



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ADA CRISTINA FRANÇA DA SILVA

**APRENDIZADO DE MÁQUINA APLICADO A PREDIÇÃO
DE ALARME DE TEMPERATURA EM ROLAMENTOS DE
VAGÕES DE TREM**

SÃO LUÍS - MA

2018



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ADA CRISTINA FRANÇA DA SILVA

APRENDIZADO DE MÁQUINA APLICADO A PREDIÇÃO DE
ALARME DE TEMPERATURA EM ROLAMENTOS DE VAGÕES DE
TREM

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Dr. João Dallyson Sousa de Almeida

SÃO LUÍS - MA

2018

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

França da Silva, Ada Cristina.

Aprendizado de máquina aplicado à predição de alarmes de temperatura / Ada Cristina França da Silva. - 2018.
53 f.

Orientador(a): João Dallyson Sousa de Almeida.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, CCET, 2018.

1. Ferrovia. 2. Hotbox. 3. MLP. 4. SVR. 5. Temperatura de Rolamentos. I. Sousa de Almeida, João Dallyson. II. Título.

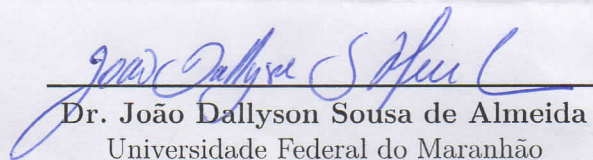
Ada Cristina França da Silva

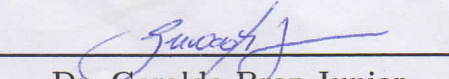
**Aprendizado de Máquina Aplicado a predição de
alarme de temperatura em rolamentos de vagões de
trem**

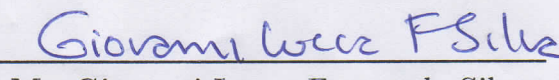
Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Data da Defesa: 11/07/2018
Conceito: **APROVADO**

Banca Examinadora


Dr. João Dallyson Sousa de Almeida
Universidade Federal do Maranhão
Orientador


Dr. Geraldo Braz Junior
Universidade Federal do Maranhão
Membro da Banca


Ms. Giovanni Lucca Franca da Silva
Universidade Federal do Maranhão
Membro da Banca

São Luís - MA
2018

Agradecimentos

Agradeço primeiramente a Deus pela vida e saúde e aos meus pais Aduino e Cristina pelos imensuráveis sacrifícios e esforços feitos por mim e meu irmãos.

A todos os amigos do UFMA, pelo companheirismo nas aulas, viagens e todos os outros momentos juntos, pelos trabalhos e pelo ambiente de amizade cultivado durante esses anos. A todos os professores da Universidade Federal do Maranhão, em especial aos do Departamento Acadêmico de Informática, que contribuem de forma imensa para a formação de todos os alunos que por lá passam. Além disso, agradeço a todos os colegas de trabalho dos estágios que passei, pois vocês tiveram paciência e dedicação em me ensinar.

“A imaginação é a faculdade da descoberta, eminentemente. É o que penetra nos mundos nunca vistos ao nosso redor, o mundo da ciência”

Ada Lovelace

Resumo

O alarme de temperatura de rolamentos é um custo elevado para as ferrovias, pois é necessário a parada do trem. A predição com antecedência dessas temperaturas pode ajudar na manutenção preditiva desse componente. Visando isso, esse trabalho propõe um estudo sobre o comportamento da temperatura de rolamento de trem e a construção de um modelo para prever futuras temperaturas. Utiliza-se de uma série temporal multivariada para compor o banco de dados e aprendizado de máquina. Para a realização da predição, o *Support Vector Regression* e Redes Neurais de Múltiplas Camadas foram escolhidas e seus resultados foram comparados. Todas as etapas são descritas, desde o processamento de dados até a etapa de avaliação dos resultados.

Palavras chave: Temperatura de Rolamentos. Hotbox. Ferrovia. Séries temporais. SVR. MLP.

Abstract

The bearing temperature alarm is a high cost for the railways as it is necessary to stop the train. Prediction of these temperatures in advance can help predictive maintenance of this component. Aiming at this, this work propose to study the behavior of train rolling temperature and proposes a model to predict future temperatures. We use a multivariate time series to compose the database and machine learning. In order to carry out the prediction, the Support Vector Regression and multi-layered neural networks were chosen and their results compared. All steps are described, from the data processing to the evaluation stage of the results.

Keywords: Bearing Temperature. Hotbox. Railway. Time Series. SVR. MLP.

Lista de Ilustrações

Figura 1 – Vagão	21
Figura 2 – Desenho de um rodeiro, eixo, roda e rolamento	22
Figura 3 – Rolamento degolado	23
Figura 4 – Série Temporal Estacionária	24
Figura 5 – Série Temporal Não Estacionária	24
Figura 6 – Separação de duas classes com um hiperplano	27
Figura 7 – (A) Conjunto de dados não linear e (B) Conjunto de dados no espaço de características	28
Figura 8 – Função de Perda SVR	29
Figura 9 – Esquema de um neurônio artificial	30
Figura 10 – Funções de Ativação	32
Figura 11 – Multilayer Perceptron	32
Figura 12 – Método proposto	34
Figura 13 – Banco de dados	35
Figura 14 – Gráfico de Temperatura do Rolamento	36
Figura 15 – Gráfico de características	37
Figura 16 – Densidade de Dados não normalizados e Dados normalizados	40
Figura 17 – Validação Cruzada	42
Figura 18 – Resultado de uma predição	44
Figura 19 – Exemplo de predição SVR	45
Figura 20 – Taxa de Erro	46
Figura 21 – Exemplo de predição Multilayer Perceptron	47
Figura 22 – Gráfico de dispersão	53

Lista de Tabelas

Tabela 1 – Entrada e saída para os modelos	37
Tabela 2 – Resultados K-Fold MLP	43
Tabela 3 – Resultados K-Fold SVR	43
Tabela 4 – Resultados do Grid	45
Tabela 5 – Resultados SVR	45
Tabela 6 – Resultados Rede Neural	46
Tabela 7 – Comparação de resultados	47

Lista de abreviaturas e siglas

SVM	Support Vector Machine
SVR	Support Vector Regression
RNA	Rede Neural Artificial
MLP	Multilayer Perceptron
LSTM	Long Short Term Memory
ACP	Análise de Componentes Principais
RBF	Radial Basis Function
RNR	Redes Neurais Recorrentes
CCO	Centro de Controle Operacional
REQM	Raiz do Erro Quadrado Médio
EAM	Erro Absoluto Médio
UFMA	Universidade Federal do Maranhão
EFC	Estrada de Ferro Carajás
SQL	Structured Query Language

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence
\rightarrow	Implica

Sumário

1	INTRODUÇÃO	15
1.1	Justificativa	16
1.2	Objetivos	17
1.2.1	Objetivo geral	17
1.2.2	Objetivo específico	17
1.3	Organização do Trabalho	17
2	TRABALHOS RELACIONADOS	19
3	FUNDAMENTAÇÃO TEÓRICA	21
3.1	Ferrovias	21
3.1.1	Séries Temporais	23
3.2	Aprendizado de Máquina	24
3.3	Support Vector Machine	26
3.3.1	Support Vector Regression	28
3.4	Redes Neurais Artificiais	30
3.4.1	Função de Ativação	31
3.4.2	Multilayer Perceptron (MLP)	32
4	METODOLOGIA	34
4.1	Base de dados	34
4.2	Pré-Processamento dos dados	37
4.2.1	Limpeza de dados	38
4.2.2	Redução de dados	38
4.2.3	Transformação dos dados	39
4.2.4	Normalização	39
4.3	Predição	40
4.3.1	Ferramentas Utilizadas	40
4.4	Validação dos Resultados	41
4.4.1	Validação Cruzada	41
4.4.2	Métricas	42
5	RESULTADOS	43
5.1	Support Vector Regression	44
5.2	Multilayer Perceptron	45
5.3	Discussão	47
6	CONCLUSÃO	48
6.1	Trabalhos Futuros	49

REFERÊNCIAS	50
APÊNDICES	52
APÊNDICE A – VISUALIZAÇÃO DE DADOS	53

1 INTRODUÇÃO

O transporte ferroviário, em função de suas características, permitiu a diminuição de custos e aumento de segurança no deslocamento de pessoas e cargas. No Brasil, as ferrovias possuem forte impacto na economia, principalmente no transporte de cargas, basicamente matéria-prima como produtos agrícolas e de extração mineral. Entre 2006 e 2014, a produção ferroviária obteve um total de 28,9% de crescimento (CNT, 2015), um cenário evolutivo favorável. Diante do cenário atual, diversos são os fatores que comprometem a eficiência do transporte ferroviário brasileiro, um desses fatores é a falha dos componentes ferroviários, assim, visando evitar defeitos, as empresas realizam a manutenção preventiva dos componentes envolvidos no transporte.

A manutenção no transporte ferroviário é uma atividade estratégica para aumentar a confiabilidade, o desempenho e a segurança neste tipo de transporte. O vagão é o componente mais relevante para a manutenção, pois é responsável pelo armazenamento da carga e compõem em maior quantidade a estrutura do trem. O gerenciamento da manutenção desses vagões é uma atividade complexa, nesse contexto, alguns sistemas tecnológicos foram desenvolvidos para monitorar especificadamente cada vagão e seus componentes, como eixos, rodas e rolamentos. Esses equipamentos de monitoramento são instalados ao decorrer da ferrovia e armazenam informações para serem utilizadas em estratégias de prevenção de falhas. Todos esses equipamentos são denominados de *Wayside* (Do inglês: Equipamentos situados ao lado dos trilho), que é composto por:

- Hot Wheel: Este equipamento tem como função, detectar temperaturas elevadas nas rodas dos vagões.
- Hotbox: Este equipamento tem como função, detectar temperaturas elevadas nos rolamentos das rodas dos vagões.
- Railbam: Este equipamento tem como função detectar deformações e possíveis problemas no rolamento com um sensor de som.

Os dados produzidos por cada equipamento são utilizados por equipes de manutenção para tentar prevenir possíveis acidentes ferroviários, como o descarrilamento, que consiste em um acidente no qual o trem sai de um trilho e pode ocasionar prejuízos pessoais e econômicos. Sendo assim, quando uma roda ou rolamento de um vagão de um trem atinge um nível alto de temperatura, um alarme é acionado e há a necessidade de investigar o problema. Nesse ato de verificar as causas do alarme para evitar um possível acidente pode ocorrer a parada do trem para a manutenção dos seus componentes, e o trajeto ocupado na ferrovia é fechado para outros trens circularem, ocasionando perdas operacionais em cada evento desse tipo. Visando prevenir a parada, as empresas realizam

uma manutenção preventiva, no qual alguns vagões são retirados antes para verificar se há uma possível falha. A falha pode ser causada por diferentes mecanismos mecânicos e/ou elétricos, que na maioria dos casos resultam em uma temperatura crescente a uma certa taxa (PETERS, 2017). Porém, com a grande quantidade de trens, vagões, rodas e rolamentos de um trem, a atividade de análise dos dados de todos os equipamentos se torna complexa. Além disso, frequentemente quando nota-se o crescimento elevado da temperatura, o trem já está circulando e torna-se mais difícil retirar os vagões para a manutenção.

Nesse contexto, este trabalho propõe realizar um estudo sobre os dados obtidos pelos equipamentos do hotbox do Wayside, e assim aplicar técnicas de aprendizado de máquina para prever um possível alarme do aumento de temperatura do rolamento de um trem e evitar a interrupção do tráfego ferroviário. Na etapa de aprendizado de máquina, o processo de regressão será aplicado em uma base de dados com série temporal e variáveis de temperatura, peso, velocidade e localidade. A meta é estimar uma função para relacionar o crescimento da temperatura e verificar a tendência dessa temperatura em futuras leituras do equipamento. Este trabalho aborda desde a formação da base de dados, extração de característica, aprendizado de máquina com o *Support Vector Machine* e Redes Neurais, e por fim, a comparação dos resultados dos modelos escolhidos.

