UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
COLEGIADO DO CURSO DE ENGENHARIA AMBIENTAL

RAYANE CARDOZO GAMA BOTELHO

ESTIMATIVA DE PARTÍCULAS INALÁVEIS (MP10) COM BASE NAS VARIÁVEIS DA QUALIDADE DO AR, DADOS METEOROLÓGICOS E NÚMERO DE VEÍCULOS USANDO APRENDIZAGEM DE MÁQUINA:

ESTUDO DE CASO DA ESTAÇÃO DE MONITORAMENTO DA QUALIDADE DO AR (EMQAR - RGV 4)

VITÓRIA
2022

RAYANE CARDOZO GAMA BOTELHO

ESTIMATIVA DE PARTÍCULAS INALÁVEIS (MP_{10}) COM BASE NAS VARIÁVEIS DA QUALIDADE DO AR, DADOS METEOROLÓGICOS E NÚMERO DE VEÍCULOS USANDO APRENDIZAGEM DE MÁQUINA

ESTUDO DE CASO: ESTAÇÃO DE MONITORAMENTO DA QUALIDADE DO AR (EMQAR - RGV 4)

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Ambiental, como parte dos requisitos necessários à obtenção do grau de Bacharel em Engenharia Ambiental.

Orientador: Frederico Damasceno Bortoloti

VITÓRIA
2022

RAYANE CARDOZO GAMA BOTELHO

ESTIMATIVA DE PARTÍCULAS INALÁVEIS (MP_{10}) COM BASE NAS VARIÁVEIS DA QUALIDADE DO AR, DADOS METEOROLÓGICOS E NÚMERO DE VEÍCULOS USANDO APRENDIZAGEM DE MÁQUINA

ESTUDO DE CASO: ESTAÇÃO DE MONITORAMENTO DA QUALIDADE DO AR (EMQAR - RGV 4)

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Ambiental, como parte dos requisitos necessários à obtenção do grau de Bacharel em Engenharia Ambiental.
Orientador: Frederico Damasceno Bortoloti

Aprovada em _____ de _____ de _____.

COMISSÃO EXAMINADORA

_____
Prof. Frederico Damasceno Bortoloti
Universidade Federal do Espírito Santo
Orientador

_____
Prof. Neyval Costa Reis Júnior
Universidade Federal do Espírito Santo

_____
Prof. Elisa Valentim Goulart
Universidade Federal do Espírito Santo

AGRADECIMENTOS

Ao meu orientador que sempre acompanhou todo o trabalho com compromisso e paciência para que eu conduzisse da melhor forma ao longo de toda a construção do TCC.

A minha família que mesmo de longe e indiretamente me ajudou durante toda a trajetória escolar para que eu estivesse chegado a esta etapa da graduação e compreensão de inúmeros finais de semanas dedicados a horas de estudos.

Ao meu namorado, por sempre ser compreensível e por fornecer apoio emocional, durante toda a graduação e por ter chegado até esta fase final do curso, sempre me motivando nesta importante etapa da minha vida.

RESUMO

O presente trabalho tem por objetivo a estimativa do material particulado MP10 a partir da criação de modelos de aprendizagem de máquina como rede neural profunda (DNN), *Random Forest* (RF) e *eXtreme Gradient Boosting* (XGBoost). Para o treinamento dos modelos, utilizou-se dados de frota veicular como: automóveis, caminhões, motos, ônibus, utilitários e outros, dados de poluentes atmosféricos como: dióxido de enxofre (SO_2), dióxido de nitrogênio (NO_2), monóxido de carbono (CO), ozônio (O_3) e dados meteorológicos como: direção escalar do vento (°) e velocidade escalar do vento (m/s). Os parâmetros foram selecionados para os anos de 2010 a 2021. Com os modelos gerados foi possível comparar os valores estimados do MP10, que apresentou valores próximos dos reais com erro médio absoluto (MAE) de 3,76 para o melhor modelo sendo ele RF. O modelo DNN também apresentou valores próximos dos reais mas não alcançou um melhor resultado quando comparado ao RF assim como ao modelo de XGBoost.

Palavras-chave: MP10. Aprendizagem de máquina. Poluentes. Deep Neural Networks. Random Forest. XGBoost.

ABSTRACT

The present work aims to predict the PM10 particulate matter by creating machine learning models such as deep neural network (DNN), Random Forest (RF) and eXtreme Gradient Boosting (XGBoost). In order to train the models, we used vehicle fleet data such as: cars, trucks, motorcycles, buses, utilities and others, air pollutant data such as: sulfur dioxide (SO_2), nitrogen dioxide (NO_2), carbon monoxide (CO), ozone (O_3) and meteorological data such as: scalar wind direction (°) and scalar wind speed (m/s). The parameters were selected for the years 2010 to 2021. We compared the estimate values of the MP10 of the generated models, which presented values close to the real ones with mean absolute error (MAE) of 3.76 for the best model, the RF. The DNN model also presented values close to the real ones but did not reach a better result when compared to the RF as well as to the XGBoost model.

Keywords: PM10. Machine learning. Pollutants. Deep Neural Networks. Random Forest. XGBoost.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

1. INTRODUÇÃO

A poluição do ar, principalmente nos ambientes urbanos, é um dos grandes problemas ambientais enfrentados em muitas cidades, principalmente nas grandes metrópoles. As partículas inaláveis grossas ou material particulado grosso (MP10) é um dos principais poluentes atmosféricos advindos de processos mecanizados (FREITAS, SOLCI; 2009). Estes poluentes estão associados a diversas doenças, especialmente às doenças respiratórias.

De acordo com o Instituto Estadual de Meio Ambiente e Recursos Hídricos (IEMA; 2021), a qualidade do ar está relacionada com a maneira e a distribuição das emissões de poluentes atmosféricos advindos de indústrias e de veículos automotores. Na região metropolitana da Grande Vitória, além de contar com um número significativo de veículos e um complexo industrial, há também o setor de logística devido ao aeroporto e ao complexo portuário. Todos esses parâmetros contribuem significativamente na qualidade do ar, através da emissão de poluentes que são classificados de acordo com a resolução do CONAMA nº 491, de 19 de novembro de 2018.

Avaliando os impactos ocasionados pelos poluentes atmosféricos, principalmente o poluente MP10 e visando como saber lidar com o problema da poluição atmosférica que afeta o ambiente físico e consequentemente a saúde humana, há uma preocupação em como se conhecer melhor o MP10 e controlar os problemas causados por este poluente.

