Universidade Federal Fluminense

LUIS AUGUSTO LÜDERS MEZA

Modelo de Aprendizado de Máquina Combinado com Algoritmo de Exame de Partículas para a Emissão de Alerta de Inundações no Município de Volta Redonda.

VOLTA REDONDA

2021

LUIS AUGUSTO LÜDERS MEZA

Modelo de Aprendizado de Máquina Combinado com Algoritmo de Exame de Partículas para a Emissão de Alerta de Inundações no Município de Volta Redonda.

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Orientador:

Eliane Da Silva Christo

Coorientador:

Kelly Alonso Costa

Universidade Federal Fluminense

VOLTA REDONDA

2021

Ficha catalográfica automática - SDC/BEM
Gerada com informações fornecidas pelo autor

M617m Meza, Luis Augusto Lüders
Modelo de Aprendizado de Máquina Combinado com Algoritmo de Exame de Partículas para a Emissão de Alerta de Inundações no Município de Volta Redonda. / Luis Augusto Lüders Meza ; Eliane Da Silva Christo, orientadora ; Kelly Alonso Costa, coorientadora. Volta Redonda, 2021.
108 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Volta Redonda, 2021.

DOI: http://dx.doi.org/10.22409/PGMEC.2021.m.02421337976

1. Aprendizado de Máquina. 2. Modelo Linear Generalizado. 3. Otimização por Exame de Partícula. 4. Previsão de Vazão. 5. Produção intelectual. I. Christo, Eliane Da Silva, orientadora. II. Costa, Kelly Alonso, coorientadora. III. Universidade Federal Fluminense. Escola de Engenharia Industrial e Metalúrgica de Volta Redonda. IV. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

Modelo de Aprendizado de Máquina Combinado com Algoritmo de Exame de Partículas para a Emissão de Alerta de Inundações no Município de Volta Redonda.

Luis Augusto Lüders Meza

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Modelagem Computacional.

Aprovada por:

Prof. Eliane da Silva Christo, D.Sc. / MCCT-UFF

Prof. Kelly Alonso Costa, D.Sc. / PPGEP-UFF

Prof. Tiago Araujo Neves, D.Sc / MCCT-UFF

Prof. Weslley Luiz da Silva Assis, D.Sc. / MCCT-UFF

Prof. Rafael Alves Bonfim de Queiroz, D.Sc. / DECOM-UFOP

Prof. Leonardo Goliatt da Fonseca, D.Sc. / MAC-UFJF

Volta Redonda, 11 de Agosto de 2021.

Dedicatória. Para toda a minha família

Agradecimentos

A todos os professores do MCCT pela dedicação e esforço em transmitir conhecimento.

A minha esposa que sempre compreendeu minhas ausências e tanto me incentivou nesta caminhada.

A todos os demais que de forma indireta contribuíram para a realização deste trabalho.

A Prof. Eliane Da Silva Christo MCCT/UFF pela paciência e atenção.

E por último um especial agradecimento ao Prof. Diomar César Lobão MCCT/UFF por seus conselhos e disponibilidade durante todo o curso.

Resumo

Com o aumento da população e o crescimento desordenado das cidades as enchentes tornam-se cada vez mais frequentes, principalmente em regiões que margeiam os rios. Pensando nisso, esse trabalho baseia-se em um estudo de Série Temporal com o objetivo de prever inundações na cidade de Volta Redonda (VR), às margens do rio Paraíba do Sul. Analisam-se os dados da vazão horária, através dos modelos de Aprendizado de Máquina (do inglês *Machine Learning*, ML), como Regressão Linear e o Modelo Linear Generalizado (do inglês *Generalized Linear Model*, GLM), combinado com a Otimização por Enxame de Partículas (do inglês *Particle Swarm Optimization*, PSO) para antecipar estas inundações, alcançando o menor Erro Percentual Absoluto Médio (do inglês *Mean Absolute Percentage Error*, MAPE) de 0,1758% em relação ao valor real. O algoritmo de ML com Regressão Linear faz parte do procedimento de seleção das variáveis de entrada, enquanto o ML com GLM é o algoritmo dedicado para realizar os treinamentos e consequentemente as previsões. O PSO é utilizado como um ajuste fino dos parâmetros do GLM, buscando melhorar as previsões dentro de uma faixa que corresponde a $\pm 10\%$ dos parâmetros do GLM, sendo estes inclusive utilizados como valores iniciais do PSO. Diferente de outros estudos com Série Temporal que utilizam apenas os valores da série para estabelecer as entradas. Neste estudo incluiu-se como entrada o histórico de vazões de três pontos distintos, sendo dois localizados a montante de VR e um situado na própria cidade. E por fim, desenvolve-se um aplicativo *web* que de maneira confiável e em tempo hábil, alerta antecipadamente sobre risco de inundações do rio Paraíba do Sul na cidade de Volta Redonda.

Abstract

The increase in population and the disorderly growth of cities, floods become more and more frequent, especially in regions bordering rivers. With this in mind, a work based on a Time Series study was developed with the objective of forecasting floods in the city of Volta Redonda (VR) on the banks of the Paraíba do Sul River. The hourly flow data was analyzed using models of Machine Learning (ML), as Linear Regression and Generalized Linear Model (GLM), together with Particle Swarm Optimization (PSO) to anticipate these floods, reaching the lowest Percentage Absolute Error (MAPE) of 0,1758% in relation to the real value. The ML algorithm with Linear Regression is part of the input variables selection procedure, while the ML with GLM is the algorithm dedicated to perform the training and consequently the predictions. The PSO is used as a fine adjustment of the GLM parameters, seeking to improve the forecasts within a range that corresponds to $\pm 10\%$ of the GLM parameters, which are even used as initial values of the PSO. Unlike other time series studies that use only series values to establish inputs. In this study, the flow series of three different points was included as input, being two located upstream of VR and one located in the city. And finally, a web application is developed that reliably and in a timely manner, warns in advance of the risk of flooding in the Paraíba do Sul river in the city of Volta Redonda.

Keywords: Machine Learning; Generalized Linear Model; Artificial Intelligence; Particle Swarm Optimization; Flow Forecast.

Palavras-chave

1. Aprendizado de Máquina
2. Modelo Linear Generalizado
3. Inteligência Artificial
4. Otimização por Exame de Partícula
5. Previsão de Vazão

Lista de Acrônimos

ADF	:	*Augmented* Dickey-Fuller
AIC	:	*Akaike's Information Criterion*
ANA	:	Agência Nacional de Águas
ANFIS	:	*Adaptative Network Fuzzy Inference System*
ANFIS-GP	:	*Adaptative Network Fuzzy Inference System with Grid Partition*
ANFIS-SC	:	*Adaptative Network Fuzzy Inference System with Subtractive Clustering*
App	:	Aplicação computacional
ARIMA	:	*Autoregressive Integrated Moving Average*
ARMA	:	*Autoregressive Moving Average*
AutoML	:	*Automatic Machine Learning*
BD	:	Banco de Dados
CAPES	:	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEEMDAN	:	*Complete Ensemble Empirical Mode Decomposition with Adaptive Noise*
CoxPH	:	*Cox Proportional Hazards*
CPU	:	*Central Processing Unit*
CSV	:	*Comma Separated Values*
DF	:	Dickey-Fuller
DRF	:	*Distributed Random Forest*
DWT	:	*Discrete Wavelet Transform*
EBT	:	*Empirical Bayes Threshold*
FIV	:	Fator de Inflação da Variância
GAM	:	*Generalized Additive Models*
GBM	:	*Gradient Boosting Machine*
GLM	:	*Generalized Linear Models*
GLRM	:	*Generalized Low Rank Models*
GUI	:	*Graphical User Interface*
HTML	:	*HyperText Markup Language*
HTTP	:	*HyperText Transfer Protocol*

Lista de Acrônimos

HWT	:	Holt-Winters-Taylor
IA	:	Inteligência Artificial
IMF	:	*Intrinsic Mode Functions*
IoT	:	*Internet of Things*
LM	:	*Multi-task ElasticNet Linear Model*
MAE	:	*Mean Absolute Error*
MAPE	:	*Mean Absolute Percentage Error*
MCCT	:	Modelagem Computacional em Ciência e Tecnologia
MEC	:	Ministério da Educação
ML	:	*Machine Learning*
MM	:	*Multi Models*
MSE	:	*Mean Square Error*
NSE	:	*Nash–Sutcliffe Model Efficiency Coefficient*
OSI	:	*Open Source Initiative*
PCA	:	*Principal Component Analysis*
PSO	:	*Particle Swarm Optimization*
R	:	Coeficiente de Correlação de Pearson
RAM	:	*Random Access Memory*
RF	:	*Random Forest*
RMSE	:	*Root Mean Square Error*
RMSLE	:	*Root Mean Square Logarithmic Error*
RNA	:	Redes Neurais Artificiais
R^2	:	Coeficiente de Determinação
SARIMA	:	*Seasonal Autoregressive Integrated Moving Average*
SNIRH	:	Sistema Nacional de Informações sobre Recursos Hídricos
SVM	:	*Support Vector Machine*
SVR	:	*Support Vector Regression*
UFF	:	Universidade Federal Fluminense
UML	:	*Unified Modeling Language*

Lista de Acrônimos

URL	:	*Uniform Resource Locator*
VR	:	Volta Redonda
XML	:	*eXtensible Markup Language*
WA	:	*Wavelet*

Sumário

Lista de Figuras

Lista de Quadros

Lista de Tabelas

Capítulo 1

Introdução

Neste capítulo, descrevem-se os objetivos que motivam este estudo, apresentam-se situações e partes envolvidas. Com o intuito de esclarecer o problema enfrentado e preparar para o entendimento de todo o conteúdo. E a Figura 1.1 ilustra os principais assuntos discutidos aqui. No final um breve relato dos próximos capítulos abordados neste trabalho.

1.1 Considerações Iniciais

Previsões são necessárias em muitas situações: decidir se construirá outra usina de geração de energia nos próximos cinco anos requer previsões de demanda futura; definir o número de pessoas em uma equipe de um *call center* na próxima semana requer previsões de volumes de chamadas; estabelecer o tamanho de um estoque requer previsões de demandas dos itens. As previsões podem ser necessárias com vários anos de antecedência (para o caso de investimentos de capital) ou apenas alguns minutos antes (para roteamento de telecomunicações). Quaisquer que sejam as circunstâncias ou horizontes de tempo envolvidos, a previsão é uma ajuda importante para o planejamento eficaz e eficiente [24].

A previsão de inundação se faz importante, pois, no Brasil, todos os anos registram-se vários casos de inundações e enchentes, mostrando a necessidade de haver estudos prévios para mitigar os riscos causados por esses acontecimentos. Nesse caso, o monitoramento e a previsão das vazões dos rios tornam-se importantes para colaborar com esse processo de prevenção de acidentes e desastres causados pelas cheias dos rios [31].

Em [17], comenta-se que segundo as Nações Unidas, o Brasil possui um dos maiores índices envolvendo casos de inundações no mundo, em decorrência disso compromete-se

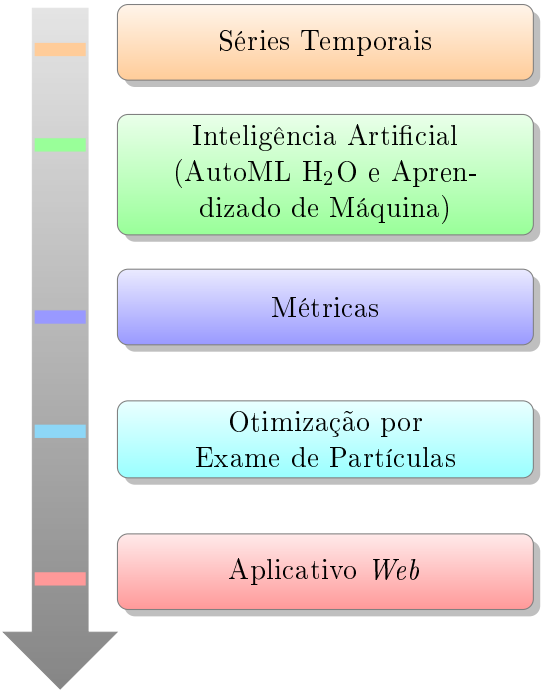

Figura 1.1: Temas principais abortados no trabalho.
Fonte: [29] e Autor

cerca de 3% do seu PIB nesses acontecimentos. Segundo [15] o país registra muitos casos de inundações em todo o seu território que atingem diretamente a população, ocasionando mortes e problemas econômicos, com isso enfatizam a necessidade de monitoramentos para diminuírem os impactos dessas ocorrências [31].

Da mesma forma, a bacia hidrográfica do rio Paraíba do Sul também sofre com os episódios de inundações, prejudicando a população que vive as suas margens e de todos que direta e/ou indiretamente dependem do seu uso. Essa bacia está localizada nos estados de São Paulo, Rio de Janeiro e Minas Gerais possuindo grande relevância nessa área, pois concentram-se entre os principais polos industriais e populacionais do Brasil [31].

As claras evidências de que o rio apresenta ocorrências de inundações e enchentes revelam a necessidade de análises mais profundas de monitoramento e previsão desses tipos de eventos. Esse fato motivou um estudo mais detalhado das possibilidades de inundações para a cidade de Volta Redonda no sul fluminense, já que o município se desenvolveu as margens do rio Paraíba do Sul. A pesquisa baseiam-se nas previsões dos dados de vazão horária das águas do rio e no desenvolvimento de um sistema de alerta para antecipar as possíveis ocorrências de cheias [31].

Para este caso, a modelagem e previsão de séries temporais baseia-se no emprego de algoritmos de Aprendizado de Máquina, considerando-se o bom desempenho em previsão

destes algoritmos e buscando-se uma comparação com modelos estatísticos, mas precisamente o modelo Holt Winters Taylor (HWT, [31]). Opta-se então, por analisar a série temporal da vazão horária da estação fluviométrica de Volta Redonda em conjunto com outras duas estações fluviométricas. São elas, as estações do Funil Jusante 1 e Funil Jusante 2, assim com dados temporais destas estações e os da estação de VR, definiu-se como objetivo a previsão de inundações na cidade Volta Redonda, o que possibilita um tempo hábil para tomar medidas que amenizem os seus efeitos.

O Quadro 1.1 relaciona as estações de medição com seus respectivos códigos no repositório do Sistema Nacional de Informações sobre Recursos Hídricos (SNIRH), de onde os dados são extraídos [4].

Estação Fluviométrica	Código SNIRH
UHE FUNIL JUSANTE 1	58242000
UHE FUNIL JUSANTE 2	58300000
UEL SANTA CECÍLIA VR	58305000

Quadro 1.1: Código SNIRH da estação fluviométrica.
Fonte: [4]

Ao obter a previsão da vazão torna-se necessário relacioná-la com a cota (nível d'água) do rio para acompanhar o fluxo dele. Assim, [28] comenta que a metodologia de curva-chave é usualmente empregada para relacionar vazão e a cota do rio, sendo a precisão dessa análise importante para avaliação dos recursos hídricos e antecipação de enchentes [31].

1.2 Objetivo

Prever inundações na cidade de Volta Redonda - RJ, através dos dados das vazões horária do rio Paraíba do Sul, permitindo reduzir os danos causados pelas enchentes na cidade.

Objetivos Específicos:

- Utilizar ML combinado com PSO para prever a vazão horária do Rio Paraíba do Sul na cidade de Volta Redonda, com 1 hora de antecedência;

- Desenvolver um aplicativo *web* para alertar inundações do rio Paraíba do Sul na cidade de Volta Redonda.

1.3 Descrições dos Capítulos

As demais partes deste trabalho estão organizadas da seguinte maneira:

- No Capítulo 2 abordam-se os principais conceitos para o entendimento do desenvolvimento do trabalho;
- No Capítulo 3 apresenta-se uma revisão bibliográfica do tema;
- Já no Capítulo 4 descreve-se a metodologia da pesquisa e como são implementadas as estratégias de análises;
- No Capítulo 5 expõem-se os resultados obtidos;
- No Capítulo 6 apresenta-se o aplicativo *web* desenvolvido para realizar previsões;
- O Capítulo 7 contém as conclusões do trabalho e as sugestões de melhorias;
- No Apêndice A têm-se os principais algoritmos utilizados;
- Por último no Apêndice B apresentam-se os *Models* que gerenciam o Banco de Dados.

Capítulo 2

Fundamentação Teórica

Neste capítulo, abordam-se as características básicas das Séries Temporais, da Regressão Linear e do GLM. Depois mencionam-se sobre as métricas responsáveis por avaliar o desempenho das previsões. Em seguida, apresentam-se fundamentos da Inteligência Artificia e conceituam-se o Aprendizado de Máquina e a técnica de otimização denominada Otimização por Enxame de Partículas. E por fim, abordam-se a linguagem de programação *Python* e a plataforma AutoML H_2O.

2.1 Séries Temporais

Uma série temporal é um conjunto de amostras tomadas no tempo e muitos destes dados aparecem como séries temporais: uma sequência mensal da quantidade de mercadorias enviadas de uma fábrica, uma série semanal do número de acidentes de aviação, observações horárias feitas sobre o rendimento de um processo químico e assim por diante. Existem vários exemplos de séries temporais em áreas como economia, negócios, engenharia, ciências naturais (especialmente geofísica e meteorologia) e ciências sociais. Exemplos de dados do tipo são exibidos como gráfico de série temporal na Figura 2.1 [11].

Uma característica de uma série temporal é que, normalmente, as observações adjacentes são dependentes. A natureza dessa dependência entre as observações de uma série temporal é de considerável interesse prático. A análise de séries temporais preocupam-se com técnicas para a análise dessa dependência. Para isso desenvolvem-se modelos estocásticos e dinâmicos para dados de séries temporais e usam-se tais modelos em áreas importantes de aplicação [11].

De acordo com [37], são exemplos de séries temporais:

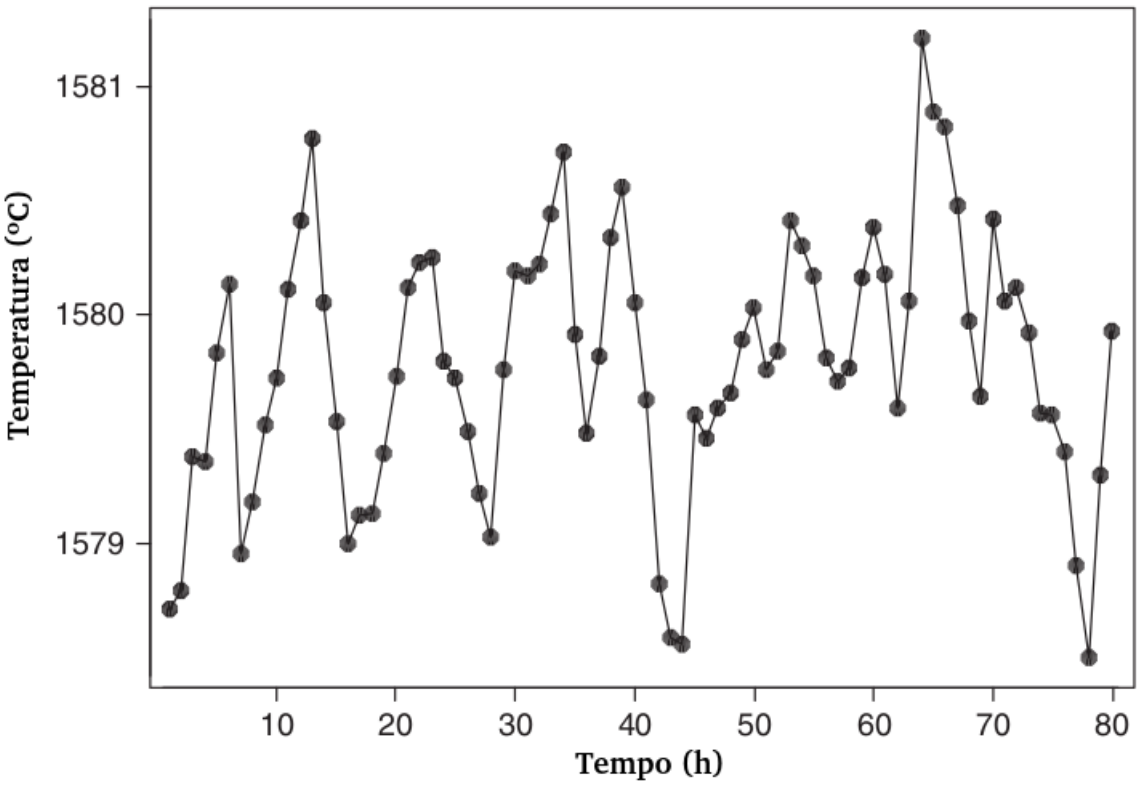

Figura 2.1: Temperatura por hora de um forno de cerâmica.
Fonte: [10]

(i) valores diários de poluição na cidade de São Paulo;

(ii) valores mensais de temperatura na cidade de Cananéia-SP;

(iii) índices diários da Bolsa de Valores de São Paulo;

(iv) precipitação atmosférica anual na cidade de Fortaleza;

(v) número médio anual de manchas solares;

(vi) registro de marés no porto de Santos.

Nos exemplos (i) - (v) têm-se séries discretas, enquanto (vi) é um exemplo de uma série contínua. Muitas vezes, obtém-se uma série temporal discreta através da amostragem de uma série temporal contínua em intervalos de tempos iguais, Δt. Assim, para analisar a série (vi) necessita-se amostrá-la (em intervalos de tempo de uma hora, por exemplo), convertendo a série contínua, observada no intervalo $[0, T]$, em uma série discreta com N pontos, onde $N = \dfrac{T}{\Delta t}$. Em outros casos, como para as séries (v) ou (vi), obtém-se o valor da série num dado instante acumulando (ou agregando) valores em intervalos de tempos iguais [37].

Existem dois enfoques na análise de séries temporais e em ambos constroem-se modelos com aplicações específicas para as séries. No primeiro enfoque, analisa-se o domínio temporal e propõem-se modelos paramétricos (com um número finito de parâmetros). No segundo, analisa-se o domínio de frequência e propõem-se modelos não-paramétricos [37].

No domínio de frequências tem-se a análise espectral com inúmeras aplicações em ciências físicas e engenharia, e que consiste em decompor a série dada em componentes de frequência, onde a existência dos espectro é a característica fundamental [37].

Ao ajustar modelos dinâmicos, uma análise teórica às vezes pode dizer não apenas a forma apropriada para o modelo, mas também pode fornecer boas estimativas dos valores numéricos de seus parâmetros. Esses valores podem ser verificados posteriormente pela análise dos dados [11].

A Figura 2.2 resume a abordagem iterativa para criar modelos de previsão e controle. E suas etapas são descritas abaixo [11]:

(a) A partir da interação entre teoria e prática, estabelece-se um grupo com modelos úteis para solucionar o problema em questão;

(b) Como esse grupo de modelos acaba sendo muito extenso para ser convenientemente ajustados diretamente aos dados, desenvolvem-se métodos aproximados para identificar os melhores modelos. Tais métodos de identificação de modelos empregam dados e conhecimento do sistema para sugerir os modelos que podem ser testados experimentalmente. Além disso, o processo de identificação pode ser usado para gerar estimativas preliminares aproximadas dos parâmetros no modelo;

(c) Ajusta-se o modelo experimental aos dados e seus parâmetros estimados. As estimativas aproximadas obtidas durante o estágio de identificação, agora podem ser usadas como valores iniciais em métodos iterativos mais refinados para estimar melhor os parâmetros;

(d) Aplica-se a avaliação do modelo com o objetivo de descobrir uma possível falta de ajuste e diagnosticar a causa. Se nenhum desvio for detectado, o modelo estará pronto para uso. Se alguma inadequação for encontrada, o ciclo iterativo retorna para a etapa b) e é repetido até que uma representação adequada seja encontrada;

(e) Realizam-se previsões e controle utilizando o modelo e os parâmetros determinados.

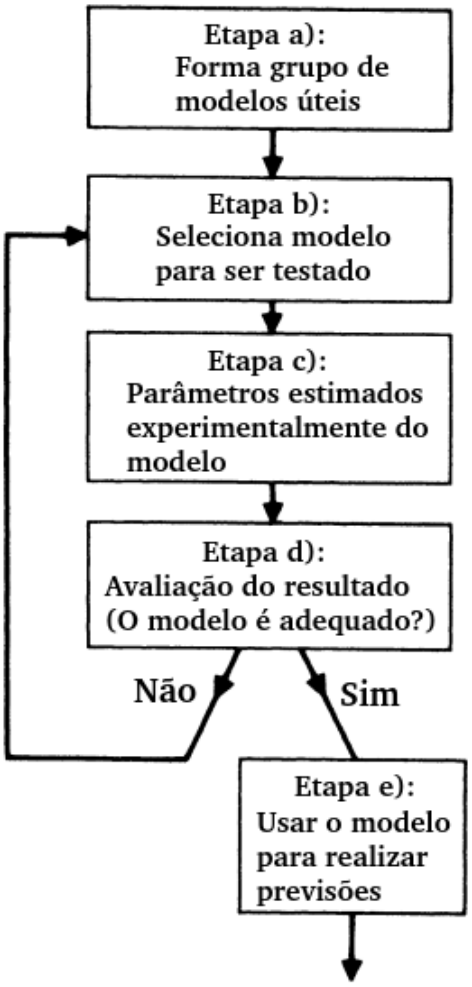

Figura 2.2: Etapas para determinar um modelo.
Fonte: [11]

2.1.1 Notação

Ao medir a temperatura do ar, de dado local, durante 24 horas; pode-se obter um gráfico semelhante ao da Figura 2.3 [37].