1.1 Justificativa

A redução das despesas operacionais é uma prioridade para a maiorias das empresas e indústrias. Metodologias para evitar o prejuízo com manutenção da falha de componentes são adotadas para evitar desperdício de recursos. Em Ferrovias, o custo de uma parada de um trem gera prejuízos financeiros e uma desorganização no planejamento de outros trens. No cenário de estudo, cada parada de trem ocasiona uma mudança na trajetória de todos os outros e o custo de hora parada de cada trem é em torno de 20 mil dólares. Nesse contexto, existem na literatura alguns estudos e ferramentas desenvolvidos para evitar a ocorrência de paradas.

O alarme de temperatura por rolamento é uma maneira de evitar um acidente e prejuízos maiores, no qual ocorrem reparos e trocas de peças. Porém, ainda assim gera custos pela parada do trem e pela perda do componente. Atualmente, muitas empresas tem adotado a realização de **manutenção preventiva**, que segundo kardec e Nascif (2001) consiste na "Atuação realizada de forma a reduzir ou evitar a falha ou queda no desempenho, obedecendo a um plano previamente elaborado, baseado em intervalos definidos de tempo", nessa caso é esperado que ocorram falhas ocasionalmente. Há também a **manutenção preditiva**, com análise de dados e acompanhamento dos componentes, onde é possível antecipar a verificação dos equipamentos antes que ocorra uma falha, com a verificação precoce de um problema, e assim, aumentar a confiabilidade da trajetória

dos trens e diminuir a parada com manutenções não programadas. O uso de técnicas de inteligência artificial e análise de dados para prever futuros valores é empregado em indústrias para ajudar na manutenção preditiva e tem resultados significativos na diminuição de custos.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo desse trabalho é propor um modelo computacional para prever valores futuros de temperaturas de rolamentos de vagões de trens visando prevê um possível alarme, utilizando técnicas de aprendizado de máquina. Para a realização desse propósito, os algoritmos de Redes Neurais de Múltiplas Camadas e *Support Vector Regression (SVR)* foram selecionados para a criação do modelo.

1.2.2 Objetivo específico

- Formar uma base de dados contendo um histórico de temperatura, velocidade, peso e localidade de medição de cada rolamento de trem.
- Prever as possíveis temperaturas dos rolamentos dos vagões de trem que irão ocorrer durante uma viagem, adotando métodos de regressão.
- Verificar se as variáveis escolhidas para o método de regressão conseguirão prever as temperaturas futuras de uma viagem e assim obter resultados satisfatórios.
- Avaliar os resultados da regressão com os algoritmos utilizados e fazer uma análise comparativa dessas abordagens.

1.3 Organização do Trabalho

O presente trabalho está estruturado em 6 capítulos, que serão desenvolvidos da seguinte forma:

- Capítulo I: Introdução e apresentação dos objetivos.
- Capítulo II: Apresentação de trabalhos relacionados com o tema proposto neste trabalho.
- Capítulo III: Conceitos teóricos para o entendimento da pesquisa realizada, explicação de temas sobre ferrovia, séries temporais, *Support Vector Regression* e Redes Neurais.

- Capítulo IV: Descrição da metodologia utilizada, visualização da base de dados, apresentação das ferramentas escolhidas: Python e suas bibliotecas, preparação dos dados, apresentação das métricas e predição com os algoritmos escolhidos.
- Capítulo V: Análise dos resultados: performance dos algoritmos, comparação dos resultados e visualização gráfica das predições.
- Capítulo VI: Considerações finais e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Este capítulo apresenta trabalhos da literatura que utilizam aprendizado de máquina em séries temporais para prever valores futuros, além de conter alguns trabalhos específicos e metodologias desenvolvidas para facilitar o processo de identificação de alarmes de temperatura em rolamentos de trem.

Tendo em vista os benefícios que a manutenção preditiva de vagões de trem possui, alguns estudos já foram realizados com o objetivo de prever alarmes de temperatura nos componentes do trem, utilizando técnicas de regressão e classificação como o SVM, árvores de decisões e a rede neural recorrente LSTM.

Um estudo realizado em uma ferrovia dos Estados Unidos (LI; PARIKH; HE, 2014) utilizou vários equipamentos de ferrovia, ao todo 1000 detectores instalados em 32 quilômetros, e processou cerca de 1.5 TB de dados correspondente a um período de 6 meses. No desenvolvimento do modelo, o *Support Vector Machine* foi selecionado como técnica de classificação para classificar em duas classes, alarmado ou não alarmado, com a antecedência de 3 e 7 dias. No total, 55 características foram escolhidas, porém, não se utilizou apenas dados de temperatura, mas também dados de outros sensores como o Railbam para a composição das características. Os resultados foram satisfatórios, na previsão de 7 dias antes, a maior taxa de acertos positivos foi 91.54%, e para 3 dias antes uma taxa de 92.56%.

Outro trabalho focado em uma ferrovia da Holanda, utilizou somente os dados de temperatura do equipamento do Hotbox durante dois anos, o que resultou em 60 milhões de medições de 131 trens. A primeira solução foi um sistema neuro fuzzy, porém a ocorrência de falhas foi infrequente e os resultados não foram bons. Uma árvore de decisão foi selecionada como alternativa, a técnica verificou a ocorrência de falhas e classificou em 7 diferentes níveis de alarme de temperatura para a construção de um sistema de monitoramento do estado físico dos rolamentos de trem. O trabalho concluiu que o algoritmo permitiu prever mais da metade das falhas. (PETERS, 2017).

Em (LUO et al., 2017) é apresentado um modelo de regressão utilizando a rede neural recorrente *LSTM*, com históricos de temperatura e outras características para a predição de temperatura de eixos de locomotivas de trem em várias etapas temporais a frente. A base de dados foi coletada de 10 locomotivas o que gerou um total 5.500 amostras, na etapa de predição a biblioteca TensorFlow foi utilizada e os resultados obtidos mostraram que o erros máximos de temperatura foram 1.2 °C, 1.7 °, 1.9 °C para respectivamente 1,6 e 12 etapas a frente.

Há outros trabalhos com diversos objetivos em que históricos de medições foram utilizados para prever valores futuros. Um dos estudos analisados selecionou o *Support Vector Regression* para a predição da temperatura atmosférica(Y.RADHIKA; M.SHASHI,

2009) e usou de diversos parâmetros de clima por um período de 5 anos para a criação do banco de dados. Além do *SVR*, o trabalho implementou uma *Multilayer Perceptron* como alternativa para solucionar o problema, os resultados foram favoráveis e o *SVR* obteve melhores resultados.

Em outro estudo, Roman e Jameel (1996) mostraram uma comparação da utilização de redes neurais recorrentes *Elman* e uma rede neural *MLP* em uma série temporal para predição de valores de diversos mercados de ações. Embora as redes neurais recorrentes sejam mais especializadas nesse tipo de *dataset*, neste trabalho, a *MLP* obteve melhores resultados.

No artigo de Silva (2017), os algoritmos de *SVR* e uma rede neural *MLP* foram comparados na solução do problema de predição de consumo de energia do nordeste. Os autores utilizaram um histórico de consumo anteriores e do valor do PIB como características. Os resultados foram favoráveis para o *Support Vector Regression*.

Através dos trabalhos apresentados, percebe-se que as séries temporais são objetos de estudo em várias áreas de conhecimento e que são de difícil otimização. Na maioria dos trabalhos pesquisados redes neurais recorrentes são escolhidas para esse tipo de série e frequentemente e alcançaram bons resultados na predição, devido a sua característica de relacionar valores novos de entrada com valores passados. O *Support Vector Regression* obteve melhores resultados se comparado as redes neurais simples em séries temporais, principalmente devido a simplicidade do *SVR* e sua característica de generalização. Porém, os resultados dependem bastante do tipo de problema a ser solucionado e da base de dados.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os fundamentos teóricos que permitem entender as técnicas e métodos escolhidos para a solução do problema proposto.

3.1 Ferrovia

A Estrada de Ferro Carajás possui cerca de 892 quilômetros de extensão e os equipamentos utilizados para a obtenção dos dados, cerca de 24 *hotboxes*, estão instalados em pontos estratégicos em toda a trajetória da ferrovia do início até o destino final.

Os trens de estudo são responsáveis pelo transporte de carga de minério de ferro e são considerados os maiores trens do mundo, cada trem possui em média cerca de 3,3 quilômetros de extensão. A parada de um trem dessa proporção gera custos elevados.

Os vagões (vide Figura 1) compõe a estrutura do trem e são responsáveis pelo armazenamento da carga. Em média, um vagão vazio possui 28 toneladas e carregado de minério de ferro, 128 toneladas. A estrada de ferro Carajás possui uma frota de aproximadamente 18.000 vagões, e para o estudo desse trabalho foram selecionados somente dois tipos de vagões, o GDU e o GDT, que são vagões de carga. Cada vagão possui cerca de 4 eixos, e cada eixo tem 2 rodas e dois rolamentos, o qual totaliza 8 rolamentos e 8 rodas para serem monitorados em cada vagão.

Figura 1 – Vagão

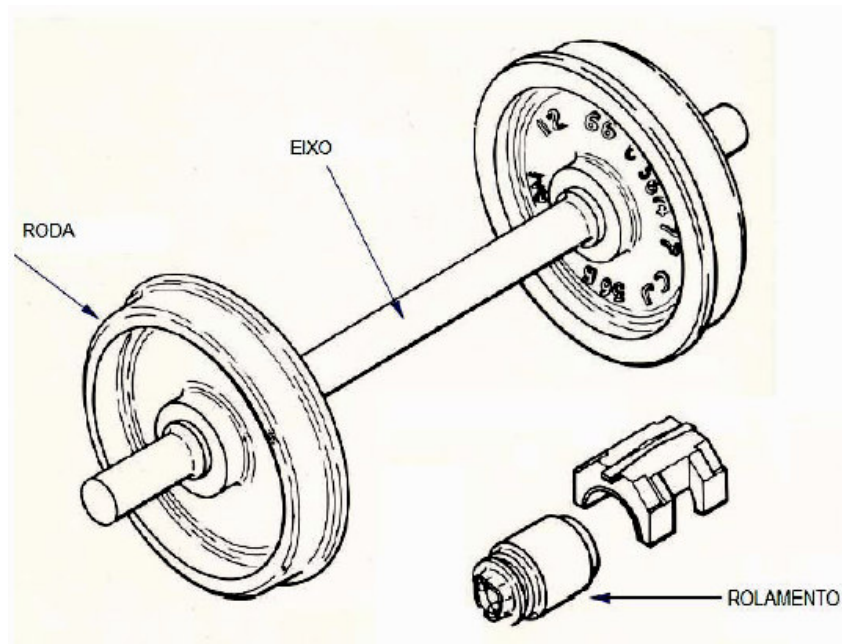


Fonte: Baseado em [EFCarajás \(2018\)](#)

O rolamento é uma das partes que compõe um rodeiro. Um rodeiro ferroviário é composto de duas rodas acopladas, um eixo e um par de rolamentos acoplados na

extremidade do eixo, conforme Figura 2. Esses rolamentos podem sofrer diversas falhas que consequentemente ocasionam um aumento de temperatura, como: imperfeições no material, corrosão, contaminação por óleo lubrificante, desalinhamentos, folgas internas, etc. Logo, com a verificação do histórico de temperatura de cada rolamento pode se diagnosticar que há uma falha nesse componente e que ocasionará em um alarme de temperatura ($\geq 70^\circ$).

Figura 2 – Desenho de um rodeiro, eixo, roda e rolamento



Fonte: [Furtado \(2005\)](#)

Em alguns casos quando ocorre temperaturas elevadas, o prejuízo e a solução do problema pode ser mais complexa, devido a total perda do rolamento e a dificuldade para retirar o componente. Nesse tipo de situação ocorre a degola do rolamento, conforme a Figura 3.