Há basicamente duas formas de se lidar com dados monitorados, a predição futura de um determinado parâmetro e também o preenchimento de falhas de um determinado parâmetro. Exemplos de predição futura são os trabalhos de (REISEN *et al.*, 2014), (SOARES *et al*., 2015), (LIRA, 2009), (DELAVAR *et al.*, 2019), (NIU *et al*., 2016), (WU; LIN, 2019) e (WANG *et al.*, 2018). E exemplos te preenchimento de falhas temos os trabalhos de (TZANIS *et al.*, 2021) e (NOOR, *et al.*, 2015).

1.1 JUSTIFICATIVA

Considerando situações em que o MP10, seja por falha de sensor, ou por

indisponibilidade do mesmo, não seja monitorado, vê-se necessário a estimativa deste poluente atmosférico a partir de outras leituras disponíveis para preenchimento de falhas num mesmo instante. Por esse motivo, um modelo de estimativa do MP10 pode ser viável desde que se tenha uma série histórica mínima do MP10 e de outras leituras para a localidade.

Dessa forma pode-se utilizar dados de poluentes atmosféricos, dados meteorológicos e de veículos, do período de 2010 a 2021 associados pela estação de monitoramento EMQAr – RGV 4, para estimar o MP10, após a criação de um modelo estimativo. Esta modelagem poderá auxiliar em planejamentos urbanos visando uma melhoria na qualidade do ar nos ambientes afetados pela poluição do MP10, preenchendo falhas quando o mesmo não estiver sendo monitorado.

1.2 OBJETIVO GERAL

O presente trabalho tem por objetivo geral desenvolver um modelo de regressão do quantitativo de material particulado de 10 μ (MP10) com base nas variáveis da qualidade do ar, dados meteorológicos e número de veículos usando rede neural profunda, *eXtreme Gradient Boosting* (XGBoost) (CHEN e HE, 2014), e o *Random Forest* (RF) (Denisko & Hoffman, 2018). Será feito um estudo de caso para a estação de monitoramento da qualidade do ar (EMQAr - RGV 4) situada no bairro Enseada do Suá localizado no município de Vitória, Espírito Santo, em que foram coletados dados históricos do período de 2010 a 2020 de alguns parâmetros e também dados de frota de veículo de 2010 a 2021, para estimar o MP10 nos anos de 2019 e 2020.

1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos são: selecionar cinco melhores características para obter os parâmetros mais relevantes para o modelo estimativo, usando métodos de seleção de características; e verificar correlações entre as variáveis monitoradas e o MP10.

Este trabalho está organizado da seguinte forma. O Capítulo 2 apresenta uma revisão de literatura a respeito da poluição do ar e quais parâmetros são

estabelecidos dentro do índice de qualidade do ar, o que são partículas inaláveis grossas, trabalhos correlatos sobre o material particulado, os tipos de regressão dentro do aprendizado supervisionado, conceitos de *deep neural network,* o que são, redes neurais convolucionais, camada de *pooling*, funções de ativação, batch *normalization*, f*ully connected network,* regularização, *random forest, extreme gradient boosting, e*rro médio absoluto, erro médio quadrático*, root mean propragation, F-value e i*nformação mútua. O Capítulo 3 apresentará como foi realizado o processo de coleta de dados, quais tratamentos foram aplicados, em que ambiente foi desenvolvido e qual ferramenta usada e como foi obtido o modelo preditivo. No Capítulo 4 são apresentados os resultados e discussões a respeito dos testes realizados e no Capítulo 5 o que se concluiu ao final deste trabalho.

2. REVISÃO DA LITERATURA

Essa seção fará uma revisão de literatura com tópicos sobre poluição do ar, trabalhos correlatos sobre o MP10, regressão em aprendizado supervisionado, conceitos sobre *Deep Neural Network,* incluindo *redes neurais convolucionais, camada de pooling, funções de ativação, batch normalization, fully connected network,* regularização, *random forest, extreme gradient boosting,* erro médio absoluto, erro médio quadrático*, root mean propragation, F-value e* informação mútua.

2.1 POLUIÇÃO DO AR

BRASIL (1976) estabeleceu-se inicialmente os padrões de qualidade do ar estaduais, e os padrões nacionais foram estabelecidos pelo Instituto Brasileiro de Meio Ambiente e aprovados pela resolução do CONAMA n° 03/90 (BRASIL, 1990). Os parâmetros contemplados que compõem o índice de qualidade do ar são: partículas inaláveis grossas (MP10), partículas inaláveis finas (MP2,5), ozônio (O_3), monóxido de carbono (CO), dióxido de nitrogênio (NO_2), dióxido de enxofre (SO_2) entre outros. Para cada tipo de poluente é calculado o índice de qualidade do ar e assim o ar recebe uma classificação numérica acompanhada de uma cor que indica a situação e classe da qualidade do ar (CETESB, 2021).

As partículas inaláveis grossas (MP10) advém de processos mecanizados como cinzas de combustão, da poeira do solo, do sal marinho e de emissões biogênicas industriais. Já as partículas respiráveis < MP2,5, podem ter origem de processos de combustão no setor industrial, da frota veicular além de outras fontes. Esse material particulado é um poluente que vem recebendo atenção devido aos problemas respiratórios principalmente que tem causado na saúde da população. Há estudos que relacionam graves problemas do material particulado grosso, agravando problemas de asma pelo acúmulo desse particulado nas vias respiratórias superiores. (FREITAS; SOLCI, 2009).

Há uma grande relação entre o material particulado (MP) e os seus efeitos com a saúde humana e os impactos ambientais. Segundo relatório da Organização

Mundial da Saúde (OMS), dentre as causas de mortes atribuídas por poluição atmosférica no ano de 2012, estão: doença aguda do trato respiratório inferior, doença pulmonar obstrutivo crônico e cardiopatia isquêmica, sendo esta última doença juntamente ao derrame, soma-se oitenta por cento dos óbitos. Homens e mulheres com vinte e cinco anos ou mais, representam o maior percentual de mortes totalizando noventa e sete por cento neste mesmo ano. (OMS, 2012).

2.2 TRABALHOS CORRELATOS DE MATERIAL PARTICULADO (MP)

Trabalhos sobre estimativa por preenchimento de falhas de poluentes foram feitos nas regiões de Atena-Grécia (TZANIS *et al.*, 2021) e na Malásia (NOOR, *et al.*, 2015).

Também foram feitos trabalhos sobre a predição de materiais particulados (MP2,5 e MP10) foram feitos nas regiões de Cariacica – Brasil (REISEN *et al.*, 2014) Uberlândia - MG, (SOARES et al., 2015) e (LIRA, 2009), na capital da cidade de Teerã no Irã (DELAVAR et al., 2019), nas cidades de Harbin e Chongqing - China (NIU *et al*., 2016), Pequim e Guilin - China (WU; LIN, 2019) e também em Pequim, Xangai e Cantão - China (WANG *et al.*, 2018).

