PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Marcelo Pereira Cunha

APLICAÇÃO DE TÉCNICAS DE DEEP LEARNING EM PREVISÃO DE VARIÁVEIS CLIMÁTICAS

Belo Horizonte
2021

Marcelo Pereira Cunha

APLICAÇÃO DE TÉCNICAS DE DEEP LEARNING EM PREVISÃO DE VARIÁVEIS CLIMÁTICAS

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte
2021

SUMÁRIO

1. Introdução

1.1. Contextualização

A previsão do tempo através de métodos numéricos possui Vilhelm Bjerknes (1904) como um importante pesquisador, pois o mesmo identificou que o estado futuro da atmosfera poderia, ser obtido pela integração das equações diferenciais que governam o seu comportamento, sendo a atmosfera modelada como um sistema dinâmico (Ynoue, 2011). As condições iniciais das equações diferenciais utilizadas, descreveriam um estado observado da atmosfera. O pesquisador britânico Lewis Fry Richardson foi o primeiro a realizara integração numérica compreensível dessas equações.

Richardson, partindo de observações dos dados meteorológicos das 7 UTC do dia 20 de maio de 1910, calculou a derivada da pressão na Alemanha em função do tempo. A variação prevista pelo seu modelo numérico para a variável pressão no período de 6 horas foi de 146 hPa, valor muito maior do que aquele realmente observado. Todavia, mesmo com um resultado aquém do esperado, o trabalho de Richardson mostrou quais seriam os obstáculos a sobrepujar no campo de previsões climáticas através de modelos numéricos.

Para realizar a previsão de uma variável atmosférica para o intervalo de apenas um dia, mesmo nos dias atuais, se faz necessário uma exorbitante quantidade de cálculos, sendo demandada considerável rapidez no processamento destes cálculos. Além disso, os dados utilizados para representar o estado inicial da atmosfera não eram suficientes na experiência realizada por Richardson. Através dos experimentos realizados, pode-se observar também que, caso as técnicas numéricas utilizadas não fossem devidamente implementadas, poderiam ser gerados erros que se propagariam ao longo de toda a computação do modelo numérico proposto, o que reduziria a precisão e acurácia da previsão.

Ao longo da década de 1960, iniciou-se a utilização dos modelos numéricos globais de tempo em previsões climáticas, possibilitando realizar a previsão de como se faria a evolução da atmosfera a longo prazo. Considerando que o tempo atmosférico e o clima possuem escalas temporais distintas, ajustes nos modelos se tornaram necessários para a representação dos processos físicos que regem cada escala de

tempo. A previsão de tempo e de clima é um ramo altamente especializado e em constante evolução.

Um aspecto importante a se ressaltar é a importância em diferenciar a previsão de tempo da previsão de clima. Conforme citado anteriormente, a previsão de tempo é realizada para poucos dias consecutivos, fato este que se deve ao escasso conhecimento das condições iniciais (estado observado da atmosfera) fornecidas aos modelos numéricos utilizados.

Na previsão climática não se tem interesse em prever com exatidão o local e o momento da ocorrência de um sistema atmosférico como na previsão do tempo e, sim, que seja realizada uma simulação pelo modelo. Caso o objetivo seja prever como será a precipitação no outono austral, uma simulação é feita para essa estação nesta região; após isto é calculado o total acumulado de precipitação e, como último passo, este é comparado com o valor climatológico (que é proveniente da média de um longo período de precipitação observada). Desta maneira, é possível descobrir se tal estação será mais úmida ou mais seca do que a climatologia observada previamente.

A principal ferramenta utilizada atualmente no campo das previsões de tempo e de clima são os modelos numéricos conhecidos como Modelos Numéricos de Circulação Geral – MCG, sendo um conjunto de equações físicas descritas em forma numérica e resolvidas com o auxílio de computadores.

Os procedimentos gerais para uma previsão de tempo ou de clima incluem 3 etapas (análise, previsão e pós-processamento).

Na fase da análise, as observações meteorológicas são fornecidas a programas computacionais, que preparam os dados para os modelos de previsão. Uma vez que a rede de observações global de dados não cobre regularmente a superfície da Terra, as observações meteorológicas são submetidas a métodos matemáticos para se tornarem uniformes, isto é, valores iniciais com espaçamento horizontal regular. Embora isso seja somente um passo preparatório, a tarefa é difícil, pois há milhões de dados provenientes de diferentes fontes (satélites, navios, estações meteorológicas de superfície etc.) e que não necessariamente foram medidos no mesmo horário. Além disso, nenhuma das medidas é completamente livre de erros. Assim, é necessário o máximo possível dos erros para produzir campos atmosféricos consistentes. O procedimento da análise também inclui a junção de condições iniciais observadas e previsões do modelo numérico para a elaboração de condições iniciais para esse modelo. Em síntese, a fase da análise tem como objetivo criar um conjunto de dados

com espaçamento horizontal uniforme para ser fornecido aos modelos de previsão como condição inicial. Terminada a etapa da análise, o modelo pode ser executado para produzir as previsões.

Ao longo da etapa de previsão, utiliza-se um modelo numérico cujo objetivo é resolver as equações básicas que descrevem o comportamento da atmosfera. Abaixo são exibidas as equações. Vale ressaltar que o vento possui 3 componentes nas direções ortogonais x, y e z, ou seja, V = (u , v , w).

Figura 1: Equações do modelo numérico de previsão do tempo

Equação	Descrição
$\frac{\partial \vec{V}}{\partial t} + \vec{V}.\vec{\nabla}\vec{V} = -\frac{\vec{\nabla}p}{\rho} - 2\vec{\Omega} \times \vec{V} + \vec{g} + \vec{F}v$	Conservação do movimento: Força = massa . aceleração Descreve como o movimento horizontal do ar (o vento meridional: norte-sul e o vento zonal: leste-oeste) evolui ao longo do tempo cronológico.
$C_p\left(\frac{\partial T}{\partial t} + \vec{V}.\vec{\nabla}T\right) = \frac{1}{\rho}\frac{dp}{dt} + Q + F_T$	Conservação da energia: entrada de energia = aumento da energia interna + trabalho realizado Descreve quais mudanças na temperatura do ar resultam da adição/subtração de calor ou devido à expansão/compressão do ar.
$\frac{\partial \rho}{\partial t} + \vec{V}.\vec{\nabla}\rho = -\rho\vec{\nabla}.\vec{V}$	Conservação da massa: A soma dos gradientes do produto da densidade e velocidade do vento nas 3 direções ortogonais é zero. Descreve que a massa do volume de ar não muda, ou seja, a massa que entra no volume de ar é igual à massa que sai.
$\frac{\partial q}{\partial t} + \vec{V}.\vec{\nabla}q = \frac{S_q}{\rho} + F_q$	Conservação da água: Descreve o complexo transporte de água em suas diversas formas e estágios dentro do ciclo hidrológico.
$p = \rho.R.T$	Conservação do estado: Descreve a relação entre a pressão, o volume, a temperatura e quantidade de um gás ideal.

Fonte: USP (Universidade de São Paulo) Departamento de Geografia

As equações utilizadas nos modelos de previsão de tempo e clima são resolvidas em pontos de grade, ou seja, a superfície do planeta é dividida em latitudes e longitudes e a intersecção delas fornece os pontos de grade. Já a distância entre 2 pontos de grade vizinhos fornece a resolução horizontal do modelo. A atmosfera também é dividida em níveis verticais, sendo esta divisão conhecida como resolução vertical.

Para uma previsão futura, o modelo é iniciado com as condições iniciais observadas e a previsão para aquele momento. Também é informado no intervalo de tempo (resolução temporal previamente determinada), que ele vai fazer a integração das equações até chegar ao final do tempo desejado (por exemplo, 24 h, 48 h, 72 h). As condições da atmosfera observadas são utilizadas apenas na primeira integração, pois nas demais o modelo utilizará a previsão feita anteriormente como condição

inicial. Com isso, é elaborado um conjunto de previsões: para algumas horas, um dia ou alguns dias etc.

Figura 2: Modelo de Temperatura, Pressão e Umidade

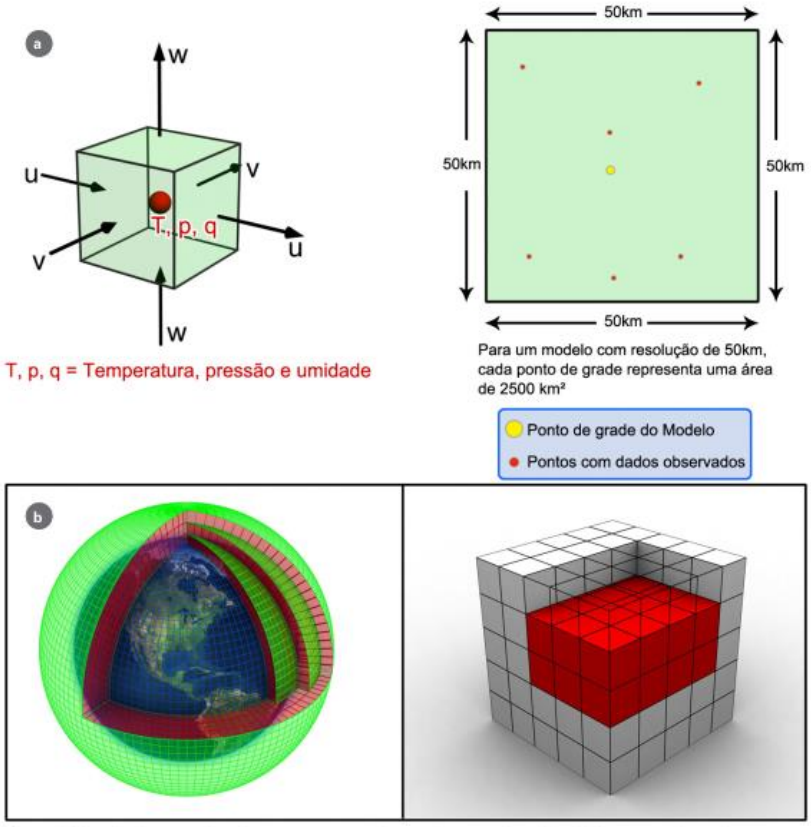


Figura 12.2: a. Lado esquerdo: volume representando parte da atmosfera; lado direito: ponto de grade e observações vizinhas; **b.** Vários volumes representando toda a extensão horizontal e vertical da atmosfera no globo.

Fonte: USP (Universidade de São Paulo) Departamento de Geografia

Durante a etapa de pós processamento, embora os modelos numéricos façam cálculos para intervalos de tempo de minutos, os arquivos de saída com os resultados dos cálculos são reduzidos para intervalos regulares de horas (por exemplo, a cada 12 horas) e em pontos de grade. Com os arquivos de saída dos modelos, os meteorologistas elaboram mapas de diferentes variáveis atmosféricas e, após o estudo desses mapas, elaboram os boletins da previsão de tempo e clima, dependendo do modelo utilizado. Essas atividades são chamadas de pós-processamento. Os boletins são importantes para o planejamento de várias atividades, como por exemplo o setor agrícola, hidrelétricas, comercial, turismo, lazer, prevenção de desastres, entre outras.