Figura 3 – Rolamento degolado



Fonte: Autor

3.1.1 Séries Temporais

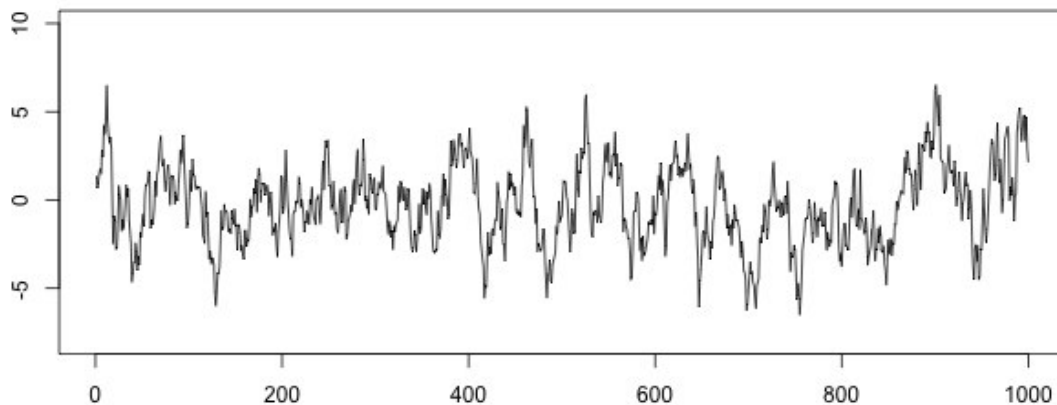
Em várias áreas de conhecimento, os dados são obtidos em instantes sucessivos de tempo, seja por horas ou por medições frequentes. Esse tipo de dados é caracterizado como uma série temporal, que simplificada de acordo com a estatística é uma sequência de observações ao longo do tempo. Em outros tipos de séries, a sequência das características é irrelevante para a análise, porém em séries temporais, os valores antecedentes mantem uma relação com os valores sucessores. Para a criação desse tipo de série, é necessário definir uma janela de tempo de valores anteriores ao que será previsto e outra janela de tempo para os valores que serão previsto no futuro. São exemplos de séries temporais:

- Valores diários de uma temperatura de uma cidade;
- Cotação de uma moeda;
- Índices da Bolsa de valores;
- Vendas mensais de um produto.

Em um série temporal é necessário destacar algumas definições que são importantes para a construção do modelo. Esses tipos séries podem ser classificadas como estacionárias, ou seja, ao longo do tempo, os valores se mantêm ao redor de uma média constante, e sua caracterização é mais simples. Outras apresentam variações que podem ser tendências, onde há uma inclinação positiva ou negativa, e são chamadas de séries não estacionárias.

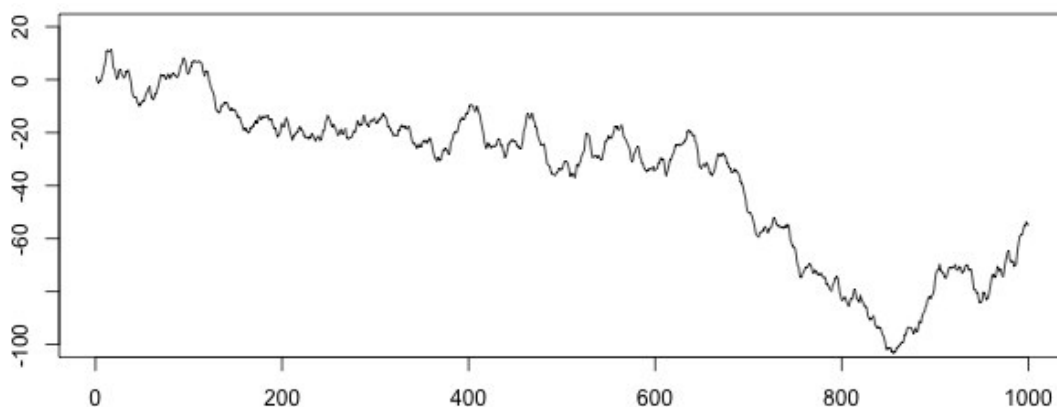
E há também tipos de séries que contém a sazonalidade, no qual determinados valores se repetem em certos períodos. Seguem imagens para comparação:

Figura 4 – Série Temporal Estacionária



Fonte: Baseado em [Jolliffe \(2011\)](#)

Figura 5 – Série Temporal Não Estacionária



Fonte: Baseado em [Jolliffe \(2011\)](#)

No trabalho proposto a base de dados é uma série temporal, onde cada rolamento possui um histórico de medições, e dependendo do seu comportamento existe uma tendência (crescimento ou queda de valores) da temperatura.

O objetivo da análise desse tipo de série geralmente é para descrever o comportamento, investigar o mecanismo gerador da série e fazer a previsão de valores futuros.

3.2 Aprendizado de Máquina

O aprendizado de máquina (AM) é uma área da inteligência artificial que busca o desenvolvimento de modelos utilizando o aprendizado. A partir de hipóteses, funções e

criações de regras utilizando experiências anteriores é possível resolver problemas complexos. Logo, o AM é amplamente utilizada como técnica para a construção de sistemas capazes de aprender de forma automática.

O aprendizado de máquina é usado na resolução de uma grande variedade de problemas do mundo real. Alguns exemplos de aplicações podem ser citados como diagnósticos médicos, classificação de textos, reconhecimento de palavras faladas, predição de valores futuros e condução de automóveis.

Para o desenvolvimento de sistemas de aprendizado, é necessário entender o problema e seus objetivos. Segundo [Carvalho \(2011\)](#), a tarefa de aprendizado de máquina pode ser dividida em:

- **Preditivas:** Nesse tipo de tarefa, a meta é encontrar um modelo a partir dos dados utilizados para treinamento que possa prever um novo valor. Para isso, cada objeto no conjunto de treinamento deve ter valores de entrada e de saída.
- **Descritivas:** Essa tarefa busca descrever um conjunto de dados, sendo assim, as entradas dos algoritmos de aprendizado de máquina não necessitam de atributos de saída.

As técnicas de aprendizado de máquina devem ser capazes de lidar com dados imperfeitos, como a presença de rótulos incorretos. E deve-se também minimizar a influência de dados muito distintos dos demais conjunto de dados do *dataset*, essa capacidade permite definir o poder de generalização do modelo escolhido ([LORENA; CARVALHO, 2007](#)). No caso em que o modelo se adapta a detalhes do dados da base de treinamento e ocorre um sobre-ajuste, apresentando uma baixa taxa de acerto para novos valores, têm-se a ocorrência do *overfitting*. Há também casos em que ocorre o *underfitting*, onde os resultados da predição são ruins tanto no treinamento como nos casos de testes, isso ocorre principalmente quando poucos exemplos representativos foram dados para o modelo.

O aprendizado de máquina pode ser dividido em três categorias, aprendizado supervisionado, não supervisionado e aprendizado por reforço. Nesse trabalho, somente as duas primeiras categorias serão detalhadas.

O aprendizado supervisionado é feito a partir de exemplos, em que os dados de entrada são rotulados com suas respectivas saídas e utilizadas para treinamento. A técnica de algoritmo tenta criar uma função para relacionar os valores de entrada (características) com os seus respectivos valores rotulados, fornecidos por um professor. Problemas nessa abordagem podem ser subdivididos em problemas de classificação ou regressão.

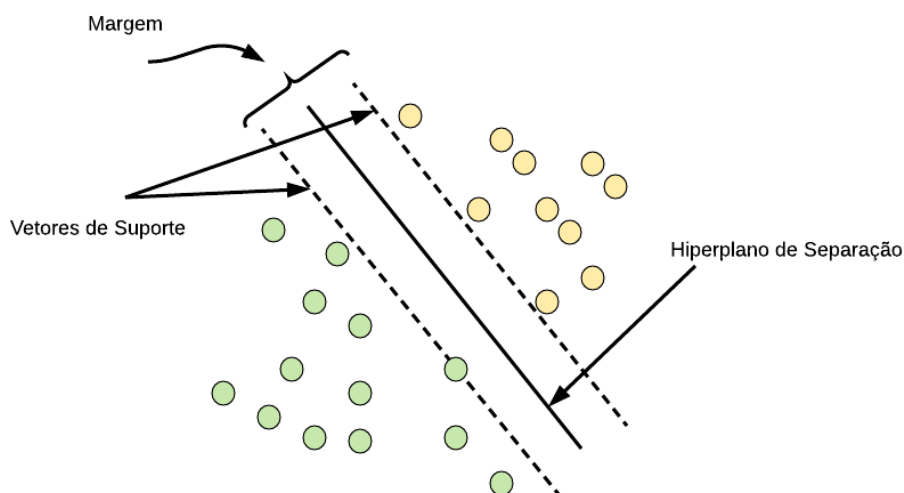
Na classificação, os resultados são rotulados em categorias ou classes e representam valores discretos. Contudo, se o problema resulta em valores contínuos, há um caso de regressão.

Já o aprendizado não supervisionado objetiva descrever o conjunto de dados. O algoritmo não recebe entradas previamente rotuladas para a criação do modelo. A ideia é que o algoritmo seja capaz de extrair padrões sem a ajuda de um tutor. A finalidade desse tipo de aprendizagem pode ser divididas em: agrupamento, no qual os dados são separados em grupos distintos de acordo com suas similaridade; sumarização, que busca encontrar uma descrição para um conjunto de dados.

3.3 Support Vector Machine

O *Support Vector Machine (SVM)* é um algoritmo de aprendizado baseado na técnica da teoria do aprendizado estatístico desenvolvido por [Vapnik \(1998\)](#). No contexto do SVM, a teoria de aprendizado estatístico busca a habilidade de generalização, no qual bons resultados são alcançados quando ocorre um equilíbrio entre a precisão alcançada durante o treinamento e a habilidade do modelo. Essas condições levam em conta o desempenho do classificador no conjunto de treinamento e na complexidade, para obter bons resultados mesmo com novos dados ([LORENA; CARVALHO, 2007](#)). A ideia principal do SVM é construir um hiperplano de separação ótimo entre os exemplos. Um modelo muito simples que é utilizado como problema inicial para a criação de SVMs mais sofisticados lida com problemas linearmente separáveis, onde os classificadores só necessitam de um hiperplano para separar os dados. Por exemplo, um problema de classificação binária, dado por (x_i, y_i) , com $x_i \in \mathbb{R}$ e $y_i \in -1, 1$, $i = 1, \dots, N$, No qual x_i é o vetor de entradas de características e y_i é o valor resultante. O objetivo é encontrar uma função $f: \mathbb{R}^n \rightarrow 1, -1$ na fase de treinamento, que relacione o vetor de caraterísticas com o resultado. Segue exemplo na [Figura 6](#) da separação em duas classes.

Figura 6 – Separação de duas classes com um hiperplano



Fonte: Autor

A função objetivo do *SVM* busca encontrar a distância mínima entre os vetores de suporte e as fronteiras das classes, uma vez encontrados esses vetores, é escolhida a fronteira com maior distância para todas as fronteiras, e assim obtemos o hiperplano. A equação de um hiperplano pode ser definida por meio do produto escalar entre os vetores \mathbf{w} e \mathbf{x} , que pode ser representada pela equação:

$$f(x) = (\mathbf{w} \cdot \mathbf{x}) + b, \quad (3.1)$$

onde, w é um vetor de pesos e o escalar b é o bias. O vetor peso define uma direção perpendicular e com o acréscimo do bias, o hiperplano se move paralelamente a ele mesmo.

A equação divide o espaço dos dados X em duas regiões: $w \cdot x + b > 0$ e $w \cdot x + b < 0$. Usando uma função, podemos separar os valores dessas equações para obter a classificação, conforme a equação abaixo:

$$w \cdot x + b \geq 0, y = 1 \quad (3.2)$$

$$w \cdot x + b < 0, y = -1 \quad (3.3)$$

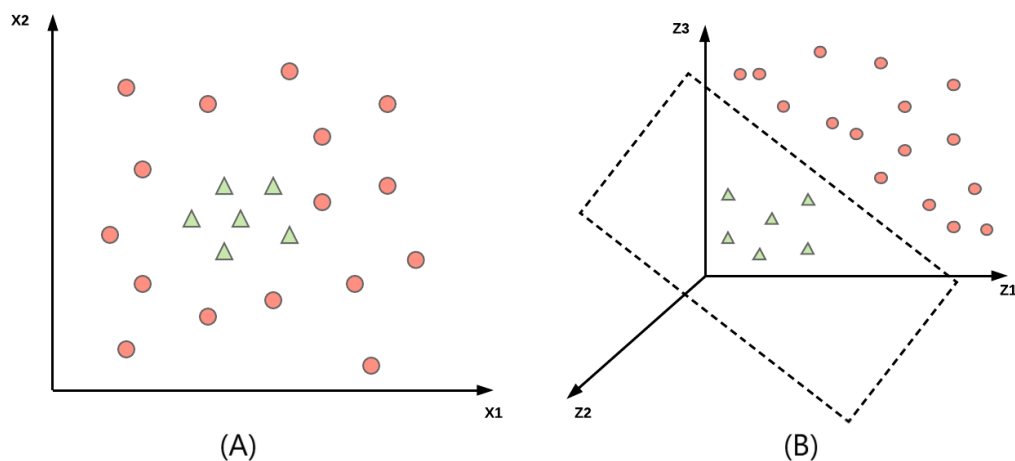
A distância euclidiana entre o hiperplano e os pontos resultantes, é determinada pela seguinte equação:

$$f(x) = \frac{|(\mathbf{w} \cdot \mathbf{x}) + b|}{\|\mathbf{w}\|}, \quad (3.4)$$

Logo, minimizar $\|w\|$ irá maximizar as margens do hiperplano.

