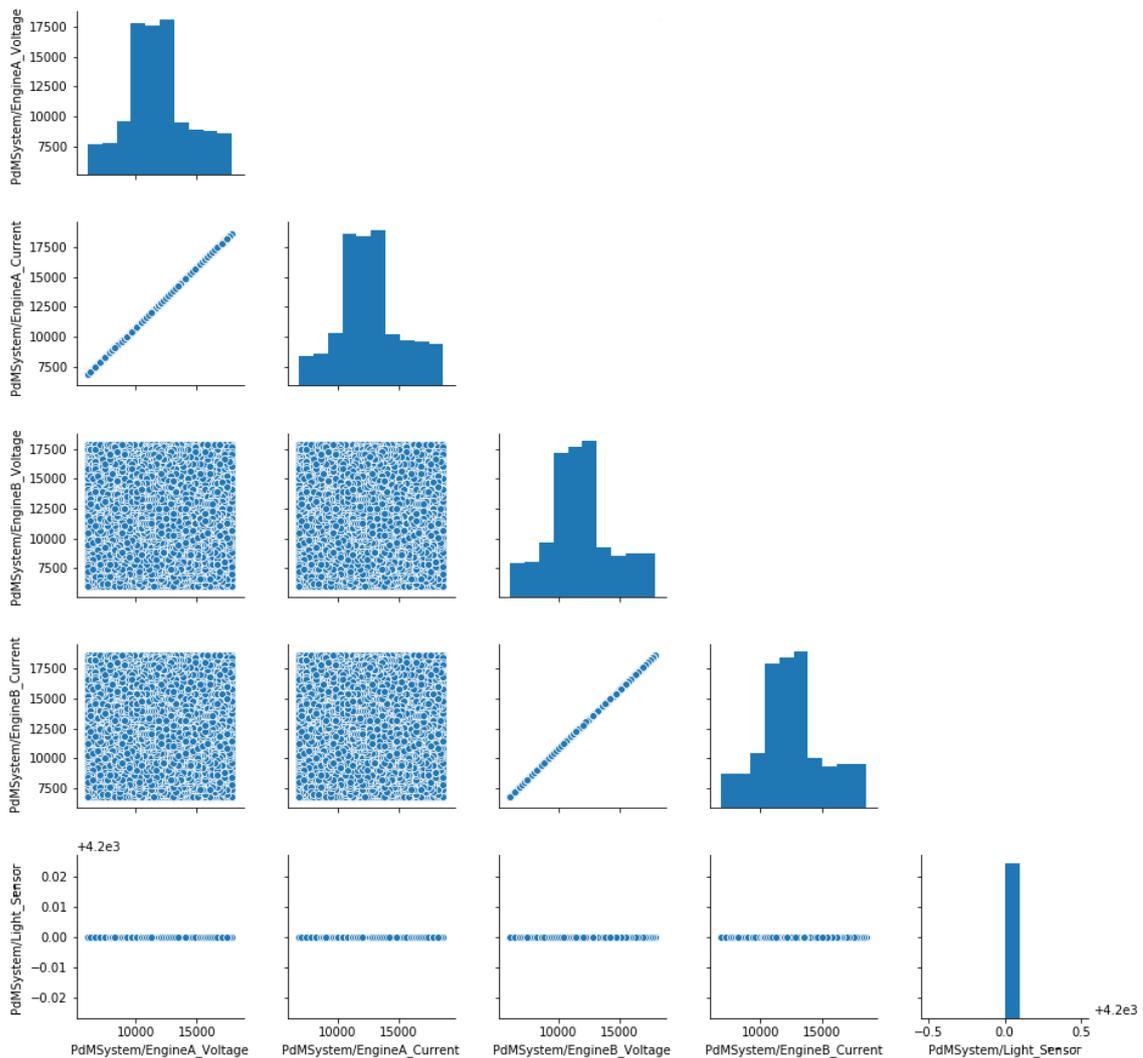


Fonte – Autor

Após a execução por mais de quatro dias, o sistema realizou um total de seis treinamentos, predizendo 286.834 amostras de um total de 340.870. Nota-se que o valor de predições é inferior ao número de amostras, isto é, tomando como exemplo a predição entre o primeiro e o segundo treinamento, das 49.999 amostras (visto que ele não prediz a 50.000^a amostra pelo fato do sistema realizar um novo treinamento) apenas 49.206 foram preditas. A perda de amostras ocorre devido a uma proteção de acesso ao *dataset* implementado para a *thread* MQTT e o código de processamento, onde foi empregado um *mutex*. Este *mutex* bloqueia o acesso ao conjunto de dados enquanto o mesmo é acessado, desta forma, uma outra *thread* aguarda a liberação, o que pode ocasionar uma entrada adicional nesse meio tempo, ignorando a amostra anterior. Contudo, por se tratar de um sistema dinâmico sem variações tão bruscas, isto não é um problema para o sistema proposto.