Os modelos utilizados para a previsão de tempo e clima podem ser globais ou regionais. Os modelos globais simulam as condições atmosféricas globalmente. Em

sua maioria tem grade com resolução horizontal em torno de 200 km, 28 níveis verticais e são inicializados com condições iniciais de todo o planeta. Esses modelos possuem grande eficiência em simular as características gerais da circulação atmosférica de larga escala, mas não as características locais como brisas de vale e de montanha. Atualmente, é inviável aumentar a resolução horizontal de tais modelos, pois isso envolve elevado custo computacional devido ao maior tempo de processamento das previsões que envolvem um número massivo de dados e mais espaço físico de disco necessário para armazená-las. Por isso, para obtenção de informações mais detalhadas da atmosfera, são utilizados os modelos regionais (ou de área limitada), que simulam as condições atmosféricas em pequenas porções do planeta e possuem grade com resolução horizontal, em torno de 20 km. Esses modelos, por terem resolução horizontal maior do que os globais, representam melhor a superfície e os fenômenos mais regionais, tornando os resultados mais precisos. Os modelos de área limitada, além das condições iniciais, precisam de condições de fronteira lateral, já que abrangem pequenas porções do globo. As fronteiras laterais têm possibilidade de serem provenientes das saídas de modelos de circulação geral da atmosfera.

1.2. O problema proposto

Considerando a elevada complexidade em resolver equações diferenciais parciais computacionalmente, tendo em vista também o massivo volume de dados referentes à condições iniciais que são demandados para uma resolução eficaz das equações que compõem os modelos climáticos, bem como os grandes erros provocados por possíveis falhas e/ou inconsistências nos dados que compõe as condições iniciais supracitadas, o objetivo do presente trabalho é propor um modelo simplificado, que possua um número reduzido de variáveis de entrada para realizar a predição do valor da variável temperatura do ar, simplificando o processo de coleta armazenamento, análise e processamento das diversas variáveis utilizadas nos modelos numéricos de equações diferenciais usadas no processo de previsão do tempo.

Para a realização de tal intento, serão utilizados os ítens listados abaixo:

- Realizar análise exploratória e descritiva dos dados fornecidos pelo Instituto Nacional de Meteorologia, doravante INMET (2020 e 2021) e Climatempo (2021).

- Desenvolver um modelo de machine learning utilizando o algoritmo LSTM (Long Short Term Memory).
- Desenvolver um modelo preditivo utilizando o modelo estatístico ARIMA (Auto Regressive Moving Average).
- Comparar os valores de predição obtidos pelo LTSM e ARIMA com os dados previstos pelo Clima Tempo, obtidos através da API de previsão do tempo do Clima Tempo.
- Os dados analisados e coletados compreendem a região de Belo Horizonte, Minas Gerais, entre 01/01/2020 e 14/07/2021. (INMET) e 14/07/2021 e 16/07/2021 (Clima Tempo).

2. Coleta de Dados

A tabela a seguir exibe o detalhamento dos dados climáticos do INMET para 2020, obtidos através do link abaixo, obtidos em 07/07/2021:

https://portal.inmet.gov.br/dadoshistoricos

Tabela 1: Dados Meteorológicos INMET 2020

Nome da coluna/campo	Descrição	Tipo
Data	Data.	OBJECT
Hora UTC	Hora.	OBJECT
PRECIPITACAO TOTAL. HORARIO (mm)	Valor de precipitação.	float64
PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA (mB)	Pressão atmosférica medida do nível da estação meteorológica.	float64
PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)	Pressão atmosférica máxima medida do nível da estação meteorológica.	float64
PRESSAO ATMOSFERICA MIN.NA HORA ANT. (AUT) (mB)	Pressão atmosférica mínima medida do nível da estação meteorológica.	float64

RADIACAO GLOBAL (Kj/m3)	Valor de radiação mensurado pela estação meteorológica.	float64
TEMPERATURA DO AR - BULBO SECO. HORARIA (Celsius)	Média horária da temperatura do ar.	float64
TEMPERATURA DO PONTO DE ORVALHO (Celsius)	Temperatura na qual o vapor de água que está em suspensão no ar começa a se condensar	float64
TEMPERATURA MINIMA NA HORA ANT. (AUT) (Celsius)	Temperatura mínima medida na hora anterior.	float64
TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius).1	Temperatura máxima medida na hora anterior.	float64
TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (Celsius)	Temperatura máxima do ponto de orvalho na última hora.	float64
TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (Celsius)	Temperatura mínima do ponto de orvalho na última hora.	float64
UMIDADE REL. MAX. NA HORA ANT. (AUT)	Umidade relativa do ar máxima na hora anterior.	float64
UMIDADE REL. MIN. NA HORA ANT. (AUT)	Umidade relativa do ar mínima na hora anterior.	float64
UMIDADE RELATIVA DO AR. HORARIA (%)	Porcentagem horária de umidade relativa do ar.	float64
VENTO. DIRECAO HORARIA (gr) ((gr))	Direção do vento horária.	float64
VENTO. RAJADA MAXIMA (m/s)	Velocidade máxima do vento.	float64
VENTO. VELOCIDADE HORARIA (m/s)	Média horária de velocidade do vento.	float64

A tabela a seguir exibe o detalhamento dos dados climáticos do INMET para 2021, obtidos através do link abaixo, obtidos em 14/07/2021:

https://bdmep.inmet.gov.br/$2a$10$HxsOeG1EWoLlEoMx3gi6eS-JdzuuYD7iOA8qVyfKi.tjqYmx3QE..zip

Tabela 2: Dados Climáticos INMET 2021

Nome da coluna/campo	Descrição	Tipo
Data Medicao	Data da medição.	object
Hora Medicao	Hora da medição.	int64
PRECIPITACAO TOTAL. HORARIO(mm)	Precipitação total horária em milímetros.	int64
PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA(mB)	Pressão ATM horária.	float64
PRESSAO ATMOSFERICA REDUZIDA NIVEL DO MAR. AUT(mB)	Pressão Atmosférica considerando o nível do mar.	float64
PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT)(mB)	Valor máximo de pressão atmosférica obtido na hora anterior	float64
PRESSAO ATMOSFERICA MIN.NA HORA ANT. (AUT)(mB)	Valor mínimo de pressão atmosférica obtido na hora anterior.	float64
TEMPERATURA DA CPU DA ESTACAO(°C)	Temperatura méria horária medida na CPU da estação meteorológica.	int64
TEMPERATURA DO AR - BULBO SECO. HORARIA(°C)	Temperatura média do ar.	float64
TEMPERATURA DO PONTO DE ORVALHO(°C)	Temperatura na qual o vapor de água que está em suspensão no ar começa a se condensar.	float64
TEMPERATURA MINIMA NA HORA ANT. (AUT)(°C)	Valor mínimo de temperatura ambiente obtido na hora anterior.	float64

TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT)(°C)	Temperatura máxima do ponto de orvalho na última hora.	float64
TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT)(°C)	Temperatura mínima do ponto de orvalho na última hora.	float64
TENSAO DA BATERIA DA ESTACAO(V)	Valot de tensão elétrica na bateria da estação meteorológica em (V).	float64
UMIDADE REL. MAX. NA HORA ANT. (AUT)(%)	Umidade relativa do ar máxima na hora anterior.	int64
UMIDADE REL. MIN. NA HORA ANT. (AUT)(%)	Umidade relativa do ar mínima na hora anterior.	int64
UMIDADE RELATIVA DO AR. HORARIA(%)	Porcentagem horária de umidade relativa do ar.	int64
VENTO DIRECAO HORARIA (gr)(° (gr))	Direção do vento horária.	int64
VENTO. RAJADA MAXIMA(m/s)	Velocidade máxima do vento.	float64
VENTO. VELOCIDADE HORARIA(m/s);	Média horária de velocidade do vento.	float64

Os dados do Climatempo foram obtidos via chamadas HTTP em código python. O cadastro para uso dessa API pode ser feito no site, obtidos em 14/07/2021:

https://advisor.climatempo.com.br/

Tabela 3: Dados Obtidos Utilizando a API Clima Tempo

Nome da coluna/campo	Descrição	Tipo
Data	Data e Hora da medição	object
Temperatura	Temperatura do ar (Ambiente)	int64

A leitura e os médotos disponibilizados por essa API podem ser visualizados no notebook “apiClimaTempo.ipynb”.

Figura 3: Código Python de Consulta da API Clima Tempo

```python
#inputs: cod cidade e token
#outputs: dicionario contendo previsao para as proximas 72 horas na cidade em questao
def buscaPrevisao72HorasRegiao(iCIDADE, iTOKEN):
    iURL = "http://apiadvisor.climatempo.com.br/api/v1/forecast/locale/" + iCIDADE + "/hours/72?token=" + iTOKEN
    iRESPONSE = requests.request("GET", iURL)
    iRETORNO_REQ = json.loads(iRESPONSE.text)
    #print(iRETORNO_REQ)

    cont = 0
    dadosPrevistos ={}

    for iCHAVE in iRETORNO_REQ['data']:
        iDATA = iCHAVE.get('date_br')
        iTEMPERATURA = iCHAVE['temperature']['temperature']
        #rint("data:" + str(iDATA) + " " + str(iTEMPERATURA) + "º" + "\n")
        dadosPrevistos[cont] = 'Data: ' + str(iDATA) + ' ' + 'Temperatura ºC: ' + str(iTEMPERATURA)
        cont = cont+1
    return dadosPrevistos
```

```python
#TESTE buscaPrevisao72HorasRegiao(iCIDADE, iTOKEN)
buscaPrevisao72HorasRegiao('6879', "8c2f52e9921a6556790d9ff7a3f0275c")
```

Fonte: Autor

Para o desenvolvimento do projeto, utilizou-se a plataforma Anaconda Navigator 1.9.12. O Anaconda é um projeto open-source que já contém a linguagem Python e centenas de bibliotecas “embutidas”.

Utilizando o Anaconda, todas as bibliotecas “obrigatórias” para fazer ciência de dados. Além das diversas bibliotecas disponíveis, o software é Anaconda é multiplataforma, ou seja, funciona em Windows, Linux e MacOS.

No presente trabalho, utilizou-se o sistema operacional Linux UBUNTU 19.04. Utilizou-se a ferramenta Jupyter Notebook através do ambiente Anaconda Navigator. O Jupyter Notebook é uma evolução do bloco de notas. De forma prática, é possível compilar trechos de códigos de diversas linguagens de programação. No trabalho em questão, utilizou-se a linguagem python na versão 3.7.6.

3. Processamento/Tratamento de Dados

Ao longo da etapa de processamento e tratamento dos dados, utilizou-se as bibliotecas pandas e numpy. Alguns ajustes foram neessários usando o editor de textos "MS Visual Studio Code". Os arquivos “DadosBHCercadinho2020.csv” e “DadosBHCercadinho2021.csv” passaram pelos ajustes.

- Substituição de ";" por “,” para fazer a separação das colunas, para evitar falhas de leitura ao longo do código.
- Casas decimais eram separadas por vírgulas (seguindo o padrão utilizado no Brasil) sendo convertido para ponto, para evitar falhas de leitura ao longo do código.