Para a classificação de problemas não lineares, o SVM mapeia os dados para um espaço de característica onde esses valores possam ser separados, como visto na Figura 7.

Figura 7 – (A) Conjunto de dados não linear e (B) Conjunto de dados no espaço de características



Fonte: Autor

Para realizar essa mudança no espaço, uma função *kernel* é utilizada, e as mais utilizadas são:

- Polinomial $\rightarrow K(x, y) = ((x_i, y_j) + 1)^p$
- Linear $\rightarrow K(x, y) = x \cdot y$
- Gaussiana(RBF) $\rightarrow e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$

O parâmetro p é especificado a priori pelo usuário.

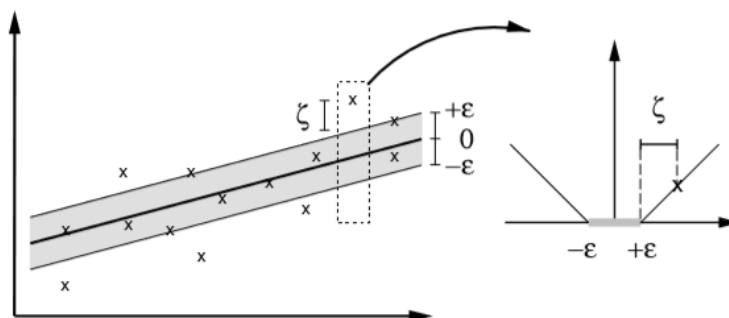
3.3.1 Support Vector Regression

Segundo Verdério (2015), máquinas de vetores de suporte para regressão diferem da técnica de classificação no sentido de que, enquanto a segunda apenas busca dividir os dados em diferentes classes e classificar corretamente os dados futuros, a primeira busca encontrar um preditor que aproxime bem os dados. O SVR utiliza uma função de perda para o cálculo entre o erro cometido na predição, o algoritmo ignora os erros que estão a uma certa distância de diferença, logo, somente são permitidos os erros menores que um valor ϵ , assim, é mantida a generalização, pois no final do processo de treinamento se

termina com várias soluções, a generalização, porém é utilizada aquela que apresenta um menor erro (LUCENA, 2016).

Um objetivo do SVR é buscar uma função que encontre um valor máximo de desvio ϵ em relação aos valores alvos. Assim, encontrar valores entre $y_i - \epsilon$ e $y_i + \epsilon$. Essa função permite introduzir variáveis de folgas não negativas ϵ_i para penalizar os valores fora da margem encontrada. A Figura 8 representa graficamente os parâmetros apresentados nesse parágrafo.

Figura 8 – Função de Perda SVR



Fonte: AlexJ.Smola e Scholkopf (2003)

A Função de regressão pode ser definida pela seguinte fórmula:

$$F(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{3.5}$$

No qual, K é a função Kernel e os coeficientes b, α_i e α_i^* são resultados da seguinte minimização:

$$\min_{\forall w, b, \epsilon, \epsilon^*} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^n (\epsilon_i + \epsilon_i^*) \tag{3.6}$$

sujeito a:

$$y_i - w \cdot x_i - b \geq \epsilon + \epsilon_i \tag{3.7}$$

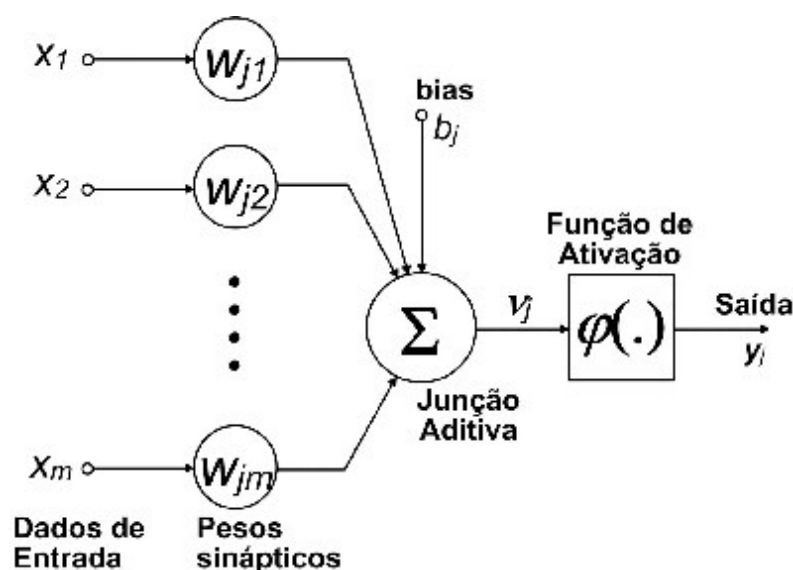
$$w \cdot x_i + b - y_i \geq \epsilon + \epsilon_i^* \tag{3.8}$$

Neste trabalho, o SVR será utilizado de forma a prever a temperatura seguinte, dado um conjunto de características obtidas em um momento anterior, sendo assim múltiplas séries temporais.

3.4 Redes Neurais Artificiais

Uma rede neural artificial é um modelo inspirado no sistema nervoso humano, no qual sua estrutura é composta pela conexão de várias unidades de processamento similares a um neurônio do cérebro. Essas unidades são conectadas por canais de comunicação e associados a determinados pesos, cada neurônio artificial da rede faz operações sobre os dados inseridos e o comportamento de aprendizado é realizado através da interação entre os canais de comunicação. Uma estrutura de um neurônio artificial simples proposta por McCulloch e Pitts (1990) pode ser visualizada graficamente no diagrama seguinte:

Figura 9 – Esquema de um neurônio artificial



Fonte: Baseado em Haykin (2001)

Esse modelo é composto por três elementos básicos:

- **Conjunto de sinapses ou elos de conexão**, cada um possui seu peso específico. De forma detalhada, uma entrada X em uma sinapse j conectada a um neurônio k é multiplicada pelo peso w_{kj} .
- **Um somador** para somar todos os valores de entrada do neurônio.
- **Uma função de ativação** para restringir os valores de saída da rede. Geralmente, os intervalos de saída podem ser descritos como um intervalo fechado de $[0,1]$ ou $[-1,1]$.

Além disso, o modelo da Figura 9 inclui o valor de um bias, que é utilizado para aumentar ou diminuir os valores de entrada para a função de ativação, ele permite uma melhor adaptação da rede em relação ao aprendizado (HAYKIN, 2001).

Em termos matemáticos, uma equação que pode descrever um neurônio k é escrita da seguinte forma:

$$u_k = \sum_{j=1}^n w_{kj} x_j \quad (3.9)$$

e

$$y_k = \varphi(u_k + b_k) \quad (3.10)$$

onde y_k , é resultado de uma função de ativação aplicada ao valor resultante do neurônio u_k mais a o valores de um bias b_k .

Em relação a estrutura de uma rede neural, ela é organizada em camadas, onde as unidades podem estar conectadas com as unidades das camadas anteriores. Essas camadas podem ser classificadas em:

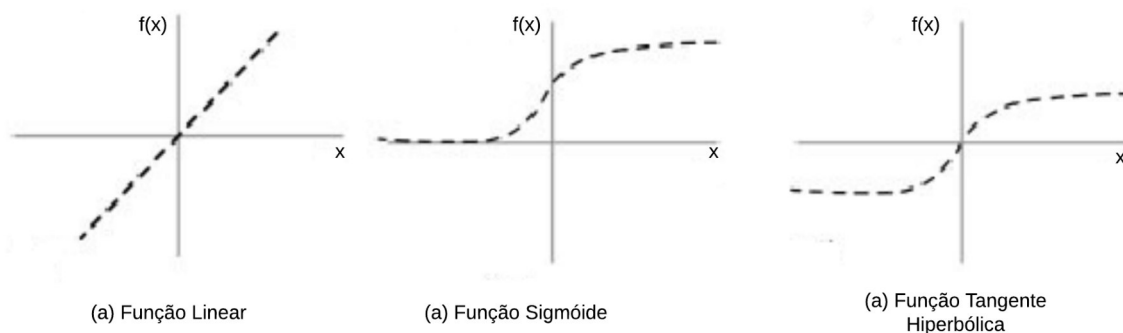
- **Camada de Entrada**, onde ocorre a entrada dos dados, serve para a distribuição dos valores para a rede e não realiza nenhum tipo de cálculo;
- **Camada Escondida**, a maior parte do processamento é feito nessa camada, é responsável por atribuir a resolução de problemas não lineares na rede;
- **Camada de Saída**, onde o resultado de todo processamento é apresentado.

3.4.1 Função de Ativação

Uma função de ativação é uma transformação não linear que é realizada sobre os dados de entrada. Essa função permite que as redes neurais possam aprender mais do que relações lineares e assim resolver problemas mais complexos. Elas basicamente, decidem se um neurônio será ativado ou não. As funções de ativação mais utilizadas são:

- **Função Linear**: É uma simples equação linear, dada pela fórmula: $f(x) = ax$;
- **Função Sigmóide**: A saída dessa função varia entre 0 e 1, tendo um formato de S. Basicamente, a função tenta empurrar os valores de y para o extremo. Pode ser escrita pela formula: $f(x) = \frac{1}{1+e^{-\lambda x}}$;
- **Função Tangente Hiperbólica**: A função tahn é muito semelhante a função sigmóide, porém seus valores variam de -1 e 1. É da forma: $f(x) = \frac{2}{1+e^{-2x}} - 1$.

Figura 10 – Funções de Ativação

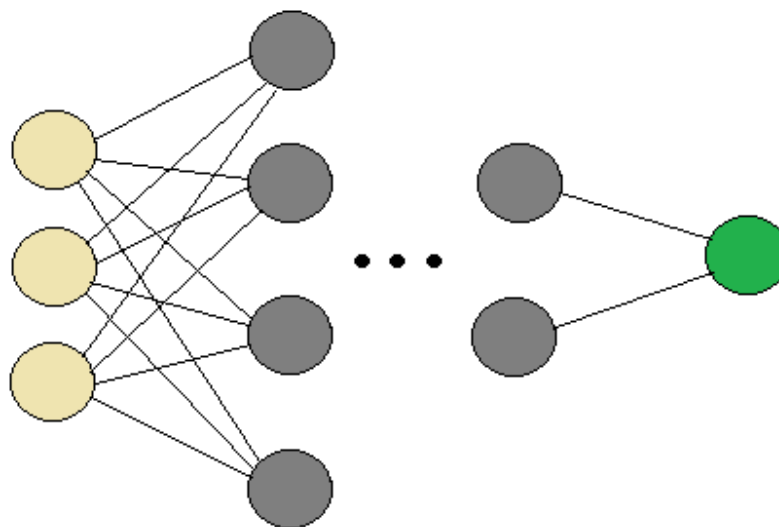


Fonte: Autor

3.4.2 Multilayer Perceptron (MLP)

Na rede *Multilayer Perceptron* há uma modificação em relação a uma rede neural simples. A grande diferença, é que a *MLP* adiciona uma camada intermediária ou camada escondida na estrutura da rede e permite a solução de problemas não lineares. Com o desenvolvimento do algoritmo de *backpropagation*, o treinamento desse tipo de rede tornou-se mais eficiente. A Figura 11 mostra o exemplo de uma rede neural com multicamadas.