Analisando a relação entre as variáveis que caracterizam o sistema: tensão, corrente e sensor LDR, tem-se os gráficos mostrados pela Figura 28. Estes gráficos representam a relação, par a par, entre as características utilizadas para o processo de treinamento e predição, considerando todo o *dataset* levantado. Quando uma variável está em par com ela mesma, a sua distribuição é mostrada, enquanto diferentes variáveis indicam a correlação entre elas. A ordem (esquerda para direita e cima para baixo) é dada pela tensão do motor A, corrente do motor A, tensão do motor B, corrente do motor B e o sensor LDR.

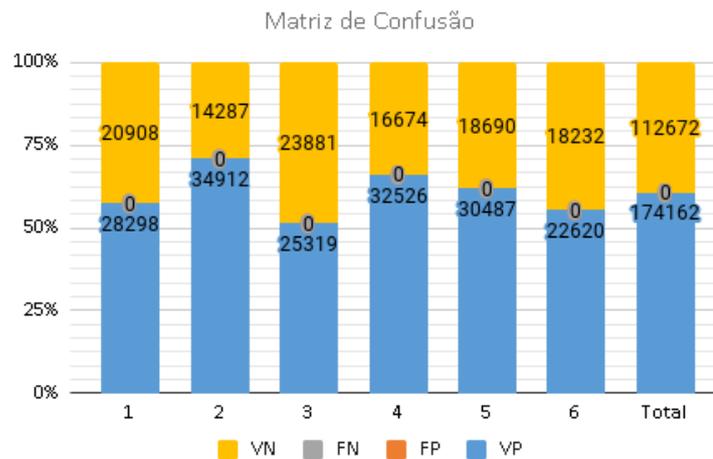
Figura 28 – Relação entre Pares de Características



Fonte – Autor

Verificando a Figura 28, nota-se que as principais características (tensão e corrente), apresentam uma distribuição semelhante à Gaussiana (normal). A relação entre corrente e tensão de um mesmo motor (A ou B) apresenta uma correlação total, indicando que o sistema de predição realizou a eliminação de uma destas variáveis para o treinamento. Por outro lado, caso se verifique a relação entre motores distintos, vê-se uma dispersão total (quadrado). O sensor LDR apresenta um valor contínuo, como esperado, sem correlação com as demais variáveis. Portanto, o sistema de predição utilizou de apenas três características para o seu treinamento. Assim como feito para a seleção do algoritmo, uma matriz de confusão foi elaborada para os resultados obtidos na execução do projeto, mostrado na Figura 29.

Figura 29 – Matriz de Confusão - Validação



Fonte – Autor

A Figura 29 apresenta o resultado final deste trabalho, onde é possível verificar que o sistema de predição apresentou uma precisão de 100%. Isto é justificável pelas distribuições e pelo fato do conjunto de dados conter sempre comportamentos de falhas e normais com abundância, o que não é verificado em sistemas operando no mundo real, onde as falhas ocorrem de forma mais ocasional. Desta forma, o sistema proposto apresentou 60,71874% de valores Verdadeiro-Positivo e 39,28126% de Verdadeiro-Negativo, sem nenhum valor falso, totalizando os 100%.

Enfim, tendo em vista o desempenho adequado, o estudo de caso será encerrado com algumas conclusões sobre todo o desenvolvimento e resultados apresentados. No próximo capítulo, será efetuado o fechamento do trabalho em si, com as devidas considerações tomadas no decorrer do estudo. Além disto, propostas de melhorias para trabalhos futuros serão descritos, a fim de tornar o produto proposto mais robusto.

5 CONCLUSÃO

O estudo de caso apresentou uma proposta diferente das relacionadas ao seu Estado da Arte, onde um sistema de predição não foi utilizado em um computador, mas sim em um sistema embarcado dedicado à aquisição e treinamento dos dados, a fim de realizar a estimativa de falhas. A iniciativa de embarcar um sistema de predição não havia sido verificada em nenhum dos artigos presentes no banco de dados da IEEE (*Institute of Electrical and Electronic Engineers*) até a redação deste documento, caracterizando este trabalho como uma nova visão de aplicação. No que tange a viabilidade da proposta, devido aos avanços da microeletrônica e o aumento de desempenho dos sistemas embarcados, a solução tornou-se totalmente viável, sem nenhum empecilho quanto ao processamento, além de atender todos os requisitos levantados e descritos pela metodologia.