Na etapa de processamento e tratamento dos dados, foi realizada usando o notebook “INMETDATACLEANING.ipynb” Primeiramente, realizou-se o carregamento dos dados, como segue na imagem abaixo:

Figura 4: Carregamento de Dados do INMET

```
#TRATAMENTO DE DADOS DE METEOROLOGIA DO INMET(2020)
import pandas as pd
import numpy as np

#Os dados tiveram de ser manipulados no NOTEPAD++, ponto e vírgul ";" que separavam colunas foram substituídas por
#Casas decimais foram utilizavam o método brasileiro ",", agora utiliza ponto ".".
dados_inmet_bh_2020 = pd.read_csv("/home/marcelo/Documentos/PosGraduacaoDataScience/Disciplinas/Monografia/DadosMon
dados_inmet_bh_2021 = pd.read_csv("/home/marcelo/Documentos/PosGraduacaoDataScience/Disciplinas/Monografia/DadosMon
```

Fonte: Autor

Visualizou-se que os dados das variáveis “Data” e “Hora UTC” estavam completos ao longo de todos os registros para o dataset de 2020, porém, as demais variáveis, apresentavam um valor equivalente a 0,204% dos dados faltantes, exceto pela variável “RADIACAO GLOBAL (Kj/m3)”, possuindo 44,48% dos dados nulos. Optou-se por deletar a variável “RADIACAO GLOBAL (Kj/m3)”, como segue na imagem abaixo. Essa variável foi eliminada do datasets de 2020 e 2021, devido ao grande percentual de dados nulos que esta possuia. No dataset de 2021 apenas a variável “PRESSAO ATMOSFERICA REDUZIDA NIVEL DO MAR. AUT(mB)” apresentou 3,09% de dados nulos.

Figura 5: Detalhamento dos dados do INMET de 2021 durante tratamento de dados

```
dados_inmet_bh_2021=dados_inmet_bh_2021.drop(['RADIACAO GLOBAL(Kj/m²)'], axis=1)
dados_inmet_bh_2021.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 326 entries, 0 to 325
Data columns (total 21 columns):
 #   Column                                         Non-Null Count  Dtype
---  ------                                         --------------  -----
 0   Data Medicao                                   326 non-null    object
 1   Hora Medicao                                   326 non-null    int64
 2   PRECIPITACAO TOTAL. HORARIO(mm)                326 non-null    int64
 3   PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA(mB)  326 non-null  float64
 4   PRESSAO ATMOSFERICA REDUZIDA NIVEL DO MAR. AUT(mB)    313 non-null  float64
 5   PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT)(mB)        326 non-null  float64
 6   PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT)(mB)       326 non-null  float64
 7   TEMPERATURA DA CPU DA ESTACAO(°C)              326 non-null    int64
 8   TEMPERATURA DO AR - BULBO SECO. HORARIA(°C)    326 non-null    float64
 9   TEMPERATURA DO PONTO DE ORVALHO(°C)            326 non-null    float64
 10  TEMPERATURA MAXIMA NA HORA ANT. (AUT)(°C)      326 non-null    float64
 11  TEMPERATURA MINIMA NA HORA ANT. (AUT)(°C)      326 non-null    float64
 12  TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT)(°C)  326 non-null  float64
 13  TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT)(°C)  326 non-null  float64
 14  TENSAO DA BATERIA DA ESTACAO(V)                326 non-null    float64
 15  UMIDADE REL. MAX. NA HORA ANT. (AUT)(%)        326 non-null    int64
 16  UMIDADE REL. MIN. NA HORA ANT. (AUT)(%)        326 non-null    int64
 17  UMIDADE RELATIVA DO AR. HORARIA(%)             326 non-null    int64
 18  VENTO DIRECAO HORARIA (gr)(° (gr))             326 non-null    int64
 19  VENTO. RAJADA MAXIMA(m/s)                      326 non-null    float64
 20  VENTO. VELOCIDADE HORARIA(m/s);                326 non-null    float64
dtypes: float64(13), int64(7), object(1)
memory usage: 53.6+ KB
```

FONTE: Autor

Figura 6 Detalhamento dos dados do INMET de 2020 durante tratamento de dados

```
dados_inmet_bh_2020.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8781 entries, 0 to 8780
Data columns (total 19 columns):
 #   Column                                                 Non-Null Count  Dtype
---  ------                                                 --------------  -----
 0   Data                                                   8781 non-null   object
 1   Hora UTC                                               8781 non-null   object
 2   PRECIPITACAO TOTAL. HORARIO (mm)                       8764 non-null   float64
 3   PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA (mB)  8764 non-null   float64
 4   PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)        8764 non-null   float64
 5   PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)       8764 non-null   float64
 6   RADIACAO GLOBAL (Kj/m3)                                4881 non-null   float64
 7   TEMPERATURA DO AR - BULBO SECO. HORARIA (Celsius)      8764 non-null   float64
 8   TEMPERATURA DO PONTO DE ORVALHO (Celsius)              8764 non-null   float64
 9   TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius)        8764 non-null   float64
 10  TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius).1      8764 non-null   float64
 11  TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (Celsius)  8764 non-null   float64
 12  TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (Celsius)  8764 non-null   float64
 13  UMIDADE REL. MAX. NA HORA ANT. (AUT) (%)               8764 non-null   float64
 14  UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)               8764 non-null   float64
 15  UMIDADE RELATIVA DO AR. HORARIA (%)                    8764 non-null   float64
 16  VENTO. DIRECAO HORARIA (gr) ((gr))                     8763 non-null   float64
 17  VENTO. RAJADA MAXIMA (m/s)                             8763 non-null   float64
 18  VENTO. VELOCIDADE HORARIA (m/s);                       8763 non-null   float64
dtypes: float64(17), object(2)
memory usage: 1.3+ MB
```

Fonte: Autor

Para as demais variáveis do arquivo de 2020, exceto “RADIACAO GLOBAL (Kj/m3)” que foi eliminada, utilizou-se a substituição dos valores nulos pela média, levando em consideração os pontos:

Positivos:

1. Resultado satisfatório quando o volume de dados não é massivo.
2. Melhor tratativa do que eliminar todas as ocorrências onde as valores estão nulos, pois geraria considerável perda de dados.

Negativos:

1. Inserir aproximações adiciona BIAS e variância aos dados.
2. Por se tratar de um método de implementação mais simples e rudimentar, tem o resultado inferior ao de outros métodos mais modernos, como por exemplo utilizar regressão linear para prever os dados faltantes.

Foi realizada a tratativa de substituição dos dados nulos pelos valores de média das variáveis usando o código exibido na imagem abaixo.

Figura 7: Tratamento de dados nulos

```
for x in range(2, 18):
  dados_inmet_bh_2020.iloc[:, x] = dados_inmet_bh_2020.iloc[:, x].replace(np.NaN, dados_inmet_bh_2020.iloc[:, x].mea
  #print(dados_inmet_bh_2020.iloc[:, 2].mean()
```

```
for x in range(2, 21):
  dados_inmet_bh_2021.iloc[:, x] = dados_inmet_bh_2021.iloc[:, x].replace(np.NaN, dados_inmet_bh_2021.iloc[:, x].mea
  #print(dados_inmet_bh_2020.iloc[:, 2].mean()
```

Fonte: Autor

Após a tratativa proposta, observou-se que todos os dados nulos foram substituídos pelos devidos valores médios das variáveis, como segue abaixo.

Figura 8: Dados INMET após tratamento

```
#Analisando volume de dados nulos
dados_inmet_bh_2021.info()
dados_inmet_bh_2021.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 326 entries, 0 to 325
Data columns (total 21 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   Data Medicao                                  326 non-null    object
 1   Hora Medicao                                  326 non-null    int64
 2   PRECIPITACAO TOTAL. HORARIO(mm)               326 non-null    int64
 3   PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA(mB)  326 non-null    float64
 4   PRESSAO ATMOSFERICA REDUZIDA NIVEL DO MAR. AUT(mB)    326 non-null    float64
 5   PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT)(mB)        326 non-null    float64
 6   PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT)(mB)       326 non-null    float64
 7   TEMPERATURA DA CPU DA ESTACAO(°C)             326 non-null    int64
 8   TEMPERATURA DO AR - BULBO SECO. HORARIA(°C)   326 non-null    float64
 9   TEMPERATURA DO PONTO DE ORVALHO(°C)           326 non-null    float64
 10  TEMPERATURA MAXIMA NA HORA ANT. (AUT)(°C)     326 non-null    float64
 11  TEMPERATURA MINIMA NA HORA ANT. (AUT)(°C)     326 non-null    float64
 12  TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT)(°C)       326 non-null    float64
 13  TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT)(°C)       326 non-null    float64
 14  TENSAO DA BATERIA DA ESTACAO(V)               326 non-null    float64
 15  UMIDADE REL. MAX. NA HORA ANT. (AUT)(%)       326 non-null    int64
 16  UMIDADE REL. MIN. NA HORA ANT. (AUT)(%)       326 non-null    int64
 17  UMIDADE RELATIVA DO AR. HORARIA(%)            326 non-null    int64
 18  VENTO DIRECAO HORARIA (gr)(° (gr))            326 non-null    int64
 19  VENTO. RAJADA MAXIMA(m/s)                     326 non-null    float64
 20  VENTO. VELOCIDADE HORARIA(m/s);               326 non-null    float64
dtypes: float64(13), int64(7), object(1)
memory usage: 53.6+ KB

Data Medicao                                  0
Hora Medicao                                  0
PRECIPITACAO TOTAL. HORARIO(mm)               0
PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA(mB)  0
PRESSAO ATMOSFERICA REDUZIDA NIVEL DO MAR. AUT(mB)    0
PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT)(mB)        0
PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT)(mB)       0
TEMPERATURA DA CPU DA ESTACAO(°C)             0
TEMPERATURA DO AR - BULBO SECO. HORARIA(°C)   0
TEMPERATURA DO PONTO DE ORVALHO(°C)           0
TEMPERATURA MAXIMA NA HORA ANT. (AUT)(°C)     0
TEMPERATURA MINIMA NA HORA ANT. (AUT)(°C)     0
TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT)(°C)       0
TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT)(°C)       0
TENSAO DA BATERIA DA ESTACAO(V)               0
UMIDADE REL. MAX. NA HORA ANT. (AUT)(%)       0
UMIDADE REL. MIN. NA HORA ANT. (AUT)(%)       0
UMIDADE RELATIVA DO AR. HORARIA(%)            0
VENTO DIRECAO HORARIA (gr)(° (gr))            0
VENTO. RAJADA MAXIMA(m/s)                     0
VENTO. VELOCIDADE HORARIA(m/s);               0
dtype: int64
```

Fonte: Autor

Após realizada a tratativa proposta, os dados foram salvos nos arquivos “dados_inmet_filtrados_2020.csv” e “dados_inmet_filtrados_2021.csv” para serem utilizados nas etapas posteriores.

Figura 9: Salvando os dados no formato CSV

```
#Salvando os Dados em CSV
dados_inmet_bh_2020.info()
dados_inmet_bh_2020.to_csv (r'/home/marcelo/Documentos/PosGraduacaoDataScience/Disciplinas/Monografia/DadosMonograf
```

Fonte: Autor

Figura 10: Salvando os dados no formato CSV

```
#Salvando os Dados em CSV
dados_inmet_bh_2021.info()
dados_inmet_bh_2021.to_csv (r'/home/marcelo/Documentos/PosGraduacaoDataScience/Disciplinas/Monografia/DadosMonograf
```

Fonte: Autor

Os dados obtidos pela API Climatempo vieram a presença de dados nulos e não demandaram tratamento nesta etapa.

4. Análise e Exploração dos Dados

Ao longo da etapa de análise, processamento e exploração de dados, utilizou-se as bibliotecas pandas, numpy, matplotlib e seaborn.