Figura 11 – Multilayer Perceptron



Fonte: Autor

A RNA pode utilizar de diversos algoritmos para o reconhecimento de padrões e a solução de problemas não lineares. Um dos algoritmos mais utilizados pelas redes é o **algoritmo de *backpropagation***, esse algoritmo consiste no cálculo de um erro com

os valores gerados e os valores reais da entrada, o erro é então propagado em um sentido inverso, ele pode ser dividido em duas fases:

- A primeira fase consiste no processo de *forward*, onde as entradas percorrem a rede e os resultados na camada de saída são obtidos (fase de propagação), durante essa etapa os pesos dos neurônios não apresentam mudança.
- A segunda fase é denominada de *backward*, no qual o gradiente da função é calculado no final da rede e esse valor é utilizado para atualizar os pesos da rede em um sentido inverso, assim os pesos das conexões vão sendo atualizados na medida que o erro é retropropagado, buscando reduzir o erro durante o treinamento da rede. O *backpropagation* é repetido até que o erro seja o menor possível ou até um número definido de iterações.

A atualização dos peso da rede pode ser associado com a seguinte regra:

$$w_i \leftarrow w_i + \Delta w_i \quad (3.11)$$

$$\Delta w_i = \eta(t - o)x_i \quad (3.12)$$

onde, t é a saída desejada, o é a saída gerada pela rede neural, e η é uma constante pequena chamada de taxa de aprendizado. Essa taxa corresponde a velocidade do aprendizado do modelo, no qual, um valor pequeno desse parâmetro gera mudanças pequenas no valor dos pesos e conseqüentemente maior tempo para o treinamento do modelo. Um problema de uma taxa de treinamento baixa é a possibilidade de cair em uma região de ótimo local, pois a rede não consegue calcular valores para fazer o modelo sair da região, uma forma seria encontrar uma taxa de aprendizado intermediária, pois se for muito grande há o risco também de perder o valor ótimo para a solução do problema.

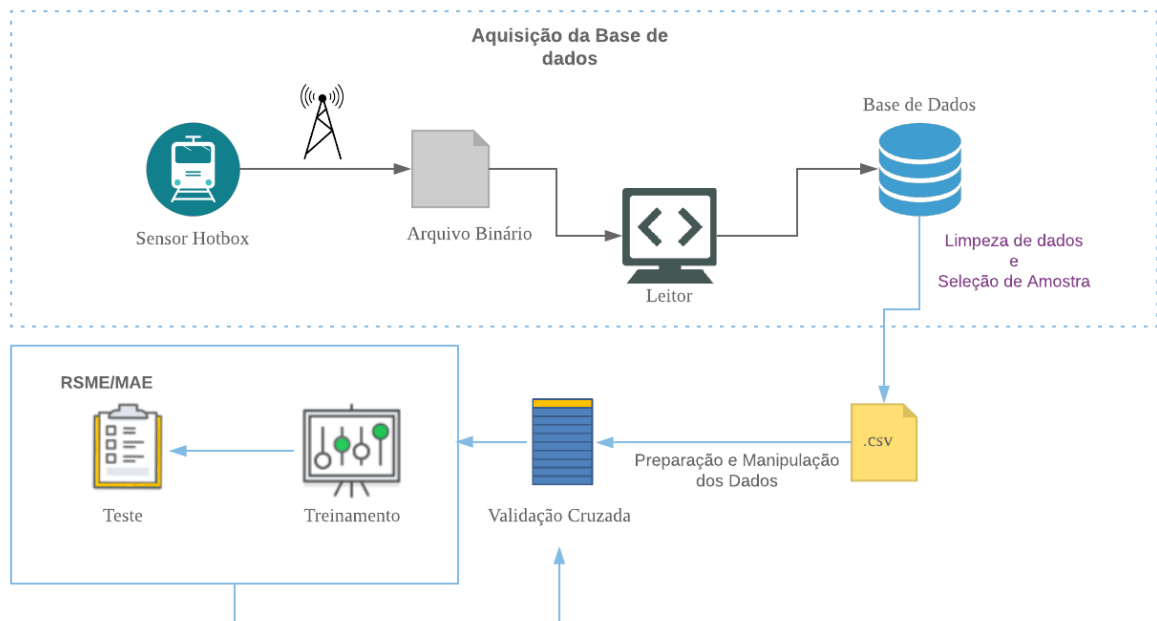
Neste trabalho, a *MLP* será utilizada de forma a predizer a temperatura seguinte, dado um conjunto de características obtidas em um momento anterior, utilizando-as como valores de entrada em sua primeira camada.

4 METODOLOGIA

O presente capítulo descreve as ferramentas, método proposto e etapas utilizadas para o processo de regressão utilizado neste trabalho.

A Figura 12 apresenta um diagrama ilustrativo da solução proposta em alto nível de abstração, nessa Figura, o processo de aquisição da base de dados inicia com a detecção de temperaturas pelo sensor do Hotbox, os dados são transferidos via rádio para um computador central que gera arquivos binários. Um leitor realiza a conversão dos dados e armazena no banco de dados. No próprio banco, ocorre uma das etapas de preparação dos dados, o que gera um arquivo .csv, posteriormente, a etapa de predição e validação dos resultados é realizada.

Figura 12 – Método proposto



Fonte: Autor

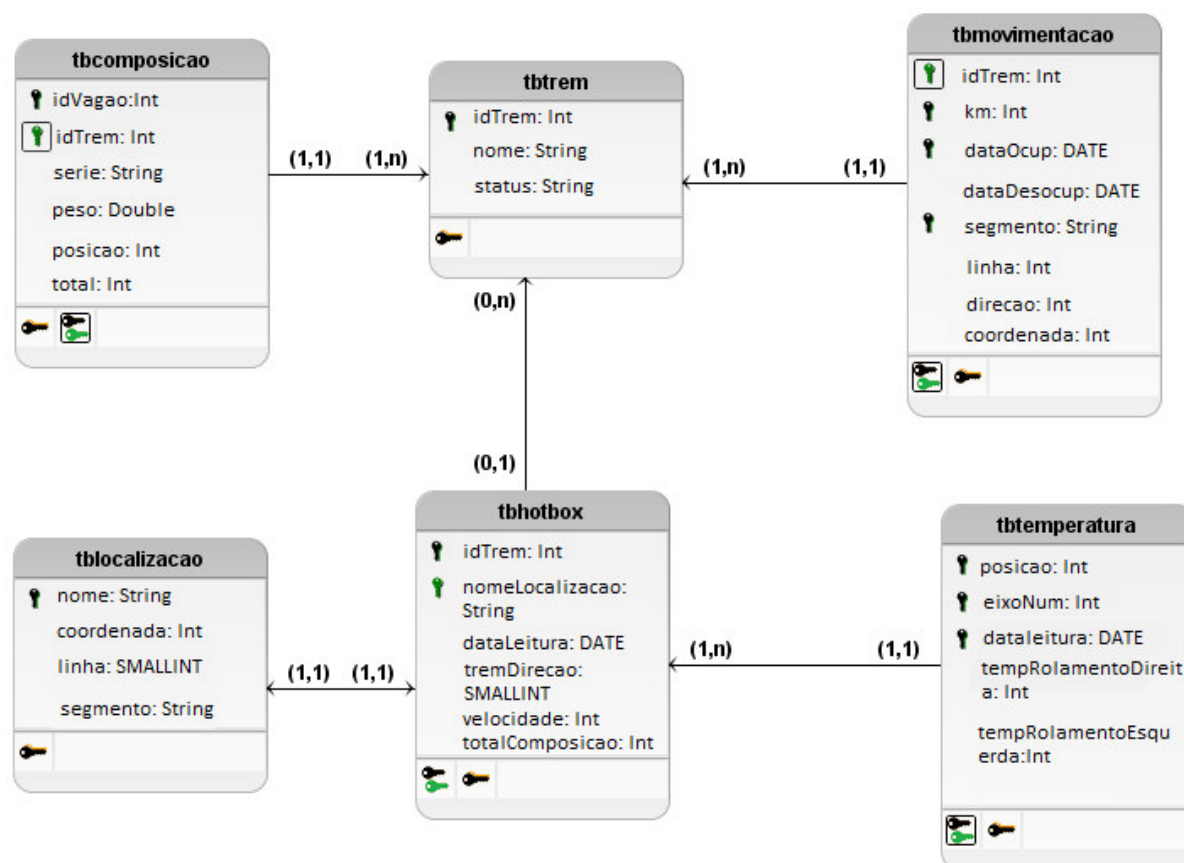
4.1 Base de dados

A base de dados utilizada nesse trabalho foi obtida diretamente de equipamentos de uma ferrovia do Brasil no período de 2016 até 2017. No cenário de estudo, ocorreram cerca de 12 alarmes de temperatura de rolamento no ano de 2017 e 24 alarmes no ano de 2016, um número pequeno se comparado as outras ferrovias do mundo.

O equipamento *hotbox* foi utilizado para medir as temperaturas de cada rolamento do trem através de radiação infravermelha. Toda vez que um trem passa ao lado do

equipamento com uma velocidade de 9 a 192Km/h, as medidas são realizadas e os dados são enviados para um software central. O histórico de cada rolamento não estava pronto e o equipamento do *hotbox* não identifica trem, vagão ou rolamento, o que dificultou a criação da base de dados. Para solucionar o problema, relações no banco de dados foram feitas para relacionar o trem e a posição de cada vagão, assim como identificar cada rolamento.

Figura 13 – Banco de dados



Fonte: Autor

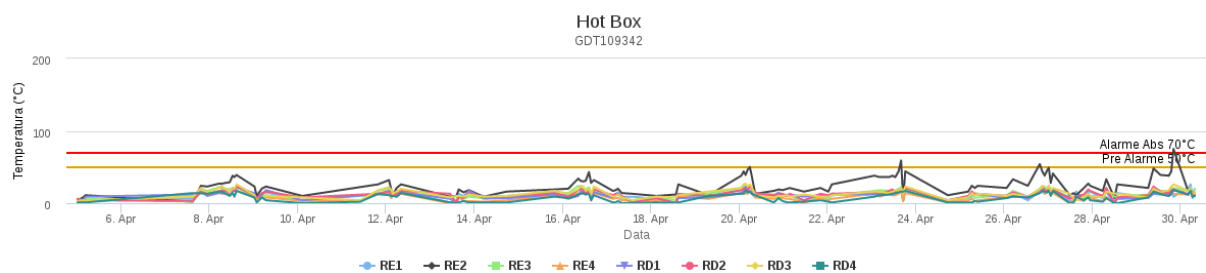
Os valores de medições de pelo menos 2 anos do equipamento do *hotbox* foram lidos e relacionados com uma base da movimentação de trens da ferrovia, e cada trem foi associado. Para a descoberta de cada vagão, uma outra base com a composição do trem foi responsável por mapear o trem e a contagem da composição para a criação de uma chave de cada vagão.

No final, a base foi constituída por um histórico de temperatura com 38 vagões, além disso possui os valores relacionados a peso, velocidade do trem e a localização onde realizou-se a medição. Para a busca dos dados relevantes realizou-se consultas com a linguagem SQL (*Structured Query Language*), os dados escolhidos foram especificadamente selecionados para a composição de uma base de dados variada, há exemplos de vagões com rolamentos alarmados e não alarmados.

O processo para obter esses dados foi constituído de 4 etapas: na primeira a leitura dos dados pelo equipamento e o armazenamento no banco de dados, no qual a cada momento em que um trem passa pelo sensor de um dos equipamentos distribuídos pela ferrovia, todos os valores de temperatura de cada rolamento de cada vagão são medidos; em uma etapa seguinte os vagões são identificados e é possível criar um histórico para uma análise de tendência. Os próximos passos, como a seleção dos melhores dados e a criação de dados supervisionados foram realizados com o intuito de preparar os valores para a entrada dos algoritmos escolhidos. A seleção dos melhores conjuntos de exemplos foi construída com uma aplicação Java e foram adotados critérios de rolamentos alarmados, rolamentos com temperaturas acima de 50° e históricos de exemplos com valores normais. A finalização do processo se deu com a criação de um arquivo `.csv` com um histórico de pelo menos 50 medições de cada rolamento.