Designa-se por $Z(t)$ a temperatura no instante t (dado em horas, por exemplo). Nota-se que para dois dias diferentes, obtém-se duas curvas que não são, em geral, as mesmas. Estas curvas são chamadas trajetórias do processo físico que está sendo observado e este (o processo estocástico) nada mais é do que o conjunto de todas as possíveis trajetórias que podem ser observadas. Cada trajetória é também chamada de série temporal ou função amostral. Designando-se por $Z^{(1)}(15)$ o valor da temperatura no instante $t = 15$, para a primeira trajetória (primeiro dia de observação), tem-se um número real, para o segundo dia tem-se outro número real, $Z^{(2)}(15)$. Em geral, denota-se uma trajetória

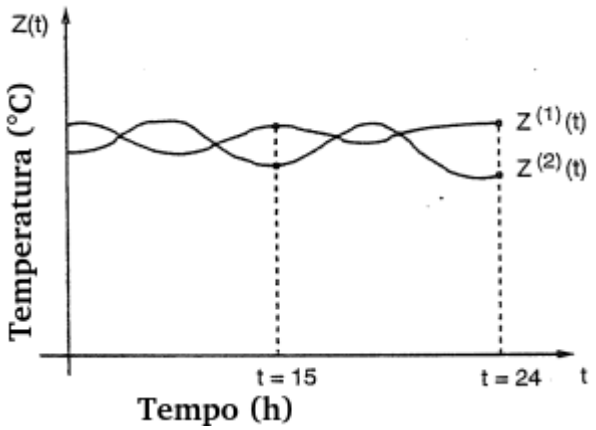

Figura 2.3: Temperatura do ar, de dado local, durante 24h.
Fonte: [37]

qualquer por $Z^{(j)}(t)$. Para cada t fixo, têm-se os valores da variável aleatória $Z(t)$, que terá certa distribuição de probabilidades [37].

Na realidade, a série temporal é uma parte de uma trajetória, dentre muitas que podem ser observadas. Em algumas situações (como em Oceanografia, por exemplo), quando têm-se dados experimentais, é possível observar algumas trajetórias do processo sob consideração, mas na maioria dos casos (como em Economia ou Astronomia), quando não é possível fazer experimentação, tem-se uma só trajetória para análise [37].

Tem-se o referido parâmetro t como sendo o tempo, mas a série $Z(t)$ poderá ser função de algum outro parâmetro físico, como espaço ou volume. De modo bastante geral, uma série temporal poderá ser um vetor $Z(t)$, de ordem $r \times 1$, onde, por sua vez, t é um vetor $p \times 1$. Por exemplo, considere a série, apresentada pela Equação 2.1 [37].

$$Z(t) = [Z_1(t), Z_2(t), Z_3(t)]', \tag{2.1}$$

onde as três componentes denotam, respectivamente, a altura, a temperatura e a pressão de um ponto do oceano e $t = (tempo, latitude, longitude)$. Diz-se que a série é multivariada ($r = 3$) e multidimensional ($p = 3$). Como outro exemplo, considera-se $Z(t)$ como sendo o número de acidentes ocorridos em rodovias do Estado de São Paulo, por mês. Aqui, $r = 1$ e $p = 2$, com $t = (mês, rodovia)$ [37].

2.1.2 Objetivos da Análise de Séries Temporais

De acordo com [37], obtida a série temporal $Z(t_1), ..., Z(t_n)$, observada nos instantes $t_1, ..., t_n$, pode-se estar interessado em:

(a) investigar o mecanismo gerador da série temporal; por exemplo, analisando uma série de alturas de ondas, pode-se querer saber como estas ondas são geradas;

(b) fazer previsões de valores futuros da série; estas podem ser a curto prazo, como para séries de vendas, produção ou estoque, ou a longo prazo, como para séries populacionais, de produtividade etc.;

(c) descrever apenas o comportamento da série; neste caso, a construção do gráfico, a verificação da existência de tendências, ciclos e variações sazonais, a construção de histogramas e diagramas de dispersão etc., podem ser ferramentas úteis;

(d) procurar a periodicidade relevante nos dados; aqui a análise espectral, mencionada anteriormente, pode ser de grande utilidade.

Em todos os casos, constroem-se modelos probabilísticos ou modelos estocásticos no domínio temporal ou de frequências. Estes modelos devem ser simples e parcimoniosos (no sentido que o número de parâmetros envolvidos deve ser o menor possível) e, se possível, sua utilização não deve apresentar dificuldades às pessoas interessadas em manipulá-los [37].

Muitas situações em ciências físicas, engenharia, ciências biológicas e humanas envolvem o conceito de sistema dinâmico, caracterizado por uma série de entrada $X(t)$, uma série de saída $Z(t)$ e uma função de transferência $v(t)$ (Figura 2.4) [37].

De particular importância são os sistemas lineares, os quais relacionam-se a saída com a entrada através de um funcional linear envolvendo $v(t)$. Um exemplo típico é a Equação 2.2 [37].

$$Z(t) = \sum_{\tau=0}^{\infty} v(\tau) X(t - \tau) \tag{2.2}$$

Segundo [11], o uso das Séries Temporais e de Modelos Dinâmicos apresentam cinco áreas de interesse:

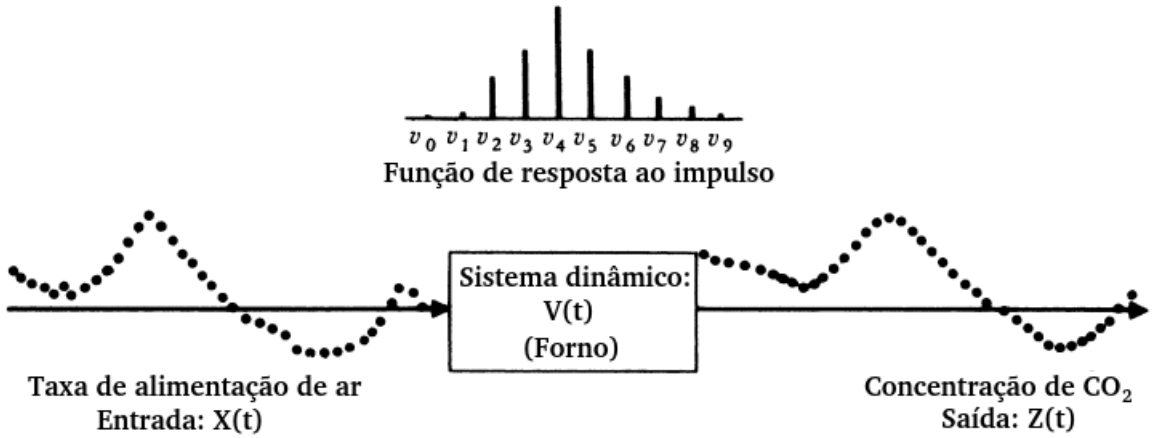

Figura 2.4: Séries de entrada e saída em relação a um sistema dinâmico.
Fonte: [11]

(a) Previsão: A previsão de valores futuros de uma série temporal $Z(t)$, a partir de valores atuais e passados $X(t)$;

(b) Função de transferência: A determinação da função de transferência $v(t)$ de um sistema a partir da entrada $X(t)$ e saída $Z(t)$, mostrando o efeito na saída do sistema para série de entrada;

(c) Variações: Simular as variáveis de entrada em modelos de funções de transferência para representar e avaliar os efeitos de eventos incomuns no comportamento de uma série temporal;

(d) Multivariado: Avaliar a inter-relações entre várias variáveis de interesse da série temporal, e a determinação de modelos dinâmicos multivariados para representar essas relações conjuntas entre as variáveis ao longo do tempo;

(e) Controle: Controlar a série de saída $Z(t)$ de tal forma que os desvios potenciais da saída do sistema, de um alvo desejado, possam ser compensados pelo ajuste dos valores da série de entrada $X(t)$.

2.1.3 Estacionariedade

A base para qualquer análise de série temporal é a série temporal estacionária. É essencialmente para séries temporais estacionárias que pode-se desenvolver modelos e previsões. No entanto, é a série cronológica não estacionária que é a mais interessante em muitas aplicações, especialmente em negócios e economia. Da mesma forma, em aplicações industriais, quando os processos não sofrem ação de controle, esperam-se que eles mostrem

comportamento não-estacionário simplesmente seguindo a segunda lei da termodinâmica. De fato, a estacionariedade na maioria dos processos é garantida apenas por ações de controle realizadas em intervalos regulares ou manutenção contínua dos componentes do sistema. Sem essas ações deliberadas para tornar os processos estacionários, a solução é deixar de lado os dados originais que exibem não estacionariedade, e focar por exemplo, nas variações das observações sucessivas [10].

Em muitas aplicações com dados não estacionários, as alterações do tempo $t-1$ para o tempo t da série temporal $\Delta Z(t)$, denotadas pela Equação 2.3, podem ser estacionárias. Se for esse o caso, pode-se modelar as mudanças, fazer previsões sobre os valores futuros dessas mudanças e, a partir do modelo das mudanças, criar modelos e criar previsões da série temporal não estacionária originalmente. Portanto, enquanto nas aplicações da vida real isso ocorre apenas em situações específicas, as séries temporais estacionárias desempenham um papel fundamental como base para a análise de séries temporais [10].

Uma série pode ser estacionária durante um período muito longo, como a série (vi) da seção 2.1, mas pode ser estacionária apenas em períodos muito curtos, mudando de nível e ou inclinação. Existem modelos estatísticos capazes de descreverem de maneira satisfatória as séries estacionárias e séries não estacionárias, mas que não apresentam comportamento explosivo. Chama-se este tipo de não-estacionariedade de homogêneo; a série pode ser estacionária, flutuando ao redor de um nível, por certo tempo, depois muda de nível e flutua ao redor de um novo nível e assim por diante, ou então muda de inclinação, ou ambas as coisas. A Figura 2.5 ilustra esta forma de não-estacionariedade [37].

Como a maioria dos procedimentos de análise estatística de séries temporais supõe que estas sejam estacionárias, será necessário transformar os dados originais, se estes não formam uma série estacionária. A transformação mais comum consiste em tomar diferenças sucessivas da série original, até se obter uma série estacionária. A primeira diferença de $Z(t)$ é definida pela Equação 2.3 [37].

$$\Delta Z(t) = Z(t) - Z(t-1), \tag{2.3}$$

a segunda diferença é a Equação 2.4

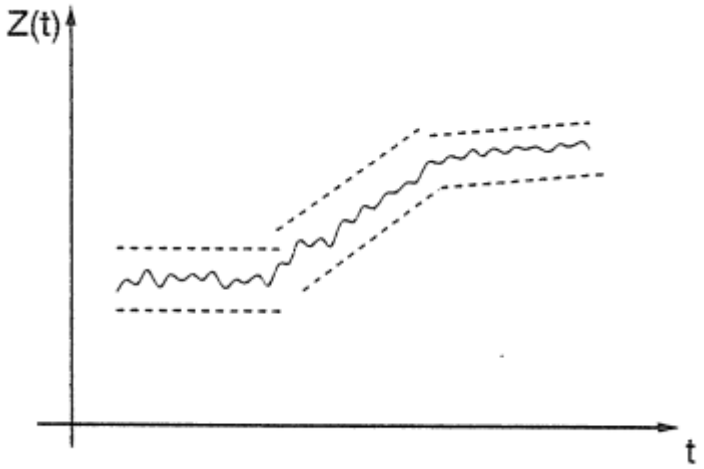

Figura 2.5: Série não-estacionária quanto ao nível e inclinação.
Fonte: [37]

$$\Delta^2 Z(t) = \Delta[\Delta Z(t)] = \Delta[Z(t) - Z(t-1)], \tag{2.4}$$

ou seja,

$$\Delta^2 Z(t) = Z(t) - 2Z(t-1) + Z(t-2), \tag{2.5}$$

De modo geral, a n-ésima diferença de $Z(t)$ é a Equação 2.6

$$\Delta^n Z(t) = \Delta[\Delta^{n-1} Z(t)]. \tag{2.6}$$

Em situações normais, será suficiente tomar uma ou duas diferenças para que a série se torne estacionária [37]. A Figura 2.6 apresenta a série da temperatura de uma planta piloto, acompanhada das suas três diferenças.

2.1.3.1 Método de Dickey-Fuller

De acordo com [46], desde de 1979 quando introduziu-se o teste Dickey-Fuller (DF), existe uma considerável atenção sobre a generalização de resíduos e o efeito da generalização sobre a inferência. E segundo [30], o teste Dickey-Fuller aumentado (do inglês

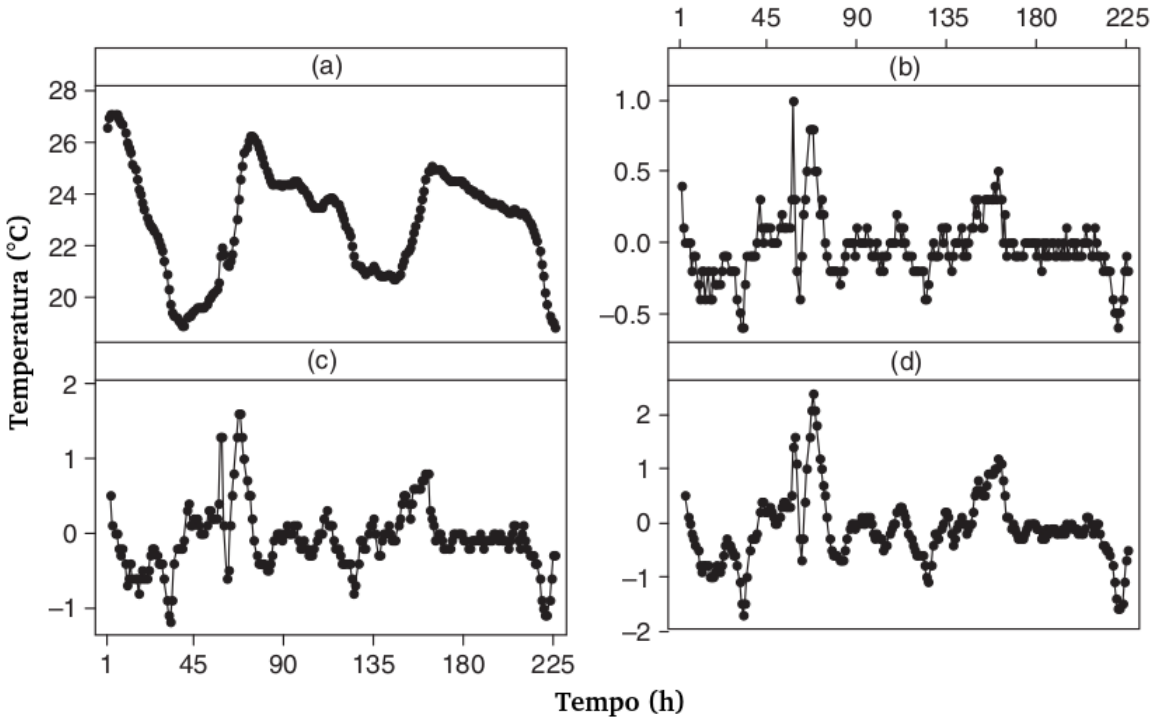

Figura 2.6: (a) Temperatura de uma planta piloto. (b) Primeira diferença. (c) Segunda diferença. (d) Terceira diferença.

Fonte: [10]

Augmented Dickey-Fuller, ADF) é o teste de raiz unitária mais usado em econometria. Por isso existem diversos estudos sobre o assunto, como [30], [39] e [46].

Conforme [12], desenvolveu-se o método de Dickey-Fuller (DF) para detectar a não estacionariedade das séries temporais. E através do modelo autorregressivo representado pela Equação 2.7 verifica-se o problema de raiz unitária, ou de não estacionariedade.

$$y_t = \rho y_{t-1} + u_t \tag{2.7}$$

onde u_t é o termo de erro estocástico também chamado de ruído branco. A raiz unitária existe se $\rho = 1$. E escreve-se a equação anterior de forma alternativa como as Equações 2.8 e 2.9 [12]:

$$\Delta y_t = (\rho - 1) y_{t-1} + u_t \tag{2.8}$$

$$= \gamma y_{t-1} + u_t \tag{2.9}$$

onde Δ é o operador de primeira diferença e $\gamma = (\rho - 1)$. As hipóteses a serem testadas são as Equações 2.10 e 2.11 [12]:

$$H_0 : \rho = 1 \Leftrightarrow H_0 : \gamma = 0 \tag{2.10}$$

$$H_1 : \rho < 1 \Leftrightarrow H_1 : \gamma < 0 \tag{2.11}$$

Se $\gamma = 0$ tem-se $\Delta y_t = (y_t - y_{t-1}) = u_t$, isto é, a primeira diferença da série temporal com caminho aleatório é estacionária, pois por hipótese, u_t é puramente aleatório. Portanto, se a hipótese nula for rejeitada, a série temporal não tem uma raiz unitária [12].

Para testar as hipóteses das Equações 2.10 e 2.11, estima-se a Equação 2.9 por mínimos quadrados e examina-se a estatística t para a hipótese de que $\gamma = 0$. Porém, se a hipótese nula é verdadeira, y_t segue um caminho aleatório e a estatística t não tem mais uma distribuição t. A estatística gerada deve então ser comparada com valores críticos especialmente construídos [12].

Dickey e Fuller (1979) tabularam os valores críticos através de simulações de Monte Carlo, e desenvolveram uma estatística τ (Tau) para testar formalmente o problema de raiz unitária. Se o valor absoluto da estatística τ calculado for maior que o valor absoluto tabulado por DF, aceita-se a hipótese nula, logo a série é não estacionária [12].

Para verificar a presença de raiz unitária em séries temporais o teste de Dickey-Fuller estima um dos seguintes modelos [12]:

(i) Modelo Puramente Aleatório, Equação 2.12:

$$\Delta y_t = \gamma y_{t-1} + u_t \tag{2.12}$$

(ii) Modelo com Constante, Equação 2.13:

$$\Delta y_t = \beta_0 + \gamma y_{t-1} + u_t \tag{2.13}$$

(iii) Modelo com Constante e Tendência, Equação 2.14:

$$\Delta y_t = \beta_0 + \beta_1 t + \gamma y_{t-1} + u_t \tag{2.14}$$

onde β_0 é o termo independente (constante de intersecção da reta) e β_1 é o coeficiente de tendência (inclinação da reta).

2.1.3.2 Método de Dickey-Fuller Aumentado

O teste ADF ou o teste de DF aumentado introduz um operador de defasagens para resolver o problema de autocorrelação do termo de erro u_t, tal como a Equação 2.15 [12]:

$$\Delta y_t = \beta_0 + \beta_1 t + \gamma y_{t-1} + \sum_{i=1}^{m} \Delta y_{t-i} + u_t \tag{2.15}$$

A equação acima pode ser estimada também sem a constante ou a tendência. A hipótese nula é mesma que a do teste DF e os valores críticos da estatística τ do teste DF são válidos para o teste ADF.

2.2 Regressão Linear

Segundo [19], a interpretação moderna da regressão estabelece que a análise de regressão diz respeito ao estudo da dependência de uma variável, a variável dependente, em relação a uma ou mais variáveis, as variáveis explanatórias, visando estimar ou prever o valor médio (da população) da primeira em termos dos valores conhecidos ou fixados (em amostragens repetidas) das segundas.

Como exemplo de função que representa uma Regressão Linear tem-se a Equação 2.16.

$$y = \beta_0 + \beta_1 x \tag{2.16}$$

onde, de acordo com [43] têm-se:

(a) x é a variável independente ou variável preditora;

(b) y é a variável dependente ou resposta;

(c) β_0 é a constante de intersecção da reta;

(d) β_1 é a inclinação da reta: esta é uma das grandezas mais importantes em qualquer análise de Regressão Linear. Um valor muito próximo de 0 indica pouco ou nenhum relacionamento. E grandes valores positivos ou negativos indicam grandes relacionamentos positivos ou negativos, respectivamente.

2.2.1 Modelo para Dados Linearmente Relacionados

Observam-se os pontos dados $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, assume-se que em função de x_i, gera-se cada y_i usando alguma reta como $y = \beta_0 + \beta_1 x$, e então adiciona-se algum ruído gaussiano. Formalmente, tem-se a Equação 2.17 [43]:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{2.17}$$

O ruído ϵ_i representa o fato de que os dados não se adaptam ao modelo perfeitamente. Então modela-se ϵ_i como sendo gaussiano: $\epsilon \sim N(0, \sigma^2)$. Observe-se ainda que a constante de intersecção β_0, a inclinação β_1 e a variância do ruído σ^2 são todos tratados como fixos (ou seja, determinísticos), mas quantidades desconhecidas [43].

2.2.2 Estimação por Mínimos Quadrados

Supondo os dados gerados $(x_1, y_1), ..., (x_n, y_n)$, então é possível encontrar uma reta para a qual a probabilidade dos dados é alta, através da solução do problema de otimização da Equação 2.18 [43]:

$$min_{\beta_0,\beta_1} = \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i)]^2 \tag{2.18}$$

O min_{β_0,β_1} significa "minimizar sobre β_0 e β_1". Isso é conhecido como o problema de Regressão Linear dos Mínimos Quadrados. Dado um conjunto de pontos, as Equações 2.19 e 2.21 são a solução [43]:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n} \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n} (\sum_{i=1}^{n} x_i)^2} \tag{2.19}$$

$$\hat{\beta}_1 = r \frac{S_y}{S_x} \tag{2.20}$$

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x} \tag{2.21}$$

onde $\overline{x}$, $\overline{y}$, S_x e S_y são as médias da amostra e os desvios padrão para os valores de x e os valores de y, respectivamente, e R é o coeficiente de correlação, definido pela Equação 2.22 [43]:

$$R = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i-\overline{x}}{S_x}\right)\left(\frac{y_i-\overline{y}}{S_y}\right) \tag{2.22}$$

Na sequência, em vez de apenas um único valor escalar x_i, tem-se um vetor $(x_{i1}, ..., x_{in})$ para cada ponto de dados i, o que denomina-se Regressão Linear Múltipla . Portanto, têm-se n pontos de dados (exatamente como antes), cada um com n variáveis ou recursos preditores diferentes. Assim, para prever y para cada ponto de dados como uma função linear das diferentes variáveis x tem-se a Equação 2.23 [43]:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} +, ..., +\beta_n x_{in} + \epsilon_i \tag{2.23}$$

Para facilitar, representam-se os dados de entrada na forma de matriz como X, uma matriz onde cada linha corresponde a um ponto de dados e cada coluna corresponde a uma variável de entrada. Como cada saída y_i é apenas um único ponto, então representa-se o conjunto resposta como um vetor de n-elementos denominado y. E expressa-se o modelo linear através da Equação 2.24 [43]:

$$y = \beta X + \epsilon \tag{2.24}$$

Onde β é um vetor de coeficientes de n-elemento, e ϵ é uma matriz de n-elementos onde cada elemento, ϵ_i, é normal com média 0 e variância σ^2. Observa-se, que não se escreve explicitamente um termo constante como β_0 de antes [43].

Para resolver este problema de otimização tem-se a Equação 2.25:

$$min_\beta = \sum_{i=1}^{n}(y_i - X_i\beta)^2 \tag{2.25}$$

Com isso, pode-se aplicar álgebra linear básica para resolver este problema e encontrar as estimativas ideais representadas pela Equação 2.26 [43]:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{2.26}$$

2.2.3 Detecção de Multicolinearidade

Para detectar a multicolinearidade entre as variáveis explicativas usa-se o fator de inflação da variância (FIV) [19].

O FIV mostra como a variância de um estimador é inflada pela presença da multicolinearidade. Considerando as variáveis explicativas X_1 e X_2 e a correlação ${R_{12}}^2$ aproxima-se de 1, o FIV aproxima-se do infinito. Ou seja, quando a colinearidade aumenta, a variância de um estimador aumenta e, no limite, pode tornar-se infinita. Se não houver colinearidade entre X_1 e X_2, o FIV será 1. Para calcular o FIV tem-se a Equação 2.27 [19].

$$FIV = \frac{1}{1 - {R_{12}}^2} \tag{2.27}$$

Alguns autores, usam o FIV como indicador de multicolinearidade. Quanto maior for o valor de FIV, mais "problemática" ou colinear será a variável X. Como regra prática, se o FIV de uma variável for maior que 10 (o que acontecerá se R^2 for maior que 0,90), considera-se essa variável como altamente colinear [19].

2.2.4 Teste de Significância dos Coeficientes

Em termos gerais, um teste de significância é um procedimento em que usam-se os resultados amostrais para verificar a veracidade ou a falsidade de uma hipótese nula. A ideia fundamental por trás dos testes de significância é a de um teste estatístico (estimador) e a distribuição amostral dessa estatística sob a hipótese nula. A decisão de aceitar ou rejeitar H_0 é tomada com base no valor do teste estatístico dos dados disponíveis [19].

Para mostrar a aplicação deste método em Regressões Lineares Múltiplas, avaliam-se as hipóteses dadas pelas Equações 2.28 e 2.29:

$$H_0 : \beta_j = 0, \qquad se \qquad p > 0,05 \tag{2.28}$$

$$H_1 : \beta_j \neq 0, \qquad se \qquad p < 0,05 \tag{2.29}$$

Se H_0 for rejeitada, conclui-se que a variável relacionada ao coeficiente β_j possui uma relação estatística relevante com a variável de saída, ou seja, deve-se utilizar esta variável na Regressão Linear. Caso contrário, descarta-se a mesma. Utiliza-se o teste estatístico que segue uma distribuição t com $n-2$ graus de liberdade [43].

2.2.4.1 Para Regressões Lineares Simples

Para realizar o teste estatístico t para o coeficiente de intersecção têm-se as Equações de 2.30 a 2.32 [43]:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n-2} \tag{2.30}$$

$$s_{\beta_0} = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \tag{2.31}$$

$$t_{\beta_0} = \frac{\hat{\beta}_0 - \beta_0}{s_{\beta_0}} \tag{2.32}$$

Onde:

- y_i: Valor real;
- $\hat{y}_i$: Valor previsto;
- n: Número de amostras;
- x_i: Variáveis de entrada;
- $\bar{x}$: Valor médio da variável de entrada x_i;
- t_{β_0}: Teste estatístico t para o coeficiente de intersecção;

- s_{β_0}: É desvio padrão estimado para o coeficiente de intersecção;
- $\hat{\sigma}^2$: É a estimativa da variância da previsão dada pelo erro quadrático médio.

Para calcular a significância do coeficiente linear apresentam-se as Equações 2.33 e 2.34 [43]:

$$s_{\beta_j} = \frac{\hat{\sigma}}{\sqrt{\sum_{i=1}^{n}(x_{ji} - \bar{x_{ji}})^2}} \tag{2.33}$$

$$t_{\beta_j} = \frac{\hat{\beta}_j - \beta_j}{s_{\beta_j}} \tag{2.34}$$

Já para regressões lineares múltiplas as variâncias dos elementos $\hat{\beta}$'s são expressas em termos dos elementos da inversa da matriz $X'X$, apresentada pela Equação 2.35. A inversa de $X'X$ vezes a constante σ^2 representa a matriz de covariância dos coeficientes da regressão $\hat{\beta}$. Os elementos da diagonal de $\sigma^2(X'X)^{-1}$ são as variâncias de $\hat{\beta}_j$ e os elementos que estão fora da diagonal da matriz são as covariâncias [36].