Foi utilizado o método dataframe.describe() do pandas para analisar as métricas estatísticas média, desvio padrão, mediana e etc.

Figura 11: Análise estatística dos dados

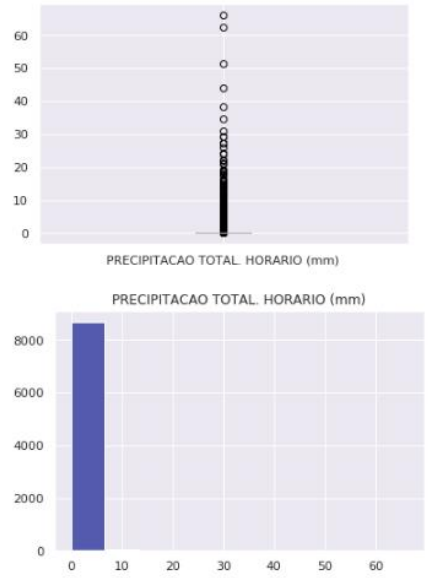
```
df.describe()
```

	PRECIPITACAO TOTAL. HORARIO (mm)	PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA (mB)	PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)	PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)	TEMPERATURA DO AR - BULBO SECO. HORARIA (Celsius)	TEMPERATURA DO PONTO DE ORVALHO (Celsius)	TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius)	TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius).1	TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (Celsius)	T
count	8781.000000	8781.000000	8781.000000	8781.000000	8781.000000	8781.000000	8781.000000	8781.000000	8781.000000	
mean	0.273642	882.947980	883.178959	882.713316	19.655306	14.740062	20.362631	18.996029	15.252111	
std	2.052392	2.885436	2.866747	2.895329	4.034454	3.272392	4.332765	3.724635	3.309822	
min	0.000000	872.300000	872.800000	872.200000	8.100000	1.700000	8.800000	7.700000	3.100000	
25%	0.000000	881.100000	881.400000	880.900000	17.000000	12.400000	17.400000	16.500000	12.800000	
50%	0.000000	882.800000	883.100000	882.600000	19.200000	15.100000	19.800000	18.800000	15.600000	
75%	0.000000	884.800000	885.000000	884.600000	22.100000	17.500000	23.200000	21.100000	18.000000	
max	66.200000	892.500000	892.500000	892.300000	34.900000	22.100000	35.300000	33.400000	23.200000	

Fonte: Autor

Nessa etapa foi observou-se que as variáveis apresentavam um baixo nível de dispersão, possuindo um desvio padrão entre 2 e 5, de maneira geral. Exceto pelas variáveis “Vento Direção Horária", “Umidade Relativa Horaria”, “Umidade Rel Min na hora anterior” e “Umidade Rel Max na hora anterior”, que apresentavam os valores 85.580479, 17.612537, 18.415922 e 16.635963, respectivamente. Uma análise de box plot e distribuição dos dados se faz mais detalhadamente, á seguir.

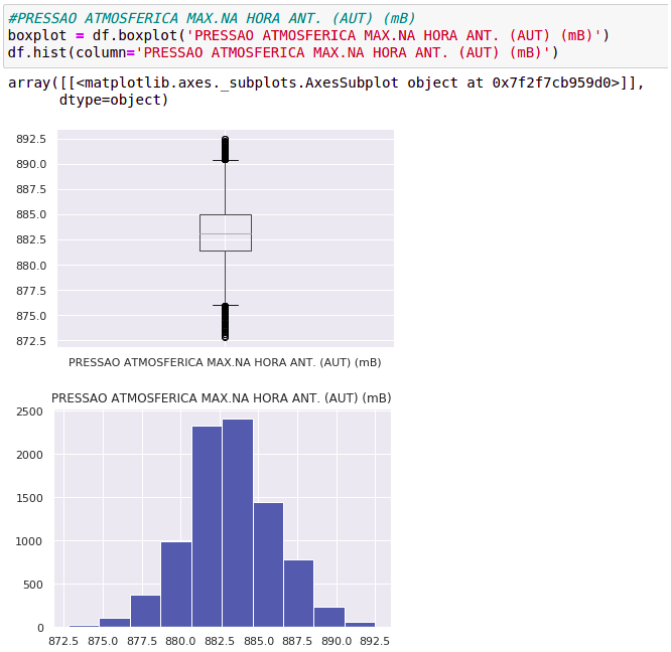
Figura 12: Análise da Precipitação Total



Fonte: Autor

A variável relacionada à precipitação total horária, apresentou distribuição não-normal e a grande maioria dos dados entre 0 e 10 mm, sendo os demais valores acima de 10 considerados outliers. Isso se deve ao fato do índice pluviométrico no período de tempo que compreende o volume de dados analisado ter sido baixo.

Figura 13: Análise da Pressão Atmosférica Máxima na Hora Anterior

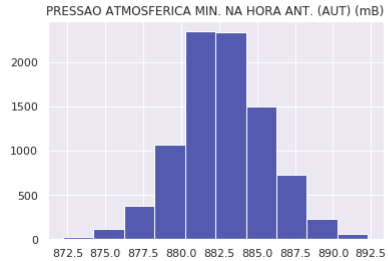


Fonte: Autor

Figura 14: Análise da Pressão Atmosférica Mínima na Hora Anterior

```
#PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)VENTO. RAJADA MAXIMA (m/s)
boxplot = df.boxplot('PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)')
df.hist(column='PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7c9817d0>]],
      dtype=object)
```

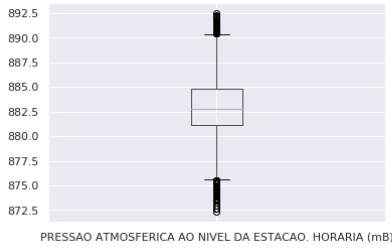


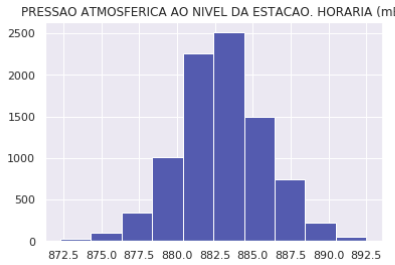
Fonte: Autor

Figura 15: Análise da Pressão Atmosférica Média

```
#PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA (mB)
boxplot = df.boxplot('PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA (mB)')
df.hist(column='PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO. HORARIA (mB)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7ce65650>]],
      dtype=object)
```





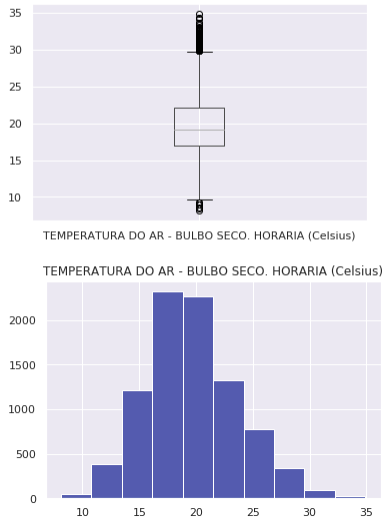
Fonte: Autor

As variáveis relacionadas à pressão atmosférica, apresentaram comportamento bastante semelhante, tendo sua distribuição aproximadamente normal e o número de outliers acima e abaixo do valor máximo semelhantes.

Figura 16: Análise da Temperatura do Ar

```
#TEMPERATURA DO AR - BULBO SECO. HORARIA (CelsiVENTO. DIRECAO HORARIA (gr) ((gr))us)
boxplot = df.boxplot('TEMPERATURA DO AR - BULBO SECO. HORARIA (Celsius)')
df.hist(column='TEMPERATURA DO AR - BULBO SECO. HORARIA (Celsius)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7c86c910>]],
      dtype=object)
```

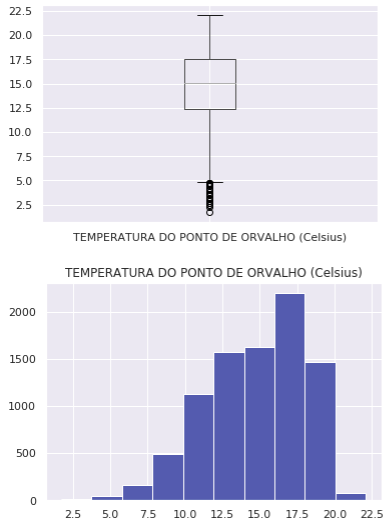


Fonte: Autor

Figura 17: Análise da Temperatura do Ponto de Orvalho

```
#TEMPERATURA DO PONTO DE ORVALHO (Celsius)
boxplot = df.boxplot('TEMPERATURA DO PONTO DE ORVALHO (Celsius)')
df.hist(column='TEMPERATURA DO PONTO DE ORVALHO (Celsius)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7c7dd610>]],
      dtype=object)
```

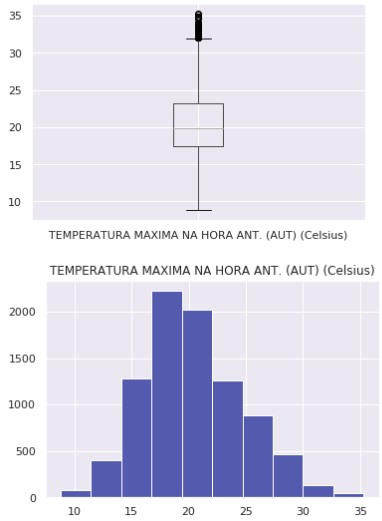


Fonte: Autor

Figura 18: Análise da Temperatura Máxima do Ar na Hora Anterior

```
#TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius)
boxplot = df.boxplot('TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius)')
df.hist(column='TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7c6cdfd0>]],
      dtype=object)
```

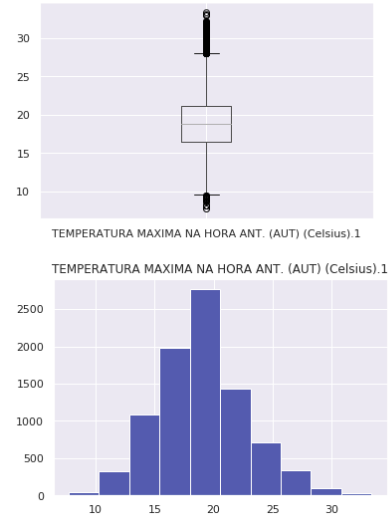


Fonte: Autor

Figura 19: Análise da Temperatura Mázima do Ar na Hora Anterior

```
#TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius).1
boxplot = df.boxplot('TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius).1')
df.hist(column='TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius).1')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fae2d810790>]],
      dtype=object)
```

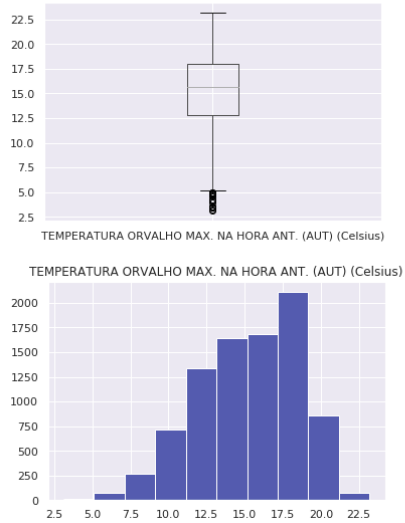


Fonte: Autor

Figura 20: Análise da Temperatura de Orvalho Máxima

```
#TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (Celsius)
boxplot = df.boxplot('TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (Celsius)')
df.hist(column='TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (Celsius)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7c647350>]],
      dtype=object)
```