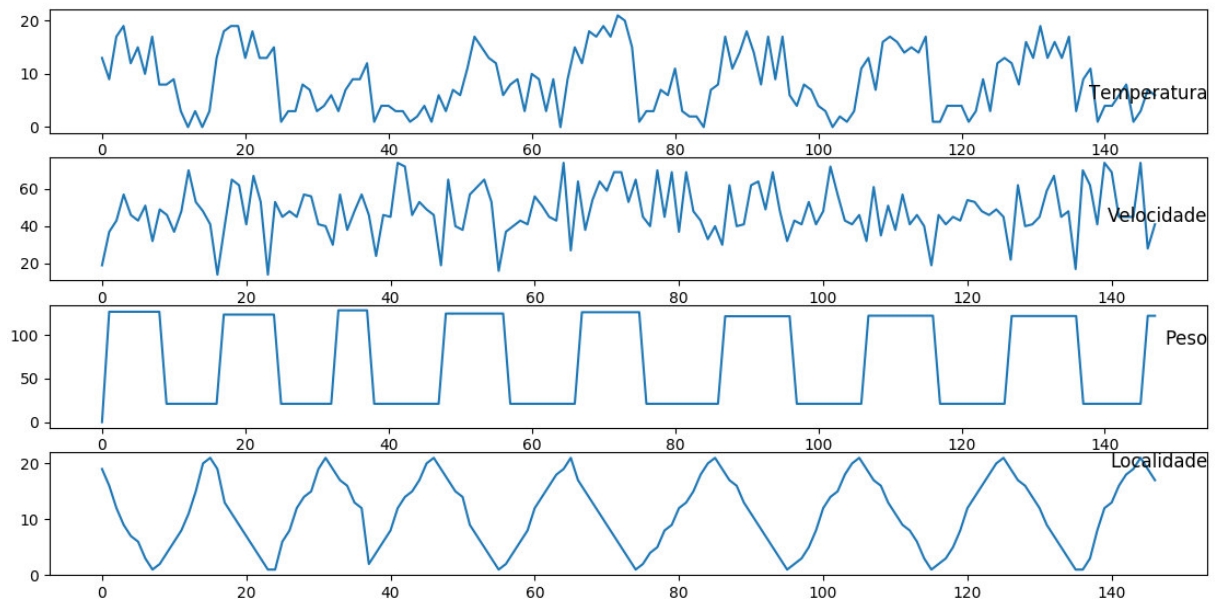
O objeto de estudo desse trabalho não apresenta uma regularidade tão clara como em outras séries temporais, por isso, a predição futura pelo algoritmos de aprendizado torna-se complexa, um exemplo do comportamento da temperatura pode ser visto na Figura 14. Visando tentar relacionar características que influenciem o valor da temperatura(T), dados de peso do vagão(P), velocidade(V) e Localidade(L) do equipamento na ferrovia também foram inseridas como variáveis para o modelo e podem ser visualizadas na Figura 15 , um outro gráfico de dispersão das características pode ser visualizado no apêndice A. Esse tipo de série temporal com várias características é denominada de série **multivariada**.

Figura 14 – Gráfico de Temperatura do Rolamento



Fonte: Autor

Figura 15 – Gráfico de características



Fonte: Autor

A série foi preparada no formato da Tabela 1. Cada linha possui um histórico com 50 medições de cada característica em tempos definidos, construído através de uma janela deslizante de tempo. Cada medição é composta pela Temperatura (T), Peso (V), Localidade (L) e Velocidade (V), Além das 50 medições, as características de Peso, Localidade e Velocidade da saída também são inseridas como características. A saída do modelo é a próxima temperatura da sequência.

Tabela 1 – Entrada e saída para os modelos

Entrada	Saída
$T_1, P_1, V_1, L_1, T_2, P_2, V_2, L_2, \dots, T_{50}, P_{50}, V_{50}, L_{50}, V_{51}, P_{51}, L_{51}$	T_{51}
$T_2, P_2, V_2, L_2, T_3, P_3, V_3, L_3, \dots, T_{51}, P_{51}, V_{51}, L_{51}, V_{52}, P_{52}, L_{52}$	T_{52}
.....
$T_{(n-50)}, P_{n-50}, V_{n-50}, L_{n-50}, \dots, T_{(n-1)}, P_{n-1}, V_{n-1}, L_{n-1}, V_n, P_n, L_n$	$T_{(n)}$

4.2 Pré-Processamento dos dados

A qualidade dos dados é fundamental para a construção de um modelo com bons resultados. Frequentemente, os dados possuem diversos problemas, como grande quantidade de ruídos, características que não possuem valor significativo para a predição, valores desconhecidos, entre outros. Diante disso, a etapa de pré-processamento visa aprimorar a qualidade dos dados para a etapa seguinte, a predição.

De uma forma geral, o pré-processamento de dados é uma etapa semi-automática, pois depende da capacidade de uma pessoa para identificar os problemas e avaliar as melhores alternativas para solucionar cada um desses problemas (BATISTA, 2003).

4.2.1 Limpeza de dados

O primeiro passo no pré-processamento foi a retirada de ruído dos dados inseridos na base. Devido a base não está pronta, essa etapa foi essencial para a qualidade dos dados. A atividade de limpeza de dados exige um especialista de domínio para verificar os possíveis ruídos que podem comprometer a construção do modelo. Algumas atividades podem ser descritas como:

- Verificação de divergências nos dados;
- Verificação de contradições na conversão de valores numéricos errados;
- Verificação de contradições na conversão de valores categóricos errados;
- Duplicação de valores;
- Falta de valores.

Na base de dados do *hotbox*, alguns problemas na leitura dos equipamentos ou na falta de informações no momento de junção com outras base de dados, gerou tanto duplicação como a falta de alguns valores. Para solucionar esse problema, os objetos com valores ausentes e as duplicações foram eliminados usando processos no banco de dados. Além disso, o range de alguns dados numéricos era inconsistente e teve que ser alterado.

4.2.2 Redução de dados

Os algoritmos de aprendizado podem ter dificuldade em lidar com uma base de dados muito grande, existe um balanço entre a eficiência computacional e a acurácia do modelo. Sendo assim, quanto mais dados são utilizados, maior tende a ser a acurácia do modelo e menor a eficiência computacional. (CARVALHO, 2011). No cenário deste trabalho, em média, cerca de 17 trens circulam diariamente. Cada leitura do equipamento gera em torno de 1360 novas linhas no banco de dados, e um valor diário de 45.492 dados. Com um histórico de um ano, foi necessário escolher os melhores exemplos para diminuir a dimensão da base. Essa amostra de dados foi escolhida de forma manual para a retirada de históricos com bastante ruído, por isso, com base em valores de temperatura que alarmaram, valores com temperaturas de pré-alarmede e dados com valores normais (rolamento sem falha) algumas amostras foram selecionadas.

4.2.3 Transformação dos dados

As técnicas utilizadas, redes neurais e *support vector machine*, lidam apenas com valores numéricos. Assim, quando existe valores simbólicos na base de dados, é necessário converter para valores numéricos. Neste trabalho, os valores de localidade, foram os únicos que precisaram passar por essa etapa. Cada localidade do equipamento era mapeada por letras do alfabeto e indicavam uma sequência na rota de um trem, esses valores foram transformado para números de 1 até 24.

4.2.4 Normalização

Qualquer tarefa de mineração de dados sem a etapa de normalização de séries temporais é sem sentido (KEOGH, 2003). A normalização apenas modifica cada característica para um intervalo definido, neste trabalho o intervalo escolhido foi $[0,1]$. Essa etapa é essencial para os algoritmos otimizarem seus resultados, pois é necessário evitar que os valores de grandezas diferentes não influenciem de maneira diversas os ajustes dos pesos da rede neural durante o treinamento (GAMBOGI, 2013).

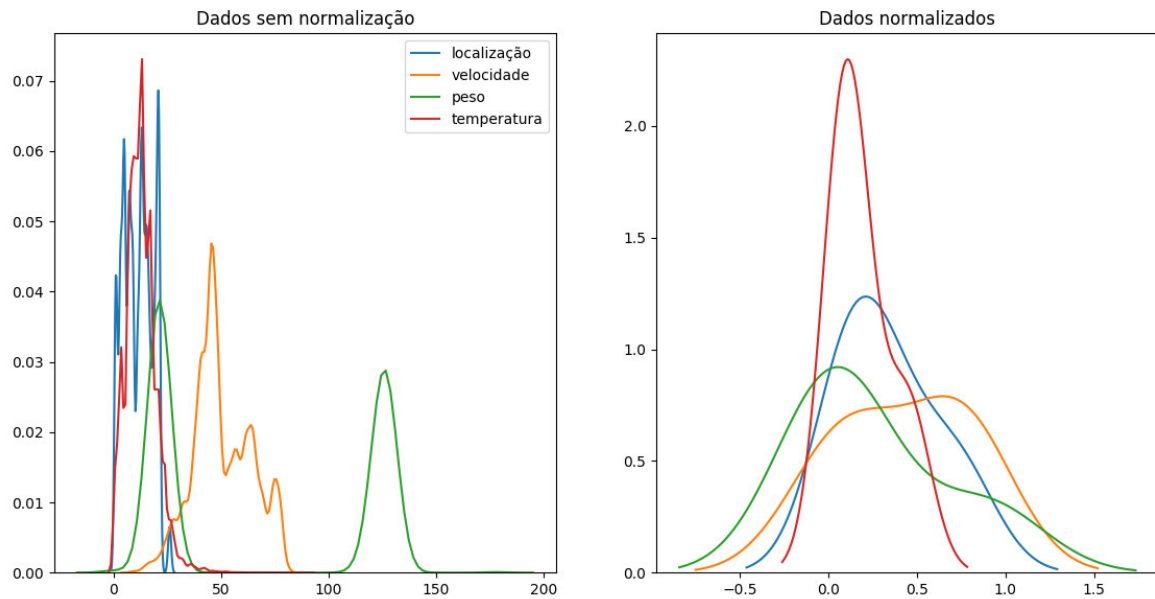
Neste trabalho, os dados foram normalizados utilizando a equação:

$$z'(t) = \frac{z(t) - \min(z)}{\max(z) - \min(z)} \quad (4.1)$$

Em que $\max(z)$ e $\min(z)$ são os valores de máximo e mínimo de cada atributo, e o valor $z(t)$ é o valor a ser normalizado.

Segue gráfico abaixo de densidade kernel dos valores antes de serem normalizados, e depois da normalização, onde o eixo x mostra a densidade de cada valor e o eixo y o range de valores reais de cada variável. É possível verificar que depois da normalização todas as variáveis estão na mesma escala em relação uma a outra.

Figura 16 – Densidade de Dados não normalizados e Dados normalizados



Fonte: Autor

4.3 Predição

Finalizado o processo de pré-processamento dos dados, a etapa seguinte é a predição com os algoritmos de aprendizado de máquina escolhidos, o *support vector regression* e as redes neurais. Um modelo preditivo pode ser caracterizado como uma função matemática (f) para a aproximação de valores de saída (y) e seus resultados verdadeiros com o mapeamento de características (x) de entrada. No modelo de predição desse trabalho foi utilizado dados históricos para realizar a predição de uma temperatura seguinte.

4.3.1 Ferramentas Utilizadas

A parte de aprendizado de máquina foi implementada utilizando a linguagem Python 3, para ela existem bibliotecas que disponibilizam algoritmos de aprendizado de máquina. A biblioteca Scikit-learn foi destacada por ser de código aberto e possuir algoritmos necessários para a implementação do método proposto neste trabalho, além disso o Keras foi escolhido para trabalhar com as rede neurais. Segue abaixo a lista de ferramentas utilizadas:

- **Scikit-learn:** O Scikit-learn é uma biblioteca de software para aprendizado de máquina desenvolvida para a linguagem Python. Ela possui diversos algoritmos para classificação, regressão e clustering, como o *support vector machines*, *k-means*, *multilayer perceptron* e *random forests*.

- **Keras:** é uma biblioteca de código aberto própria para a modelagem com redes neurais. Foi desenvolvida para a linguagem python e é capaz de trabalhar com redes neurais profundas.
- **Seaborn e Matplotlib:** Essas bibliotecas foram utilizadas para a visualização dos dados e plotagem dos gráficos.
- **LibSVM:** É uma biblioteca de código aberto desenvolvida por [Chang e Lin \(2011\)](#). Nesse trabalho o gridSearch da libsvm foi utilizado para a seleção dos melhores hiperparâmetros do *support vector regression*.