De acordo com os resultados observados, tanto na comparação entre o desempenho do sistema embarcado e um computador tradicional quanto no decorrer da execução do sistema proposto, é possível caracterizar o sistema como um produto mínimo viável, que seria empregado junto a máquinas e/ou sistemas para monitoramento de possíveis falhas. Apesar do desempenho muitas vezes inferior ao da maioria dos computadores, utilizar um embarcado de forma dedicada à aquisição e processamento de dados é totalmente possível, como foi devidamente demonstrado no desenvolvimento deste estudo de caso. Contudo, há algumas propostas de melhorias que podem ser empregadas para que o sistema de predição seja melhor ofertado como um produto:

- Considerar outros algoritmos de *Machine Learning* em outros possíveis cenários: este ponto é importante para determinar qual o melhor algoritmo existente para ser empregado no produto final. O *Random Forest* mostrou-se competente em realizar a função proposta neste trabalho, atingindo o 100% de precisão na validação do estudo de caso. Contudo, para diferentes tipos de máquinas ou sistemas, este pode não apresentar um bom resultado. Portanto, considerar outros cenários com outros algoritmos (e suas variações) pode tornar o sistema de predição mais robusto e capaz de atuar em diferentes situações.
- Utilizar um método probabilístico para prever o próximo valor: a utilização de alguns métodos probabilísticos, como Monte Carlo, pode ser eficaz na tentativa de indicar qual seria o próximo valor que a máquina poderia apresentar, isto é, dado o histórico da máquina, se um medidor de tensão indicar 5V, qual a probabilidade do mesmo ir para 4,9V ou 5,1V. Desta forma, o sistema poderia apresentar uma melhor capacidade de predição, visto que anteciparia o próximo valor.
- Elaborar uma interface para o usuário: como um produto, este deveria conter uma interface para o usuário, onde o mesmo poderia entrar com os limites e condições de funcionamento

da máquina analisada. Com isso, o sistema de predição realizaria a lógica implementada pelo CLP na validação deste estudo de caso, realizando a aquisição direta dos dados e aplicando os limites para indicar a falha (ou possível falha), sempre armazenando os dados em seus conjuntos de dados e apresentando um comparativo entre as predições e as falhas verdadeiras, podendo indicar a qualidade do sistema por meio de um percentual de precisão. Além disto, caso o estudo do primeiro item não indique um único algoritmo, mas um conjunto, estes poderiam ser ofertados a escolha do usuário.

- Utilizar outro sistema embarcado: verificar as diferentes plataformas pode enriquecer as possibilidades de desenvolvimento do sistema de predição embarcado. Fazer uso de dispositivos mais modernos, com maior poder de processamento, pode tornar o sistema proposto incrivelmente poderoso, com altas capacidades, possibilitando, até, a utilização de conceitos não implementados neste trabalho, como *Big Data*.

Os pontos levantados podem ser estudados e desenvolvidos sem problemas em trabalhos futuros, onde seria possível delimitar as possibilidades que os sistemas de predição podem ter como algo dedicado a esta função. Os autores abordados no Estado da Arte não entram em detalhes sobre as máquinas utilizadas, assim, abordar as diferentes metodologias propostas por eles em sistemas embarcados também é uma forma enriquecedora de conhecimento para estes dispositivos que estão cada vez mais presentes em diversas aplicações industriais e domésticas. Como visto na apresentação das diferentes formas de manutenção, a predição está em seus passos iniciais, ainda passando por fase de desenvolvimento, sendo este um excelente momento para experimentar estas diferentes propostas.

É notável que a evolução tecnológica é constante, não é mais necessário esperar muito tempo para que uma nova descoberta ou forma de uso para algum dispositivo apareça. A indústria deve aproveitar esta modernização e adentrar à chamada Indústria 4.0, permitindo um maior controle e segurança para execução de seus processos. Contudo, este passo deve ocorrer de forma gradual, uma vez que muitos processos são resistentes a modernização. O presente trabalho trouxe com sucesso uma proposta de melhoria para as indústrias, fazendo com que a economia da manutenção preditiva seja alcançada com a simplicidade e alto desempenho de um sistema embarcado.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, P. S. d. *Manutenção mecânica industrial : princípios técnicos e operações*. 1. ed. São Paulo: Érica, 2015. 152 p. ISBN 978-85-365-1980-7. Citado 8 vezes nas páginas 14, 15, 16, 17, 18, 19, 20 e 21.