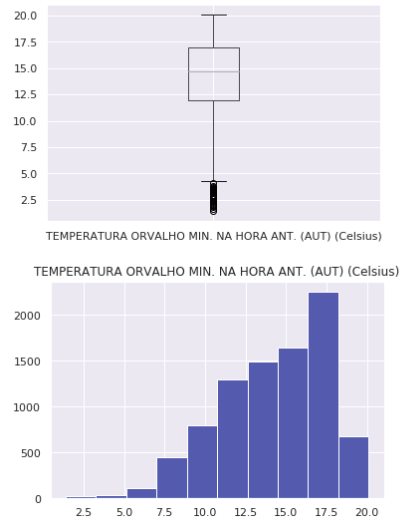


Fonte: Autor

Figura 21: Análise Temperatura de Orvalho Mínima Na Hora Anterior

```
#TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (Celsius)
boxplot = df.boxplot('TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (Celsius)')
df.hist(column='TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (Celsius)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7c972550>]],
      dtype=object)
```



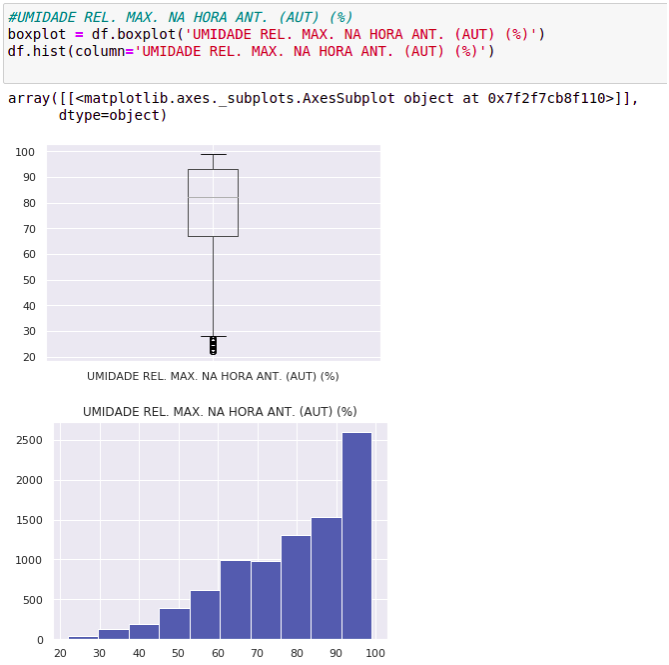
Fonte: Autor

Analisou-se que as variáveis relacionadas à temperatura e observou-se que não possuíam as mesmas características de distribuição de probabilidades e outliers.

Todas possuem uma distribuição que se aproxima da distribuição normal e em relação à outliers, segue análise obtida.

- Temperatura do Ar – Bulbo seco apresentou um volume considerável de outliers acima do valor máximo e poucos outliers abaixo do valor mínimo.
- Temperatura do ponto de Orvalho apresentou um grande volume de outliers abaixo do valor mínimo.
- Temperatura máxima na Hora anterior apresentou outliers acima do valor máximo.
- Temperatura Orvalho Máximo na hora anterior e Temperatura Orvalho Mínimo na hora anterior apresentaram um valor considerável de outliers abaixo do valor mínimo.

Figura 22: Análise Umidade Relativa Máxima Na Hora Anterior

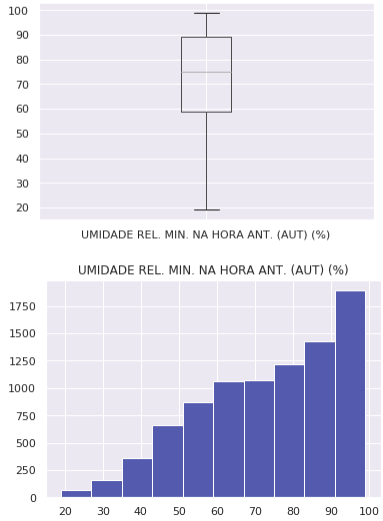


Fonte: Autor

Figura 23: Análise Umidade Mínima na Hora Anterior

```
#UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)
boxplot = df.boxplot('UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)')
df.hist(column='UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7cdeded0>]],
      dtype=object)
```



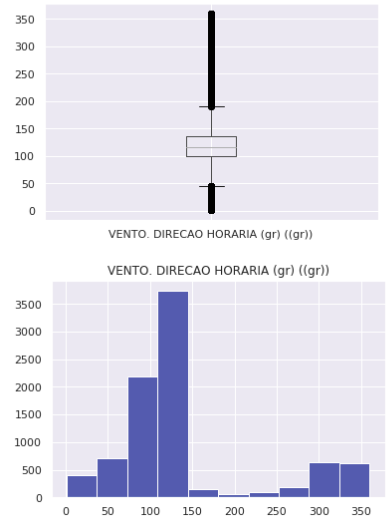
Fonte: Autor

As variáveis relacionadas a Umidade relativa do ar, apresentaram comportamento parecido. Todas as 3 variáveis apresentaram distribuição não normal e apenas a variável referente à umidade relativa máxima na hora anterior apresentou outliers abaixo do valor mínimo. As demais não apresentaram outliers.

Figura 24: Análise Vento Direção Horária

```
#VENTO. DIRECAO HORARIA (gr) ((gr))
boxplot = df.boxplot('VENTO. DIRECAO HORARIA (gr) ((gr))')
df.hist(column='VENTO. DIRECAO HORARIA (gr) ((gr))')
```

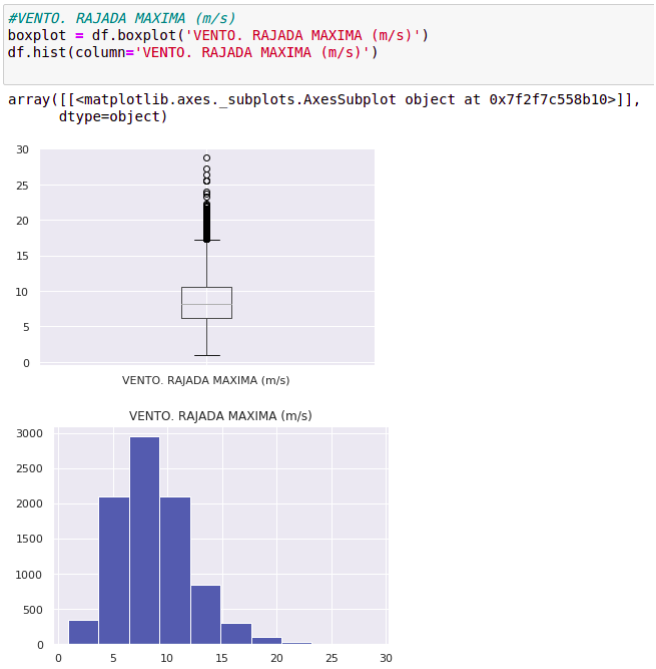
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f2f7d065950>]],
      dtype=object)
```



Fonte: Autor

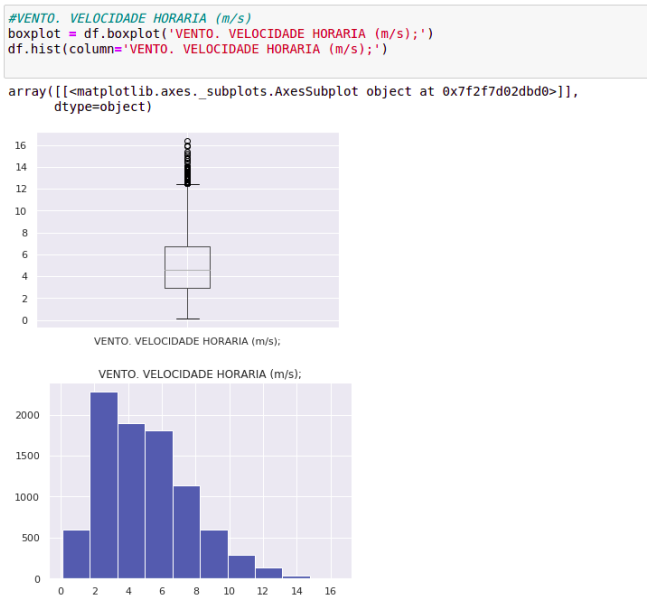
A variável relacionada a direção horária do vento apresentou um grande volume de outliers acima e abaixo dos valores máximo e mínimo, respectivamente.

Figura 25: Análise de Velocidade Máxima do Vento



Fonte: Autor

Figura 26: Análise da Velocidade Média do Vento

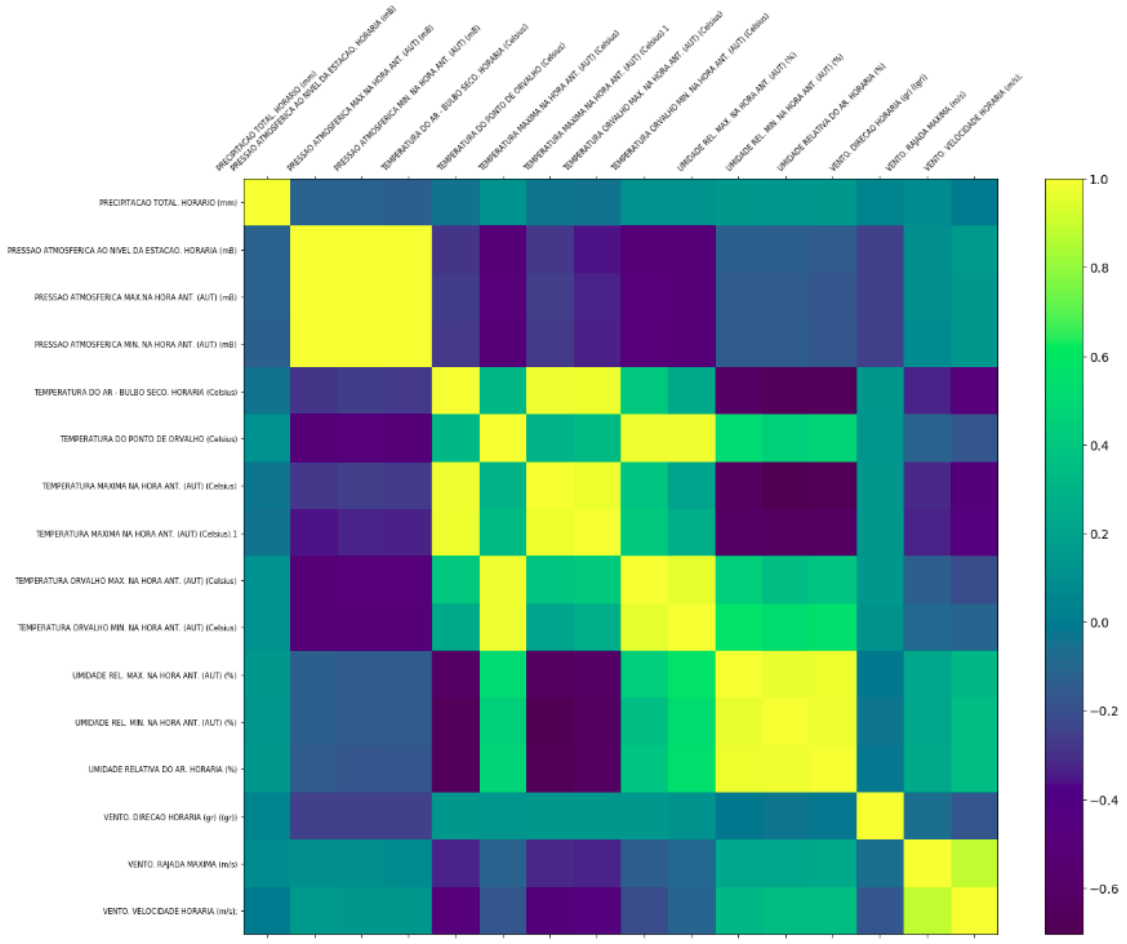


Fonte: Autor

Observou-se que as variáveis relacionadas à velocidade média horária e velocidade máxima horária apresentaram um certo número de outliers acima do valor máximo e uma distribuição estatística bastante semelhante. Os valores médios, máximo e mínimo foram diferentes pois uma das variáveis considera o valor médio horário do vento, enquanto outra considera o valor máximo horário.

Após as análises de distribuição e boxplot, realizou-se a análise de correlação dos dados e, afim de reduzir o número de variáveis à serem consideradas no processo de modelagem utilizando machine learning e ferramentas estatísticas, decidiu-se utilizar a análise de correlação e eliminar as variáveis que possuíssem correlação maior que 0,95 entre si.

Figura 27: Gráfico de Correlação Entre as Variáveis



Fonte: Autor

Através da análise realizada, pode-se observar correlação entre às variáveis de temperatura do Ar e Temperatura Máxima na Hora anterior. Na etapa de criação do modelo de machine learning, optou-se por utilizar a Temperatura Máxima na hora anterior e umidade relativa máxima na hora anterior como variáveis independentes, devido à correlação que apresentaram.

5. Criação de Modelos de Machine Learning

Levou-se em consideração, a fim de maior simplicidade do modelo, o menor número possível de variáveis de entrada para se obter a variável dependente, que no caso proposto seria a temperatura do ar. Considerando a análise de correlação e a premissa supracitada, optou-se por utilizar as variáveis abaixo como entrada do modelo(x_1 e x_2) pois de acordo com estudos da área, as variáveis relacionadas à umidade estão fortemente conectadas com mudanças de temperatura atmosférica, chuvas e etc. Optou-se por utilizar a temperatura máxima na hora anterior por esta variável possuir forte correlação com a variável dependente. Foram carregados 5000 linhas do dataset “dados_inmet_filtrados_2020.csv” para as etapas de treinamento e testes. 300 linhas carregadas para a etapa de validação do modelo.

Figura 30: Variáveis Utilizadas no Modelo

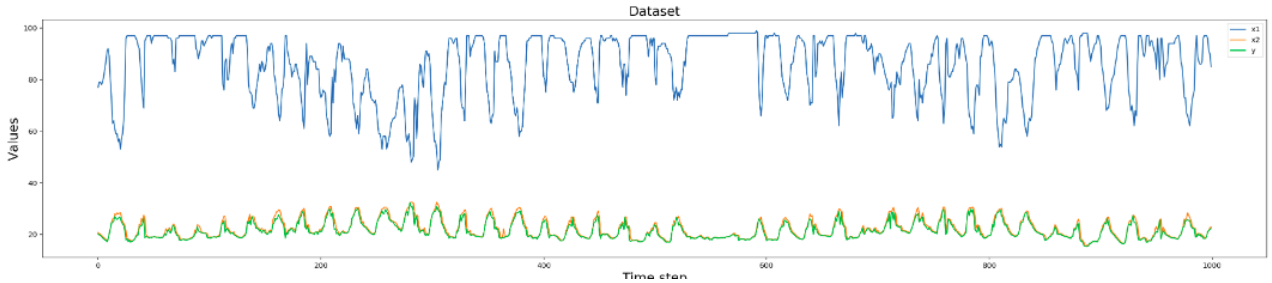
```
x_1 = dataset['UMIDADE REL. MAX. NA HORA ANT. (AUT) (%)']
x_2 = dataset['TEMPERATURA MAXIMA NA HORA ANT. (AUT) (Celsius)']
y = dataset['TEMPERATURA DO AR - BULBO SECO. HORARIA (Celsius)']