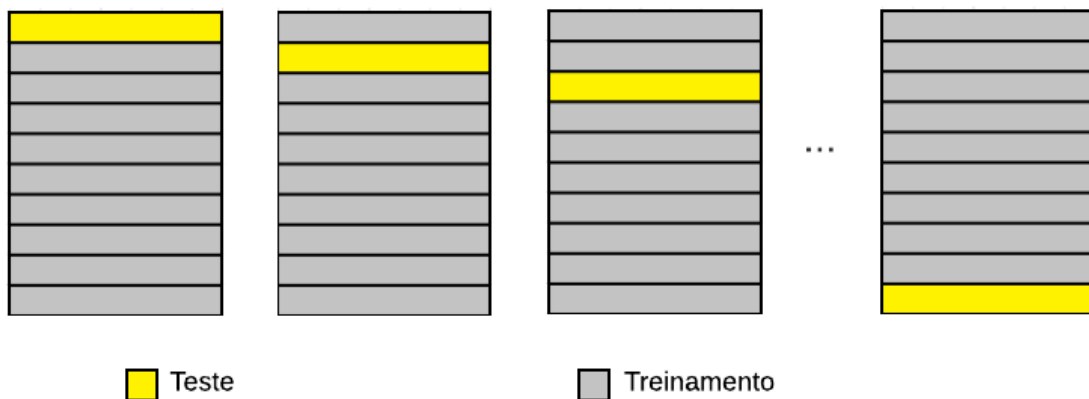
4.4 Validação dos Resultados

Após o treinamento e criação do modelo de aprendizado de máquina, é necessário verificar se os resultados são consistentes. A verificação com somente um conjunto de dados de treinamento pode trazer resultados ilusórios, pois o modelo pode se sobreajustar (overfitting) nos valores de treinamento e ter péssimos resultados em novos conjuntos de dados. Portanto a validação cruzada foi aplicada, de forma a aumentar a capacidade de generalização do modelo.

4.4.1 Validação Cruzada

Para evitar problemas de overfitting e underfitting, a técnica de **validação cruzada** é utilizada para a criação de um modelo de dados adequado. Um dos métodos mais utilizado da validação cruzada e o escolhido neste trabalho é o **método k-fold**, no qual consiste em dividir o conjunto total de dados em k subconjuntos de mesmo tamanho, sendo assim um subconjunto é utilizado para teste, e os k - 1 que restaram servem para o treinamento e definição de parâmetros. Essa medida torna os resultados das predições mais confiáveis, garantindo que os dados utilizados para treinamento não estejam nos dados de teste. A Figura 17 ilustra a metodologia da validação cruzada pelo método k-fold.

Figura 17 – Validação Cruzada



Fonte: Autor

4.4.2 Métricas

Para avaliar o resultado do preditor produzido, algumas métricas foram selecionadas para o treinamento dos algoritmos e pontuação dos testes realizados. O problema desse trabalho é caracterizado como um problema de regressão, logo, as medidas de avaliações de resultados não são as mesmas que um problema de classificação. Para a regressão, as seguintes medidas foram utilizadas:

- Raiz do Erro Quadrado Médio(REQM): Essa métrica calcula a distância entre os valores numéricos previstos e a resposta ideal(resultado verdadeiro). Quanto menor o valor do REQM, melhor será a capacidade preditiva do modelo, um resultado ideal seria o valor 0. A seguinte fórmula é utilizada:

$$REQM = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4.2)$$

No qual, n é a quantidade de exemplos utilizados, y_i o valor real e \hat{y}_i o resultado da máquina. No REQM, as diferenças menores tem pouco importância no resultado, enquanto diferenças maiores recebem mais peso.

- Erro Absoluto Médio (MAE): Parecido com o REQM, contudo ao invés de elevar a diferença do valor real e o valor previsto ao quadrado, ele toma o valor absoluto. Logo, em vez de atribuir um peso de acordo com a diferença, ocorre a atribuição de peso de forma linear. Segue a fórmula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.3)$$

No qual, n é quantidade de exemplos utilizados, y_i o valor real e \hat{y}_i o resultado da máquina.

5 RESULTADOS

Este capítulo apresenta os testes realizados em cada método proposto com os seus respectivos parâmetros, bem como os resultados obtidos. Além disso, há a visualização de gráficos de resultados, e a comparação de cada algoritmo com as métricas de *REQM* e *MAE*.

Inicialmente, os modelos foram testados utilizando a validação cruzada com o método k-fold, no qual $k = 10$. No final de todas as etapas, a média dos resultados foi calculada para encontrar o resultado final. As Tabelas 2 e 3 mostram um exemplo de um dos testes com a *Multilayer Perceptron* e do *Support Vector Regression*, respectivamente.

Todos os algoritmos foram testados em um computador com processador i5, 4 núcleos e 8GB de memória Ram.

Tabela 2 – Resultados K-Fold MLP

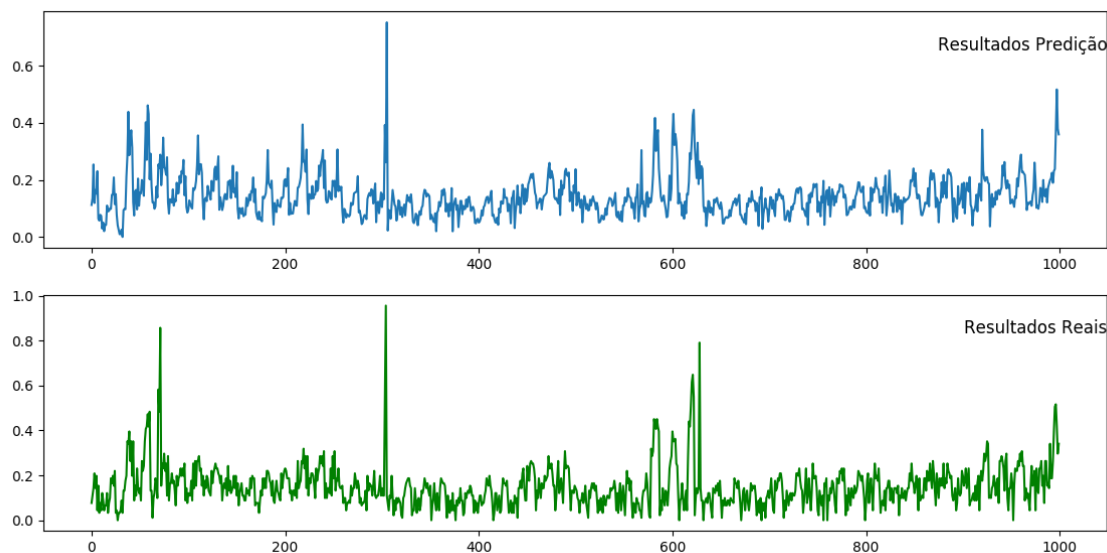
Fold	REQM	MAE
1	0.04405	0.04405
2	0.04146	0.04146
3	0.03750	0.03750
4	0.04346	0.04346
5	0.04975	0.04975
6	0.04825	0.04825
7	0.04370	0.04370
8	0.04624	0.04624
9	0.05992	0.05992
10	0.05047	0.05047
Média	0.04650	0.04650

Tabela 3 – Resultados K-Fold SVR

Fold	REQM	MAE
1	0.06652	0.04582
2	0.06086	0.04285
3	0.06320	0.04363
4	0.05910	0.04588
5	0.06241	0.04647
6	0.06257	0.04738
7	0.05455	0.04101
8	0.05420	0.03992
9	0.06421	0.04339
10	0.06016	0.04207
Média	0.06078	0.04384

Alguns gráficos das predições serão apresentadas no modelo da Figura 18. No qual há o resultado da predição pelo algoritmo e os resultados reais.

Figura 18 – Resultado de uma predição



Fonte: Autor

5.1 Support Vector Regression

Para a seleção dos parâmetros do *SVR*, utilizou-se o algoritmo *grid search* (CHANG; LIN, 2011), no qual exaustivamente testa os melhores resultados para cada parâmetro definido para a otimização dos hiperparâmetros do algoritmo. Assim, os seguintes parâmetros foram otimizados para cada função *kernel*:

- C: Esse parâmetro busca um equilíbrio entre a largura da margem do hiperplano com a minimização do erro (PEDREGOSA et al., 2011).
- Gamma: Esse parâmetro define quais pontos ao redor da fronteira de decisão terão maior peso. Caso o valor de gamma seja alto, os valores mais próximos a fronteira terão maior importância na inferência dos pontos restantes. Se o valor de gamma for pequeno, os pontos mais distantes da fronteira terão maior peso (PEDREGOSA et al., 2011).

Os valores otimizados dos parâmetros do *SVR* foram encontrados depois de uma exaustiva busca, e a Tabela 4 mostra o resultado final encontrado.

O *Support Vector Regression* foi analisado com dois tipos de *kernel*, o *Gaussiano(RBF)* e o *Polinomial*. Os resultados usando validação cruzada são apresentados na

Tabela 4 – Resultados do Grid

GridSearch SVR		
C	Gamma	p
0.5	0.125	0.0078125

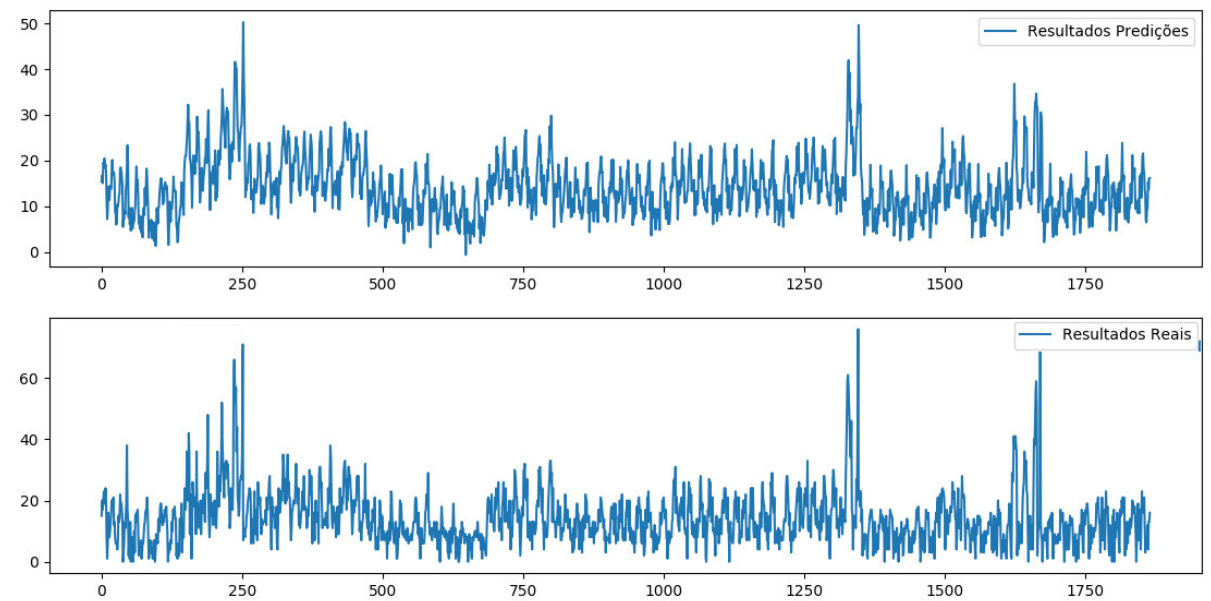
Tabela 5. Pode-se notar um erro médio de 4% , onde os dados variam de [0-1] para o EAM e 6% para o REQM.

Tabela 5 – Resultados SVR

SVR		
Kernel	EAM	REQM
RBF	0.0438	0.0607
Polinomial	0.0438	0.0608

A Figura 19 demonstra graficamente 1750 previsões realizadas por um 1 Fold da etapa do K-fold pelo *SVR* em comparação com os resultados reais.

Figura 19 – Exemplo de previsão SVR



Fonte: Autor

5.2 Multilayer Perceptron

O modelo de rede neural proposto foi construído com uma camada oculta ,*taxa de aprendizado* igual 0.001, batch-size igual 10 e o *adam* foi escolhido como otimizador. Os números de neurônios foram testados em diversos valores para identificar a melhor configuração, assim como, a quantidade de épocas, e testados com k-fold para cada

configuração. Os resultados são mostrados na Tabela 6 A função de ativação selecionada foi a tangente hiperbólica.