Em um estudo feito em Antenas na Grécia (TZANIS *et al.*, 2021) foi realizado uma estimativa espacial de concentrações de poluentes usando dados de monitoramento de uma rede de estações operando em Atenas. As variáveis com falhas trabalhadas são: NO_2, O_3, MP10, MP2,5, SO_2. Foi estimado um total de 12.526 dados ausentes, desse total mais de 40% são dados ausentes de MP_{10} e $MP_{2,5}$, também mais de 20% para O_3 e SO_2, e mais de 15% para o NO_2. Foi estipulado um número de estações para cada poluente como dados de entradas para desenvolver modelos de redes neurais. O critério dessa seleção se baseou em atender um mínimo de percentual de 80% de dados na estação porém algumas exceções foram incluídas para a na análise e estações com dados ausentes de 1 anos foram desconsideradas. Foram utilizadas quatorze estações para o NO_2, treze para O_3, onze para MP10, seis para MP2,5 e seis para o SO_2,essas variáveis foram monitoradas por hora um período de 3 anos (2016–2018). Utilizou-se modelos de redes neurais para realizar a interpolação dos dados para estimar os dados ausentes das variáveis trabalhadas. Os resultados apresentados que o desempenho

dos modelos é afetado pelo número de estações e dados por estações, além dos padrões característicos de cada poluente.

Para o trabalho realizado na Malásia (NOOR, *et al.*, 2015) a concentração do MP10 foi obtida por hora em 8 estações de monitoramento de 2000 a 2009. Os dados ausentes correspondem a no máximo 12 horas. Foram utilizados métodos de imputação interpolação linear (LI), média acima abaixo (MAB), interpolação do vizinho mais próximo (NN), média diária (DM), média de 12 horas (12M), média de 6 horas (6M), média de linha (RM) e ano anterior (PY) onde foram calculados para preencher os dados ausentes que foram simulados. Foram utilizados 4 critérios estatísticos (MAE, MSE, precisão da previsão e índice de concordância) para avaliar o desempenho dos métodos. O método 6M apresentou bons resultados comparativamente com a interpolação linear. Os métodos RM e MI apresentaram desempenho moderado. E a métrica MAE de menor valor se apresentou para o método LI, com valor a cerca de 13.

Neste trabalho realizado em Cariacica, ES – BRASIL (REISEN *et al*., 2014) a concentração média diária do MP10 é o conjunto de dados trabalhos e essa concentração é considerada como um processo sazonal fracionalmente integrado. O modelo Sazonal Autoregressivo Fracionadamente Integrado a Média Móvel (SARFIMA) é utilizado aplicado a concentração de MP10 na cidade de Cariacica. A série de dados bruta possui 1826 observações que compreende o período de 1º de janeiro de 2005 até 31 de dezembro de 2009. Os dados de 23 de maio a 31 de dezembro do ano de 2009, equivalentes a 223 observações foram descartados da modelagem e foram usados para predição a frente da amostra. Foram utilizadas as métricas erro quadrático médio (MSE) e erro simétrico de Percentual Absoluto Médio (sMAPE) e modelo SARFIMA-GARCH se apresentou com alto nível de precisão.

No trabalho de estudo de caso realizado na cidade de Uberlândia (SOARES *et al*., 2015) para a predição do material particulado MP10 de origem veicular para o ano de 2012, utilizou-se o modelo matemático CAL3QHC que avalia a dispersão de monóxido de carbono (CO) e material particulado. Como parâmetros de entrada foram considerados a frota de veículos movidos a diesel e gasolina, possuindo ano de fabricação entre 1982 e 2012. Os parâmetros de concentrações de MP10 usados

foram composto por uma média mensal com coleta realizada de três em três dias no período de amostragem de 24 horas, durante o ano de 2012. O modelo CAL3QHC apresentou um desempenho moderado na predição do MP10.

Em outro estudo na cidade de Uberlândia (LIRA, 2009), objetivou desenvolver modelos empíricos para prever a concentração de MP10, usando modelos lineares como ARX, ARMAX, Box-Jenkins e erro na saída e também modelos baseados em redes neurais de 4, 5 e 6 neurônios. Como entrada foram considerados o fluxo de veículos e variáveis meteorológicas e como saída a predição da concentração de MP10 com horizonte de predição de três dias à frente. Foram obtidas médias mensais para os anos de 2003, 2004, 2005, 2006 e 2007 dos parâmetros entrada e saída. Como resultado as melhores estimativas derivaram do modelo Box-Jenkins, o RMSE apresentado para esse modelo é de 0,2297. No geral os resultados apresentaram boas estimativas exceto o modelo linear erro da saída.

Para o trabalho feitos nas cidades de Pequim e Guilin (WU; LIN, 2019) foi desenvolvido um modelo híbrido contendo a decomposição secundária (SD), a técnica de decomposição transfomada wavelets (WD) é escolhida para gerar uma sequência de detalhes de alta frequência - WD(D) e também uma sequência de aproximação de baixa frequência - WD(A). Foi adotado a decomposição em modo variacional (VMD) melhorado pela entropia da amostra (SE) para suavizar a amostra e então é aplicado LSTM para facilitar a previsão. Foi usado LSSVM com os parâmetros otimizados pelo algoritmo Bat (BA) que utiliza os fatores de poluentes atmosféricos de: MP2,5, MP10, O_3, SO_2, CO e NO_2, sendo esses fatores adequados para prever WD(A), extraindo informações de Índice de Qualidade do Ar (IQA). Foram coletados séries de dados diários de (IQA) no período de 1° de dezembro de 2016 a 31 de dezembro de 2018, nas cidades de Pequim e Guilin. O erro médio absoluto (MAE) encontrado para a cidade de Pequim foi de 6,6885 e para Guilin foi de 3,8036. O modelo híbrido proposto compreende as características das séries originais de IQA e apresenta uma alta taxa de previsão correta das classes de IQA.