x_1 = x_1.values
x_2 = x_2.values
y = y.values
```

Fonte: Autor

Segue abaixo o gráfico das tendẽncias das 3 variáveis em questão.Sendo a de Temperatura do Ar à variável à ser predita pelo modelo.

Figura 31: Visualização dos Variáveis Dependentes e Independente

```
plt.figure(figsize=(30, 6))
plt.plot(x_1[:1000] , label='x1')
plt.plot(x_2[:1000] , label='x2')
plt.plot(y[:1000] , label='y')
plt.legend(loc='upper right')
plt.title("Dataset" ,  fontsize=18)
plt.xlabel('Time step' ,  fontsize=18)
plt.ylabel('Values' , fontsize=18)
plt.legend()
plt.show()
```



Fonte: Autor

Os próximos passos foram a formatação dos dados e definição do dataset padronizado para ser utilizado pelo LTSM, tendo sido o método “MinMaxScaler” utilizado para normalização dos dados entre 0 e 1.

Figura 32: Normalização das Variáveis Independentes e Dependente

```
# convert to [rows, columns] structure
x_1 = x_1.reshape((len(x_1), 1))
x_2 = x_2.reshape((len(x_2), 1))
y = y.reshape((len(y), 1))

print ("x_1.shape" , x_1.shape)
print ("x_2.shape" , x_2.shape)
print ("y.shape" , y.shape)
```

```
x_1.shape (5000, 1)
x_2.shape (5000, 1)
y.shape (5000, 1)
```

```
# normalization features
scaler = MinMaxScaler(feature_range=(0, 1))
x_1_scaled = scaler.fit_transform(x_1)
x_2_scaled = scaler.fit_transform(x_2)
y_scaled = scaler.fit_transform(y)

# horizontally stack columns
dataset_stacked = hstack((x_1_scaled, x_2_scaled, y_scaled))

print ("dataset_stacked.shape" , dataset_stacked.shape)
```

```
dataset_stacked.shape (5000, 3)
```

Fonte: Autor

A função split_sequences() recebe os dados da sequência, o valor de quantos dados do passado deve ser usado para prever cada step à frente e o número de steps à ser previsto.

Figura 33: Preparação de Dados para Inseri-los no Modelo LTSM

```
# split a multivariate sequence into samples
def split_sequences(sequences, n_steps_in, n_steps_out):
    X, y = list(), list()
    for i in range(len(sequences)):
        # find the end of this pattern
        end_ix = i + n_steps_in
        out_end_ix = end_ix + n_steps_out-1
        # check if we are beyond the dataset
        if out_end_ix > len(sequences):
            break
        # gather input and output parts of the pattern
        seq_x, seq_y = sequences[i:end_ix, :-1], sequences[end_ix-1:out_end_ix, -1]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)
```

```
# choose a number of time steps #change this accordingly
n_steps_in, n_steps_out = 96 , 6

# covert into input/output
X, y = split_sequences(dataset_stacked, n_steps_in, n_steps_out)

print ("X.shape" , X.shape)
print ("y.shape" , y.shape)
```

Fonte: Autor

Em seguida, foi realizada a divisão entre dados de treinamento do modelo e dados de teste, usando a variável split com o valor atribuído de 4000, que indica 4000 dados para treinamento e 1 para teste.

Figura 34: Separação dos Dados em Treinamento e Teste

```
split = 4000
train_X , train_y = X[:split, :] , y[:split, :]
test_X , test_y = X[split:, :] , y[split:, :]

n_features = train_X.shape[2]
```

Fonte: Autor

Definiu-se a taxa de aprendizado como 0,01. Este parâmetro representa a velocidade com o qual a rede neural aprenderá, sendo valores próximos à zero um aprendizado mais lento e gradual e valores próximos a um mais acelerado, mas também com grandes probabilidades de aumentar o erro no modelo gerado.

Figura 35: Definição e Configuração do Modelo

```
# Definição da taxa de aprendizado
opt = keras.optimizers.Adam(learning_rate=0.01)

# define model
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(n_steps_in, n_features)))
model.add(LSTM(50, activation='relu'))
model.add(Dense(n_steps_out))
model.add(Activation('linear'))
model.compile(loss='mse' , optimizer=opt , metrics=['mse', 'mae'])
```

Fonte: Autor

Utilizou-se o modelo Sequential(), que é uma pilha linear e sequencial de camadas formando a rede neural recorrente, definiu-se a função de ativação da rede como linear, a função de perda e usou as métricas mean square error e mean average error para validação do modelo em treinamento.

Figura 36: Configuração do Modelo e Inicio da Etapa de Treinamento

```
# Definição da taxa de aprendizado
opt = keras.optimizers.Adam(learning_rate=0.01)

# define model
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(n_steps_in, n_features)))
model.add(LSTM(50, activation='relu'))
model.add(Dense(n_steps_out))
model.add(Activation('linear'))
model.compile(loss='mse' , optimizer=opt , metrics=['mse', 'mae'])
```

```
# Fit network
#history = model.fit(train_X , train_y , epochs=60 , steps_per_epoch=25 , verbose=1 ,validation_data=(test_X, test_
history = model.fit(train_X , train_y , epochs=60 , steps_per_epoch=25 , verbose=2 ,validation_data=(test_X, test_y
```

Fonte: Autor

Configurou-se o modelo a treinar com 60 épocas de treinamento e 25 passos de treinamento por época. Verificou-se que pelo método de análise Elbow, o valor de 60 épocas para o número de épocas como adequado, tendo em vista o decaimento exponencial do erro e um valor mínimo obtido próximo à 60 épocas de treino. O treinamento da rede LSTM é feito com a função model.fit().

Figura 37: Geração dos Gráficos de Erros de Treinamento

```
pyplot.plot(history.history['val_mse'])
pyplot.xlabel('MSE', fontsize=18)
pyplot.ylabel('Épocas de Treinamento', fontsize=16)
pyplot.title('Valor do MSE em 60 épocas de Treinamento', fontsize=12)
pyplot.show()

pyplot.plot(history.history['val_mae'])
pyplot.xlabel('MAE', fontsize=18)
pyplot.ylabel('Épocas de Treinamento', fontsize=16)
pyplot.title('Valor do MAE em 60 épocas de Treinamento', fontsize=12)
pyplot.show()
```

Fonte: Autor

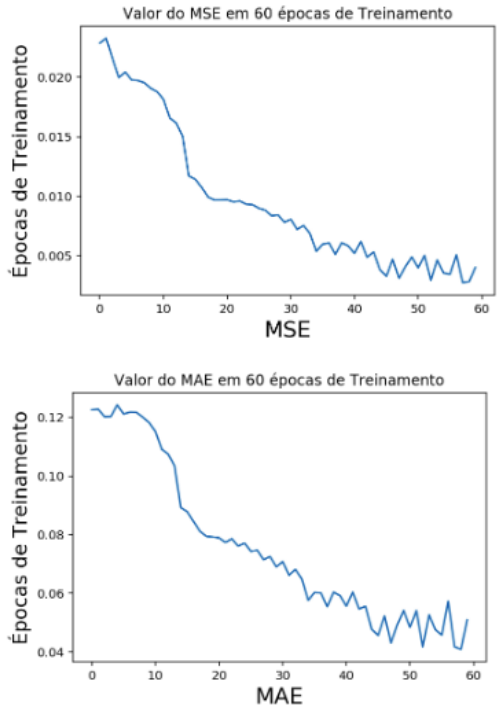
A biblioteca matplotlib foi utilizada para plotar os gráficos de MSE e MAE. Seguem abaixo os valores das métricas calculados para a última época de treinamento.

Figura 38: Erros Obtidos ao Fim das 60 Épocas de Treinamento