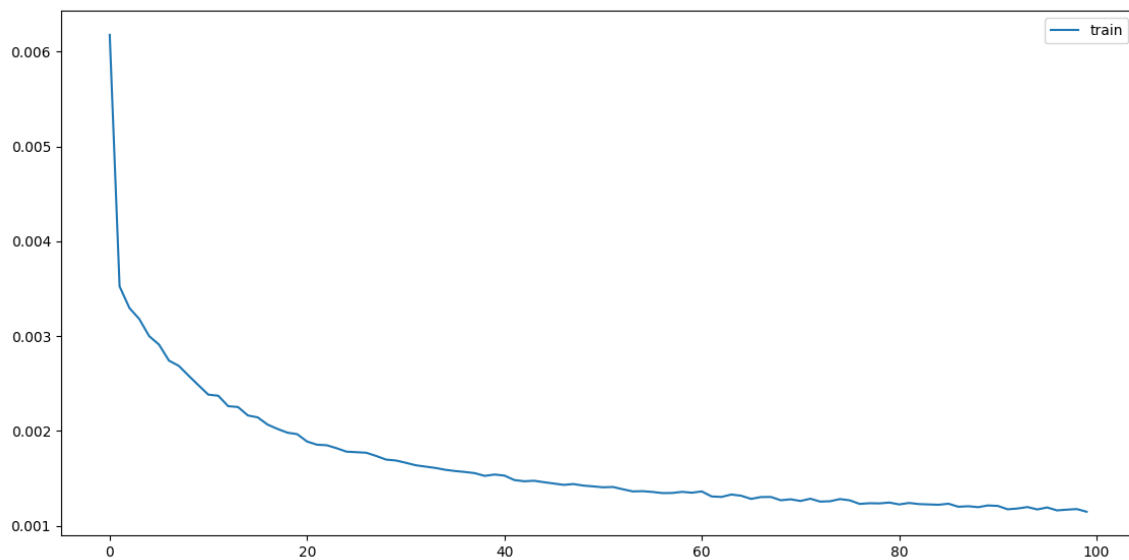
Tabela 6 – Resultados Rede Neural

MLP			
Neurônios	Épocas	EAM	REQM
100	100	0.0468	0.0468
100	150	0.0465	0.0465
150	100	0.0478	0.0478
150	150	0.0471	0.0471

Os resultados da rede neural foram satisfatórios para as duas métricas selecionadas. A configuração de 100 neurônios e 150 épocas encontrou os valores e será utilizada para comparar com o *Support Vector Regression*.

A Figura 20 apresenta a curva de aprendizado da rede neural ao decorrer de 100 épocas. A partir de 80 iterações a taxa de erro permanece constante.

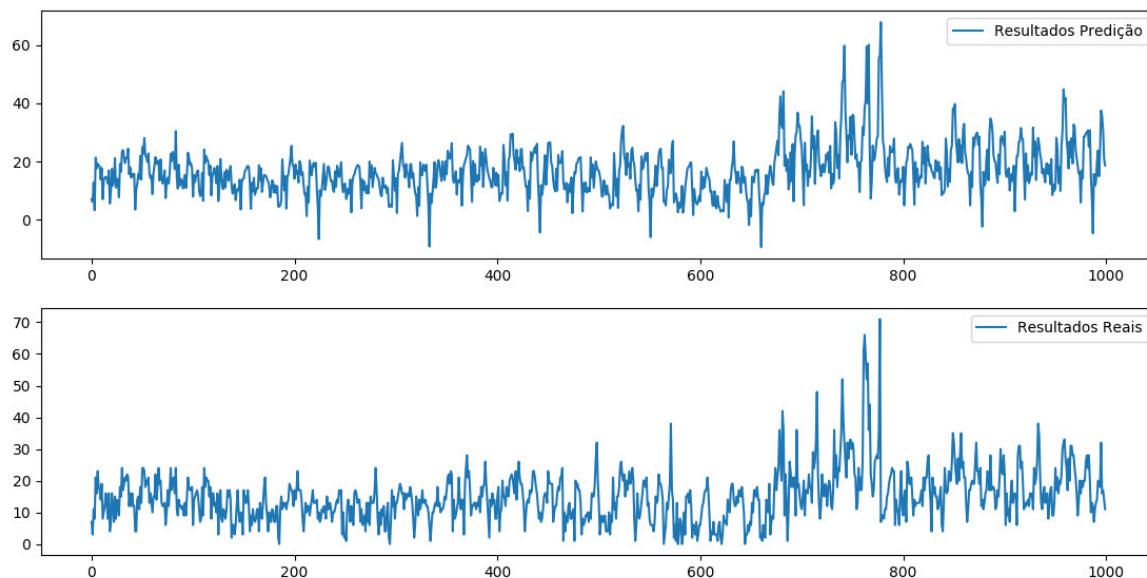
Figura 20 – Taxa de Erro



Fonte: Autor

A Figura 21 demonstra graficamente 1000 predições realizadas pela *MLP* em comparação com os resultados reais. É possível verificar que a *MLP* foi capaz de generalizar um aumento da temperatura e identificar um possível alarme.

Figura 21 – Exemplo de predição Multilayer Perceptron



Fonte: Autor

5.3 Discussão

Os algoritmos selecionados tiveram bons resultados, isso pode ser afirmado dado os valores das métricas aplicadas. O tipo de problema desse trabalho requer um mínimo de erro possível e uma busca para encontrar valores mais altos de temperatura. A *Multilayer Perceptron* conseguiu melhores resultados no *Raiz do Erro Quadrado Médio* se comparado ao *Support Vector Regression*, o que indica que a

MLP não há erros tão grandes na predição, pois no *REQM* diferenças maiores possuem mais peso. O *SVR* teve valores de *REQM* bem maiores do que o *MAE*, isso se deve a um erro maior na predição de temperatura mais altas. Graficamente, é possível perceber que o *SVR* não conseguiu generalizar e detectar temperaturas anormais, considero que um pior resultado desse algoritmo na detecção de variações é devido o desbalanceamento da base de dados, pois não há tantos exemplos de temperaturas anormais. A Tabela 6 apresenta os melhores resultados do dois algoritmos.

Tabela 7 – Comparação de resultados

SVR e MLP		
Algoritmo	EAM	REQM
SVR	0.0438	0.0607
MLP	0.0465	0.0465

6 CONCLUSÃO

A manutenção preditiva nas indústrias garante uma redução de custos e aumento na confiabilidade. Os alarmes de temperatura pelo equipamento do *hotbox* geram paradas não programadas na logística das ferrovias, por isso há relevância na detecção de altas temperaturas antes de um evento desse tipo acontecer. O campo de pesquisa para a predição de falhas nos componentes do trem é de grande utilidade, porém com quantidade de fatores que podem influenciar as falha em rolamentos, é um problema de alta complexidade.

Este documento apresentou uma aplicação de técnicas de aprendizado de máquina visando resolver problemas de paradas de trem por alarmes de temperatura em rolamentos. Algumas outras pesquisas foram realizadas para solucionar o problema com outras modelagens diferentes, como os seguintes trabalhos já citados (LUO et al., 2017), (LI; PARIKH; HE, 2014) e (PETERS, 2017). A maioria das soluções busca uma classificação para a predição de um alarme no futuro. Neste trabalho, o objetivo foi solucionar esse problema com uma técnica de regressão para a predição dessas temperaturas em etapas a frente com uma série temporal, uma vantagem em relação a classificações, é a possibilidade de disponibilizar o gráfico de temperatura para um especialista analisar o comportamento também. Um outro motivo para a escolha da metodologia, foi a questão da base de dados, que seria reduzida para poucos dados se fosse um problema de classificação.

A predição de séries temporais é caracterizada por ser um problema no qual é difícil encontrar soluções ótimas. Pois, geralmente na maioria dos problemas do mundo reais, as séries temporais não são estacionárias. Além disso, há diversos fatores/características que podem influenciar nos resultados das predições, o que causa uma grande quantidade de *outliers*. Outro problema é a falta de dados para a construção da sequência, a qualidade dos dados é essencial para a construção de um modelo eficaz, de modo que não há como exigir dos algoritmos a solução, se não há relação entre as características e o resultado.

Neste trabalho, a base de dados tinha bastante ruído e falta de informações, além de pouco exemplos com temperaturas mais altas. Porém, mesmo com as limitações, foi possível treinar os algoritmos de aprendizado de máquina selecionados para a predição de temperatura. Assim, pode se concluir através do resultados apresentados que a predição de alarmes de temperatura com um histórico e as características sugeridas são viáveis para a detecção com antecedência de um alarme.

Assim, a detecção com antecedência do crescimento de temperatura é uma solução para detectar possíveis falhas e evitar a parada de ferrovias.

6.1 Trabalhos Futuros

Durante a realização desse estudo, verificou-se que há a possibilidade de várias melhorias e estudos. Segue algumas delas:

- Pretende-se estudar a utilização de outros equipamentos do *Wayside*, como o *Railbam*, para a detecção de falhas no rolamento;
- A construção de uma modelagem diferente para a resolução do problema, como a construção de um classificador para verificar quais rolamentos irão alarmar ou não;
- A construção de uma base de dados mais robusta e com menos ruído, além de fazer testes com a base inteira e com amostras aleatórias;
- O estudo da viabilidade da implementação de um sistema automatizado com a técnica estudada.

Referências

- ALEXJ.SMOLA; SCHOLKOPF, B. A tutorial on support vector regression. **Statistics and Computing**, v. 14, p. 199–222, 2003.
- BATISTA, G. E. de A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. 2003. Tese.(Doutorado em Ciências) - ICMC - USP, São Carlos. p. 3.
- CARVALHO, A. C. de. **Inteligência Artificial. Uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011. 2-51 p.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, v. 2, p. 27:1–27:27, 2011. Software disponível em <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- CNT. **Pesquisa CNT de Ferrovias 2015**. 2015. Confederação Nacional de Transporte.
- EFCARAJÁS, D. V. G. 2018.
- FURTADO, P. M. C. **Curso de Especialização em Transporte Ferroviário de Cargas**. 2005.
- GAMBOGI, J. A. **Aplicação de redes neurais na tomada de decisão no mercado de ações**. 2013. Dissertação (Mestrado em Engenharia Elétrica). Universidade de São Paulo.
- HAYKIN, S. **Redes Neurais Princípios e práticas**. [S.l.]: bookman, 2001. 28-74 p.
- JOLLIFFE, I. Principal component analysis. In: **International encyclopedia of statistical science**. [S.l.]: Springer, 2011. p. 1094–1096.
- KARDEC, A.; NASCIF, J. **Manutenção função Estratégica**. 3. ed. [S.l.]: Qualitymark, 2001.
- KEOGH, S. K. E. On the need for time series data mining benchmarks: A survey and empirical demonstration. **Data Mining and Knowledge discovery**, v. 7, 2003.
- LI, H.; PARIKH, D.; HE, Q. Improving rail network velocity: A machine learning approach to predictive maintenance. **Elsivier**, 2014. United States. Disponível em: <https://pdfs.semanticscholar.org/eb6c/ffd6308f9fe479d59807efd7d1727c0aabe8.pdf>.
- LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **RITA - Revista de Informática Teórica e Prática**, XIV, n. 2, 2007.
- LUCENA, A. H. B. da F. **Uma Abordagem Híbrida para a Predição de Séries Temporais**. 2016. Dissertação (Dissertação em Sistemas de Informação). Universidade Federal de Pernambuco.
- LUO, C. et al. Lstm-based temperature prediction for hot-axles of locomotives. **ITM Web of Conferences.**, 2017. China. Disponível em: https://www.itm-conferences.org/articles/itmconf/pdf/2017/04/itmconf_ita2017_01013.pdf.
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biology**, v. 52, n. 1, 1990.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PETERS, M. Early warnings for failing train axle bearings based on temperature. **Annual conference of the prognostics and health management society**, 2017. Disponível em: <https://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2017/phmc_17_041.pdf>.

ROMAN, J.; JAMEEL, A. **Backpropagation and Recurrent Neural Networks in Financial Analysis of Multiple Stock Market Returns**. 1996.

SILVA, L. P. V. e José Viterbo Filho e Igor Machado Coelho e João Marcos Meirelles da. **Uma comparação de técnicas de regressão para a previsão de consumo de energia residencial no cenário nacional**. 2017.

VAPNIK, V. N. **Statistical Learning Theory**. [S.l.]: John Wiley and Sons, 1998.

VERDÉRIO, A. **Sobre o uso de regressão por vetores suporte para a construção de modelos em um método de região de confiança sem derivadas**. 2015.

Y.RADHIKA; M.SHASHI. Atmospheric temperature prediction using support vector machines. **International Journal of Computer Theory and Engineering**, v. 1, n. 1, 2009.

Apêndices

APÊNDICE A – Visualização de Dados

Figura 22 – Gráfico de dispersão