Com isso a estatística do teste t é dada pela Equação 2.37 e os graus de liberdade são iguais a $n - k - 1$, onde k é número de variáveis de entrada. Assim, tem-se que o desvio padrão estimado para β_j é dado pela Equação 2.36 [36]:

$$C_{jj} = (X'X)^{-1} \tag{2.35}$$

$$s_{\beta_j} = \sigma\sqrt{C_{jj}} \tag{2.36}$$

$$t_{\beta_j} = \frac{\hat{\beta}_j - \beta_j}{s_{\beta_j}} \tag{2.37}$$

2.2.5 Teste de Durbin Watson

De acordo com [19], um fator que prejudica o desempenho de modelos de previsão utilizando Regressão Linear é a autocorrelação dos resíduos. Assim a Equação 2.38 representa o teste d de Durbin-Watson, que permite identificar se existe ou não a correlação dos resíduos.

Ainda de acordo com [19], existem outros métodos de detecção de autocorrelação dos resíduos, mas nenhum deles pode ser considerado o melhor. Sendo o teste d de Durbin-Watson o mais famoso e o método indicado para defasagem de 1 período. Que coincide exatamente com o caso avaliado neste trabalho.

$$d = \frac{\sum_{t=2}^{t=n}(\hat{\epsilon}_t - \hat{\epsilon}_{t-1})^2}{\sum_{t=1}^{t=n}\hat{\epsilon}_t^2} \tag{2.38}$$

Segundo [19], pode-se reescrever a Equação 2.38 como sendo a Equação 2.39 [19].

$$d \simeq (1 - \hat{\rho}) \tag{2.39}$$

onde $\hat{\rho}$ é o coeficiente de autocorrelação de primeira ordem, um estimado de ρ sendo dado pela Equação 2.40. Já ρ é conhecido como coeficiente de autocovariância do esquema auto-regressivo de primeira ordem de geração dos resíduos. E é representado pela Equação 2.41 [19].

$$\hat{\rho} = \frac{\sum \hat{\epsilon}_t \hat{\epsilon}_{t-1}}{\sum \hat{\epsilon}_t^2} \tag{2.40}$$

$$\epsilon_t = \rho\epsilon_{t-1} + u_t \tag{2.41}$$

Como $-1 \leq \rho \leq 1$ tem-se que $0 \leq d \leq 4$, com isso pode-se utilizar a Figura 2.7 para determinar se existe ou não autocorrelação de resíduos.

Na Figura 2.7 consideram-se [19]:

(a) H_0: Ausência de autocorrelação positiva;

(b) H_0^*: Ausência de autocorrelação negativa;

(c) d_L: Limite inferior, consultar valor tabulado por Durbin e Watson;

(d) d_U: Limite superior, consultar valor tabulado por Durbin e Watson;

(e) Zona de Indecisão: não se pode concluir se há ou não autocorrelação de primeira ordem.

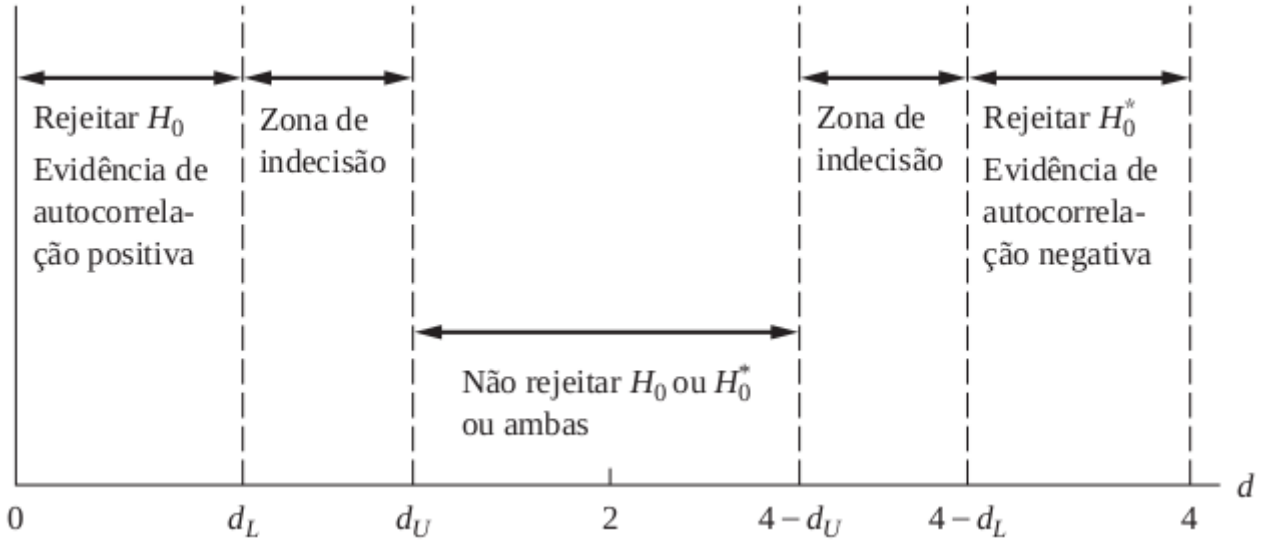

Figura 2.7: Estatística d de Durbin-Watson.
Fonte: [19]

2.2.6 Correção da Autocorrelação do Resíduo

Ao conhecer as consequências da autocorrelação, principalmente a falta de eficiência dos estimadores, deve-se corrigir o problema. A correção depende do conhecimento que se tem da natureza da interdependência entre os termos de erro, ou seja, do conhecimento da estrutura da autocorrelação [19].

Como exemplo, considera-se o modelo de regressão de duas variáveis da Equação 2.42 [19]:

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t \tag{2.42}$$

Considere a Equação 2.42 no tempo (t - 1):

$$y_{t-1} = \beta_0 + \beta_1 x_{t-1} + \epsilon_{t-1} \tag{2.43}$$

Multiplica-se a Equação 2.43 por ρ:

$$\rho y_{t-1} = \rho\beta_0 + \rho\beta_1 x_{t-1} + \rho\epsilon_{t-1} \tag{2.44}$$

Subtrai-se a Equação 2.44 da 2.42:

$$(y_t - \rho y_{t-1}) = (1-\rho)\beta_0 + (x_t - \rho x_{t-1})\beta_1 + (\epsilon_t - \rho\epsilon_{t-1}) \tag{2.45}$$

Substitui-se a Equação 2.41 na 2.45:

$$(y_t - \rho y_{t-1}) = (1-\rho)\beta_0 + (x_t - \rho x_{t-1})\beta_1 + u_t \tag{2.46}$$

Considerando $\beta_0^* = \beta_0(1-\rho)$, $\beta_1^* = \beta_1$, $y_t^* = y_t - \rho y_{t-1}$, $x_t^* = x_t - \rho x_{t-1}$ e substituindo na Equação 2.46 tem-se a Equação 2.47:

$$y_t^* = \beta_0^* + \beta_1^* x_t^* + u_t \tag{2.47}$$

O [19] cita um método iterativo de Cochrane e Orcutt, que consiste nos passos abaixo:

(a) Calcula-se a Equação 2.42 pelo Método dos Quadrados Mínimos e obtenha os resíduos, ϵ_t .

(b) Utilizam-se os resíduos obtidos na etapa (a), calcula-se a seguinte regressão: $\epsilon_t = \hat{\rho}\epsilon_{t-1} + v_t$;

(c) Com o ρ obtido na etapa (b), calcula-se a Equação 2.47 de diferenças generalizadas;

(d) Como não se sabe se o ρ obtido por meio da etapa (b) é o melhor estimador de ρ, substitui-se os valores de β_0^* e β_1^* calculados na etapa (c) na regressão original, Equação 2.42, e obtem-se os novos resíduos;

(e) Executam-se as etapas acima até ρ quando estiver com um valor aceitável ou estiver entre 0,01 e 0,005.

2.2.7 Modelo Linear Generalizado

Utiliza-se este modelo para avaliar e quantificar a relação entre uma variável de saída e as variáveis explicativas. O GLM difere da regressão simples em dois aspectos importantes [26]:

(i) Escolhe-se a distribuição da resposta proveniente família exponencial. Assim, a distribuição da resposta não precisa ser normal ou próxima do normal e pode ser explicitamente não normal;

(ii) Uma transformação da média da resposta está linearmente relacionada às variáveis explicativas;

Uma consequência de permitir que a resposta seja da família exponencial é que a resposta pode ser, e geralmente é, heterocedástica. Assim, a variância muda de acordo com a média que pode, por sua vez, variar com as variáveis explicativas. Isso contrasta com a suposição homocedástica de regressão normal [26].

Como exemplo da importância dos modelos lineares generalizados tem-se a análise de dados de seguros. Com os dados de seguros, as premissas do modelo normal frequentemente não são aplicáveis. Os tamanhos dos sinistros, as frequências dos sinistros e a ocorrência de um sinistro em uma única apólice são resultados que não são normais. Além disso, a relação entre resultados e fatores de risco costuma ser multiplicativa, em vez de aditiva [26].

Dada uma resposta y, tem-se o modelo linear generalizado (GLM) na Equação 2.48 [26]:

$$f(y) = c(y, \phi)exp(\frac{y\theta - a(\theta)}{\phi}), \qquad g(\mu) = x'\beta \tag{2.48}$$

$f(y)$ especifica que a distribuição da resposta está na família exponencial. A segunda equação especifica que uma transformação da média, $g(\mu)$, está linearmente relacionada às variáveis explicativas contidas em x. O θ é denominado parâmetro canônico e ϕ parâmetro de dispersão. [26].

Dada uma variável de resposta y, construir um GLM consiste nas seguintes etapas [26]:

(i) Escolhe-se uma distribuição de resposta $f(y)$ e, escolhe-se $a(\theta)$ em 2.48. A distribuição da resposta adapta-se à situação dada;

(ii) Escolhe-se a função *link* $g(\mu)$. Esta escolha às vezes é simplificada escolhendo o chamado *link* "canônico" correspondente a cada distribuição de resposta;

(iii) Escolhem-se as variáveis explicativas x em termos das quais deve-se modelar $g(\mu)$. Aplicam-se considerações semelhantes à modelagem de regressão comum;

(iv) Coletam-se observações $y_1, ..., y_n$ da resposta y e os valores correspondentes $x_1, ..., x_n$ das variáveis explicativas x. As observações sucessivas são consideradas independentes, ou seja, considera-se uma amostra aleatória da população analisada;

(v) Ajusta-se o modelo estimando β e, se desconhecido, ϕ. Geralmente realiza-se o ajuste com a estimativa de máxima verossimilhança ou suas variantes;

(vi) Dada a estimativa de β, geram-se previsões (ou valores ajustados) de y para diferentes configurações de x. E examina-se o quão bem o modelo se ajusta examinando o desvio dos valores ajustados em relação aos valores reais, bem como outros diagnósticos do modelo. Além disso, usa-se o valor estimado de β para ver se as variáveis explicativas fornecidas são importantes na determinação de μ.

As etapas extras em comparação com a modelagem de regressão comum são escolher $a(\theta)$ (implicando na distribuição da resposta) e o *link* $g(\mu)$. Guia-se a escolha de $a(\theta)$ pela natureza da variável de resposta. Sugere-se a escolha do *link* pela forma funcional da relação entre a resposta e as variáveis explicativas [26].

As etapas acima raramente ocorrem sequencialmente. Por exemplo, os dados são frequentemente coletados antes da especificação de um modelo e podem ser simplesmente um banco de dados existente. A exploração inicial dos dados provavelmente indicará diferentes modelos e diferentes distribuições de respostas. Os ajustes são frequentemente seguidos por refinamentos adicionais dos ajustes, em que algumas das variáveis explicativas podem ser descartadas ou transformadas. Os dados podem ser analisados criteriosamente pela exclusão de vários casos ou pela incorporação de diferentes efeitos [26]. Os Quadros 2.1 e 2.2 apresentam os principais parâmetros do modelo GLM.

Função *Link*	$g(\mu)$	*Link* Canônico
Identity	μ	Normal
Log	$\ln(\mu)$	Poisson
Power	μ^p	Gamma(p=-1)
Square Root	$\sqrt{\mu}$	—
Logit	$\ln(\frac{\mu}{1-\mu})$	Binomial

Quadro 2.1: *Links* comumente usados.
Fonte: [26]

Distribuição	θ	$a(\theta)$	ϕ	$E(y)$	$V(\mu) = \frac{Var(y)}{\phi}$
$B(\eta, \pi)$	$\ln(\frac{\pi}{1-\pi})$	$n\ln(1+e^\theta)$	1	$n\pi$	$n\pi(1-\pi)$
$P(\mu)$	$\ln(\mu)$	e^θ	1	μ	μ
$N(\mu, \sigma^2)$	μ	$\frac{1}{2}\theta^2$	σ^2	μ	1
$G(\mu, \nu)$	$-\frac{1}{\mu}$	$-\ln(-\theta)$	$\frac{1}{\nu}$	μ	μ^2
$IG(\mu, \sigma^2)$	$-\frac{1}{2\mu^2}$	$-\sqrt{-2\theta}$	σ^2	μ	μ^3
$NB(\mu, \kappa)$	$\ln(\frac{\kappa\mu}{1-\kappa\mu})$	$-\frac{1}{\kappa}\ln(1-\kappa e^\theta)$	1	μ	$\mu(1+\kappa\mu)$

Quadro 2.2: Famílias de Distribuições Exponenciais e seus parâmetros.
Fonte: [26]

2.3 Métricas

Nesta seção detalham-se os índices utilizados para medir o desempenho dos modelos de Aprendizado de Máquina.

2.3.1 Erro Percentual Absoluto Médio

Este índice de erro é dado pela Equação 2.49, e quanto mais próximo de zero este índice estiver melhor será a previsão [31]. Utiliza-se este índice para comparar o resultado deste trabalho com o trabalho de referência [31].

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{|Y_t - \hat{Y}_t|}{Y_t} \tag{2.49}$$

Onde Y_t é o valor real e $\hat{Y}_t$ o valor previsto.

2.3.2 Coeficiente de Eficiência do Modelo Nash–Sutcliffe

De acordo com [32], o coeficiente *Nash–Sutcliffe Efficiency* (NSE) representa o quanto os valores previstos ($\hat{Y}_t$) e os valores reais (Y_t) estão ajustados, sendo $\bar{Y}_t$ o valor médio da série analisada. A faixa de trabalho do NSE é de $-\infty$ a 1 e valores maiores que 0 são considerados aceitáveis. O NSE é dado pela Equação 2.50:

$$NSE = 1 - \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{\sum_{t=1}^{n}(Y_t - \bar{Y}_t)^2} \tag{2.50}$$

2.3.3 Coeficiente de Correlação

Conforme [14], o coeficiente de correlação (R) é uma medida adimensional da associação linear entre um par de variáveis (Y_t, $\hat{Y}_t$), obtida dividindo a covariância pelo produto dos desvios-padrão das variáveis. A correlação assume um valor entre -1 e $+1$, com o valor 0 indicando que não há associação linear e o valor $|1|$ indica uma correlação perfeita. Define-se a correlação R entre um par de variáveis (Y_t, $\hat{Y}_t$) pela Equação 2.51:

$$R = \frac{n\sum_{t=1}^{n} Y_t\hat{Y}_t - \sum_{t=1}^{n} Y_t \sum_{t=1}^{n} \hat{Y}_t}{\sqrt{n\sum_{t=1}^{n} Y_t^2 - (\sum_{t=1}^{n} Y_t)^2}\sqrt{n\sum_{t=1}^{n} \hat{Y}_t^2 - (\sum_{t=1}^{n} \hat{Y}_t)^2}} = \frac{cov(Y,\hat{Y})}{\sqrt{var(Y)\cdot var(\hat{Y})}} \tag{2.51}$$

2.4 Inteligência Artificial

A Inteligência Artificial (IA), comparada com a inteligência natural do ser humano, visa imitar, estender e aumentar a inteligência humana através de meios e técnicas artificiais para alcançar certa inteligência de máquina. A ciência da IA se concentra em modelos computacionais de comportamentos inteligentes, desenvolve sistemas de computador para atividades noéticas, como percepção, facilitação, aprendizado, associação, tomada de decisão, etc., e resolve problemas complexos que somente especialistas humanos podem resolver [42].

Alguns pesquisadores classificam a pesquisa em IA em duas categorias: a inteligência simbólica e inteligência computacional. Inteligência simbólica, também conhecida como IA tradicional, resolve problemas através do raciocínio baseado no conhecimento. A inteli-

gência computacional resolve problemas com base em conexões treinadas a partir de dados de exemplo. Redes neurais artificiais (RNA), algoritmos genéticos, sistemas nebulosos, programação evolutiva, vida artificial, etc. estão incluídos na inteligência computacional [42].

A maioria das pessoas não percebem o quanto de suas vidas são afetadas pela IA e não podem imaginar como ela se expandirá nos próximos anos. As pessoas navegam na Internet com mecanismos de busca de IA e encontram informações através de agentes inteligentes quando compram um produto em um site. Suas contas bancárias e cartões de crédito são monitorados com software de IA, e os sistemas de transporte funcionam sem problemas com programas de IA. Todo e-mail e toda chamada de telefone celular são roteados usando redes de IA. Os aparelhos na cozinha e o carro na garagem todos têm peças que usam tecnologia de IA [45].

Com o passar do tempo a IA foi evoluindo e foi dividida em áreas de estudo ou interesse, a Figura 2.8 ilustra as principais partes da IA.

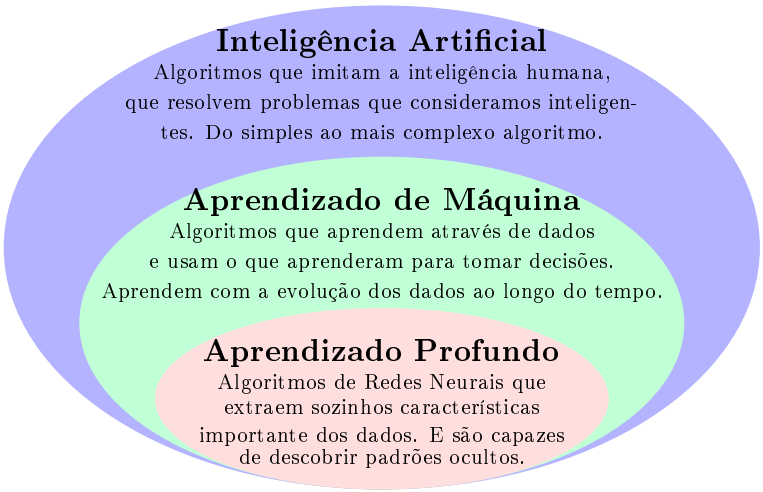

Figura 2.8: Inteligência Artificial e suas divisões.
Fonte: [29] e [13]

Já [25], na Figura 2.9, reduziu a história recente da IA a três segmentos, com base na evolução das linguagens de programação. Esses segmentos são os anos iniciais (1954-1973), épocas turbulentas (1974-1993) e a era moderna (1994 até o presente).

2.4.1 Aprendizado de Máquina

O conhecimento, a representação do conhecimento e os algoritmos de raciocínio baseados no conhecimento são sempre considerados o coração da IA, enquanto o Aprendizado de

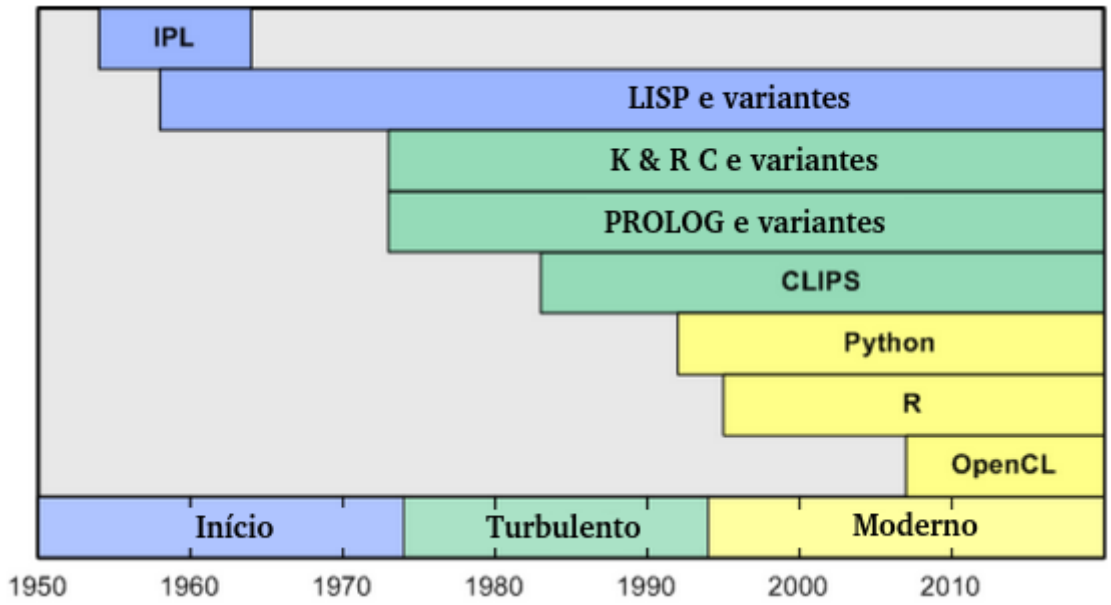

Figura 2.9: Evolução da IA com base nas linguagens de programação.
Fonte: [25]

Máquina pode ser visto como um problema mais crítico. Os psicólogos e filósofos sustentam que o mecanismo básico da aprendizagem está em tentar transferir comportamentos bem-sucedidos em uma prática para outras práticas semelhantes [42].

Aprender é o processo de aquisição de conhecimento, obtenção de experiência, melhoria de desempenho, descoberta de regras e adaptação aos ambientes. A Figura 2.10 ilustra um modelo simples de aprendizado com quatro elementos básicos de um sistema de aprendizado. O Ambiente fornece informações externas, semelhantes a um supervisor. A Unidade de Aprendizado processa as informações fornecidas pelo ambiente, correspondendo a vários algoritmos de aprendizado. A Base de Conhecimento armazena as informações em certos formalismos de representação do conhecimento. A Unidade de Execução realiza determinadas tarefas com base nas informações da Base de Conhecimento e envia os resultados da Unidade de Execução para a Unidade de Aprendizagem por meio da Realimentação. Com isso, melhora-se o sistema gradualmente através do aprendizado[42].

A pesquisa em Aprendizado de Máquina não apenas permite que as máquinas adquiram automaticamente conhecimento e obtenham inteligência, mas também descobre princípios e segredos do pensamento e da aprendizagem dos humanos, e até ajuda a melhorar a eficiência da aprendizagem humana. A pesquisa em Aprendizado de Máquina também tem um grande impacto nos padrões de armazenamento de memória, métodos

de entrada de informações e arquiteturas de computadores [42].

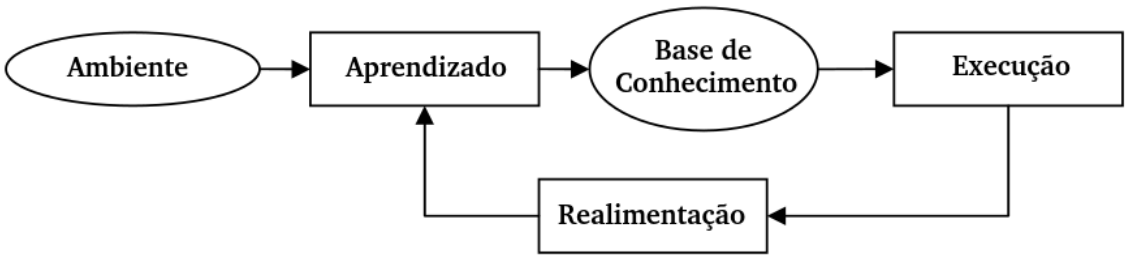

Figura 2.10: Modelo simples de aprendizado.
Fonte: [42]

Os modelos de Aprendizado de Máquina dividem-se em quatro categorias de aprendizado: supervisionado, não supervisionado, por reforço, rede neural.

2.4.1.1 Aprendizado Supervisionado

No aprendizado supervisionado, rotula-se o conjunto de dados com a resposta que o algoritmo deve apresentar. O aprendizado supervisionado leva as variáveis de entrada (X) junto com uma variável de saída (Y). A variável de saída representa a coluna na qual deseja-se prever. O algoritmo usa essas variáveis para aprender e aproximar a função de mapeamento da entrada com a saída. Os algoritmos de aprendizado supervisionado suportam problemas de classificação e regressão. O Quadro 2.3 mostra alguns exemplos de aprendizado supervisionado [2].

Variáveis Independentes(X)	Variáveis Dependentes(Y)
Anos de Carreira, Formação, Idade	Salário
Idade do Carro, Idade do Motorista	Risco de Acidente Automotivo
Histórico Escolar	Nota no ENEM

Quadro 2.3: Exemplos de aprendizado supervisionado.
Fonte: [20]

2.4.1.2 Aprendizado não Supervisionado

No aprendizado não supervisionado, fornece-se o modelo com um conjunto de dados que não é rotulado, ou seja, sem um resultado explícito que o algoritmo deve retornar. Nesse caso, o algoritmo tenta encontrar padrões e estrutura nos dados, extraindo recursos

úteis. O modelo organiza os dados de diferentes maneiras, dependendo do algoritmo (*clustering*, detecção de anomalias, auto-codificadores, etc.). O Quadro 2.4 mostra alguns exemplos de aprendizado não supervisionado [2].

Variáveis Independentes(X)	Variáveis Representativas(Y)
Registros de Compras	Associação entre produtos
Registros de Compras	Perfil dos consumidores
Transações bancárias	Normalidade da transação

Quadro 2.4: Exemplos de aprendizado não supervisionado.
Fonte: [20]

2.4.1.3 Aprendizado por Reforço

O aprendizado por reforço é um modelo de aprendizado comportamental. O algoritmo recebe *feedback* da análise dos dados para que o usuário seja orientado para o melhor resultado. O aprendizado por reforço difere de outros tipos de aprendizado supervisionado porque o sistema não é treinado com o conjunto de dados de amostra. Em vez disso, o sistema aprende por tentativa e erro. Portanto, uma sequência de decisões bem-sucedidas resultará no "reforço" do processo, pois melhor soluciona o problema em questão [22].

Uma das maneiras mais fáceis de pensar sobre o aprendizado por reforço é a maneira como um animal é treinado para executar ações baseadas em recompensas. Se o cão receber uma recompensa toda vez que ele estiver sob comando, ele executará essa ação todas as vezes. Esta categoria também pode ser aplicada para treinar robôs e carros autônomos [22].