```
Epoch 60/60
25/25 - 2s - loss: 0.0052 - mse: 0.0052 - mae: 0.0549 - val_loss: 0.0040 - val_mse: 0.0040 - val_mae: 0.0509
```

Fonte: Autor

Figura 39: Análise de Erros de Treinamento



Fonte: Autor

Em seguida, os dados de validação foram carregados e parametrizados à mesma maneira que os dados de treinamento da rede, utilizando-se dos mesmos métodos e funções de parametrização.

Figura 40: Carregando Dados de Validação

```
# Test Data Batch 1 , Test Data Batch 2 , Test Data Batch 3
#url_test = 'Test Data Batch 1.csv'
url_test = 'ValidateData.csv'
dataset_test_ok = pd.read_csv(url_test)
dataset_test_ok.head()
```

Fonte: Autor

Os cálculos das métricas estatísticas se fez usando funções da biblioteca sklearn. As funções descritas na imagem abaixo foram utilizadas para a preparação dos dados de validação para serem inseridos no modelo previamente treinado e também para cálculo das métricas para validar a acurácia das previsões realizadas.

Figura 41: Funções Para Cálculo de Métricas Estatísticas e Preparação de Dados

```
def calculateStatisticalMetrics(dataList1, dataList2):
    from sklearn.metrics import mean_squared_error
    from math import sqrt
    from sklearn.metrics import mean_absolute_error

    mse = sklearn.metrics.mean_squared_error(dataList1, dataList2)
    rmse = math.sqrt(mse)
    mae = mean_absolute_error(dataList1, dataList2)
    print('RMSE = %f: MSE = %f MAE = %f' % (rmse, mse, mae))



def prep_data(x1_test_scaled , x2_test_scaled , y_test , start , end , last):
    #prepare test data X
    dataset_test = hstack((x1_test_scaled, x2_test_scaled))
    dataset_test_X = dataset_test[start:end, :]
    test_X_new = dataset_test_X.reshape(1, dataset_test_X.shape[0] , dataset_test_X.shape[1])

    #prepare past and groundtruth
    past_data = y_test[:end , :]
    dataset_test_y = y_test[end:last , :]
    scaler1 = MinMaxScaler(feature_range=(0, 1))
    scaler1.fit(dataset_test_y)

    # predictions
    y_pred = model.predict(test_X_new)
    y_pred_inv = scaler1.inverse_transform(y_pred)
    y_pred_inv = y_pred_inv.reshape(n_steps_out,1)
    y_pred_inv = y_pred_inv[:,0]

    return y_pred_inv , dataset_test_y , past_data

# Calculate MAE and RMSE
def evaluate_prediction(predictions, actual, model_name , start , end):
    errors = predictions - actual
    mse = np.square(errors).mean()
    rmse = np.sqrt(mse)
    mae = np.abs(errors).mean()

    print("Test Data from {} to {}".format(start, end))
    print('Mean Absolute Error: {:.6f}'.format(mae))
    print('Root Mean Square Error: {:.6f}'.format(rmse))
    print('')
    print('')
```

Fonte: Autor

Os valores de previsão obtidos através dos dados de validação foram retornados pela função model.predict(test_X_new), que recebe como parâmetro os dados de validação das variáveis independentes parametrizadas previamente.

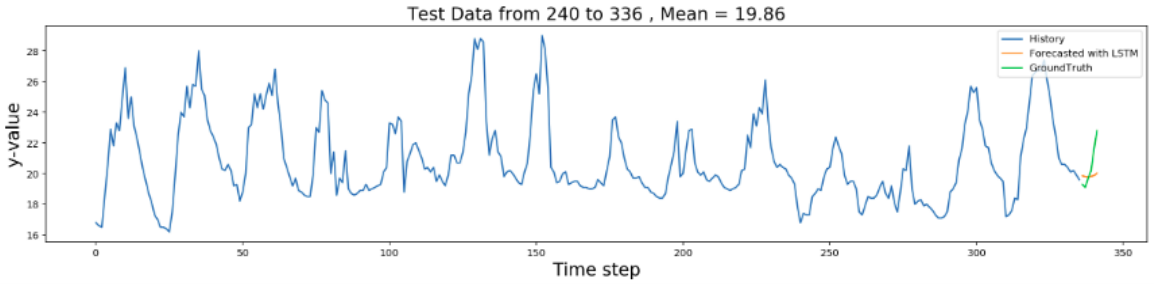
Na imagem abaixo pode-se visualizar o gráfico de valores históricos, comparados aos valores extrapolados pela previsão, juntamente às métricas estatísticas para validação do modelo.

Figura 42: Valores de Erro Obtido

```
Test Data from 240 to 336
Mean Absolute Error: 1.083333
Root Mean Square Error: 1.457981
```

Fonte: Autor

Figura 43: Gráfico Exibindo Dados Históricos e Valores das Previsões

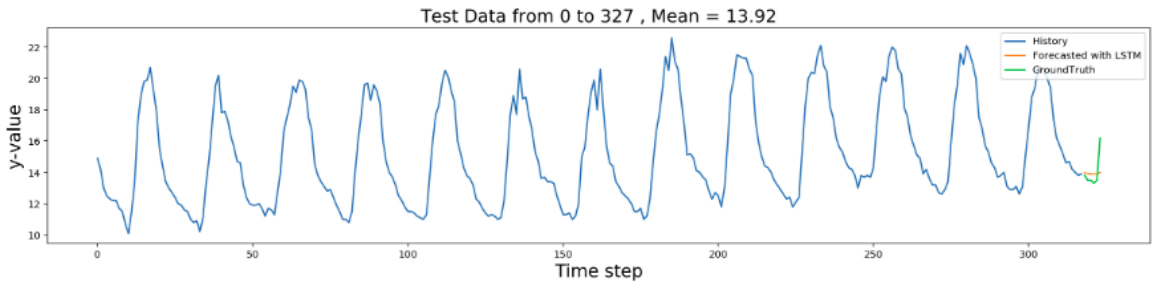


Fonte: Autor

Para o processo de previsão da variável Temperatura do Ar para 6 horas no futuro, utilizou-se como dados de entrada o dataset de dados dados_inmet_filtrados_2021.csv, composto por dados entre 01/07/2021 e 14/07/2021 e possuindo 327 registros históricos. Os métodos de preparação e inserção de dados no modelo foram os mesmos utilizados nas etapas de treinamento e validação dos dados já mencionadas e destrinchadas previamente no presente trabalho. Os valores das métricas estatísticas bem como o gráfico contendo os dados históricos e os valores previstos pelo modelo LTSM segue abaixo.

Figura 44: Valores Históricos e Predição Realizada pelo LSTM

```
Test Data from 0 to 327
Mean Absolute Error: 0.715907
Root Mean Square Error: 1.011332
```



Fonte: Autor

Paralelamente, para comparação com os resultados obtidos pelo LSTM, utilizou-se o ARIMA (Auto Regressive Model Moving Average), fortemente indicado em modelos preditivos. As bibliotecas abaixo foram utilizadas.

Figura 45: Bibliotecas Python Utilizadas

```
# Carregando as bibliotecas necessárias
from pandas import read_csv
from matplotlib import pyplot
from pandas import read_csv
from pandas import datetime
from pandas import DataFrame
from statsmodels.tsa.arima.model import ARIMA
import numpy
import pandas
import matplotlib.pyplot as plt
import math
```

Fonte: Autor

O dataset utilizado foi o “dados_inmet_filtrados_2021.csv” já descrito anteriormente no presente documento. Realizou-se a separação entre dados de treinamento e validação, como segue.

Figura 46: Preparação dos Dados de Validação

```
# Fazendo o split do dataset
#series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0)
y_series = pandas.Series(y)
split_point = len(y_series) - 6
dataset, validation = y_series[0:split_point], y_series[split_point:]
print('Dataset %d, Validation %d' % (len(dataset), len(validation)))
dataset.to_csv('dataset_Test_Arima.csv', index=False)
validation.to_csv('dataset_validation_Arima.csv', index=False)
```

Fonte: Autor

Utilizando a função Forecast(), utilizando os dados de teste foi previsto para as próximas 6 horas no futuro o valor da temperatura do ar, considerando apenas esta variável no processo de modelagem.

Figura 47: Inicialização do Modelo ARIMA

```
# Prevendo os proximos 6 steps usando a função forecast()
# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
    return yhat + history[-interval]

# Carrengando o dataset
series = read_csv('dataset_Test_Arima.csv', header=0)
# seasonal difference
X = series.values
hours_in_day = 24
differenced = difference(X, hours_in_day)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit()
# multi-step out-of-sample forecast
forecast = model_fit.forecast(steps=6)
# invert the differenced forecast to something usable
y_forecasted =[]
history = [x for x in X]
day = 1
for yhat in forecast:
    inverted = inverse_difference(history, yhat, days_in_year)
    print('Hora %d: %f' % (day, inverted))
    history.append(inverted)
    y_forecasted.append(inverted)
    day += 1
```

Fonte: Autor

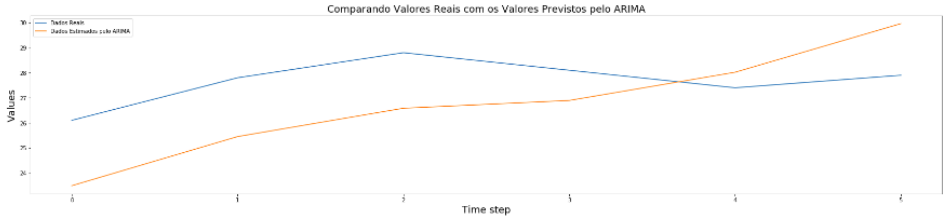
O resultado obtido foi:

Figura 48: Valores de Temperatura Preditos pelo ARIMA entre 14:00 e 19:00

```
Hora 1: 23.488886
Hora 2: 25.450610
Hora 3: 26.583281
Hora 4: 26.893117
Hora 5: 28.019567
Hora 6: 29.962166
```

Fonte: Autor

Figura 49: Gráfico de Comparação dos Valores




```
RMSE = 1.972492: MSE = 3.890726 MAE = 1.844307
```

Fonte: Autor

Utilizou-se o método Predict() para gerar as previsões, porém os resultados obtidos foram consideravelmente similares.

Figura 50: Configuração e Parametrização do ARIMA

```
# Prevendo os proximos 6 steps usando a função predict()
# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return numpy.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
    return yhat + history[-interval]