No trabalho realizado em Harbin e Chongqing cidades localizadas na China (NIU *et al*., 2016), foi utilizado um modelo híbrido de decomposição em conjunto e um otimizador chamado de *grey wolf* (GWO) que é usado para a predição da concentração do MP2,5. Em comparação com outros métodos que prevê o MP2,5,

este modelo mostrou melhoras na precisão da predição e também das taxas de acertos da predição direcional. O modelo proposto é composto por três etapas principais: primeiro decompõe o MP2,5 da série original em várias funções de modo intrínseco (IMFs) através da Decomposição Complementar em Modos Empíricos por Conjunto (CEEMD), para simplificar os dados complexos, individualmente, prevendo cada IMF com Suporte a Regressão Vetorial (SVM) que é otimizada por GWO, integrando todos os IMFs preditos para o conjunto resultado como predição final feita por outro SVM otimizado pelo GWO. De acordo com o estudo empírico, o modelo de conjunto de decomposição híbrido proposto é notavelmente superior a todos os modelos de referência considerados por sua maior precisão de previsão e taxas de acerto de previsão direcional. Os dados da amostra da concentração diária de MP2,5 são do período de 1° de novembro de 2013 até 20 de setembro de 2015, totalizando 689 dados de concentração de MP2,5. Foi utilizado a média do dia seguinte e do dia anterior para substituir os cinco dias dos valores ausentes. Depois da etapa de pré-processamento dos dados ausentes, são apresentados as características dos dados de Harbin e Chongqing respectivamente incluindo a média (69,679255 e 60,9356), o máximo (518 e 211), o mínimo (8 e 8) e a variância (4441,283 e 1529,577) para as duas séries originais de concentração de MP2,5. O erro médio absoluto (MAE) encontrado para a cidade de Harbin foi de 2.4103 Chongqing foi de 3.1741. Os dados da amostra foram divididos em duas partes, um subconjunto de treinamento contendo 490 dados (cerca de 70% dos dados totais), e um subconjunto de teste contendo o restante.

Para o trabalho realizado em Pequim, Xangai e Cantão também localizado na China (WANG *et al.*, 2018) foi construído um sistema de alerta de poluição do ar compreendendo dois módulos: um módulo de predição de poluição e um módulo de avaliação de qualidade do ar. No módulo de predição foram utilizados dois métodos de *denoising* e um algoritmo de otimização multi-objetivo, formando um modelo híbrido de predição. No módulo de avaliação foi usada a avaliação sintética *fuzzy* para avaliar a qualidade do ar. Foram utilizados dados de concentração diária de poluentes nas cidades de Pequim, Xangai e Cantão e também realizadas três simulações numéricas. O modelo L2,1RF-ELM apresentou melhor desempenho de predição em relação a predição da rede neural tradicional como resultado das simulações. O modelo híbrido proposto é melhor que o modelo estatístico tradicional

ARIMA quando comparados. Os poluentes das três cidades estudadas são: MP2,5, MP10, CO, O_3, SO_2 e NO_2 e os dados de concentração diária são do período de 1° de janeiro de 2017 a 1° de julho de 2017. Os dados foram divididos em dois subconjuntos, um subconjunto de dados de treinamento com 3480 dados e um subconjunto de teste com 888 dados. O erro médio absoluto (MAE) para o MP no modelo híbrido proposto foi de 5,0582 para a cidade de Xangai e 3,2802 para a cidade de Cantão. Logo a predição do modelo híbrido proposto (SSA-EEMD-MOALO-L2,1RFELM, SEMR) foi alcançada para este trabalho.

No trabalho realizado na capital do Irã, Teerã (DELAVAR et al., 2019), foram feitos modelos de predição fazendo uso de métodos de aprendizagem de máquina, incluindo máquina de vetores de suporte de regressão (SVR), regressão geograficamente ponderada (GWR), rede neural artificial (ANN) e rede neural não linear auto-regressiva com uma entrada externa (NARX), para predição dos poluentes MP2,5 e MP10. Foi proposto um modelo de predição para melhorar os métodos mencionados, em que o erro percentual foi reduzido e melhorado em 57%, 47%, 47% e 94%, respectivamente. O algoritmo mais confiável para a predição foi o NARX que foi usado como modelo proposto e seu erro de predição de um dia chegou a 17 $\mu g/m^3$ e apresentou erro médio absoluto (MAE) de 1,45 para o MP10.

Tendo em vista o presente trabalho, a grande diferença dos trabalhos citados até aqui, é que o mesmo faz a estimativa do MP10 imputando o seu valor baseando-se em leituras de outras variáveis, enquanto que os trabalhos de preenchimento de falhas citados utilizam valores do próprio MP10 (das proximidades ou de anteriores) para imputar seu valor.

2.3 APRENDIZADO SUPERVISIONADO - REGRESSÃO

O aprendizado de máquina utiliza diversas maneiras de se aprender com os dados. A depender do tipo de entrada fornecida e da saída esperada, pode-se classificar os algoritmos pelo seu estilo de aprendizado, categorizados como: supervisionado, não supervisionado, autosupervisionado e por reforço. Algoritmo supervisionado requer os que o analista ensine o algoritmo entrando com dados de exemplos com seus devidos atributos e classe que é o caso da classificação, ou seu valor numérico de retorno esperado caso seja regressão. O treinamento usado cria um modelo onde um algoritmo irá se ajustar aos dados, no processo de

treinamento, as previsões ou classificações irão avançar tornando-se mais precisas. Pode-se citar como um dos exemplos de algoritmos supervisionados a regressão linear ou logística (MUELLER; MASSARON, 2020). A regressão linear pode ser simples ou múltipla, quando se quer prever o valor de um dado a partir de outro dado, e quando se quer saber a relação de uma dado com o outro, usa-se a regressão linear simples.

Shang *et al.* (2019) lista como principais modelos estatísticos: a regressão linear e regressão linear generalizada, regressão não linear, modelo auto-regressivo integrado de médias móveis (ARIMA), modelo oculto de Markov (HMM), *Random Forest* (RF), regressão vetorial de suporte (SVR), e rede neural artificial (ANN), incluindo *Extreme Learning Machines* (ELM), árvores de regressão (CART), além de comentar sobre modelos híbridos. Desses, somente os cinco últimos modelos da lista são normalmente considerados como de aprendizagem de máquina. Apesar de amplamente usados, a regressão linear e o ARIMA são por definição lineares, portanto não se adaptam muito bem ao problema de poluentes do ar (NIU *et al*., 2016), enquanto que o HMM sofre de vários problemas inerentes, pois é custoso computacionalmente e depende muito da condição inicial (BUDAKALOTI; SRIVATAVA; OTEY, 2009).