2.4.1.4 Aprendizado por Redes Neurais e Profundo

Uma rede neural consiste em três ou mais camadas: uma camada de entrada, uma ou várias camadas ocultas e uma camada de saída. Inserem-se os dados através da camada de entrada. Em seguida, modificam-se os dados na camada oculta e na camada de saída com base nos pesos aplicados a esses nós (neurônios) [22].

A rede neural típica pode consistir em milhares ou até milhões de nós de processamento simples, densamente interconectados. Usa-se o termo aprendizado profundo quando existem várias camadas ocultas em uma rede neural. Usando uma abordagem iterativa,

uma rede neural se ajusta e faz inferências continuamente até que um ponto de parada específico seja alcançado [22]. A Figura 2.11 exemplifica o formato de um nó e a Figura 2.12 as camadas da rede neural.

De acordo com [42], as características básicas das redes neurais incluem:

(a) Armazenamento distribuído de informações;

(b) Processamento paralelo de informações;

(c) Capacidades de auto-organização e auto-aprendizagem.

2.4.1.5 Tipos de Algoritmos de Modelos de Aprendizado de Máquina

Selecionar o algoritmo certo é parte da ciência e parte da arte. Dois cientistas de dados, encarregados de resolver o mesmo desafio comercial, podem escolher algoritmos diferentes para abordar o mesmo problema. No entanto, entender diferentes classes de algoritmos de Aprendizado de Máquina ajuda os cientistas de dados a identificar os melhores tipos de algoritmos [22].

Abaixo tem-se uma breve visão geral dos principais tipos de algoritmos de Aprendizado de Máquina:

(a) Bayesiano: permite que os cientistas considerem resultados de outros testes, como por exemplo, uma probabilidade no modelo atual independentemente dos dados. Esses algoritmos são especialmente úteis quando você não possui grandes quantidades de dados para treinar um modelo com confiança [22];

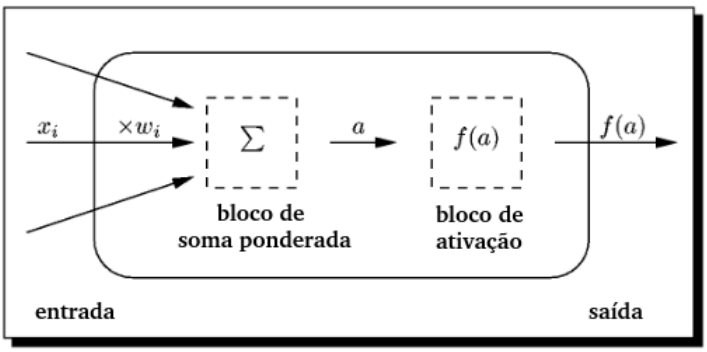

Figura 2.11: Arquitetura do Neurônio.
Fonte: [21]

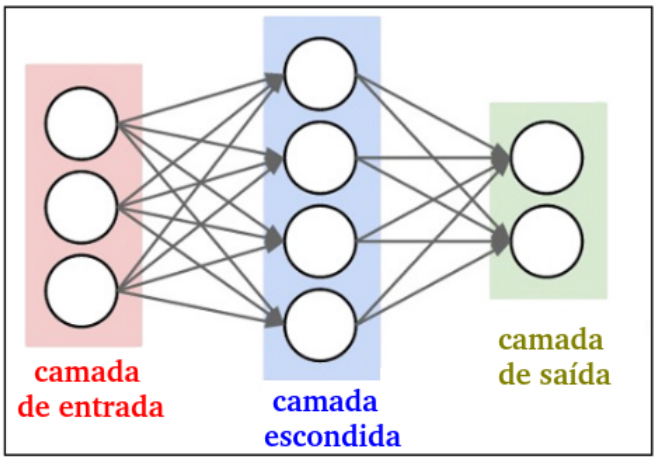

Figura 2.12: Arquitetura da Rede Neural.
Fonte: [22]

(b) Agrupamento: nesta técnica agrupam-se objetos com parâmetros semelhantes. Todos os objetos em um grupo são mais semelhantes entre si do que os objetos que estão em outros grupos. Agrupamento é um tipo de aprendizado não supervisionada porque os dados não estão rotulados. O algoritmo interpreta os parâmetros que compõem cada item e os agrupa adequadamente [22];

(c) Árvore de Decisão: usa uma estrutura de ramificação de árvore para ilustrar os resultados de uma decisão. Através da árvore de decisão mapeia-se os resultados possíveis de uma decisão. Cada nó de uma árvore de decisão representa um resultado possível. Atribuem-se porcentagens aos nós com base na probabilidade do resultado ocorrer. Por exemplo, utilizam-se em campanhas de *marketing* na qual existem vários grupos de consumidores e deseja-se saber qual grupo responderá melhor a campanha [22];

(d) Redução de Dimensionalidade: ajuda os sistemas a remover dados que não são úteis para análise. Usa-se este grupo de algoritmos para remover dados redundantes, *outliers* e outros dados não úteis. São úteis ao analisar dados de sensores e outros casos de uso da Internet das Coisas (IoT). Nos sistemas IoT, pode haver milhares de pontos de dados simplesmente dizendo que um sensor está ligado. Com isso, reduz o espaço de armazenamento, melhora o desempenho do aprendizado e facilita a visualização dos dados [22];

(e) Baseado em Instâncias: usa-se quando deseja-se categorizar novos pontos de dados baseados em semelhanças com os dados de treinamento. Este conjunto de algoritmos às vezes são chamados de aprendizes preguiçosos porque não há fase de treinamento.

Em vez disso, o algoritmo simplesmente combina novos dados com dados de treinamento e categoriza os novos pontos de dados com base na semelhança com os dados de treinamento. Não é adequado para conjuntos de dados que possuem variação aleatória, dados irrelevantes ou dados com valores ausentes. E podem ser muito úteis no reconhecimento de padrões. Por exemplo, em análise de produtos químicos, estrutura biológica e espacial [22];

(f) Rede Neural e Aprendizado Profundo: tenta imitar a maneira como um cérebro humano aborda problemas e usa camadas de unidades interconectadas para aprender e inferir relacionamentos com base nos dados observados. Uma rede neural pode ter várias camadas conectadas e quando houver mais de uma camada oculta em uma rede neural, esta é denominada aprendizagem profunda. Modelos de redes neurais são capazes de ajustar e aprender conforme os dados mudam. E são frequentemente usadas quando os dados são não marcados ou não estruturados. Possui aplicações em visão computacional, carros autônomos e análise de imagens [22];

(g) Regressão Linear: comumente usados para análise estatística e são algoritmos importantes para uso em Aprendizado de Máquina. Eles ajudam os analistas a modelar relacionamentos entre pontos de dados. Algoritmos de regressão podem quantificar a força da correlação entre variáveis em um conjunto de dados. Além disso, pode ser útil para prever os valores futuros com base em valores históricos. No entanto, é importante lembrar que a análise de regressão pressupõe que a correlação esteja relacionada à causalidade. Sem entender o contexto ao redor dos dados, a análise de regressão pode levar a previsões imprecisas [22];

(h) Regularização para evitar sobreajuste: é uma técnica para modificar modelos de Aprendizado de Máquina para evitar problemas de preditivas fracas (Sobreajuste). Pode-se aplicar a regularização a qualquer modelo de Aprendizado de Máquina. A regularização simplifica modelos complexos que são propensos a serem excessivamente ajustados. Se um modelo estiver superajustado, ele fornecerá imprecisões as previsões quando exposto a novos conjuntos de dados [22];

(i) Baseado em Regras: usa regras relacionais para descrever dados. Em resumo quer dizer: se X é um dado de entrada, faça Y. No entanto, à medida que os sistemas são operacionalizados, estas regras se tornam complexas e, com isso, surgem as exceções das regras [22].

2.5 Otimização por Enxame de Partículas

Otimização é uma disciplina científica que lida com a detecção de soluções ótimas para um problema, entre alternativas. A otimização das soluções baseia-se em um ou vários critérios que geralmente dependem do problema e do usuário [40].

Por exemplo, um problema de engenharia estrutural pode admitir soluções que aderem principalmente às especificações fundamentais de engenharia, bem como às expectativas estéticas e operacionais do projetista. As restrições podem ser impostas pelo usuário ou pelo próprio problema, reduzindo assim o número de soluções em potencial. Se uma solução atende a todas as restrições, chama-se de solução viável. Entre todas as soluções viáveis, o problema de otimização global diz respeito à detecção da solução ideal. No entanto, isso nem sempre é possível ou necessário. De fato, há casos em que soluções abaixo do ideal são aceitáveis, dependendo da qualidade em comparação com a melhor. Geralmente descreve-se isso como otimização local, embora usa-se o mesmo termo também para descrever a pesquisa local entre vizinhança do espaço de pesquisa [40].

Descreve-se a PSO com o seguinte cenário: um grupo de pássaros (partículas) está pesquisando aleatoriamente alimentos em uma área (espaço de pesquisa). Há apenas um pedaço de comida (melhor solução) na área pesquisada. Os pássaros não sabem onde está a comida, mas eles sabem área em que a comida está e as posições de seus colegas. Então, uma estratégia eficaz é se aproximar do pássaro que aparentemente está mais perto da comida [6].

Neste trabalho utiliza-se a PSO para otimizar o Aprendizado de Máquina. Esta otimização consiste em fazer um ajuste fino nos parâmetros estabelecidos pelo Aprendizado de Máquina. Escolheu-se o PSO devido a sua enorme capacidade de otimização já consagrada. E também, devido as suas características assíncronas e de paralelização, que serão importantes em implementações futuras, se houver a necessidade de processar grandes volumes de dados.

Segundo [8], indica-se a atualização assíncrona para implementação de PSO paralelizado. E a paralelização contribuí para uma maior eficiência em aplicações de tempo real, pois aumenta a capacidade de processamento. Já [33] relata que em algumas situações o PSO assíncrono é 3,5 vezes mais rápido que o PSO síncrono. E por último o [23], afirma que assim como outras meta-heurísticas baseadas em população, o PSO é intrinsecamente paralelo e pode ser implementado com eficácia em *Graphics Processing Units* (GPUs), que são arquiteturas de processamento massivamente paralelas. E com isso, aumenta-se

ainda mais a velocidade de processamento.

A figura 2.13 apresenta um fluxograma básico de um algoritmo de PSO.

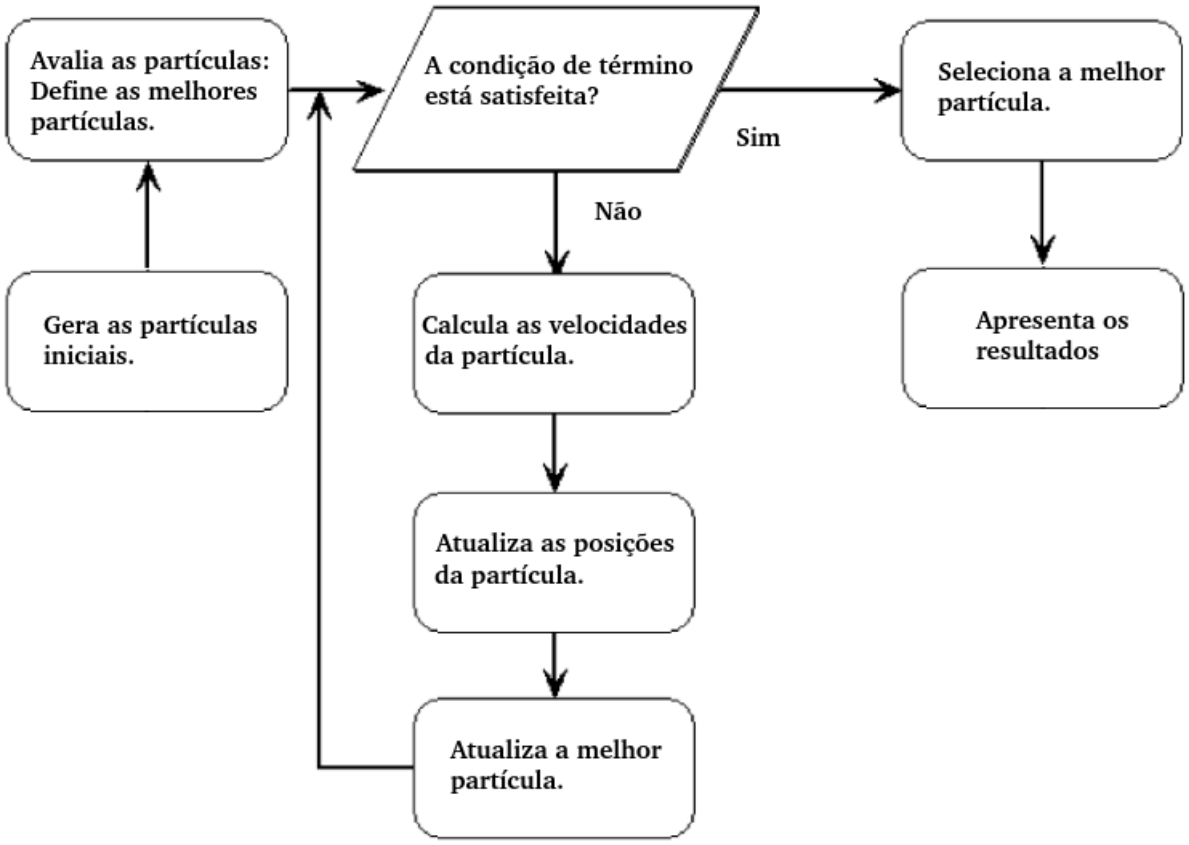

Figura 2.13: Arquitetura da PSO.
Fonte: [6]

2.6 *Python*

Python é uma linguagem de programação de propósito geral, frequentemente aplicada em funções de *script*. Ela é comumente definida como uma linguagem de *script* orientada a objetos [34].

Esta linguagem é desenvolvida sob uma licença de código aberto aprovada pela *Open Source Initiative* (OSI), tornando-a livremente utilizável e distribuível, mesmo para uso comercial. A licença do Python é administrada pela fundação *Python Software* [47].

Assim como C, C ++, Perl e Java, *Python* é um exemplo de uma linguagem de alto nível. Com isso possui vantagens enormes, como por exemplo: facilidade e rapidez na implementação, programas menores e portabilidade. Esta portabilidade significa que

podem ser executados em diferentes tipos de computadores com poucas ou nenhuma modificação [16].

De acordo com [34] é possível desenvolver com *Python* os seguintes tipos de aplicações:

(a) Programação de Sistemas;

(b) Interface gráfica com o usuário (GUI);

(c) *Scripts* de internet;

(d) Integração de componentes (C, C++ e JAVA);

(e) Programação de banco de dados;

(f) Programação numérica;

(g) Jogos, Imagens, IA e *eXtensible Markup Language* (XML).

Baseado nas características citadas acima escolhe-se *Python* como linguagem de programação e com ela desenvolvem-se algoritmos de testes, treinamento, validação e o aplicativo *Web*. Dentro das várias bibliotecas existentes no *Python* a Tabela 2.1 listas as mais importantes para o desenvolvimento do trabalho.

	Bibliotecas	
BeautifulSoup	matplotlib	seaborn
csv	numpy	selenium
datetime	os	shutil
glob	pandas	sklearn
h2o	pickle	statsmodels
html.parser	pyswarms	sys
lightgbm	re	time
math	scipy	urllib2

Tabela 2.1: Bibliotecas *Python* mais importantes para o trabalho.
Fonte: [47]

2.6.1 AutoML H_2O

O H_2O é uma plataforma de Aprendizado de Máquina e análise preditiva de código aberto, distribuída, rápida, escalonável e com armazenamento na Memória de Acesso Aleatório (do inglês *Random Access Memory*, RAM). Que permite criar modelos de Aprendizado de Máquina em *big data* e possibilita a utilização desses modelos em um ambiente corporativo [2].

O analisador de dados do H_2O possui inteligência integrada para entender o esquema do conjunto de dados recebido e suporta a inserção de dados de várias fontes em vários formatos [2].

A plataforma AutoML H_2O possibilita de forma automática a utilização de vários algoritmos supervisionados e não supervisionados, com poucas configurações e ainda apresenta uma classificação com os melhores resultados. Esta ferramenta é ideal para definir quais algoritmos investigar para encontrar a melhor solução de cada problema.

Ao escolher a plataforma AutoML H_2O consideraram-se as seguintes características: código aberto, desenvolvimento em *Python*, número de algoritmos utilizados e fácil utilização.

A configuração da plataforma é mostrada no Algoritmo 3 do Apêndice A. E abaixo apresentam-se os algoritmos suportados pela H_2O:

Algoritmos supervisionados:

(a) Aprendizado de Máquina Automático;

(b) Modelo de Riscos Proporcionais de Cox;

(c) Aprendizado Profundo;

(d) Floresta Aleatória Distribuída;

(e) Modelo Linear Generalizado;

(f) Modelo Aditivo Generalizado;

(g) Máquina de Aumento de Gradiente;

(h) Classificador Naive Bayes;

(i) Combinação de Empilhamento;

(j) Máquina de Vetores de Suporte;

(k) Aumento de Gradiente Extremo.

Algoritmos não supervisionados:

(a) Agregação;

(b) Modelo de Classificação Baixa Generalizado;

(c) Floresta de Isolamento;

(d) Agrupamento *K-Means*;

(e) Análise do Componente Principal.

Capítulo 3

Trabalhos Relacionados

Neste capítulo apresenta-se uma revisão de trabalhos publicados relativos à previsão de vazão em rio, com o objetivo de estabelecer uma referência para os resultados encontrados durante o desenvolvimento do trabalho. Estão incluídos resultados obtidos com modelos hidrológicos, estatísticos e de ML.

3.1 Revisão Geral dos Estudos

O trabalho desenvolvido por [31] é a principal referência deste trabalho, por utilizar os mesmos dados e períodos. Desta forma, ao final do corrente trabalho procuram-se resultados melhores que os desta referência. O estudo de [31] faz previsões através de séries temporais utilizando dados da vazão horária do rio Paraíba do Sul para a cidade de Volta Redonda. Por meio desses resultados, possibilita-se realizar alertas em tempo hábil para mitigar os problemas causados pelas cheias do rio. Os modelos de séries temporais para os dados com sazonalidade revelam-se satisfatórios, dos quais analisam-se os modelos Auto-Regressivo Integrado de Média Móvel Sazonal (do inglês *Seasonal Autoregressive Integrated Moving Average*, SARIMA) e Holt-Winters-Taylor com dupla sazonalidade. Sendo que esse último apresenta boa precisão da previsão ao obter o melhor valor do erro, MAPE igual a 1,69% com uma hora de antecedência. Além disso, o MAPE do modelo escolhido foi quase 80% menor que o do modelo que apresenta o maior erro [31].

No trabalho [44], realiza-se a previsão da vazão diária para o posto 266 (Itaipu) a partir da meta-heurística PSO e modelos Auto-Regressivo de Média Móvel (do inglês *Autoregressive Moving Average*, ARMA) e Auto-Regressivo Integrado de Média Móvel (do inglês *Autoregressive Integrated Moving Average*, ARIMA). No primeiro estágio utiliza-se a meta-heurística na determinação das ordens p (autorregressão) e q (médias móveis). Em

seguida, usa-se a mesma meta-heurística na obtenção dos coeficientes autorregressivos e médias móveis. Os resultados obtidos mostram que o modelo ajustado é adequado à série estudada. O erro encontrado (MAPE) para a previsão com um passo a frente é de 0,82% e para sete passos a frente 1,05%.

O estudo [18] tem como objetivo prever a vazão natural diária de curto prazo, com um horizonte de até 7 dias a frente, da estação Campos - Ponte Municipal do baixo curso do rio Paraíba do Sul. Utiliza-se uma base de dados com séries históricas de precipitação e vazão na modelagem da previsão da vazão. E investiga-se a capacidade dos métodos de Aprendizado de Máquina, tais como Floresta Aleatória (do inglês *Random Forest*, RF) e Rede Neural Artificial (RNA), em relação a um modelo linear de previsões. De acordo com os resultados, todos os métodos de Aprendizado de Máquina obtêm desempenhos satisfatórios em relação às medidas de erro utilizadas, para o horizonte de previsão, de modo que estes métodos auxiliem no acompanhamento e previsão do fluxo de bacias hidrográficas. Com objetivo de alcançar uma melhor generalização, e assim melhores resultados, cada um dos modelos RF, RNA e *Multi-task ElasticNet Linear Model* têm o processo de treinamento repetido 25 vezes, dos quais em cada iteração aplica-se a técnica de validação cruzada. Por outro lado, utiliza-se os dados referentes a precipitação e vazão de 26 anos como dados de entrada nos modelos de previsão de vazão, em um período de 1-7 dias. A Tabela 3.1 mostra os resultados obtidos.

Antecedência		MAPE(%)			NSE	
Dias	LM	RF	RNA	LM	RF	RNA
1	9,849	9,404	11,043	0,916	0,905	0,907
2	16,392	14,895	15,557	0,770	0,782	0,766
3	21,750	19,270	18,597	0,647	0,688	0,649
4	25,871	22,659	20,797	0,570	0,626	0,572
5	28,867	25,127	22,597	0,517	0,589	0,514
6	31,112	26,986	23,864	0,477	0,561	0,469
7	32,730	28,543	24,664	0,448	0,532	0,435

Tabela 3.1: Desempenho MAPE e NSE referentes aos modelos de previsão de vazão com antecedência de 1 a 7 dias.

Fonte: [18]

Já o estudo [48] explora a utilização de redes de Memória de Curto e Longo Prazo

(do inglês *Long Short-Term Memory*, LSTM) para a previsão de vazão em rios da Bacia Hidrográfica do Paraíba do Sul, um modelo notável pela insensibilidade as distâncias temporais entre eventos. Ele é apresentado, analisado e comparado, com foco na facilidade de reaplicação, demonstração da eficácia e ilustração dos resultados. A partir de comparações com outros modelos, obtêm-se medidas de desempenho melhores e resultados mais precisos. O maior destaque observado da LSTM é sua resiliência, especialmente para grandes conjuntos de dados. O uso de parâmetros ou entradas ruins é suplantado por sua capacidade de aprendizado e em muitas vezes que se espera obter resultados piores, as mudanças acabam ignoradas pela estrutura da rede [48]. Para avaliar os resultados utiliza-se o Coeficiente de Eficiência de Nash-Sutcliffe (NSE). No rio Paraíba do Sul utilizam-se os dados da estação Campos - Ponte Municipal e obtem um NSE igual a 0,9303. Para o rio Carangola, afluente do rio Paraíba do Sul, e considerando os dados da estação Carangola o NSE é igual a 0,8502 [48].

O artigo [41] demonstra a aplicação de duas técnicas diferentes do Sistema de Inferência Neuro-Difuso Adaptativo (do inglês *Adaptative Network Fuzzy Inference System*, ANFIS) para a estimativa de fluxos mensais. Na primeira parte do estudo utilizam-se dois modelos diferentes do ANFIS, o ANFIS com Partição Grade (do inglês *Grid Partition*, ANFIS-GP) e o ANFIS com Agrupamento Subtrativo (do inglês *Subtractive Clustering*, ANFIS-SC), na previsão de fluxo com um mês de antecedência. Em seguida avaliam-se os resultados considerando o efeito da periodicidade no desempenho da previsão do modelo. Neste momento trabalham-se com os dados de fluxo mensais de duas estações, a Estação Besiri no rio Garzan e a Estação Baykan no rio Bitlis na Bacia Firat-Dicle da Turquia. Na segunda parte do estudo, testam-se os desempenhos das técnicas ANFIS estimando vazões com dados de um outro rio próximo. Os resultados do Erro Absoluto Médio (do inglês *Mean Absotute Error*, MAE) e Raiz do Erro Quadrático Médio (do inglês *Root Mean Square Error*, RMSE) indicam que o desempenho do modelo ANFIS-SC é um pouco melhor que o modelo ANFIS-GP na previsão de fluxo. Como é dito, testa-se o desempenho com dados de uma outra estação próxima as estações de treinamento Besiri e Baykan. No final, para os dois modelos ANFIS-SC e ANFIS-GP, o melhor valor do Coeficiente de Determinação (R^2) foi de 0,97 e do Coeficiente de Correlação (R) igual 0,985 com o objetivo de prever um mês a frente.

De acordo com o artigo [27], são necessárias previsões confiáveis e precisas do fluxo de um rio em muitas atividades de planejamento, desenvolvimento de projeto, operação e manutenção de recursos hídricos. Por isso neste estudo, investiga-se a exatidão relativa dos modelos de RNA e de Regressão por Vetores de Suporte (do inglês *Support Vector*

Regression, SVR) acopladas à Transformação de Ondaletas (do inglês *Wavelet*, WA), na previsão mensal de vazões fluviais, e comparadas aos modelos regulares de RNA e SVR, respectivamente. E também comparam-se os desempenhos relativos dos modelos regulares de RNA e SVR. Para isso, utilizam-se dados mensais do fluxo fluvial das estações de Kharjegil e Ponel no norte do Irã. A comparação dos resultados revela que os modelos de RNA e SVR, juntamente com a transformação de *wavelet*, fornecem resultados de previsão mais precisos do que os modelos normais de RNA e SVR. No entanto, verificam-se que os modelos SVR acoplados à transformação *wavelet* fornecem melhores resultados de previsão do que os modelos RNA acoplados à transformação *Wavelet*. Os resultados indicam que os modelos SVR regulares executam um pouco melhor que os modelos normais de RNA. Sendo assim os melhores resultados para a estação Kharjegil com WA-SVR é R igual a 0,759 e para a estação Ponel com WA-SVR é R igual a 0,743.