# Carregando o dataset
series = read_csv('dataset_Test_Arima.csv', header=0)
# seasonal difference
X = series.values
hours_in_day = 24
differenced = difference(X, hours_in_day)
# fit model
model = ARIMA(differenced, order=(7,0,1))
model_fit = model.fit()
# multi-step out-of-sample forecast
start_index = len(differenced)
end_index = start_index + 5
forecast = model_fit.predict(start=start_index, end=end_index)
# invert the differenced forecast to something usable
y_predicted =[]
history = [x for x in X]
day = 1
for yhat in forecast:
    inverted = inverse_difference(history, yhat, days_in_year)
    print('Day %d: %f' % (day, inverted))
    history.append(inverted)
    y_predicted.append(inverted)
    day += 1
```

Fonte: Autor

Figura 51: Valores de Temperatura Preditos pelo ARIMA entre 14:00 e 19:00

```
Hora 1: 23.488886
Hora 2: 25.450610
Hora 3: 26.583281
Hora 4: 26.893117
Hora 5: 28.019567
Hora 6: 29.962166
```

Fonte: Autor

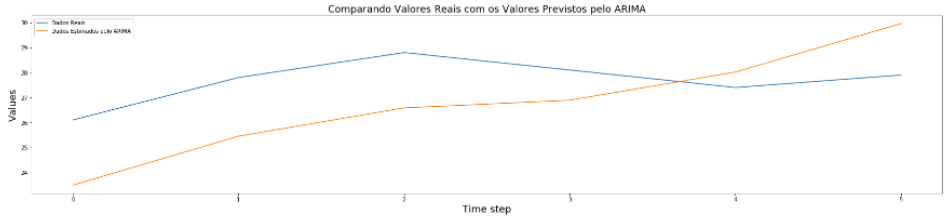
Realizando a validação do modelo ARIMA frente aos dados reais presentes no arquivo “dataset_validation_Arima.csv”, obteve-se o resultado abaixo:

Figura 52: Código Gerado Para Exibir os Resultados do Modelo

```
#Validando a previsão com o método "Predict" em relação aos dados reais

data_Real = read_csv('dataset_validation_Arima.csv')
data_Predicted = y_predicted

plt.figure(figsize=(30, 6))
plt.plot(data_Real[:7] , label='Dados Reais')
plt.plot(data_Predicted[:7] , label='Dados Estimados pelo ARIMA')
#plt.plot(y[:1000] , label='y')
plt.legend(loc='upper right')
plt.title("Comparando Valores Reais com os Valores Previstos pelo ARIMA" ,  fontsize=18)
plt.xlabel('Time step' ,  fontsize=18)
plt.ylabel('Values' , fontsize=18)
plt.legend()
plt.show()
calculateStatisticalMetrics(data_Real, data_Predicted)
```

Fonte: Autor

Figura 53: Gráfico Contendo Dados Reais e Preditos pelo ARIMA



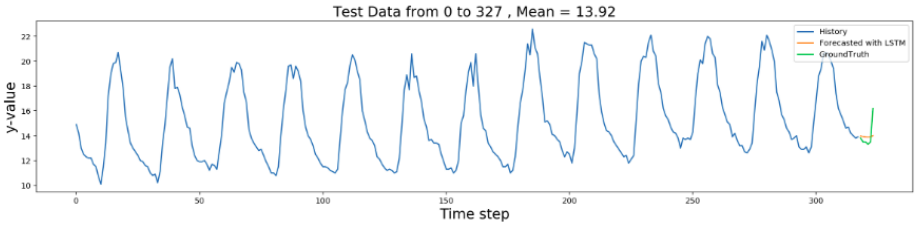

```
RMSE = 1.972492: MSE = 3.890726 MAE = 1.844307
```

Fonte: Autor

6. Apresentação dos Resultados

O resultado obtido utilizando o LSTM apresentou menores erros nas métricas estatísticas utilizadas (mean absolute error, root mean square error e mean square error) frente ao ARIMA, para realizar a previsão dos dados. Os dados previstos por ambos os modelos foram do dia 14/07/2021 das 14:00 às 19:00.

Figura 54: Resultados Obtidos pelo LSTM



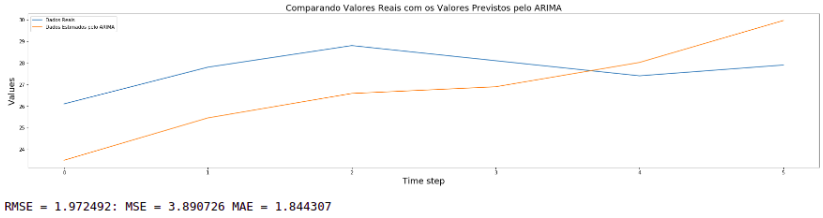
Fonte: Autor

Figura 55: Valores de Erro Obtidos pelo LSTM

```
Test Data from 0 to 327
Mean Absolute Error: 0.715907
Root Mean Square Error: 1.011332
```

Fonte: Autor

Figura 56: Resultados Obtidos Pelo ARIMA

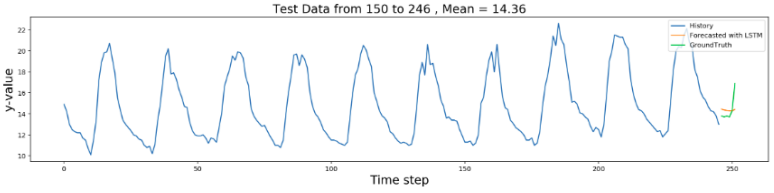


Fonte: Autor

Observou-se que o ARIMA também não acompanhou com precisão as mudanças de taxa de variação dos dados reais, tendo não obtido resultados consideravelmente precisos em regiões próximas à pontos de inflexão.

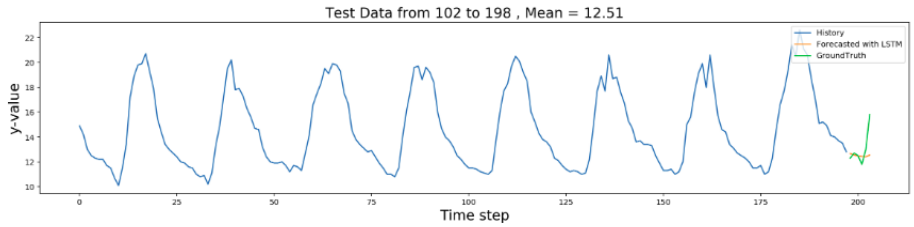
Mesmo tendo-se obtido menores erros com o Modelo LSTM, cabe salientar que este algoritmo não lidou com grande precisão com regiões próximas à pontos de inflexão, sendo que a taxa de variação dos dados previstos não conseguiu acompanhar com exatidão a taxa de variação dos dados reais, como segue nos exemplos abaixo.

Figura 57: Análise dos Resultados Obtidos pelo LSTM



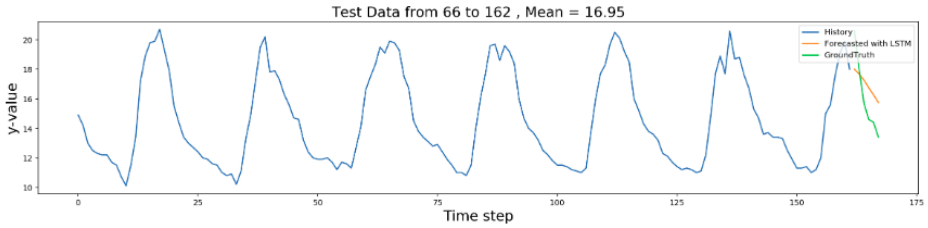
Fonte: Autor

Figura 58: Análise Parcial dos Dados Obtidos pelo LTSM



Fonte: Autor

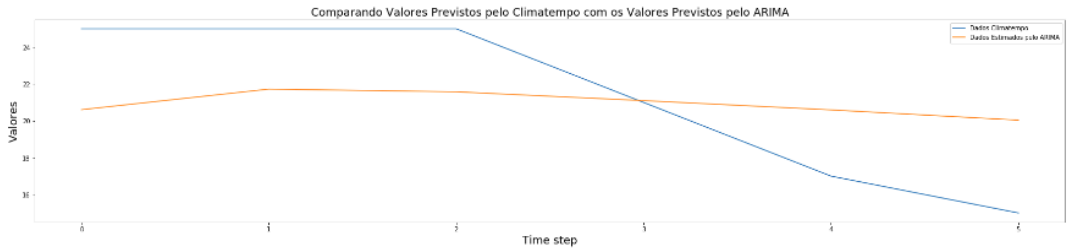
Figura 59: : Análise Parcial dos Dados Obtidos pelo LTSM



Fonte: Autor

Para fins de visualização e comparação, comparamos os resultados obtidos pelo ARIMA e o LSTM com dados previstos pelo ClimaTempo, detalhados previamente no presente trabalho no ítem referente à coleta de dados.

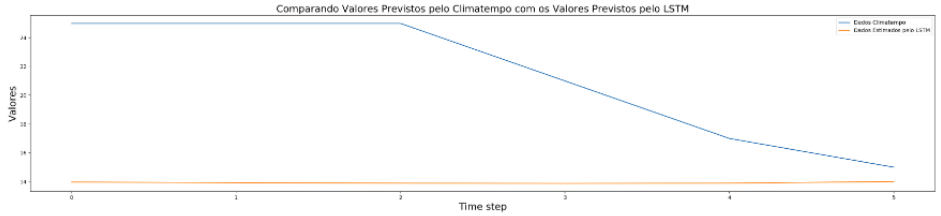
Figura 60: Gráfico Comparando dados Previstos pelo Climatempo com Dados Previstos pelo ARIMA




```
RMSE = 3.653571: MSE = 13.348581 MAE = 3.304568
```

Fonte: Autor

Figura 61: Gráfico Comparando dados Previstos pelo Climatempo com Dados Previstos pelo LSTM



Fonte: Autor

Observou-se um elevado valor de RMSE para o ARIMA e o LSTM, tendo o ARIMA apresentado melhor resultado. Entretanto, não se pode afirmar os dados obtidos pela API do Clima Tempo foram obtidos com os mesmos equipamentos de medição que os dados fornecidos pelo INMET, além do fato que não foi informado pela API Clima Tempo qual é a região da cidade onde foram obtidos os dados das medições, tendo essa comparação apenas um valor de visualização das respostas dos modelos propostos com dados obtidos através de métodos numéricos tradicionais.

Possivelmente, desenvolvendo um modelo com mais variáveis independentes relacionadas à temperatura do ar pode-se obter um modelo LSTM com menores erros, porém isso aumentaria o custo computacional e de armazenagem dos dados.

7. Links

Link para o vídeo:

https://www.youtube.com/watch?v=veCo2a3IxmA

Link para o repositório:

https://github.com/MarceloPereiraCunha/TCC_DATASCIENCE_MARCELOCUNHA

REFERÊNCIAS

Yonue, Rita Yuri. **Meteorologia.** Cidade: São Paulo, Editora USP/UNIVESP, 2012.

Preduna, T.G. **Neuro Model for Weather Forecast.** Bucarest: IEEE, 2020.

Booz, Jarret. **A Deep Learning-Based Weather Forecast System**

for Data Volume and Recency Analysis. Frotsburg: ICNC, 2019.

Mahmood, Hisham. **Short-Term Photovoltaic Power Forecasting Using**

an LSTM Neural Network and Synthetic Weather Forecast. Lakeland: IEEE Access, 2020.

Haupt, Sue Ellen. **Machine Learning for Applied Weather Prediction.** Cidade: IEEE, 2018.

Salman, Afam Galih. **Weather Forecasting using Deep Learning Techniques.** Jakarta: IEEE, 2018.