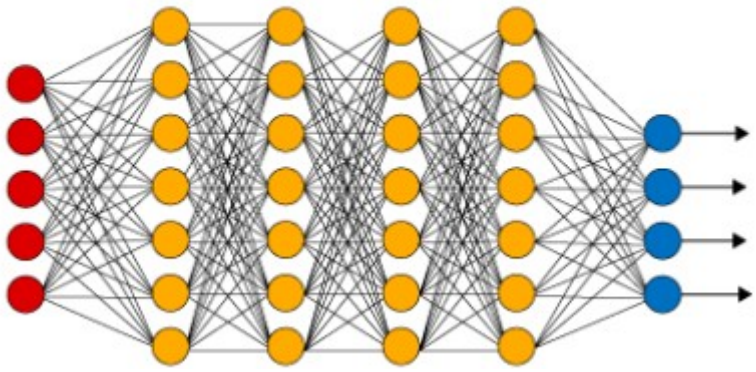
2.3.1 *Deep Neural Network*

Existem diversas pesquisas que visam a classificação e avaliação da qualidade do ar que faz uso de algoritmos de aprendizagem de máquina, utilizando diversos modelos para predição da qualidade do ar (KANG *et al.*, 2018), como por exemplo, o modelo de Rede Neural Artificial (*Artificial Neural Networks* ou ANN). Essa arquitetura busca simular estruturas e redes do cérebro humano, consistindo de neurônios que podem ou não transmitir sinais uns aos outros usando uma função de ativação.

A rede neural profunda ou *Deep Neural Network*, utiliza camadas de neurônios em que por meio dessas camadas a informação é transmitida. Em uma

rede, a primeira camada se chama camada de entrada, depois tem-se as camadas ocultas que intermedeiam a rede e por fim a camada de saída. Cada camada possui um tipo de função de ativação e pode-se também caracterizá-la como um algoritmo simples. (DATA, 2022). Pode-se ver na Figura 1 uma rede neural profunda sendo representada.

Figura 1: Rede Neural Profunda



Fonte: DATA (2022)

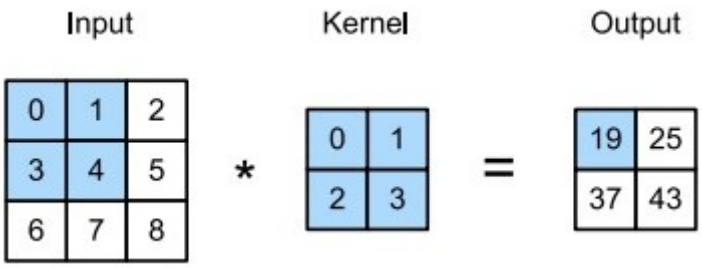
2.3.1.1 Redes Neurais Convolucionais

As redes neurais convolucionais (*Convolutional Neural Networks* - CNN) (LECUN *et al.*, 1998) contribuíram fortemente para os avanços no aprendizado profundo atualmente. As convoluções fazem uso de cálculos matemáticos para processar imagens com a finalidade de reconhecer diversas imagens com precisão de maneira otimizada. As camadas das redes neurais convolucionais realizam tarefas de maneira hierárquica. Em seu funcionamento para reconhecimento de imagem, por exemplo, as camadas iniciais irão detectar as bordas da imagem, as camadas intermediárias identificam formas mais complexas que são criadas pelas bordas e as camadas finais detectam características da imagem a ser classificada como por exemplo identificar a pata de um cachorro, ou o bico de um pato. Com base na hierarquia de padrões identificados seguida pelas convoluções, é possível explicar porquê as redes convolucionais profundas apresentam melhor desempenho que as redes rasas, quanto mais convoluções melhor será o entendimento da rede para aprender padrões mais complexos e por fim ter um reconhecimento de imagem eficiente. (MUELLER; MASSARON, 2020, p.183-195).

A rede neural convolucional é um tipo especializado de modelo de rede neural projetado para trabalhar com dados de imagem bidimensionais (2D), embora possam ser usados com dados unidimensionais (1D) e tridimensionais (3D). O item central para a rede neural convolucional é a camada convolucional. Esta camada realiza uma operação chamada de convolução. Uma convolução é uma operação linear que envolve a multiplicação de um conjunto de pesos com uma entrada, assim como em uma rede neural tradicional. A técnica foi projetada para entrada bidimensional, então a multiplicação é realizada entre uma matriz de dados de entrada e uma matriz bidimensional de pesos, chamado de filtro ou kernel (BROWNLEE, 2019).

O filtro é menor que os dados de entrada e o tipo de multiplicação aplicado entre um recorte do tamanho do filtro da entrada e o filtro é um produto escalar. Um produto escalar é a multiplicação elemento a elemento entre o recorte do tamanho do filtro da entrada e do filtro, que é então somado, sempre resultando em um único valor. Por resultar em um único valor, a operação é frequentemente chamada de produto escalar. Com isso, o mesmo filtro (conjunto de pesos) é multiplicado pela matriz de entrada várias vezes em diferentes pontos da entrada. Especificamente, o filtro é aplicado sistematicamente a cada parte sobreposta ou recorte do tamanho do filtro dos dados de entrada, da esquerda para a direita, de cima para baixo (BROWNLEE, 2019). Pode-se melhor exemplificar como funciona um filtro observando a Figura 2 onde há a operação de convolução bidimensional e os locais destacados em azul representando o primeiro elemento de saída.

Figura 2: Operação de convolução bidimensional. As partes em azul são o primeiro elemento de saída e os elementos de matriz de entrada e kernel usados em seu cálculo: 0 × 0 + 1 × 1 + 3 × 2 + 4 × 3 = 19.



Fonte: ZHANG *et al.* (2022)

Pode-se ver um filtro sendo aplicado a uma região da matriz de entrada, que tem tamanho 2x2, e produz um único elemento no mapa de características. Esse filtro irá deslocar da esquerda para a direita em uma unidade (*stride* de 1), e será novamente aplicado, gerando o segundo elemento da primeira linha do mapa de características, e depois voltará à esquerda, deslocando uma unidade para baixo, e assim por diante, até completar o mapa de características.

A saída da multiplicação do filtro pela matriz de entrada uma vez é um valor único. Como o filtro é aplicado várias vezes à matriz de entrada, o resultado é uma matriz bidimensional de valores de saída que representa uma filtragem da entrada. A matriz de saída bidimensional desta operação é chamada de mapa de características (*feature map*). Uma vez que um mapa de característica é criado, podemos passar cada valor no mapa de característica através de uma função de ativação não linear, como uma ReLU, assim como fazemos para as saídas de uma camada totalmente conectada (BROWNLEE, 2019).