O artigo [38] considera às características não estacionárias e de ruído dos dados da série temporal de vazão de rio, e adotam-se alguns métodos de pré-processamento para lidar com a complexidade de multiescala e ruído. Este artigo, propõem uma estrutura aprimorada que compreende o *Complete Ensemble Empirical Mode Decomposition with Adaptive Noise* (CEEMDAN) e *Empirical Bayes Threshold* (EBT). Emprega-se o CEEMDAN-EBT para decompor dados não estacionários da série temporal da vazão do rio nas Funções de Modo Intrínseco (do inglês *Intrinsic Mode Functions*, IMFs). Os IFMs derivados são divididos em duas partes; IMFs com predominância de ruído e IMFs sem ruído. Em primeiro lugar, separam-se os IMFs com ruído dominante usando *Empirical Bayesian Threshold* para integrar os IFMs com ruído e esparsidade. Em segundo lugar, utilizam-se os IMFs com ruído e IFMs sem ruído como entradas em *data-driven* e em modelos estocásticos simples, respectivamente, para prever a vazão baseado na série temporal. Finalmente, agregam-se os IMFs previstos para obter o resultado final da predição. Então aplica-se a estrutura proposta em quatro rios do Sistema de Bacias Indu e avalia-se o desempenho da previsão com o Erro Quadrado Médio (do inglês *Mean Square Error*, MSE), o MAE e o MAPE. O método proposto, CEEMDAN-EBT com *Multi Models* (MM), produziu o menor MAPE para todos os quatro estudos de caso em comparação com outros métodos. E dentre todos os testes atinge-se o menor MAPE (0,0025%) com os dados do rio Chenab. Com isso, os resultados sugerem que modelo híbrido proposto é uma ferramenta eficiente para fornecer uma previsão confiável, de séries temporais não estacionárias e com ruídos. Desta forma, pode ser aplicada como indicador para decisões de políticas públicas, como o planejamento da geração de energia e a gestão de recursos hídricos.geração de energia e a gestão de recursos hídricos.

O estudo [35] explora as influências remotas do clima, via padrões de variabilidades climáticas e regionais, via precipitação e vazão em bacias de contribuição. Para tanto, desenvolvem-se modelos empíricos de previsão de vazões mensais defasados no tempo da Usina Hidrelétrica de Itaipu. Testam-se estes modelos com diferentes grupos de preditores, tais como: índices climáticos; precipitação em regiões pluviométricas homogêneas; vazão em pontos a montante e em Itaipu; e o conjunto de todos os preditores anteriores. Por meio do método *stepwise* selecionam-se os preditores mais significativos, sendo destacados os seguintes preditores: índices do *El Niño* Oscilação Sul e de anomalias de temperatura da superfície do mar no Atlântico Tropical Sul; precipitação em locais na faixa sul da bacia; e a própria vazão em Itaipu defasada. Validam-se os modelos indicando de um modo geral o maior desempenho, nas defasagens mais curtas quando considerados os preditores de vazão e precipitação e nas defasagens mais longas considerando os índices climáticos. Portanto, os resultados deste estudo demonstram a importância de serem consideradas as influências remotas do clima nas estimativas de vazão, principalmente para previsões de longo prazo. Considerando a previsão da vazão com um passo de antecedência tem-se um coeficiente R igual a 0,87.

Este artigo [23] tem como objetivo revisar algumas técnicas de aprendizado de máquina e alguns *Ensembles* destas técnicas, para previsão de vazão de rio em diferentes áreas. E também discutem-se alguns métodos de mineração de dados encontrados na revisão da literatura. Assim após revisar muitos algoritmos de previsão, constroem-se alguns modelos (RNA, SVR e Cadeia de Markov) para previsão e comparação. Na etapa seguinte, constrói-se um modelo *Ensemble* com os mesmos algoritmos (RNA, SVR, cadeia de Markov). Com isso, implementam-se duas técnicas de modelagem de *Ensembles*, que são o *Bagging* e o *Voting*. Continuando, aplicam-se aos modelos um conjunto de oito dados da estação Eldeim, perto do Sudão do Nilo Azul. Conclui-se então que a técnica de *Bagging* oferece a melhor acurácia do que a *Voting*. Desta forma, comparam-se os resultados do *Ensemble* com os modelos individuais, e verifica-se que a técnica de *Bagging* fornece a maior precisão de todos os modelos, e tem R igual a 0,97. Como principal conclusão deste artigo, têm-se que os modelos individuais fornecem previsões que não consideram todas as situações e fenômenos em comparação com os modelos de *Ensembles*.

Por fim, o estudo [32] investiga o desempenho e o potencial de um modelo híbrido, que combina a Transformada de Ondaletas Discreta (do inglês *Discrete Wavelet Transform*, DWT) e o SVR para previsão diária e mensal de fluxo, o modelo híbrido chama-se de DWT-SVR. Três fatores principais da fase de decomposição da WA (a função WA mãe, o nível de decomposição e o efeito de borda) são propostos com o objetivo de melhorar

a precisão do modelo DWT-SVR. Compara-se o desempenho do modelo DWT-SVR com diferentes combinações, desses três fatores, com o modelo regular de SVR e avalia-se a eficácia com o RMSE e o NSE. Através dos dados diários e mensais de fluxo observados em duas estações em Indiana, Estados Unidos, testam-se as habilidades de previsão desses modelos. Os resultados demonstram que os diferentes modelos híbridos nem sempre superam o modelo SVR, na previsão com um dia ou um mês de antecedência. Isso sugere que é crucial considerar e comparar os três fatores principais ao usar o modelo DWT-SVR (ou outros métodos de ML acoplados à WA), em vez de escolhê-lo com base nas preferências pessoais. Em seguida, combinam-se previsões de vários modelos DWT-SVR com diferentes configurações, combinações feitas aplicando a média do Critério de Informação de Akaike (do inglês *Akaike Information Criterion*, AIC). Essa previsão do conjunto é superior ao melhor modelo DWT-SVR e ao modelo SVR regular, para previsões futuras com um dia e um mês de antecedência. Com relação as previsões com maior antecedência (ou seja, dois dias, três dias e dois meses), as previsões do conjunto usando a técnica da média da AIC, constatam-se que são consistentemente melhores do que o melhor modelo DWT-SVR e modelo SVR. Portanto, integrar técnicas de média ao modelo híbrido DWT-SVR é uma abordagem promissora para a previsão diária e mensal do fluxo. Além disso, é altamente recomendável considerar os três fatores principais ao usar modelos SVR baseados em WA (ou outros modelos de previsão baseados em WA). A Tabela 3.2 mostra um resumo dos resultados encontrados, para as estações de medição. A estação I é localizada no rio *East Fork White*, próximo a cidade de Bedford e a estação II situada no rio *Eel* próximo a Logansport.

O [7] apresenta a aplicação dos modelos ARIMA, SARIMA e a RNA Jordan-Elman na previsão de vazão mensal do rio Kizil em Xinjiang, China. Ele baseia-se em dois tipos diferentes de dados de vazão mensal (originais e dessazonalizados), que através das séries temporais e modelos de RNA Jordan-Elman utilizam-se de registros antigos da vazão como preditores. Assim avalia-se o desempenho das previsões com um mês de antecedência para todos os modelos no período de teste (1998-2005), com dados da vazão média mensal provenientes da estação de medição Kalabeili no Rio Kizil. Os modelos de RNA Jordan-Elman, que usam as vazões defazadas em um período como entradas, não apresentam nenhuma melhoria significativa em relação aos modelos de séries temporais. Com isso, os resultados sugerem aplicar os modelos de série temporais simples (ARIMA e SARIMA) no local de estudo, umas vez que apresentam desempenhos semelhantes e possuem estruturas mais simples. O coeficiente de correlação para os modelos SARIMA, ARIMA e RNA são R = 0.9381, R = 0.9434 e R = 0.9487, respectivamente.

Estação	Antecedência	SVR NSE	Melhor DWT-SVR NSE	Combinado DWT-SVR NSE
I	1 dia	—	0,928	0,929
	2 dias	0,754	0,765	0,781
	3 dias	0,572	0,631	0,639
	1 mês	—	0,666	0,671
	2 meses	0,001	0,216	0,227
II	1 dia	–	0,765	0,783
	2 dias	0,409	0,494	0,508
	3 dias	0,208	0,436	0,453
	1 mês	—	0,541	0,637
	2 meses	-0,211	0,180	0,297

Tabela 3.2: Desempenho da previsão de fluxo com NSE.
Fonte: [32]

3.2 Discussões

Durante o desenvolvimento do trabalho percebe-se que uma característica importante não é considerada por este trabalho e também pelos trabalhos deste Capítulo. Esta característica é as mudanças que ocorrem ao longo do leito do rio Paraíba do Sul e seus afluentes. Que são oriundas dos efeitos da conservação do solo, crescimento desordenado das cidades e entre outros, pois estas mudanças influenciam diretamente na previsão.

O [7] explica temas que fazem parte do corrente trabalho e do trabalho de referência. Na abordagem cita que embora os modelos tradicionais de RNA (ML) sejam não lineares eles são determinísticos e não capturam a estocasticidade da série temporal da vazão. Ao contrário dos modelos ARIMA e SARIMA que explicam a estocasticidade e são lineares. Também cita que modelos como ARIMA e SARIMA não podem incluir outras entradas que não sejam a própria série defasada. E modelos de ML ou RNA podem utilizar quantas entradas for conveniente.

Em [35] estabelece-se que para cada período específico de uma série temporal é possível utilizar diferentes preditores. Ou seja, diferente da abordagem do estudo corrente que utiliza as mesmas variáveis de entrada para qualquer período da série temporal estudada.

No trabalho [44] a meta-heurística PSO determina as ordens p e q e depois os coeficientes autorregressivos e médias móveis. Já no corrente trabalho aplica-se o ML para determinar os hiperparâmetros e o PSO apenas tenta melhorar os hiperparâmetros já estabelecidos.

Outra forma de enfrentar o problema apresentam-se nos estudos [23] e [32] que utilizam *Ensemble*, que consiste na combinação de modelos de previsão. A combinação ocorre quando um modelo utiliza o resultado de outros modelos ou do mesmo modelo com configurações diferentes, para realizar previsões. Segundo [23] esta técnica reduz o erro de generalização da previsão.

Diferente do trabalho corrente, no estudo [48] utilizam-se os dados de precipitação como entrada, mas no trabalho e para a região de estudo a inclusão desta variável de entrada não fornece uma melhora significativa na previsão.

Por fim, no cenário proposto por [18] divide-se a série temporal em partes, sendo que o modelo de previsão será o mesmo para toda a série, mas cada parte da série terá uma configuração diferente do modelo de previsão adotado. No estudo corrente utiliza-se o mesmo modelo de previsão e com as mesmas configurações ao longo de toda a série.

3.3 Comparativo entre os Estudos

Como já citado, com este trabalho busca-se um algoritmo de ML que encontre resultados melhores que o modelo estatístico HWT, apresentado no estudo [31]. Pois o estudo [31] utiliza os mesmos dados do corrente trabalho e atinge um MAPE igual a 1,69%.

Porém, além disso, necessita-se conhecer se o desempenho da previsão deste trabalho está consonância com outros trabalhos semelhantes. Então para poder avaliar o resultado final do trabalho, neste Capítulo, buscam-se artigos que fazem previsões de vazão de rio. E a Tabela 5.5 representa um resumo dos resultados dos artigos analisados neste Capítulo.

Referência	Modelo	MAPE(%)	NSE	R
Estudo [31]	HWT	1,69	—	—
Estudo [44]	ARMA-PSO	0,82	—	—
Estudo [18]	RNA	11,64	—	—
Estudo [48]	LSTM	—	0,9303	—
Estudo [41]	ANFIS-SC	—	—	0,985
Estudo [27]	WA-SVR	—	—	0,759
Estudo [38]	CEEMDAN-EBT-MM	0,0025	—	—
Estudo [35]	Empírico Linear	—	—	0,87
Estudo [23]	Ensemble Bagging	—	—	0,97
Estudo [32]	Combinado DWT-SVR	—	0,929	—
Estudo [7]	RNA	—	—	0.9487

Tabela 3.3: Comparação entre estudos com um passo de antecedência.
Fonte: Autor

Capítulo 4

Metodologia

Neste capítulo apresenta-se a metodologia utilizada para enfrentar o desafio de realizar previsões de forma eficiente, que resultará no desenvolvimento de um aplicativo *web* de alerta para inundações. A Figura 4.1 representa as etapas metodológicas.

4.1 Definir o Problema

A definição cuidadosa do problema requer uma compreensão da maneira de como utilizam-se as previsões, quem as exige e como a função de previsão encaixa-se na organização que a demanda. Um previsor precisa gastar tempo conversando com todos os envolvidos na coleta de dados, na manutenção de bancos de dados e no uso das previsões para o planejamento futuro [24].

De acordo com [1], uma boa maneira para começar a entender o problema, e se ele pode ser resolvido por Aprendizado de Máquina, é analisando e respondendo os questionamentos abaixo.

(a) Qual é o problema que o sistema deve resolver?;

(b) Qual o formato ou tipo de saída desejado?;

(c) Qual base de dados que será utilizada?;

(d) O que o ML deve fazer?;

(e) A saída está relacionada com o resultado?;

(f) É possível obter dados para treinamento?;

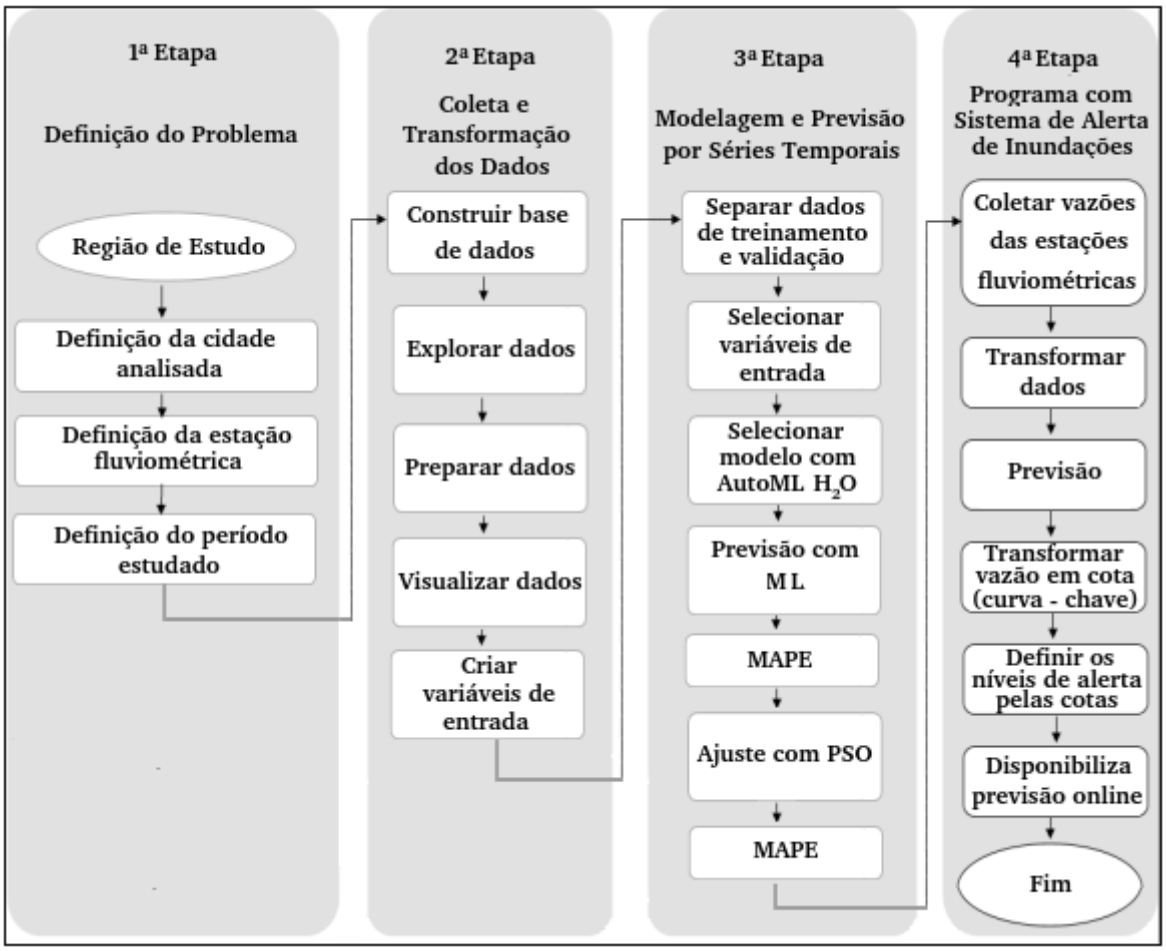

Figura 4.1: Etapas da metodologia adotada.
Fonte: [31] e Autor

(g) Que dados seu algoritmo precisará acessar quando executar o modelo de previsão?;

(h) Existe um tempo máximo para se obter a resposta?;

(i) Como será a avaliação de desempenho, qual será a condição que indicará sucesso ou falha?;

(j) Como seria resolvido este problema caso não utilizasse ML?.

O problema que se deseja resolver com este estudo são às inundação causada pela cheia do Rio Paraíba do Sul na cidade de Volta Redonda - RJ. Para isso, deseja-se prever o nível do rio com uma hora de antecedência, utilizando como dados de entrada as vazões de 3 estações de medição dispostas a jusante da cidade de Volta Redonda – RJ e extraídos do repositório do SNIRH. Com isso é possível alertar, antecipadamente, a população da cidade.

Tem-se como resultado final o nível do rio que é calculado pela vazão através da Equação 4.1. Assim o ML baixa os dados brutos do repositório, trata-os, realiza a previsão, transforma a saída para indicar o nível do rio e por último disponibiliza estas previsões em uma página *web*, a fim de alertar a população local. Como disponibilizam-se os dados com uma hora de antecedência e de hora em hora, o tempo máximo para executar todas estas etapas é de uma hora.

Por fim, define-se como critério de avaliação de desempenho o MAPE igual a 1,69%, conforme [31], e considera-se sucesso um MAPE $< 1{,}69\%$ e a falha um MAPE $\geq 1{,}69\%$.

4.2 Construir uma Base de Dados Bruta

Em primeiro lugar, deve-se considerar que o ML faz previsões com base em dados passados; portanto, caso não exista registros antigos, precisa-se obtê-los [1].

No corrente trabalho, para construir a base de dados bruta utilizam-se dados horários do SNIRH entre 01/01/2018 e 31/03/2018. Assim, considerando a Quadro 1.1 têm-se três estações que juntas fornecem três arquivos que compõem a base de dados bruta. Então para cada estação de medição baixa-se um arquivo no formato HTML. Sendo que estes arquivos possuem registros de nível de chuva, nível do rio e vazão do rio.

4.3 Transformar os Dados

De acordo com [1], existe um ditado para ML que diz que seu modelo é tão bom quanto os seus dados. Por isso precisa-se medir a qualidade do conjunto de dados e melhorá-la, além de estabelecer uma quantidade de dados necessária para obter resultados úteis. Estas definições dependem do tipo de problema a ser resolvido.

Segundo [1], alguns casos que podem diminuir a qualidade da previsão são:

(a) Incluir variáveis que não tem valor preditivo, ou seja, não estão altamente relacionadas com a variável que se deseja prever;

(b) A incorreta configuração dos parâmetros do ML;

(c) Dados com erros ou anomalias;

(d) Desvio na construção do código que trata os dados com erro.

De acordo com [1], melhora-se o desempenho do ML ao escrever algoritmos que verificam a base de dados nos seguintes aspectos:

(a) Todos os dados numéricos estão redimensionados. Como por exemplo, para valores entre 0 e 1;

(b) Os dados ausentes são substituídos por valores médios, padrão ou remove-se corretamente aquela parte da informação da base dados;

(c) As distribuições de dados após a transformação estão em conformidade com as expectativas. Por exemplo, ao normalizar usando *z-scores*, a média é 0;

(d) Os *outliers* estão manipulados, redimensionados ou removidos.

Com os dados disponíveis iniciam-se os processos de exploração, preparação, visualização e criação dos dados. Estes processos serão apresentados nas subseções 4.3.1, 4.3.2, 4.3.3 e 4.3.4 respecivamente.

4.3.1 Exploração dos Dados

Consiste em avaliar visualmente os dados brutos. Ao conhecer os dados identificam-se aqueles que tem potencial para serem utilizados como entrada, ou percebe-se a falta de dados em determinados períodos.

4.3.2 Preparação dos Dados

Desenvolve-se um algoritmo que retira todas as colunas com dados que não são utilizados. Já para as linhas, se alguma delas estiver com uma célula sem dado, toda a linha é excluída. E por último, aplica-se a normalização em todos os dados numéricos.

Na sequência cria-se um único arquivo no formato *Comma Separated Values* (CSV) com todos os dados de interesse das três estações, são eles: vazão do rio, mês, dia do mês, dia da semana e hora.

4.3.3 Visualizações

Consiste em plotar gráficos para verificar o comportamento dos dados, por exemplo, identificar se os dados podem ser definidos como uma série estacionária. E qual técnica

utiliza-se para normalização. O objetivo da visualização é checar se as suposições iniciais geram mudanças importantes na saída.

Através da análise do gráfico de histograma e da variância da série, constata-se qual técnica de normalização utilizar e de acordo com [1] e [31], a normalização recomendada é a logarítmica. Aplica-se esta normalização quando o conjunto de dados tem poucos valores com muitas repetições e muitos valores com poucas repetições. Ao analisar a Figura 4.2 verificam-se que existem muitos pontos entre 150 e 300 m^3/h e poucos pontos entre 300 e 850 m^3/h, assim como sugere[1]. A Figura 4.3 apresenta uma comparação entre três formas de normalização.

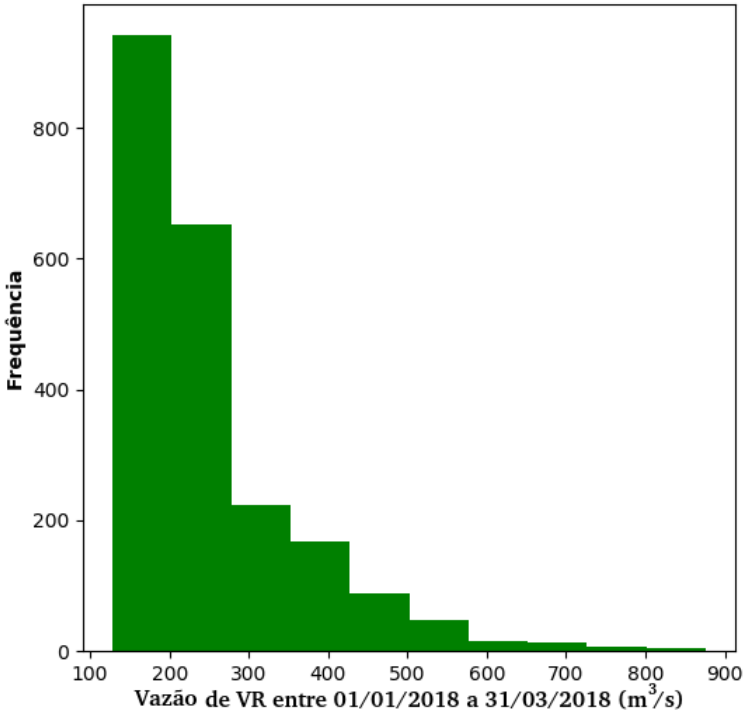

Figura 4.2: Histograma da vazão em Volta Redonda.
Fonte: Autor

Para identificar a estacionariedade da série analisa-se o gráfico da série temporal da vazão, o comportamento da série indica se esta é estacionária ou não. E conforme a Subseção 2.1.3.1, para confirmar a hipótese aplica-se o método de Dickey Fuller, que deve confirmar aquilo que já foi visualizado no gráfico. A Figura 4.4 mostra a vazão de VR normalizada e confirma a estacionariedade da série baseado no método Dickey Fuller. De acordo com este método quando Estatística ADF $<$ Valores Críticos indica estacionariedade.

Cabe ressaltar, que a análise de normalização, estacionariedade e estatística são feitas única e exclusivamente na série temporal de saída, vazão da estação de VR. As séries das

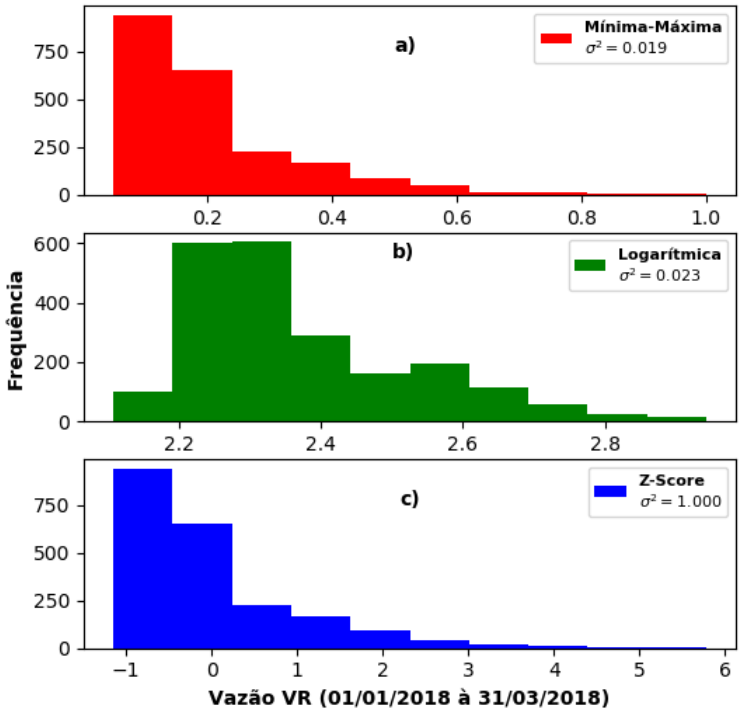

Figura 4.3: a) Normalização Mínima e Máxima. b) Normalização Logarítmica e c) Normalização Z-Score.

Fonte: Autor

entradas (vazão em outros pontos, *lag*, média e etc.) não são analisados nestes quesitos e só são efetivamente utilizadas no ML se agregarem melhoras para os resultados das previsões, durante a fase de treinamento.

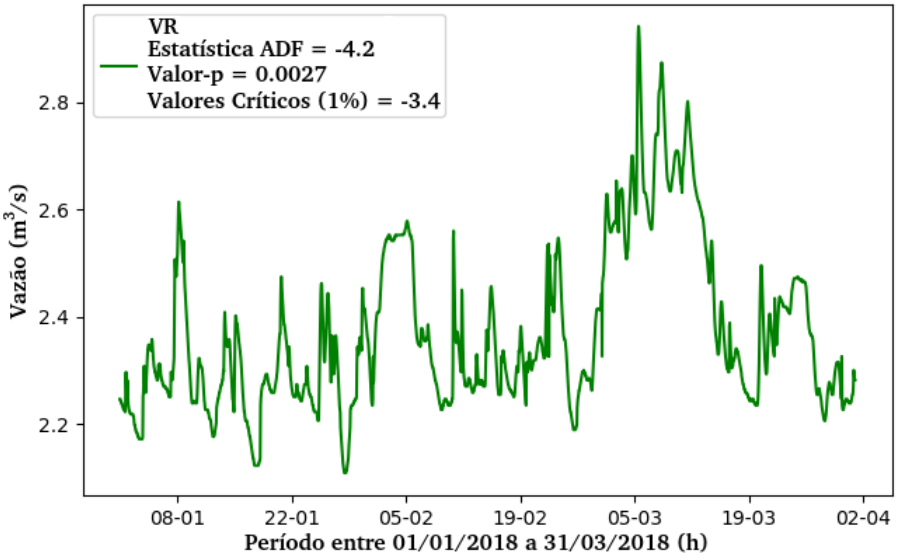

Figura 4.4: Análise de estacionariedade da Vazão de VR normalizada.