ANDRADE, E. L. d. *Introdução à pesquisa operacional: métodos e modelos para análise de decisões*. 5. ed. Rio de Janeiro: LTC, 2015. ISBN 978-85-216-2966-5. Citado na página 27.

BUTTE, S.; R, P. A.; PATIL, S. Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks. In: *2018 IEEE Workshop on Microelectronics and Electron Devices (WMED)*. IEEE, 2018. p. 1–5. ISBN 978-1-5386-4953-4. Disponível em: <<https://ieeexplore.ieee.org/document/8360836/>>. Citado 7 vezes nas páginas 4, 32, 43, 44, 45, 46 e 52.

CACHADA, A. *et al.* Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture. In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2018. p. 139–146. ISBN 978-1-5386-7108-5. Disponível em: <<https://ieeexplore.ieee.org/document/8502489/>>. Citado 4 vezes nas páginas 31, 34, 45 e 46.

CASTRO, L. N. de; FERRARI, D. *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*. 1. ed. São Paulo: Saraiva, 2016. ISBN 978-85-472-0098-5. Citado 4 vezes nas páginas 24, 25, 26 e 27.

COPPIN, B. *Inteligência artificial*. 1. ed. Rio de Janeiro: LTC, 2017. ISBN 978-85-216-2935-1. Citado 3 vezes nas páginas 28, 29 e 30.

FACELI, K. *et al.* *Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina*. 1. ed. Rio de Janeiro: LTC, 2011. ISBN 978-85-216-1880-5. Citado 3 vezes nas páginas 28, 29 e 30.

FILHO, G. F. *Automação de processos e de sistemas*. 1. ed. São Paulo: Érica, 2014. ISBN 978-85-365-1813-8. Citado 2 vezes nas páginas 22 e 23.

FOGLIATTO, F. S.; DUARTE, J. L. R. *Confiabilidade e manutenção industrial*. Rio de Janeiro: Elsevier, 2009. ISBN 978-85-352-3353-7. Citado na página 24.

GOLDSCHMIDT, R.; BEZERRA, E. *Data mining: conceitos, técnicas, algoritmos, orientações e aplicações*. 2. ed. Rio de Janeiro: Elsevier, 2015. ISBN 978-85-352-7822-4. Citado na página 28.

GROOVER, M. *Automação industrial e sistemas de manufatura*. 3. ed. São Paulo: Pearson Prentice Hall, 2011. 596 p. ISBN 978-85-7605-871-7. Citado na página 22.

JR., R. K. R.; CEGIELSKI, C. G. *Introdução a sistemas de informação*. 5. ed. Rio de Janeiro: Elsevier, 2016. 480 p. ISBN 978-85-352-7740-1. Citado 2 vezes nas páginas 27 e 28.

KINGHORST, J. *et al.* Hidden Markov model-based predictive maintenance in semiconductor manufacturing: A genetic algorithm approach. In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2017. p. 1260–1267. ISBN 978-1-5090-6781-7. Disponível em: <<http://ieeexplore.ieee.org/document/8256274/>>. Citado 6 vezes nas páginas 32, 41, 42, 45, 46 e 47.

KOVALEV, D. *et al.* Data Mining Methods and Techniques for Fault Detection and Predictive Maintenance in Housing and Utility Infrastructure. In: *2018 International Conference on Engineering Technologies and Computer Science (EnT)*. IEEE, 2018. p. 47–52. ISBN 978-1-5386-5589-4. Disponível em: <<https://ieeexplore.ieee.org/document/8420112/>>. Citado 4 vezes nas páginas 32, 44, 45 e 46.

LOBO, R. N. *Gestão de Produção*. 1. ed. São Paulo: Érica, 2010. ISBN 978-85-365-1781-0. Citado 5 vezes nas páginas 15, 16, 18, 19 e 20.

LOPES, I. L.; SANTOS, F. A. O.; PINHEIRO, C. A. M. *Inteligência artificial*. 1. ed. Rio de Janeiro: Elsevier, 2014. ISBN 978-85-352-7808-8. Citado na página 29.

MATHEW, J.; LUO, M.; PANG, C. K. Regression kernel for prognostics with support vector machines. In: *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2017. p. 1–5. ISBN 978-1-5090-6505-9. Disponível em: <<http://ieeexplore.ieee.org/document/8247740/>>. Citado 6 vezes nas páginas 4, 31, 35, 36, 45 e 46.