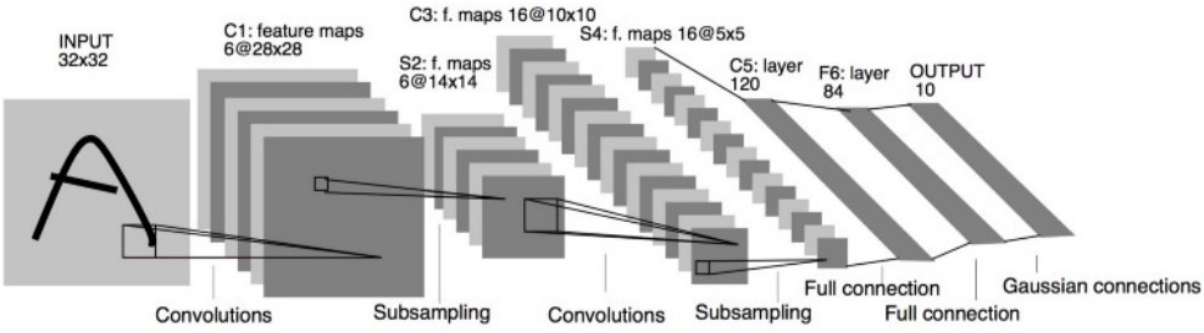
Em resumo, temos uma entrada (*input*), como uma imagem de valores de pixel, e temos um filtro (*kernel*), que é um conjunto de pesos, e o filtro é aplicado sistematicamente aos dados de entrada para criar um mapa de características (*feature map*) (BROWNLEE, 2019).

Além disso, a uma entrada, pode-se aplicar múltiplos filtros do mesmo tamanho de uma vez, gerando assim múltiplos mapas de características. Deve-se fazer isso especificando um número de filtros a ser aplicado, bem como o tamanho do filtro, que são os hiperparâmetros mais básicos de configuração da camada convolucional. Também se especifica o tamanho do deslocamento do filtro sobre a entrada (*stride*), que comumente é igual um. (IBM, 2020).

Na Figura 3, temos um exemplo de arquitetura de rede convolucional. Como *input*, temos uma matriz de entrada de dimensão 32x32, na sequência, uma camada convolucional C1 com 6 filtros de tamanho 28x28, que filtra as informações mais relevantes simplificando e gerando 6 mapas de características. Os mapas de características são subamostrados em um filtro com uma janela de 2x2 (uma operação conhecida por *pooling*), e passam por outra convolução (C3), com 16 filtros de 10x10, produzindo 16 mapas de características. Eles são novamente

subamostrados com uma janela de 2x2, e depois redimensionados para 400, passando a uma rede de 3 camadas totalmente conectadas (de 120, 84 e 10 neurônios) resultando na camada de saída.

Figura 3: Um exemplo de rede convolucional: a arquitetura da rede LeNet5

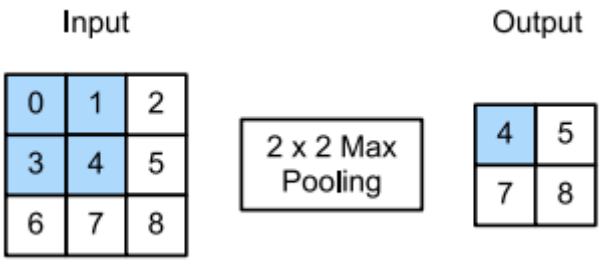


Fonte: ALVES (2018)

2.3.1.2 Camada de *Pooling*

Em redes neurais convolucionais, além de camadas convolucionais, há também as camadas de *Pooling* ou camadas de agrupamento. As camadas de *Pooling* são usadas posteriormente às camadas convolucionais, dessa maneira, simplificam as informações na saída da camada convolucional. A partir de cada saída do mapa de características da camada convolucional (saída de cada neurônio da camada), a camada de *pooling* irá preparar um mapa de características condensadas, por exemplo, aplicando a ativação máxima de uma região da imagem (kernel), o que se denomina de Max-Pooling. O Max-Pooling é uma técnica utilizada para cada mapa de características, em que, com um número menor de características agrupadas, reduz o número de parâmetros úteis para as próximas camadas. Há também o *Pooling* L2, nesta aplicação semelhante ao *Max-Pooling*, é realizada a raiz quadrada sobre a soma dos quadrados das ativações da região da imagem. Ambas as técnicas são bastante utilizadas buscando otimização de desempenho, partindo dos dados de validação é possível comparar as aplicações de pooling e escolher a melhor técnica. (DATA, 2022).

Figura 4: Max-pooling com uma janela de pooling de 2×2.



Fonte: ZHANG *et al.* (2022)

As porções sombreadas são o primeiro elemento de saída, bem como os elementos tensores de entrada usados para o cálculo de saída: max(0, 1, 3, 4) = 4.

2.3.1.3 Funções de ativação

A importância das funções de ativação está associada ao que se deve fazer com a informação recebida, ou seja, tomam a decisão da relevância ou não da informação, decidindo se um neurônio deve ou não deve ser ativado. Há diversos tipos de funções de ativação como a função linear (Equação 1), focada em problemas mais simples, função Sigmóide, bastante utilizada principalmente em classificadores (DATA, 2022).

$$f(x)=x \qquad \text{(Equação 1)}$$

A função ReLU (Equação 2) é bastante utilizada para a criação de redes neurais atualmente e seu funcionamento se dá em vantagem das demais funções devido a maneira de como acontece a ativação ou não ativação do neurônio, fazendo com que a rede seja mais adequada e apresente melhor desempenho (DATA, 2022).

$$f(x)=max(0,x) \qquad \text{(Equação 2)}$$

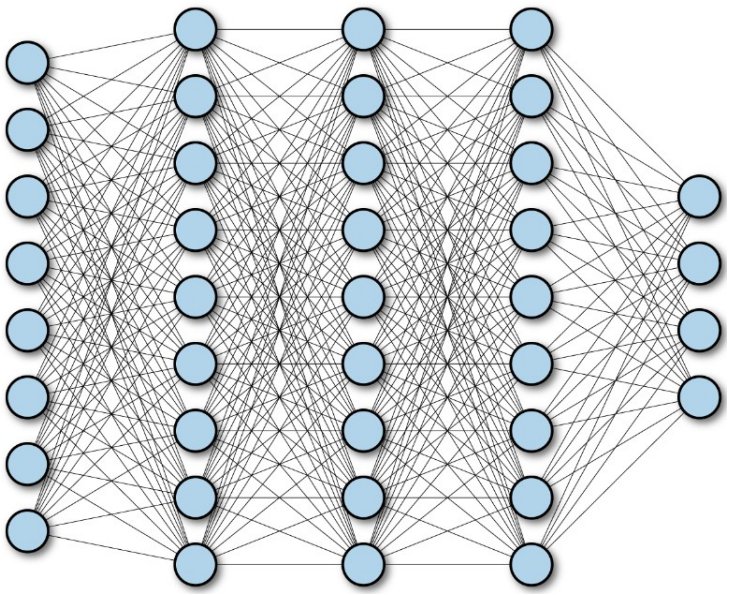
2.3.1.4 *Batch normalization*

A normalização de lote ou *batch normalization* é utilizada para padronizar as entradas para cada camada para cada mini-lotes de exemplos, com isso pretende-se estabilizar a aprendizagem durante o processo a fim de reduzir em grandes quantidades os números de épocas de treinamento úteis. Essa técnica é bastante utilizada para treinamento de redes neurais muito profundas ao acelerar o treinamento e em alguns casos reduz pela metade o número de épocas e ainda fornece alguma regularização reduzindo assim o erro generalizado (BROWNLEE, 2019).