Fonte: Autor

4.3.4 Estabelecendo as Variáveis de Entrada e Saída

Primeiramente precisa-se confirmar se existe correlação entre as vazões Jusante 1, Jusante 2 e VR. Condição confirmada ao analisar a Figura 4.5 na qual todos os índices estão acima 0,5.

Em seguida imaginam-se quais serão os dados de entrada derivados das três vazões: Jusante 1, Jusante 2 e VR. Entre as variáveis criadas têm-se o *lag* $L = y_{t-1}$, a diferença do *lag* $D = y_{t-1} - y_{t-2}$ e a média móvel $M = (y_{t-1} - y_{t-2})/2$.

Assim as variáveis de entrada sugeridas são:

(a) L1: *Lag* da vazão da estação Jusante 1;

(b) D1: Diferença do *Lag* da vazão da estação Jusante 1;

(c) M1: Média móvel da vazão da estação Jusante 1;

(d) L2: *Lag* da vazão da estação Jusante 2;

(e) D2: Diferença do *Lag* da vazão da estação Jusante 2;

(f) M2: Média móvel da vazão da estação Jusante 2;

(g) L3: *Lag* da vazão da estação Santa Cecília;

(h) D3: Diferença do *Lag* da vazão da estação Santa Cecília;

(i) M3: Média móvel da vazão da estação Santa Cecília;

(j) Mês: Mês do ano em que foi realizada a medida da vazão;

(k) Dia: Dia do mês em que foi realizada a medida da vazão;

(l) Dia_Semana: Dia da semana em que foi realizada a medida da vazão;

(m) Hora: Horário em que foi realizada a medida da vazão.

Em relação a saída do modelo de aprendizado máquina considera-se apenas uma única variável de saída, sendo esta denominada de Objetivo. E representa a previsão da vazão horária com uma hora de antecedência da estação Santa Cecília.

Com as variáveis de entrada e saída estabelecidas, opta-se por deixar todos os dados na mesma escala, neste caso utiliza-se de uma função logarítmica, conforme 4.3.3.

Ao transformar os dados das entradas e saída, cria-se um novo arquivo CSV para guardar os dados já tratados, chama-se este arquivo de base de dados.

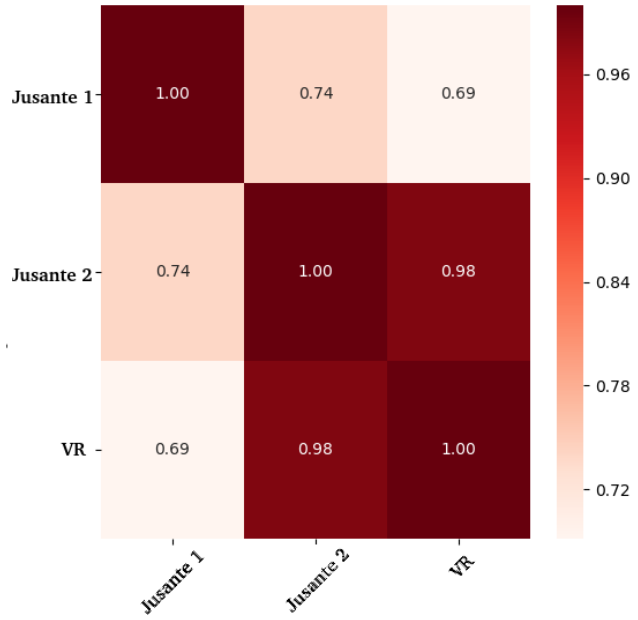

Figura 4.5: Correlação entre as três vazões utilizadas.
Fonte: Autor

4.4 Treinar o Modelo

De acordo com [1], deve-se seguir algumas recomendações ao desenvolver um modelo de ML, assim a depuração será simplificada. As práticas recomendadas são:

(a) Comece com um modelo simples que use um ou dois recursos. Começar com um modelo simples e facilmente depurável, ajuda a restringir as muitas causas possíveis para o fraco desempenho do modelo;

(b) Faça o modelo funcionar experimentando diferentes recursos e valores de hiperparâmetros. Mantenha o modelo o mais simples possível para simplificar a depuração;

(c) Otimize o modelo testando iterativamente estas alterações: adicionando recursos, ajuste de hiperparâmetros e aumento da capacidade do modelo;

(d) Após cada alteração no modelo, revise as métricas e verifique se a qualidade do modelo aumenta;

(e) Certifique-se de adicionar complexidade ao modelo de forma lenta e incremental.

4.4.1 Dados de Treinamento e Validação

Com a base de dados estabelecida e tratada precisa-se separá-la em duas partes: treinamento e validação. No trabalho de referência [31] os períodos propostos para a fase de treinamento e validação, correspondem respectivamente aos períodos entre 01/01/2018 e 24/03/2018 e entre 25/03/2018 e 31/03/2018.

Ao separar os dados entre treinamento e validação verificam-se se estas duas partes possuem as mesmas estatísticas, assim garantem-se que os dados de treinamento e validação tenham a mesma representatividade. Ou seja, caso isso não aconteça, tem-se um indicativo de que o ML não tem condições de realizar a previsão e fracassará na sua missão. Uma vez que, esta treinando com dados de características diferentes a dos dados de validação (previsão).

Busca-se então variar o tamanho destes períodos com o objetivo de aproximar suas estatísticas. Os novos períodos encontrados são 01/01/2018 a 11/03/2018 e 12/03/2018 a 31/03/2018. Uma comparação entre os períodos de referência e os novos períodos é apresentada na Figura 4.6.

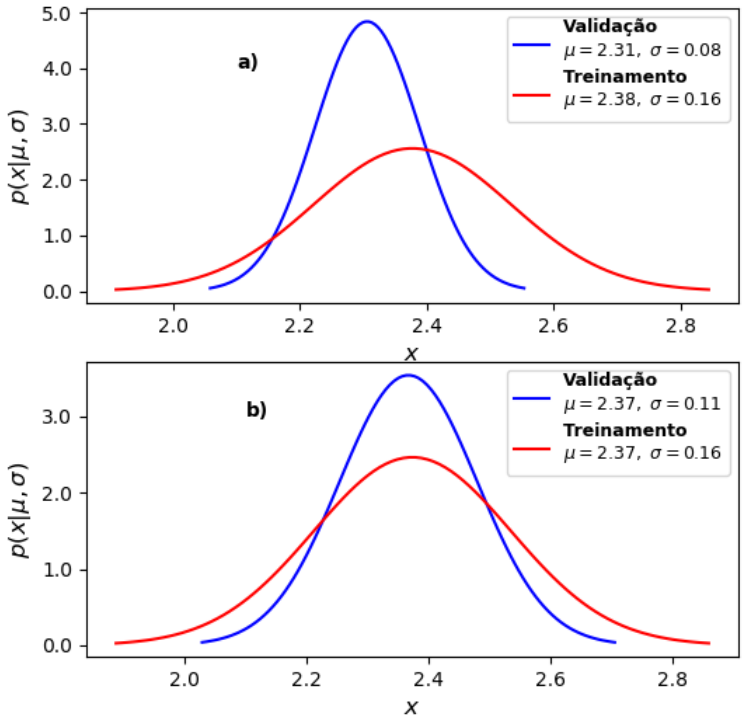

Figura 4.6: a) Períodos de referência. b) Novos períodos propostos.
Fonte: Autor

4.4.2 Escolha das Variáveis de Entrada

Escolhem-se as variáveis de entrada utilizando os critérios (a), (b) e (c), assim as variáveis selecionadas são aquelas que obtêm sucesso em todos os itens abaixo:

(a) Utiliza-se o fator FIV para eliminar a multicolinearidade. Com isso, retira-se a variável que apresenta FIV $>$ 10, eliminando uma a uma cada variável. Então primeiro retira-se a variável que apresenta o maior FIV acima de 10;

(b) Considerando que não existe mais multicolinearidade, aplica-se o Teste t de significância individual e assim as variáveis com p $>$ 0,049 são desconsideradas. Novamente as variáveis são eliminadas individualmente, ou seja, primeiro retira-se a variável com o maior p acima de p $>$ 0,049;

(c) Após os itens anteriores, testam-se as variáveis de entrada que sobraram em um modelo de ML simples. Incluem-se individualmente e em combinações as variáveis no modelo de ML. Desta forma, verifica-se qual variável ou qual combinação de variáveis contribui ou prejudica para a previsão. Isso significa testar todas as combinações possíveis entre as entradas e escolher a que fornece a melhor previsão. Esta técnica é aplicada na fase de treinamento.

4.4.3 Estratégias de Treinamento

Nesta Subseção descrevem-se as estratégias para aplicação do ML utilizando os modelos da Regressão Linear e do GLM.

4.4.3.1 Modelo de Regressão Linear

Inicialmente deve-se configurar os hiperparâmetros, por isso visando facilitar esta etapa, neste momento escolhe-se um algoritmo simples que possua poucos hiperparâmetros. Este algoritmo é o modelo de Regressão Linear, que de tão simples possui apenas a normalização como hiperparâmetro. E neste caso, os dados já foram normalizados na Seção 4.3, por tanto, não é necessário alterar ou configurar nenhum hiperparâmetro.

Um dos objetivos em utilizar um modelo de Regressão Linear é estabelecer um MAPE de referência para modelos complexos. E também, diminuir o tempo de desenvolvimento, pois quando um modelo simples atingi o resultado esperado, não existe motivo para investir em modelos com complexidade maior.

Na Subseção 4.4.1 faz-se uma pré seleção das variáveis de entrada, apesar disso uma segunda seleção é feita. No bloco a) da Figura 4.7 procura-se pela variável que apresenta a melhor métrica e depois tenta-se encontrar alguma combinação entre elas que possa reduzir ainda mais a métrica.

Para executar os treinamentos de ML utilizando a Regressão Linear aplica-se a biblioteca scikit-learn. A scikit-learn é uma biblioteca *Python* dedicada a aprendizado de máquima.

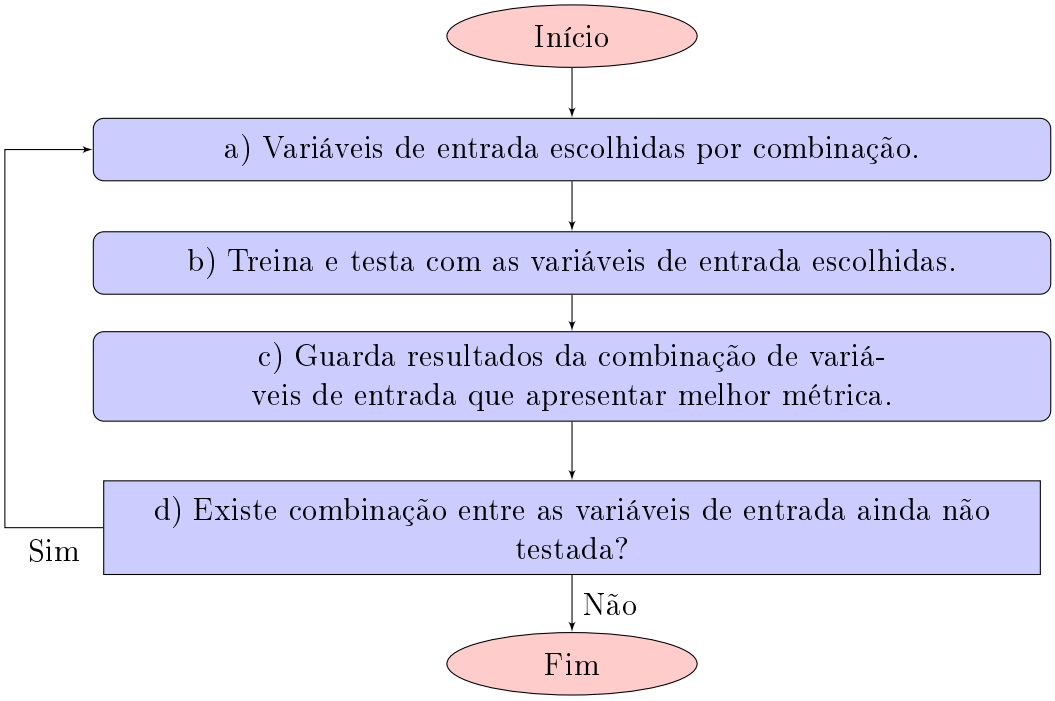

Figura 4.7: Etapas do ciclo de execução da Regressão Linear.
Fonte: Autor.

4.4.3.2 Selecionar Modelo com AutoML H2O

Como já citado no início da Seção 4.4 opta-se por iniciar os testes com um modelo de Regressão Linear simples e comparar os resultados com um modelo mais complexo. Para escolher o modelo complexo, utiliza-se da ferramenta chamada AutoML H2O. Ferramenta esta que testa automaticamente todos os algoritmos citados na Subseção 2.6.1

O AutoML estabelece o GLM como o melhor algoritmo, assim caso a Regressão Linear não apresente um desempenho satisfatório, o GLM seria uma segunda opção entre os algoritmos de ML.

4.4.3.3 Modelo de GLM

Para reduzir a complexidade do treinamento considera-se a avaliação das variáveis de entrada concluídas na Subseção 4.4.3.1, assim, nesta Subseção, preocupa-se apenas com a busca pela melhor configuração dos hiperparâmetros do GLM. Para encontrar a melhor solução do GLM, opta-se manter fixa a família de distribuição (Gaussiana) e variar a função *Link*. Sendo esta etapa representada no bloco b) da Figura 4.8.

Para executar este modelo de algoritmo no *Python* utiliza-se a biblioteca statsmodels.

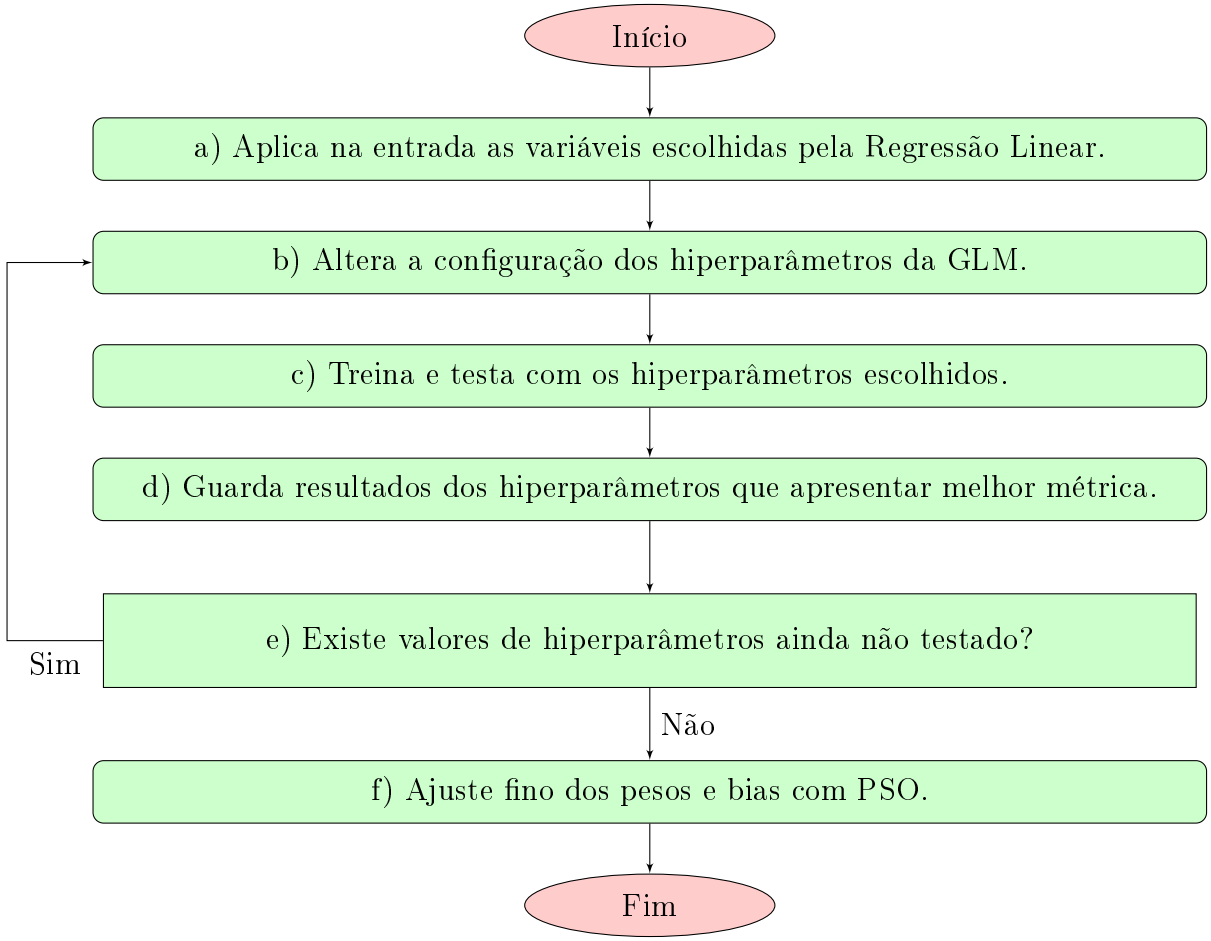

Figura 4.8: Etapas do ciclo de execução da GLM.
Fonte: Autor.

4.4.4 Métricas para Validação de Resultados

De acordo com [1], quando se fala em métricas é importante considerar os questionamentos abaixo.

(a) Qual a métrica utilizada?;

(b) Como os valores para realizar a métrica serão obtidos?;

(c) Quando os valores do item b serão obtidos?;

(d) Quanto tempo demorará para saber se o novo ML obteve sucesso ou falha?;

Como já mencionado, para avaliar o desempenho do ML, com a Série Temporal estudada, busca-se um MAPE $< 1{,}69\%$. Já para verificar de maneira geral a eficiência do ML, opta-se pela validação cruzada e por aplicar o ML treinado em 2018 nos dados de 2019 e 2020.

4.4.5 Previsão

Apesar de trabalhar com dados provenientes majoritariamente de vazões e prever a vazão do rio Paraíba do Sul na cidade de Volta Redonda, busca-se na verdade prever o nível do rio. Para isso a metodologia da curva-chave relaciona vazão com o nível através da Equação 4.1 [31].

$$Q = a(h - h_0)^b \tag{4.1}$$

onde Q é vazão, h_0 o nível quando a vazão é zero, h o nível atual, a e b coeficientes determinados pelo método dos quadrados mínimos. E seus valores são: $h_0 = 366\ m$, $a = 41,02$, $b = 1,93$ [31].

O Quadro 4.1 mostra três diferentes faixas de nível do rio e suas respectivas cores. Com esta forma, tenta-se destacar visualmente as situações que necessitam de atenção.

Identificação	Níveis para Inundação (m)
Cota Normal (Cor Verde)	Até 368,78
Cota de Alerta (Cor Amarelo)	Entre 368,78 e 370,3
Cota de Inundação (Cor Vermelho)	Acima de 370,3

Quadro 4.1: Níveis das Cotas para Inundação em relação ao nível do mar.
Fonte: [31]

4.5 Aplicar o Modelo

Para aplicar o modelo de previsão faz-se necessário utilizar um servidor web, que ficará responsável por processar as informações e divulgar as previsões.

Considerando que para realizar previsões é necessário que os dados representem aquela situação que se deseja prever. Estabelece-se que a fase de treinamento do modelo ocorrerá uma vez por ano. Sempre utilizando os dados do período de 01 de janeiro a 31 de março, período no qual existe uma maior incidência de enchentes.

Com isso, uma vez por ano o sistema automaticamente coleta os dados, transforma os dados, treina o ML e atualiza o modelo de previsão. Conforme a Figura 4.9 que representa o ciclo de funcionamento do modelo.

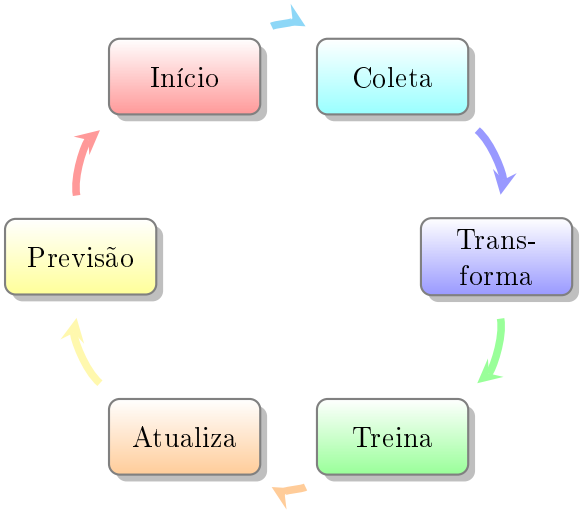

Figura 4.9: Ciclo da aplicação com treinamento (anual).
Fonte: [29] e Autor

Uma vez atualizado o modelo de previsão, seguem-se as etapas apresentadas na Figura 4.10. E passa-se a prever de acordo com a última atualização recebida, sendo que este ciclo se repete sempre que o aplicativo *web* for acessado.

No aplicativo *web* que permite o acesso as previsões, acessa-se também um histórico das vazões do dia atual.

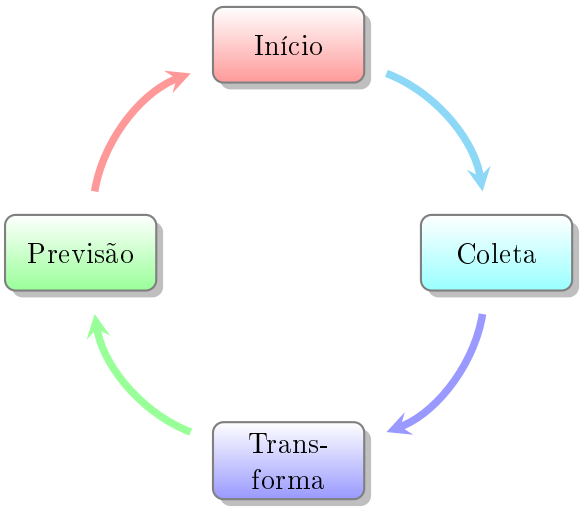

Figura 4.10: Ciclo da aplicação sem treinamento (horário).
Fonte: [29] e Autor

Capítulo 5

Resultados

Neste capítulo expõe-se os resultados obtidos utilizando a metodologia descrita no Capítulo 4.

Ao analisar os resultados deve-se considerar que todas as métricas (MAPE, RMSE, MSE, MAE e RMSLE) apresentadas neste capítulo são métricas totais. Ou seja, são métricas médias calculadas a partir do cálculo individual dos últimos 480 valores da série da vazão de VR. E o tempo de execução é o tempo de relógio, dado em segundos, que engloba os ciclos de treinamento e previsão do algoritmo de ML.

Os resultados apresentados nesta Seção foram obtidos através de simulações realizadas em um computador com as seguintes características:

- Sistema Operacional: Linux Mint, Release 18, 64bits;
- CPU: Intel(R) Core(TM)2 Duo CPU T5800 @ 2.00GHz;
- RAM: 4 GB;

5.1 Seleção das Variáveis de Entrada

Nas Subseções 5.1.1, 5.1.2 e 5.1.3 selecionam-se as variáveis de entrada de acordo com os critérios (a), (b) e (c) da Subseção 4.4.2 respectivamente.

5.1.1 Fator de Inflação da Variância

Em cada teste calcula-se o FIV de todas as variáveis e elimina-se a variável que possui o FIV > 10 e maior que os demais.

Dados dos testes realizados:

1º. Teste: [Month, Day, Weekday, Hour, M1, M2, M3, L1, **L2**, L3, D1, D2, D3]

2º. Teste: [Month, Day, Weekday, Hour, M1, M2, M3, **L1**, L3, D1, D2, D3]

3º. Teste: [Month, Day, Weekday, Hour, M1, M2, M3, **L3**, D1, D2, D3]

4º. Teste: [Month, Day, Weekday, Hour, M1, **M2**, M3, D1, D2, D3]

Após o 4º. teste todas as variáveis possuem FIV $<$ 10 e as variáveis selecionadas são apresentadas abaixo.

Resultado: [Month, Day, Weekday, Hour, M1, M3, D1, D2, D3]

5.1.2 Teste *t*

Ao aplicar o Teste *t* de significância elimina-se a multicolinearidade retirando em cada teste as variáveis com p $>$ 0,049.

Dados dos testes realizados:

1º. Teste: [**Month**, Day, Weekday, Hour, M1, M3, D1, D2, D3]

2º. Teste: [Day, **Weekday**, Hour, M1, M3, D1, D2, D3]

3º. Teste: [**Day**, Hour, M1, M3, D1, D2, D3]

4º. Teste: [Hour, M1, M3, **D1**, D2, D3]

Encerrada esta etapa restaram apenas as 5 variáveis mostradas a seguir.

Resultado: [Hour, M1, M3, D2, D3]

5.1.3 Regressão Linear

Nesta Subseção testa-se o desempenho do ML através das combinações das variáveis de entrada que restaram após as Subseções 5.1.1 e 5.1.2.

Estes testes consistem em aplicar cada variável individualmente na entrada do modelo de aprendizado de máquina. E depois testam-se como entrada todas as combinações geradas através destas variáveis. Estes testes são realizados pelo Algoritmo 2 do Apêndice A.

Dados dos testes realizados:

Teste: [**Hour**, **M1**, M3, D2, D3]

Resultado: [M3, D2, D3]

Considerando as variáveis M3, D2 e D3, a Tabela 5.1 apresenta o melhor MAPE atingido com um ML com Regressão Linear.

Antecedência da Previsão	MAPE(%)	Tempo de Execução(s)
1 hora	0,1824	0,0048

Tabela 5.1: Resultados do ML com Regressão Linear.
Fonte: Autor

5.1.4 Avaliação das Variáveis Resultantes

Ao término das Subseções 5.1.1, 5.1.2 e 5.1.3 avalia-se a correlação das variáveis de entrada M3, D2 e D3. Através da Figura 5.1 percebe-se que entre as variáveis selecionadas existem variáveis com alta e baixa correlação com a saída.