MATHEW, V. *et al.* Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning. In: *2017 IEEE International Conference on Circuits and Systems (ICCS)*. IEEE, 2017. p. 306–311. ISBN 978-1-5090-6480-9. Disponível em: <<http://ieeexplore.ieee.org/document/8326010/>>. Citado 7 vezes nas páginas 6, 32, 42, 43, 45, 46 e 52.

PAOLANTI, M. *et al.* Machine Learning approach for Predictive Maintenance in Industry 4.0. In: *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2018. p. 1–6. ISBN 978-1-5386-4643-4. Disponível em: <<https://ieeexplore.ieee.org/document/8449150/>>. Citado 5 vezes nas páginas 32, 42, 45, 46 e 52.

SELEME, R. *Manutenção industrial: mantendo a fábrica em funcionamento*. 1. ed. Curitiba: Editora Intersaberes, 2015. 148 p. ISBN 978-85-443-0341-2. Citado 6 vezes nas páginas 15, 16, 17, 18, 19 e 20.

SILVA, L. A. d.; PERES, S. M.; BOSCARIOLI, C. *Introdução à mineração de dados: com aplicações em R*. 1. ed. Rio de Janeiro: Elsevier, 2016. ISBN 978-85-352-8446-1. Citado 3 vezes nas páginas 24, 25 e 26.

SPENDLA, L. *et al.* Concept of predictive maintenance of production systems in accordance with industry 4.0. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*. IEEE, 2017. p. 000405–000410. ISBN 978-1-5090-5655-2. Disponível em: <<http://ieeexplore.ieee.org/document/7880343/>>. Citado 7 vezes nas páginas 31, 33, 34, 45, 46, 47 e 52.

SUSTO, G. A. *et al.* A predictive maintenance system based on regularization methods for ion-implantation. In: *2012 SEMI Advanced Semiconductor Manufacturing Conference*. IEEE, 2012. p. 175–180. ISBN 978-1-4673-0351-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6212884>>. Citado 7 vezes nas páginas 4, 31, 36, 37, 45, 46 e 47.

SUSTO, G. A. *et al.* A predictive maintenance system for integral type faults based on support vector machines: An application to ion implantation. In: *2013 IEEE International*

Conference on Automation Science and Engineering (CASE). IEEE, 2013. p. 195–200. ISBN 978-1-4799-1515-6. Disponível em: <<http://ieeexplore.ieee.org/document/6653952/>>. Citado 8 vezes nas páginas 4, 31, 37, 38, 45, 46, 47 e 52.

SUSTO, G. A. *et al.* Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Transactions on Industrial Informatics*, v. 11, n. 3, p. 812–820, 6 2015. ISSN 1551-3203. Disponível em: <<http://ieeexplore.ieee.org/document/6879441/>>. Citado 8 vezes nas páginas 4, 31, 38, 39, 45, 46, 47 e 52.

SUSTO, G. A. *et al.* An adaptive machine learning decision system for flexible predictive maintenance. In: *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014. p. 806–811. ISBN 978-1-4799-5283-0. Disponível em: <<http://ieeexplore.ieee.org/document/6899418/>>. Citado 7 vezes nas páginas 4, 31, 40, 41, 45, 46 e 47.

THOMAZINI, D.; ALBUQUERQUE, P. U. B. d. *Sensores industriais: fundamentos e aplicações*. 8. ed. São Paulo: Érica, 2011. ISBN 978-85-365-2026-1. Citado 3 vezes nas páginas 21, 22 e 23.

WEI, L. Prediction of the aircraft fault maintenance sorties based on least squares of linear regression. In: *2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization*. IEEE, 2012. p. 223–225. ISBN 978-1-4673-0915-8. Disponível em: <<http://ieeexplore.ieee.org/document/6340849/>>. Citado 4 vezes nas páginas 31, 35, 45 e 46.

XAYYASITH, S.; PROMWUNGKWA, A.; NGAMSANROAJ, K. Application of Machine Learning for Predictive Maintenance Cooling System in Nam Ngum-1 Hydropower Plant. In: *2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE)*. IEEE, 2018. p. 1–5. ISBN 978-1-5386-7159-7. Disponível em: <<https://ieeexplore.ieee.org/document/8612435/>>. Citado 8 vezes nas páginas 6, 31, 39, 40, 45, 46, 47 e 52.

YAN, J. *et al.* Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance. *IEEE Access*, v. 5, p. 23484–23491, 2017. ISSN 2169-3536. Disponível em: <<http://ieeexplore.ieee.org/document/8085101/>>. Citado 8 vezes nas páginas 4, 31, 32, 33, 45, 46, 47 e 52.