2.3.1.5 *Fully Connected Networks*

*Fully Connected Network*s, ou redes totalmente conectadas, são definidas como uma rede onde se tem várias camadas e, neurônios em uma camada totalmente conectada tem conexões completas com todas as ativações na camada anterior (VOULODIMOS *et al.*, 2018). Uma grande rede em que se tem diversas outras redes e estas estão se conectando, denomina-se de rede profunda ou *deep network*. Os neurônios também são chamados de ‘nós’ e os termos redes densas, redes totalmente conectadas para *Fully Connected Network*s também podem ser encontrados em diversas literaturas. (RAMSUNDAR; ZADER, 2022).

Figura 5: Rede multicamada totalmente conectada



Fonte: O’REILLY (2022)

As redes neurais são um conjunto de funções não lineares dependentes. Cada função individual consiste em um neurônio (ou um perceptron). Em camadas totalmente conectadas, o neurônio aplica uma transformação linear ao vetor de entrada por meio de uma matriz de pesos *W*. Uma transformação não linear é então aplicada ao produto através de uma função de ativação não linear *f*:

$$y = f(Wx + b) \qquad \text{(Equação 3)}$$

onde,
W: é a matriz de pesos;
x: o vetor de entrada (ou de saída da camada anterior);
b: viés (bias)

Toma-se o produto escalar entre a matriz de pesos *W* e o vetor de entrada *x*. O termo de viés (*b*) pode ser adicionado dentro da função não linear.

2.3.1.6 Regularização

A regularização L2 é uma técnica para análises estatísticas bastante utilizada em aprendizagem de máquina também é conhecida como *Ridge regression*. Em um aprendizado de máquina a importância da regularização se dá ao se trabalhar principalmente com muitas variáveis para a determinação dos melhores parâmetros, deixando assim menos favorável ao *overfitting* (CARDERELLI, 2021).

A regularização baseada em penalidades é a abordagem mais comum para reduzir o *overfitting* (AGGARWAL, 2018). Considerando uma operação de regressão de grau *d*, a predição $\hat{y}$ para um dado valor de *x* é dada pela Equação 4.

$$\hat{y} = \sum_{i=0}^{d} w_i x_i \qquad \text{(Equação 4)}$$

onde,

w_i : pesos;

x_i : um atributo de um exemplo de entrada.

A função de perda quadrática para um conjunto de exemplos de treinamento (x,y) do conjunto de dados *D* pode ser definida como na Equação 5.

$$L = \sum_{(x,y)\in D} (y - \hat{y})^2 \qquad \text{(Equação 5)}$$

Aumentando o valor de *d* tende a aumentar o *overfitting*. Uma possível solução é reduzir o valor de *d* para fazer o modelo se ajustar melhor a qualquer dado, mas perde-se na expressividade de reconhecer padrões complexos. Para reter a expressividade, sem causar *overfitting*, ao invés de reduzir o número de parâmetros (*w*), pode-se aplicar uma penalidade suave sobre o uso de parâmetros. A escolha de penalidade mais comum é a regularização L2. Nesse caso, a penalidade adicional é definida pela soma dos quadrados dos valores dos parâmetros, e assim, para o parâmetro de regularização $\lambda > 0$, a função de custo ou perda fica conforme a Equação 6.

$$L = \sum_{(x,y)\in D} (y - \hat{y})^2 + \lambda \cdot \sum_{i=0}^{d} w_i^2 \qquad \text{(Equação 6)}$$

Aumentando ou diminuindo o valor de λ altera-se a suavidade da penalidade. Em geral, observou-se experimentalmente que é mais desejável usar modelos complexos (por exemplo, redes neurais maiores) com regularização do que modelos simples sem regularização (AGGARWAL, 2018).

Com isso, *Ridge Regression* (também chamada de regularização de Tikhonov) é uma versão regularizada da regressão linear: um termo de regularização igual a $\lambda \cdot \sum_{i=0}^{d} w_i^2$ é adicionado à função de custo. Isso força o algoritmo de aprendizado a não apenas ajustar os dados, mas também manter os pesos do modelo os menores possíveis. O termo de regularização só deve ser adicionado à função de custo durante o treinamento. O hiperparâmetro λ controla o quanto se deseja regularizar o modelo. Se $\lambda = 0$, então a *Ridge Regression* é apenas regressão linear. Se λ for muito grande, então todos os pesos terminam muito próximos de zero e o resultado é uma linha reta passando pela média dos dados (GÉRON, 2019). A função de custo ou perda também pode ser escrita em função do MSE, como na Equação 7.

$$L = MSE(w) + \lambda \cdot \frac{1}{2} \sum_{i=0}^{d} w_i^2 \qquad \text{(Equação 7)}$$

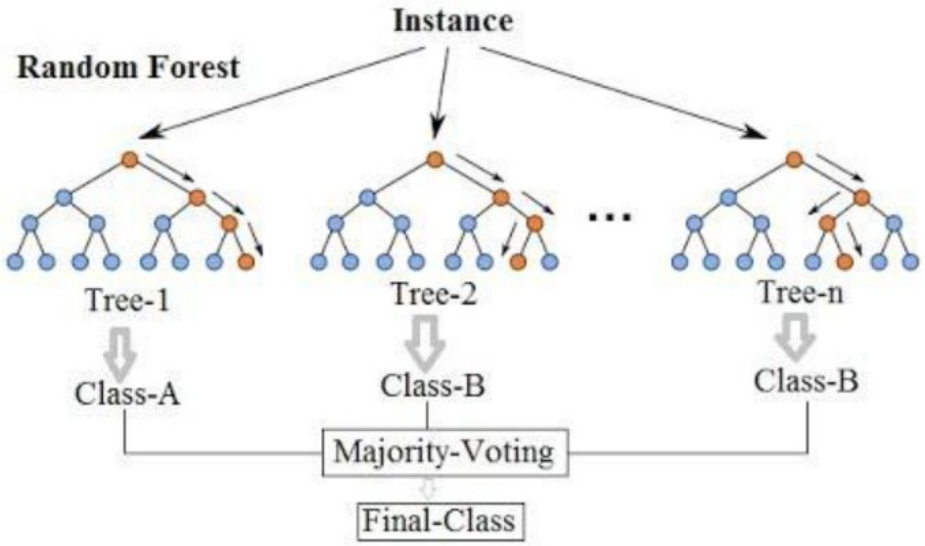
2.3.2 *Random Forest (RF)*

No modelo de *Random Forest* (RF) (BREIMAN, 2001), várias árvores de decisão são construídas baseadas em subconjunto de dados e uma agregação de predições é usada como predição final. O modelo de árvore de decisão é um modelo de árvore em que cada ramo de nó representa uma escolha entre várias alternativas e cada nó representa uma decisão (KANG et. al, 2018). O RF é um dos modelos que são normalmente considerados como de aprendizagem de máquina, assim como árvores de regressão (CART), regressão vetorial de suporte (SVR) entre vários outros (SHANG *et al*., 2019).