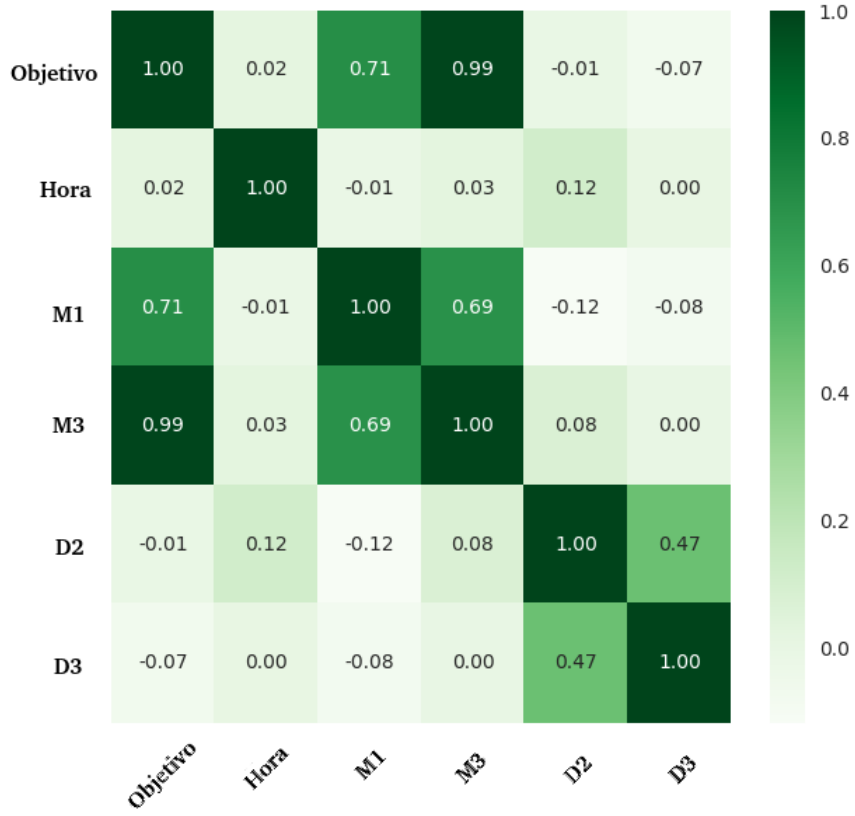

Figura 5.1: Correlação entre as principais variáveis de entrada.
Fonte: Autor

5.2 AutoML H_2O

Utiliza-se o AutoML H_2O para escolher o algoritmo de ML, que apresenta o melhor desempenho, para o problema estudado neste trabalho.

O Algoritmo 3 do Apêndice A gera os resultados apresentados na Tabela 5.2. E de acordo com estes resultados, seleciona-se o algoritmo GLM por ter a melhor performance.

Ainda durante os testes o AutoML H_2O indica a distribuição Guassiana e a função *Link Identity* como as melhores configurações.

Algoritmo	RMSE	MSE	MAE	RMSLE
GLM	0,0115	0,000132	0,0059	0,00344
XGBoost Grid 1	0,0145	0,000210	0,0097	0,00429
XGBoost Grid 4	0,0147	0,000218	0,0098	0,00438
DeepLearning	0,0148	0,000219	0,0101	0,00442

Tabela 5.2: Classificação dos melhores algoritmos gerada pelo AutoML H_2O.
Fonte: Autor

5.3 GLM

Como indica-se na Seção 5.2 o algoritmo GLM tem o melhor desempenho sobre vários outros algoritmos. Mas contrariando a expectativa e apesar do GLM possuir uma ótima performance, indicada por diversos índices, infelizmente o resultado do MAPE foi próximo ao alcançado com a Regressão Linear. E como MAPE foi estabelecido como métrica de desempenho deste trabalho, chega-se a conclusão que, por uma pequena margem, o algoritmo de ML com GLM é o mais indicado para este caso.

A Tabela 5.3 mostra a performance do GLM com 1 hora de antecedência, realizada com Algoritmo 4 do Apêndice A.

5.4 GLM com PSO

Para utilizar o PSO como forma de refinamento do resultado encontrado no GLM, utilizam-se os parâmetros encontrados no GLM como valores iniciais do PSO. E estabelece-se uma faixa de $\pm 10\%$ destes parâmetros para que o PSO busque por uma solução melhor.

Antecedência da Previsão	Modelo da Distribuição	Função *Link*	MAPE (%)	Tempo de Execução(s)
1 hora	Gaussiana	*Identity*	0,1815	0,0231
1 hora	Gaussiana	*Log*	0,5846	0,0157
1 hora	Gaussiana	*Inverse Power*	7,5994	0,0230

Tabela 5.3: Resultado do ML com GLM.
Fonte: Autor

Com isso deseja-se diminuir o tempo de processamento do PSO e melhorar o desempenho das previsões.

O resultado é gerado pelo Algoritmo 5 do Apêndice A e apresentado na Tabela 5.4.

Antecedência da Previsão	Número de Partículas	Número de Iterações	MAPE (%)	Tempo de Execução(s)
1 hora	100	200	0,1758	2,5570

Tabela 5.4: Resultado do refinamento utilizando PSO.
Fonte: Autor

5.5 Comparativo com o Trabalho de Referência

Como já citado, o objetivo do trabalho é buscar um algoritmo de ML que encontre resultados melhores que o modelo estatístico HWT, apresentado no estudo [31]. Pois este estudo utiliza os mesmos dados do corrente trabalho.

Assim, a Tabela 5.5 compara o resultado do trabalho atual com o de referência, com isso identifica-se que este trabalho possui uma MAPE dez vezes menor do que o trabalho de referência.

5.6 Comparativo entre as Previsões e o Valor Real

A correlação aplicada entre os valores previsto por Aprendizado de Máquina com Regressão Linear, GLM e GLM-PSO e o valor real, tem como objetivo comparar os

Referência	Modelo	MAPE(%)
Estudo Atual	GLM-PSO	0,1758
Estudo [31]	HWT	1,69

Tabela 5.5: Comparação entre o trabalho atual e o de referência.
Fonte: Autor

resultados obtidos por cada algoritmo, e assim identificar se essas previsões diferem significativamente entre elas e em relação ao valor real. O diagrama de calor da Figura 5.2 ilustra estas comparações.

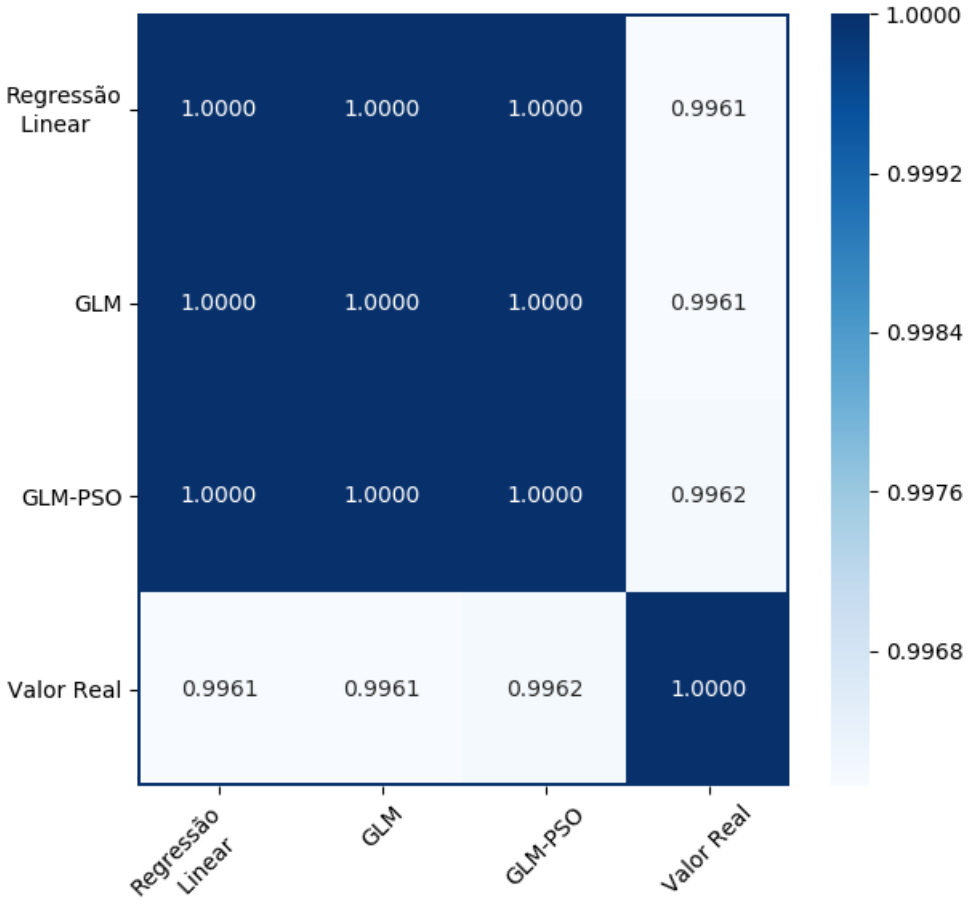

Figura 5.2: Correlação entre valores previstos e valor real.
Fonte: Autor

5.7 Consolidação das Previsões

Para confirmar o bom desempenho da metodologia, aplicam-se duas estratégias. A primeira é a validação cruzada e a segunda consiste em utilizar o modelo definido em 2018 com os dados de 2019 e 2020.

De acordo com [9], na previsão tradicional de séries temporais é prática comum reservar a parte do final de cada série temporal para teste e utilizar o resto da série para treinamento. Sendo a validação cruzada a abordagem mais comum para constatar a capacidade de generalização dos métodos preditivos. A Figura 5.3 mostra os resultados dos testes utilizando a validação cruzada, que tem como principal característica o aumento gradual dos dados utilizados.

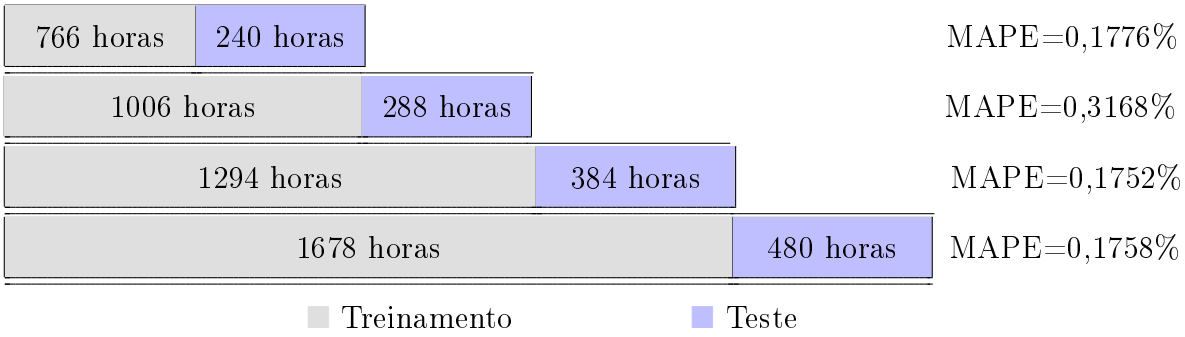

Figura 5.3: Resultados da validação cruzada para o GLM-PSO.
Fonte: Autor

Na segunda estratégia, aplicam-se os parâmetros obtidos com os dados de 2018 nos dados dos anos subsequentes, 2019 e 2020. Ou seja, treina-se a máquina com os dados de 2018 e depois realizam-se previsões com os dados de 2019 e 2020. A Tabela 5.6 apresenta os resultados do MAPE, notam-se que os valores das métricas não variam significativamente. Além disso, possuem valores extremamente baixos se comparados com o trabalho de referência. Assim ao plotar os gráficos destas previsões, conforme Figuras 5.4, 5.5 e 5.6, fica evidente a semelhança entre valor real e valor previsto.

Dados	MAPE(%)	R	NSE
2018	0,1758	0,9962	0,9924
2019	0,2139	0,9970	0,9939
2020	0,1857	0,9906	0,9812

Tabela 5.6: Resultados para o período entre 2018 a 2020.
Fonte: Autor

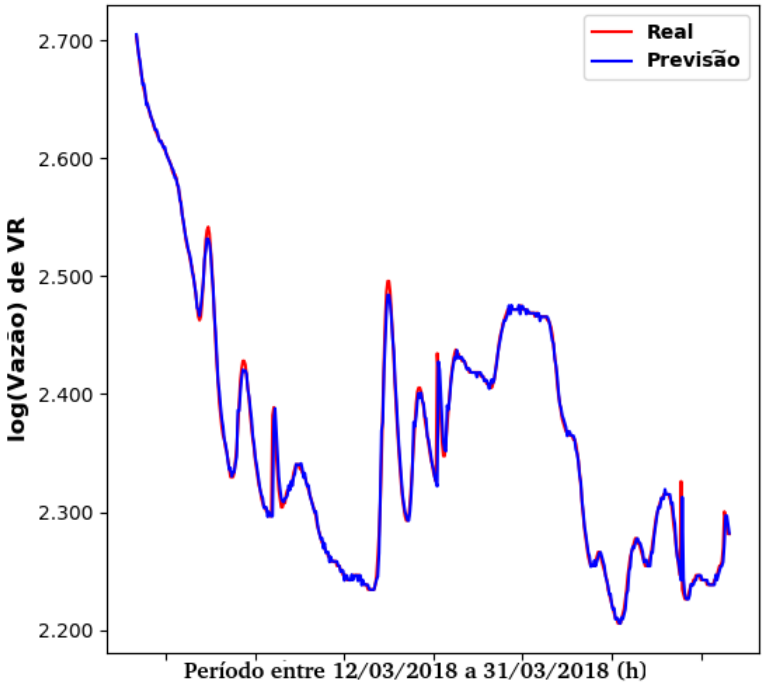

Figura 5.4: Previsão com os dados de 2018.
Fonte: Autor

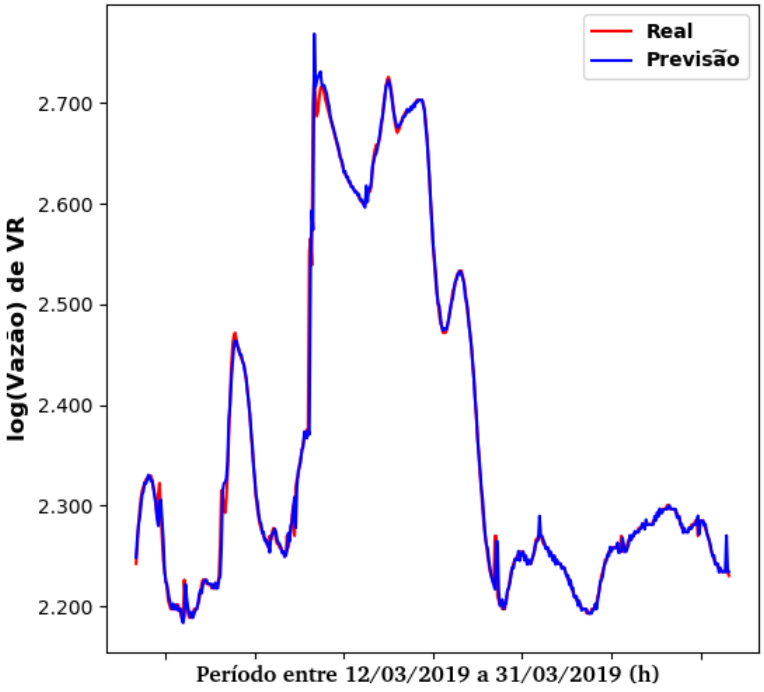

Figura 5.5: Previsão com os dados de 2019.
Fonte: Autor

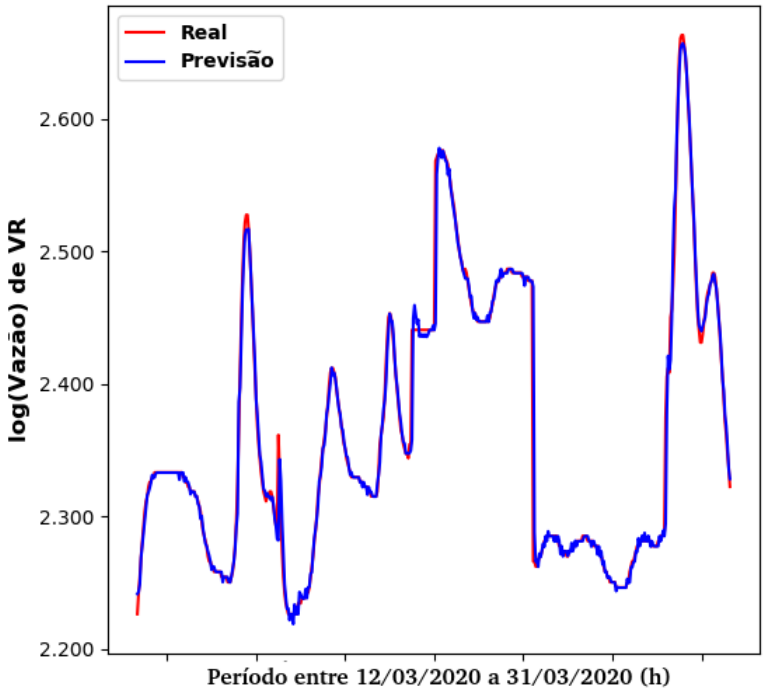

Figura 5.6: Previsão com os dados de 2020.
Fonte: Autor

Capítulo 6

Aplicativo *Web*

Neste capítulo exibem-se as informações do aplicativo *web* que realiza as previsões, com um breve relato de suas funcionalidades. Destaca-se que o aplicativo utiliza a linguagem de programação Python e o *Web FrameWork* Django, por isso pode-se hospedá-lo em qualquer servidor (Linux ou Windows) que tenha estas linguagens instaladas. O aplicativo caracteriza-se por ser uma página *web* responsiva, ou seja, que se adapta automaticamente ao dispositivo do usuário. E nele verifica-se a vazão histórica do dia atual e as previsões de vazão e nível para a próxima hora.

Na Figura 6.1 o digrama de caso de uso feito em Linguagem de Modelagem Unificada (do inglês *Unified Modeling Language*, UML), resume os detalhes das interações entre o SNIRH, aplicativo *web* e o Usuário. E através do link `http://mcct-ml.herokuapp.com` acessa-se o aplicativo.

6.1 Django

Django é um *framework web Python* de alto nível que permite o rápido desenvolvimento de *websites* seguros e de fácil manutenção. Construído por desenvolvedores experientes, o Django resolve a maior parte do trabalho de desenvolvimento *web* [3].

A escolha do Django deve-se ao fato de que este utiliza *Python* no desenvolvimento. Além de ser gratuito, de código aberto, com comunidade próspera e ativa, ótima documentação e muitas opções de suporte gratuito e pago [3].

No Django durante o funcionamento de um *website data-driven* (orientado a dados) tradicional, um aplicativo *web* aguarda solicitações *HyperText Transfer Protocol* (HTTP) do navegador da *web* (ou outro cliente). Quando recebe-se uma solicitação, o aplicativo

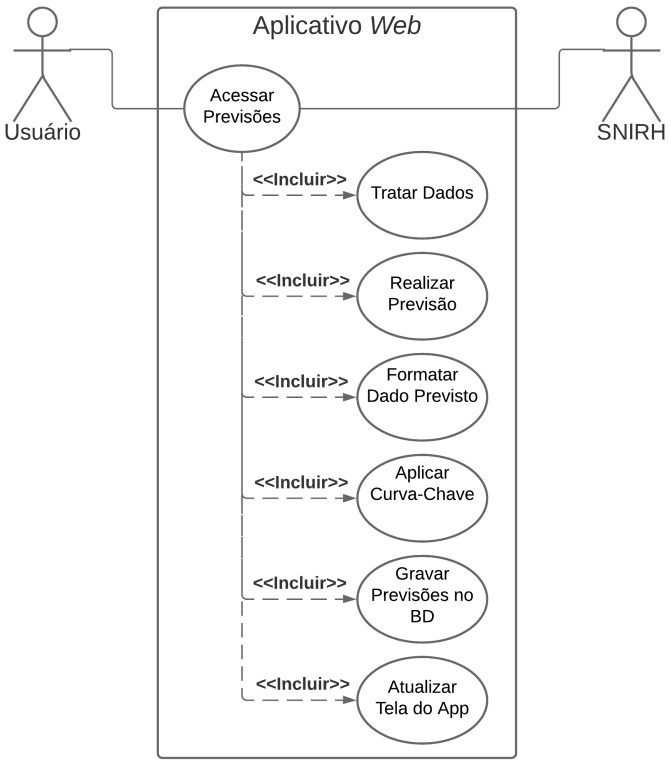

Figura 6.1: Estudo de caso.
Fonte: Autor

web calcula o que é necessário com base no *Uniform Resource Locator* (URL) e possivelmente nas informações dos dados POST ou GET. De acordo com a solicitação, pode-se ler ou gravar informações de um banco de dados ou executar outras tarefas necessárias para satisfazer a solicitação. O aplicativo *web* ao retornar com uma resposta para o navegador *web*, cria dinamicamente uma página *HyperText Markup Language* (HTML) para o navegador exibir, inserindo os dados recuperados em espaços reservados em um *template* HTML [3].

De acordo com [3], aplicativos *web* feitos com Django agrupam o código que manipula cada uma dessas etapas em arquivos separados, como representado na Figura 6.2:

(a) URLs: Embora seja possível processar solicitações de cada URL por meio de uma única função, simplifica-se a manutenção do código ao escrever uma função *view* separada para manipular cada recurso. Usa-se um mapeador de URLs para redirecionar as solicitações HTTP para a *view* apropriada, com base na URL da solicitação. O mapeador de URLs também processa padrões específicos de *strings* ou dígitos que aparecem em uma URL e os transmitem a uma função *view* como dados;

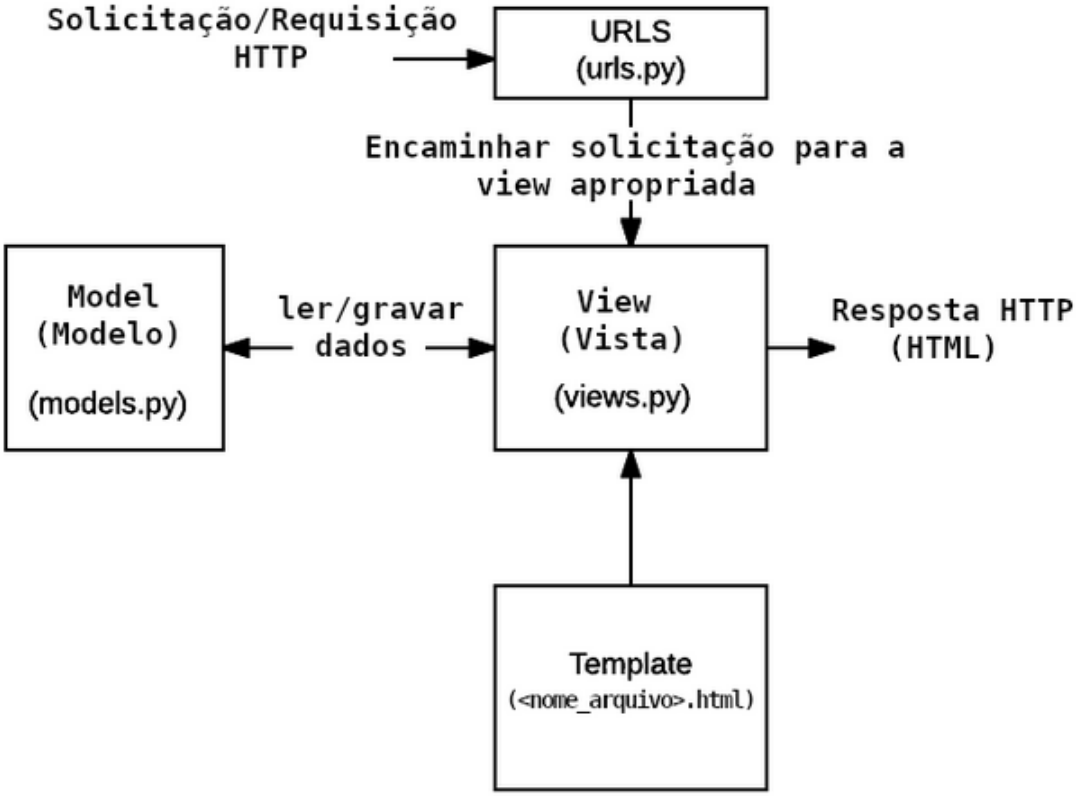

Figura 6.2: Arquitetura básica de um *website* feito em Django.
Fonte: [3]

(b) *View*: É uma função manipuladora de solicitações, que recebe solicitações HTTP e retorna respostas HTTP. As *views* acessam os dados necessários para satisfazer solicitações por meio dos modelos e utilizam-se dos *templates* para formatar as respostas;

(c) *Models*: Os modelos são objetos em *Python* que definem a estrutura dos dados de um aplicativo *web*, e fornecem mecanismos para gerenciar (adicionar, modificar e excluir) e consultar registros no banco de dados;

(d) *Templates*: É um arquivo de texto que define a estrutura ou o layout de um arquivo (como uma página HTML), com espaços reservados para representar o conteúdo real. Com uma *view* cria-se dinamicamente uma página HTML usando um tempalte HTML, preenchendo-o com os dados de um modelo.

6.1.1 Banco de Dados

O aplicativo *web* desenvolvido em Django cria, acessa e gerência o banco de dados por meio de objetos *Python* chamados de *models*. Os *models* também definem a estrutura do

armazenado dos dados, incluindo o tipo de campo, tamanho máximo do campo, valores padrão e etc. Esta é uma grande vantagem e facilidade do Django, pois só é necessário definir qual banco de dados utilizar, o resto é feito em *Python* através dos *models* [3]. No Apêndice B apresentam-se os *Models* utilizados.

Oficialmente os bancos de dados compatíveis com o Django são [3]:

(a) PostgreSQL

(b) MariaDB

(c) MySQL

(d) Oracle

(e) SQLite

Neste aplicativo utilizou-se o SQLite por ser um banco de dados relacional, na qual seu código fonte é de domínio público e pode ser utilizado gratuitamente para qualquer propósito [5].

6.2 Tabela de Previsão

Na Figura 6.3 tem-se a tabela de previsão, que é uma tabela preenchida dinamicamente de acordo com os dados extraídos do SNIRH. Nela é possível visualizar as previsões de vazão máxima e nível máximo para a próxima hora. Estas previsões são realizadas utilizando o conhecimento adquirido pelo ML durante a fase de treinamento, que ocorreu durante o desenvolvido desta dissertação.

Primeiro, realiza-se a previsão da vazão, e com a Equação 4.1 determinar-se o nível máximo do rio. E para o cálculo do MAPE considera-se os dados das 24 horas anteriores.

6.3 Imagem de Alerta

A imagem de alerta é uma animação que visa destacar o valor do nível do rio para a próxima hora, conforme a Tabela 4.1. Sendo que a imagem muda automaticamente de acordo com a previsão para o valor do nível do rio.

Previsão (18h20min)
Nível Máximo (m)
368.24
Vazão (m³/s)
194.58
MAPE (%) *
0.09

Figura 6.3: Previsão da vazão nível e cálculo MAPE.
Fonte: Autor

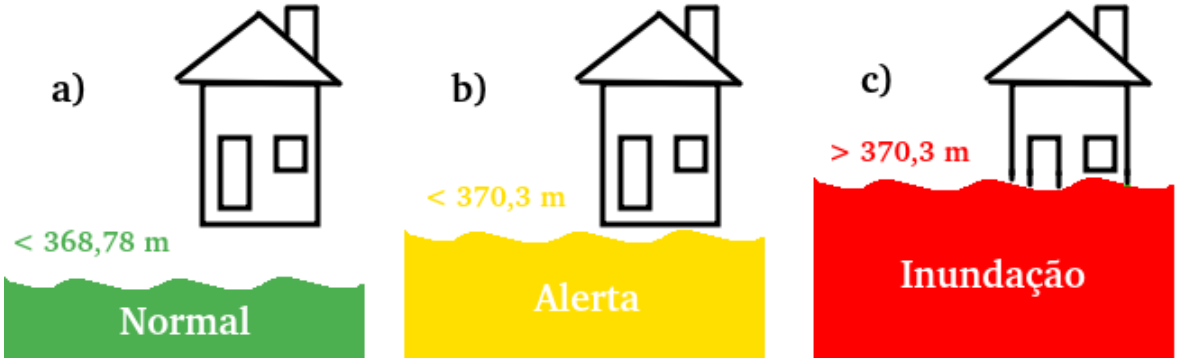

Figura 6.4: a) Cota Normal. b) Cota de Alerta. c) Cota de Inundação.
Fonte: Autor

6.4 Visualização do Histórico da Vazão

Para ajudar no acompanhamento do comportamento do rio, a Figura 6.5 mostra o histórico das vazões horárias do dia atual. O histórico é atualizado de hora em hora e tem seu início às 00h20min do dia atual. Ou seja, toda vez que um novo dia inicia, um novo histórico também inicia.

6.5 Visualização do Aplicativo *Web*

Ao acessar o aplicativo *web* o usuário visualiza as informações descritas nos itens anteriores conforme a Figura 6.6.

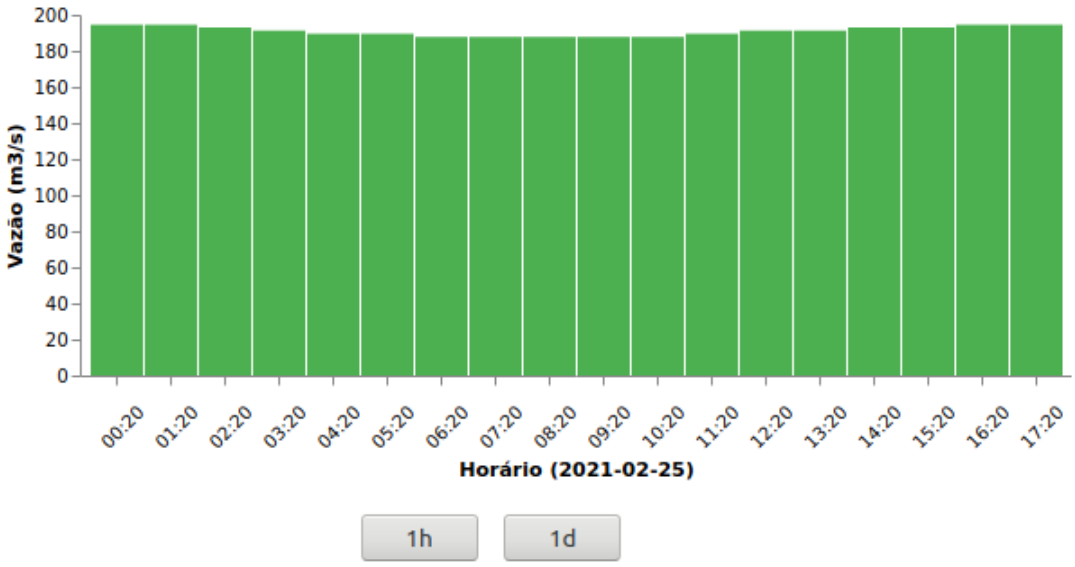

Figura 6.5: Histórico diário da vazão.
Fonte: Autor

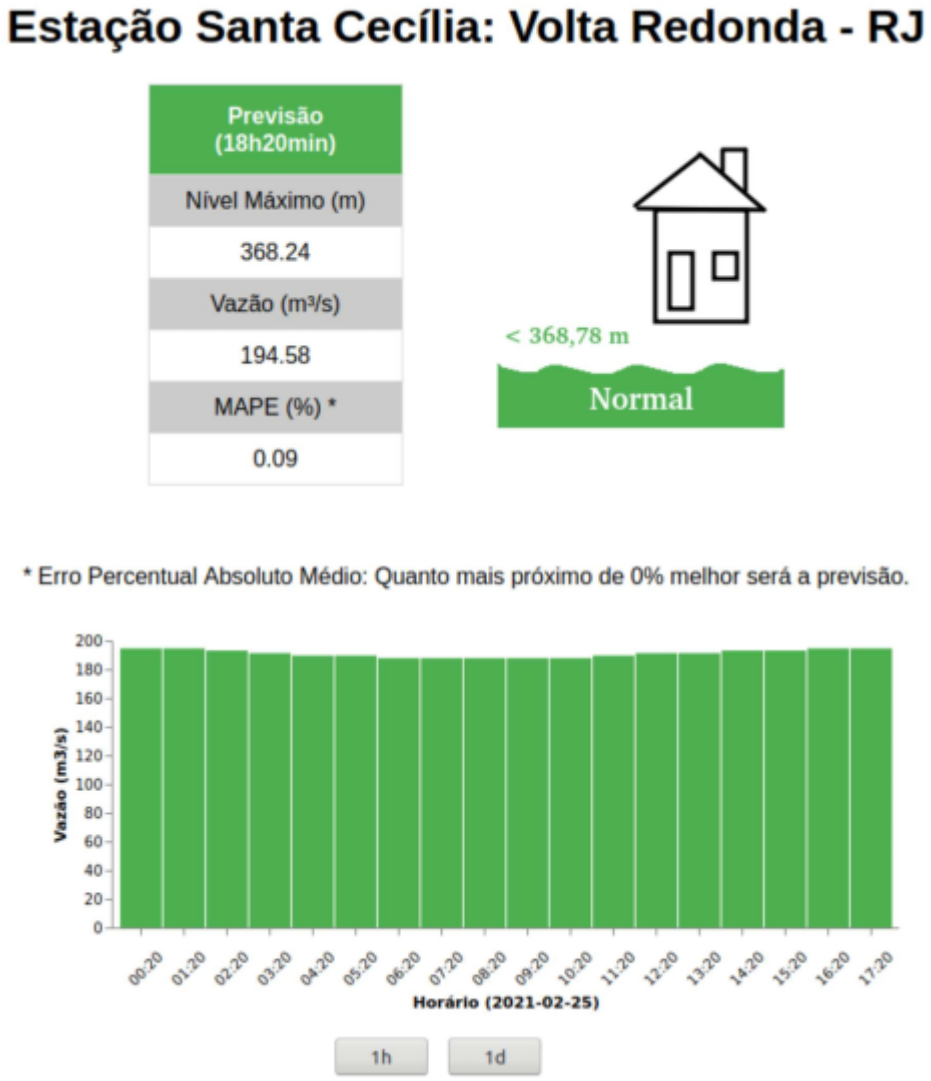

Figura 6.6: Visão geral do aplicativo *web*.
Fonte: Autor

Capítulo 7

Conclusões e Trabalhos Futuros

Neste capítulo discorrem-se sobre as conclusões do trabalho e sugestões para trabalhos futuros baseados nesta pesquisa.

7.1 Conclusões

O crescente avanço tecnológico e o aumento do desempenho dos algoritmos de Aprendizado de Máquina motivou uma busca para aplicar estes avanços na solução de um problema local. Conclui-se esta busca com o presente estudo, assim a previsão de vazão do Rio Paraíba do Sul, com o objetivo de prever enchentes, alia a tecnologia a um problema real. Que consiste em minimizar as perdas e transtornos causados pelas enchentes.

Desta forma, ao estabelecer um MAPE igual a 0,1758% para a previsão de vazão, e com isso possibilitar um alerta de inundações, o trabalho atingiu seu principal objetivo.

Provou-se ainda que a utilização de Aprendizado de Máquina, em conjunto ou não com técnicas de otimização, são excelentes opções para substituir o modelo estatístico HWT em previsões de vazão de rio. Uma vez que, o MAPE do ML foi dez vezes menor que o MAPE do HWT.

Outro ponto importante estabelecido foi a Metodologia, visto que esta influência diretamente nos resultados. Através dela verificou-se:

- A importância do tratamento dos dados de entrada, pois o MAPE só atingiu valores inferiores ao MAPE do HWT após os dados serem tratados;
- O bom desempenho do PSO, que conseguiu reduzir o MAPE estabelecido pelo GLM, que já era baixo em comparação com HWT;

- Que combinação do PSO com ML apresentou-se importante, principalmente para problemas com base de dados extensa. Pois nesta configuração é possível aproveitar a velocidade do ML com a precisão do PSO.

Entre as principais limitações do trabalho, esta a necessidade de se estabelecer critérios para considerar as modificações que ocorrem no leito do rio durante os anos. Com isso, é possível retreinar o modelo e revalidar as cotas definidas por [31], ao longo do tempo.

Por fim, o desenvolvimento de um aplicativo *web* mostrou-se viável e de fácil utilização. Assim, através do aplicativo previnem-se antecipadamente as inundações e elaboram-se ações estratégicas que minimizam seus efeitos negativos. E é importante como fonte de informação para outros setores envolvidos como, os da geração de energia, reservatório de água e empresas de transportes.

7.2 Trabalhos Futuros

Ao concluir este trabalho é possível vislumbrar várias linhas de pesquisa, mas um trabalho importante para o futuro seria aumentar a antecedência da previsão, que neste estudo é de uma hora de antecedência. É possível realizar previsões com 4 horas de antecedência, desde que tenham-se dados suficientes para treinamento. Para isso, sugerem-se duas possibilidades, na primeira incluem-se no treinamento dados com mesmo período, mas de outros anos. Na segunda utilizam-se técnicas que aumentam os dados de treinamento artificialmente.

Outras possibilidades são: testar novas variáveis de entrada como dados meteorológicos, estudar um modelo híbrido com HWT e ML(GLM) e aprimorar o aplicativo *web* inserindo outras previsões.

Referências

[1] Data preparation and feature engineering for machine learning. Google AI Education, Disponível em: <https://developers.google.com/machine-learning/data-prep>. Acessado em Julho de 2020.

[2] H2O.ai. Disponível em: <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/welcome.html#sparkling-water-users>. Acessado em Junho de 2020.

[3] Introdução ao django. MDM Web Docs, Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introduction>. Acessado em Junho de 2020.

[4] Sistema nacional de informações sobre recursos hídricos (SNIRH). Sistema HIDRO - Telemetria, Disponível em: <http://www.snirh.gov.br/hidrotelemetria/serieHistorica.aspx>. Acessado em Fevereiro de 2021.

[5] Sqlite. Disponível em: <https://www.sqlite.org/>. Acessado em Setembro de 2021.

[6] Abraham, A., Grosan, C., Ramos, V. *Swarm Intelligence in Data Mining*, vol. 34. Springer, 2006.

[7] Abudu, S., liang Cui, C., King, J. P., Abudukadeer, K. Comparison of performance of statistical models in forecasting monthly streamflow of kizil river, china. *Water Science and Engineering 3*, 3 (2010), 269–281.

[8] Aziz, N. A. A., Ibrahim, Z. Asynchronous particle swarm optimization for swarm robotics. *Engineering Proceedings Journal 41* (8 2012), 951–957.

[9] Bergmeir, C., Benítez, J. M. On the use of cross-validation for time series predictor evaluation. *Information Sciences 191* (5 2012), 192–213.

[10] Bisgaard, S., Kulahci, M. *Time Series Analysis and Forecasting by Example*, 1 ed. Wiley, 2011.

[11] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., Ljung, G. M. *Time Series Analysis: Forecasting and Control*, 5 ed. Wiley, 2016.

[12] Cazarotto, S. Teste de raiz unitária em modelo painel: Uma aplicação a teoria da paridade real de juros na américa latina. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, 2006.

[13] Ceron, R. AI, machine learning and deep learning: What is the difference? IBM IT Infrastructure Blog, Disponível em: <https://www.ibm.com/blogs/systems/ai-machine-learning-and-deep-learning-whats-the-difference/>. Acessado em Junho de 2020.

[14] Cowpertwait, P. S., Metcalfe, A. V. *Introductory Time Series with R.* Springer, 2009.

[15] de Resende Londe, L., Coutinho, M. P., Gregório, L. T. D., Santos, L. B. L., Érico Soriano. Desastres relacionados à Água no brasil: Perspectivas e recomendações. *Ambiente & Sociedade 17* (2014), 133–152.

[16] Downey, A. B. *Think Python.* O'Reilly Media, 2012.

[17] Fava, M. C., Mendiondo, E. M., Souza, V. C. B., de Albuquerque, J. P., Ueyama, J. Proposta metodológica para previsões de enchentes com uso de sistemas colaborativos. Em *XX Simpósio Brasileiro de Recursos Hídricos* (2013), p. 1–8.

[18] Gorodetskaya, Y., Tavares, G. G., da Fonseca, L. G., de Melo Ribeiro, C. B. Comparacão de métodos de aprendizado de máquina para a previsão de curto prazo de vaão do baixo curso do rio Paraíba do Sul. Em *XXI - ENMC: Encontro Nacional de Modelagem Computacional* (10 2018).

[19] Gujarati, D. N., Porter, D. C. *Econometria Básica*, 5 ed. McGraw-Hill, 2011.

[20] Honda, H., Facure, M., Yaohao, P. Os três tipos de aprendizado de máquina. LAMFO, Disponível em: <`https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/`>. Acessado em Junho de 2020.

[21] Hristev, R. M. *The ANN Book*, 1 ed. 1998.

[22] Hurwitz, J., Kirsch, D. *Machine Learning For Dummies.* John Wiley & Sons, Inc., 2018.

[23] Hussein, A., Agbinya, J., Satti, I. A survey on data mining techniques for water flow forecasting. *Australian Journal of Basic and Applied Sciences* (3 2020), 13–27.

[24] Hyndman, R. J., Athanasopoulos, G. *Forecasting: Principles and Practice*, 2 ed. OTexts, 5 2018.

[25] Jones, M. T. As linguagens da IA. IBM developerWorks, Disponível em: <`https://www.ibm.com/developerworks/br/library/cc-languages-artificial-intelligence/index.html`>. Acessado em Junho de 2020.

[26] JONG, P. D., HELLER, G. Z. *Generalized Linear Models for Insurance Data.* Cambridge University Press, 2008.

[27] Kalteh, A. M. Monthly river flow forecasting using artificial neural network and support vector regression models coupled with wavelet transform. *Computers & Geosciences 54* (4 2013), 1–8.

[28] Kavousizadeh, A., Ahamdi, A. High-performance approach for estimating stage-discharge curves in the open channels. *Journal of Hydrology 565* (7 2018), 197–213.

[29] Kottwitz, S. Smartdiagram examples. TEXample.net, Disponível em: <`https://texample.net/tikz/examples/feature/smartdiagram/`>. Acessado em Junho de 2021.

[30] Kulaksizoglu, T. Lag order and critical values of the augmented dickey-fuller test: A replication. *Journal of Applied Econometrics 30*, 6 (2015), 1010.

[31] Leandro, F. R. Sistema de alerta de inundação integrado à previsão de vazão. Dissertação de Mestrado, Universidade Federal Fluminense, Volta Redonda, RJ, Brasil, 2019.

[32] Liu, Z., Zhou, P., Chen, G., Guo, L. Evaluating a coupled discrete wavelet transform and support vector regression for daily and monthly streamflow forecasting. *Journal of Hydrology 519* (7 2014), 2822–2831.

[33] ll Koh, B., George, A. D., Haftka, R. T., Fregly, B. J. Parallel asynchronous particle swarm optimization. *International Journal for Numerical Methods in Engineering 67* (7 2006), 578–595.

[34] Lutz, M., Ascher, D. *Aprendendo Python*. O'Reilly Media, 2007.

[35] Malfatti, M. G. L., de Oliveira Cardoso, A., Hamburger, D. S. Modelo empírico linear para previsão de vazão de rios na usina hidrelétrica de itaipu - bacia do rio paraná. *Revista Brasileira de Meteorologia 33*, 2 (2 2018), 257–268.

[36] Montgomery, D. C., Runger, G. C. *Applied statistics and probability for engineers*, 3 ed. John Wiley & Sons, Inc., 2002.

[37] Morettin, P. A., Toloi, C. M. C. *Análise de Séries Temporais*, 2 ed. Blucher, 2006.

[38] Nazir, H. M., Hussain, I., Ahmad, I., Faisal, M., Almanjahie, I. M. An improved framework to predict river flow time series data. *PeerJ Journals* (7 2019).

[39] Paparoditis, E., Politis, D. N. The asymptotic size and power of the augmented dickey-fuller test for a unit root. *Econometric Reviews 37*, 9 (2018), 955–973.

[40] Parsopoulos, K. E., Vrahatis, M. N. *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference, 2010.

[41] Sanikhani, H., Kisi, O. River flow estimation and forecasting by using two different adaptive neuro-fuzzy approaches. *Water Resour Manage 26* (2 2012), 1715–1729.

[42] Shi, Z. *Advanced Artificial Intelligence*, vol. 1. World Scientific, 2011.

[43] Sridharan, R. 6.s085 statistics for research projects: Iap 2015. MIT Class, Disponível em: <`http://www.mit.edu/~6.s085/`>. Acessado em Novembro de 2020.

[44] Teixeira, L. L., Siqueira, P. H. Previsão de séries de vazões com a metaheurística pso. *Revista Ciências Exatas e Naturais 17* (7 2015), 207–224.

[45] Thomas, P. *Artificial Intelligence*. Thomson Gale, 2005.

[46] Tu, Y., Chan, N., Wang, Q. Testing for a unit root with nonstationary nonlinear heteroskedasticity. *Econometric Reviews 39*, 9 (2020), 904–929.

[47] van Rossum, G. Python. Disponível em: <`https://www.python.org/`>. Acessado em Junho de 2020.

[48] VASSALLI, L. C. Aplicação de redes neurais lstm para a previsão de curto prazo de vazão de rio paraíba de sul. Monografia (Engenharia Computacional), UFJF, Juiz de Fora, MG, Brasil, 2018.

APÊNDICE A – Algoritmos Desenvolvidos na Fase de Testes

Algoritmo 1 Algoritmo de Aquisição e Tratamento dos Dados

1: $estacaoJ1$ = leDadosSNIRH(Estação Jusante 1);
2: $estacaoJ2$ = leDadosSNIRH(Estação Jusante 2);
3: $estacaoSTA$ = leDadosSNIRH(Estação Santa Cecília);
4: $dadosEstacoes$ = unificaDadosEstações($estacaoJ1$, $estacaoJ2$, $estacaoSTA$);
5: $baseDados$ = tratamentoDados($dadosEstacoes$);
6: $Localizacao$ = salvaCSV($baseDados$, $c : \backslash projeto \backslash baseDados.csv$);
7: **return** $Localizacao$;

Algoritmo 2 Seleção de Variáveis (ML com Regressão Linear)

```
1: xtr, xval, ytr, yval = leCSV(Localizacao)          {% Carrega base de dados tratada}
2: comp = len(xtr.columns)
3: for step in range(comp) :
4: varMenorErro = None
5:     for var in xtr.columns :
6:         if var in aceitas :
7:             continue
8:         md1 = LinearRegression(nJobs = −1)          {% Configura ML}
9:         md1.fit(xtr[aceitas + [var]], ytr)          {% Treina ML}
10:        p = md1.predict(xval[aceitas + [var]])      {% Realiza previsão}
11:        mape = np.mean(np.abs((yval − p)/yval)) ∗ 100   {% Calcula o MAPE}
12:        if mape < valorMenorErro :
13:            varMenorErro = var
14:            valorMenorErro = mape
15:            yvalMenor = yval
16:            pMenor = p
17:    if varMenoErro is None :
18:        break
19:    aceitas.append(varMenorErro)                    {% Guarda variáveis selecionadas}
```

Algoritmo 3 Seleciona Algoritmo com AutoML H2O

```
xtr, xval, ytr, yval = leCSV(Localizacao)        {% Carrega dados de treinamento e validação}

h2o.init()                                       {% inicializa AutoML H2O}
hfTrain = h2o.H2OFrame(xtr)                      {% Coloca dados de treinamento em formato H2O}
hfTtest = h2o.H2OFrame(xval)                     {% Coloca dados de validação em formato H2O}

A =' Y'                                          {% Indica nome da variável de saída}
B = ['D2', 'M3', 'D3']                           {% Indica nome das variáveis de entrada}

aml = H2OAutoML(maxModels = 20)                  {% Número de máximo de modelos de treinamento}

{% Treina AutoML H2O}
aml.train(x = B, y = A, trainingFrame = hfTrain, leaderboardFrame = hfTest)
preds = aml.leader.predict(hfTest)               {% Previsão com melhor modelo da classificação}

hfResults = pd.DataFrame()                       {% Cria variável frame}
hfResults['Y'] = xval['Y'].resetIndex(drop = True)   {% Pega saída de validação original}
hfResults['P'] = h2o.asList(preds, usePandas = True) {% Pega saída de validação prevista}

{% Calcula MAPE do melhor modelo}
mape = np.mean(np.abs((hfResults['Y'] - hfResults['P'])/hfResults['Y'])) * 100

lb = aml.leaderboard                             {% Imprime classificação com resultados dos modelos}
print(lb.head(rows = lb.nrows))

print(aml.leader)                                {% Imprime dados do melhor modelo}
```

Algoritmo 4 ML com GLM

```
xtr, xval, ytr, yval = leCSV(Localizacao)        {% Carrega base de dados tratada}

{% Opções testadas da função Link}
opcoes = [sm.families.links.identity(), sm.families.links.log(), sm.families.links.inversePower()]
 for link in opcoes :
    gaussIdent = sm.GLM(ytr, xtr, family = sm.families.Gaussian(link))   {% Configura ML}
    gaussIdentResults = gaussIdent.fit()                                 {% Treina ML}
    p = gaussIdentResults.predict(xval)                                  {% Realiza previsão}
    mape = np.mean(np.abs((yval - p)/yval)) * 100                        {% Calcula o MAPE}
    print("Mape : %f\n" % mape)                                          {% Imprime MAPE de cada função Link}
```

Algoritmo 5 GLM com PSO

```
{% Testa possiveis soluções}
def testaSolucoes(xn) :
    size = len(X)
    YY = np.zeros(size)
    YY[:] = xn[0] * X[:, 0] + xn[1] * X[:, 1] + xn[2] * X[:, 2]
    mape = np.mean(np.abs(np.divide(np.subtract(Y, YY), Y))) * 100
    return mape

{% Gera soluções e armazena resultados}
def costFunction(xt) :
    nParticles = xt.shape[0]
    j = [testaSolucoes(xt[i]) for i in range(nParticles)]
    return np.array(j)

xtr, xval, ytr, yval = leCSV(Localizacao)          {% Carrega base de dados tratada}
X = np.array(xtr[0 : len(xtr.index)])
Y = np.array(ytr[0 : len(ytr.index)])

lb = [−1.198, 0.9, −0.396]                         {% Limite inferior de procura}
ub = [−1.045, 1.1, −0.309]                         {% Limite superior de procura}
bounds=(lb, ub)
options = {'c1' : 0.5, 'c2' : 0.3, 'w' : 0.9}      {% inicializa Swarm}
dimensions = 3                                     {% Estabelece número de entradas}

{% Configura PSO}
optimizer = ps.single.GlobalBestPSO(nParticles = 100, dimensions = dimensions, options =
    options, bounds = bounds)
optimizer.initPos = [−1.121627, 1.000011, −0.352618]   {% Valores iniciais do PSO}
cost, pos = optimizer.optimize(cost_function, iters = 200)   {% Realiza a otimização}
ppp = pos[0] * xval['D2'] + pos[1] * xval['M3'] + pos[2] * xval['D3']   {% Realiza previsão final}
mapep = np.mean(np.abs((yval − ppp) / yval)) * 100   {% Calcula MAPE final}
```

APÊNDICE B – Estrutura dos *Models*

Algoritmo 6 Models - Imagens de Alerta

```
1: class alertImages(models.Model):
2:     imgAlert = models.ImageField();          {% Localização da imagem}
3:     imgColor = models.TextField();           {% Nome da imagem}
```

Algoritmo 7 Models - Registros da previsão de 1 dia

```
1: class forecastDay(models.Model):
2:     flow = models.FloatField();              {% Vazão}
3:     level = models.FloatField();             {% Nível}
4:     mape = models.FloatField();              {% Métrica}
5:     dateHour=models.DateTimeField(default=datetime.now()-timedelta(hours=3)); {%Data, Hora}
6:     def _ float_ (self):
7:         return self.flow                     {% Retorna Vazão}
```

Algoritmo 8 Models - Registros da previsão de 1 hora

```
1: class forecastHour((models.Model):
2:     flow = models.FloatField();              {% Vazão}
3:     level = models.FloatField();             {% Nível}
4:     mape = models.FloatField();              {% Métrica}
5:     dateHour=models.DateTimeField(default=datetime.now()-timedelta(hours=3)); {%Data, Hora}
6:     def _ float_ (self):
7:         return self.flow                     {% Retorna Vazão}
```