Random Forest é então definido como um conjunto de árvores de decisão, onde o ‘nó’ final/folha resultante será a classe majoritária para resolver problemas de classificação ou a resultante será uma média para problemas de regressão. No modelo de RF o número de árvores de classificação irá crescer e para cada saída de cada árvore é escolhida uma classe por ela votada. Há uma ordem de etapas que é preciso seguir para que uma árvore seja construída. A primeira etapa acontece quando uma amostra aleatória de linhas de dados de treinamento é obtida para cada árvore. Na segunda etapa com a amostra colhida, será obtido um subconjunto de características para divisão em cada árvore. Por fim, na terceira etapa cada árvore cresce na maior extensão especificada para então chegar a um voto para a classe (MONTANTES, 2020).

Na Figura 6 abaixo podemos ver uma representação de uma Random Forest simplificada.

Figura 6: Random Forest simplificada



Fonte:MONTANTES (2020)

Pode-se também pensar intuitivamente no funcionamento do algoritmo Random Forest em duas grandes etapas: uma é a criação da Random Forest, a outra é fazer uma previsão a partir do classificador de Random Forest criado no primeiro estágio (POLAMURI, 2017).

Na primeira etapa, constrói-se a Random Forest da seguinte forma:

a) Selecionar aleatoriamente as *k* características do total de *m* características onde *k* << *m;*

b) Entre as *k* características, calcular o nó *d* usando o melhor ponto de divisão;

c) Dividir o nó *d* em nós filhos usando a melhor divisão;

d) Repetir os passos de (a) a (c) até que o número *l* de nós seja alcançado;

e) Construir Random Forest repetindo os passos (a) a (d) por *n* vezes para criar *n* árvores de decisão.

Na próxima etapa, com a Random Forest criada, faz-se a previsão da seguinte forma:

a) Obter as características de teste e usar as regras de cada árvore de decisão

criada aleatoriamente (estimador) para prever o resultado, e armazenar o resultado previsto (alvo);

b) Calcular os votos para cada alvo previsto;

c) Considerar o alvo previsto com alta votação como a previsão final do algoritmo Random Forest.

2.3.3 *Extreme Gradient Boosting* (XGBoost)

XGboost é uma técnica bastante eficiente e é implementada para classificação e também para regressão (CHEN; GUESTRIN, 2016). Essa técnica baseada em árvores de decisão tem sido amplamente utilizada devido ao seu bom desempenho aplicado ao aprendizado de máquina (WANG *et al*., 2018).

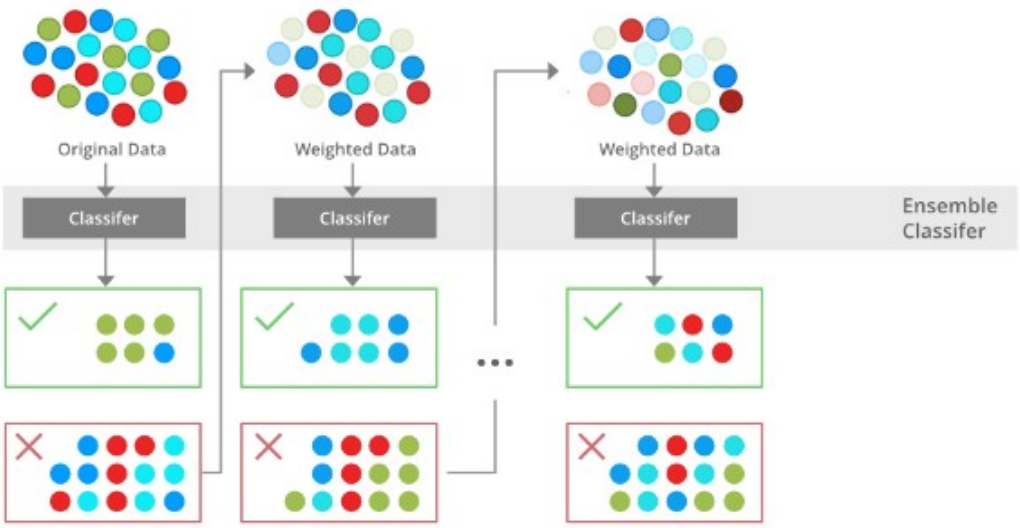
O modelo utiliza uma estrutura de *gradient boosting* baseado em árvores de decisão, que, quando aplicados em dados tabulares, apresentam melhores desempenho. O XGBoost combina várias técnicas de otimização para a produção de bons resultados, fazendo o uso de menos recursos de computação (GOMES, 2019).

Uma árvore de decisão se assemelha a um fluxograma em que cada nó interno corresponde a um teste em um atributo e cada ramificação representa uma saída do teste e cada nó terminal (nó folha) possui um rótulo de classe. A árvore é ensinada após a divisão do conjunto de origem em conjuntos menores baseados em um teste de valor do atributo, este processo se repete em cada subconjunto advindo de modo recursivo denominado de particionamento recursivo. Quando o subconjunto de um nó apresenta o mesmo valor da variável de destino ou ainda quando a divisão não é mais útil às previsões a reclusão é finalizada (SAXENA, 2022).

Para a modelagem do conjunto de dados, é usado a técnica de *boosting*, que busca criar um forte classificador partindo de classificadores mais fracos, e assim o modelo é feito a partir de dados de treinamento. Na sequência, é criado um segundo modelo que tem como propósito corrigir os erros apresentados no primeiro modelo, nesta etapa os modelos são acrescentados até que o número máximo de modelos

sejam adicionados ou até que o conjunto de dados do treinamento seja corretamente previsto (SAXENA, 2022).

Figura 7: Procedimento de Boosting

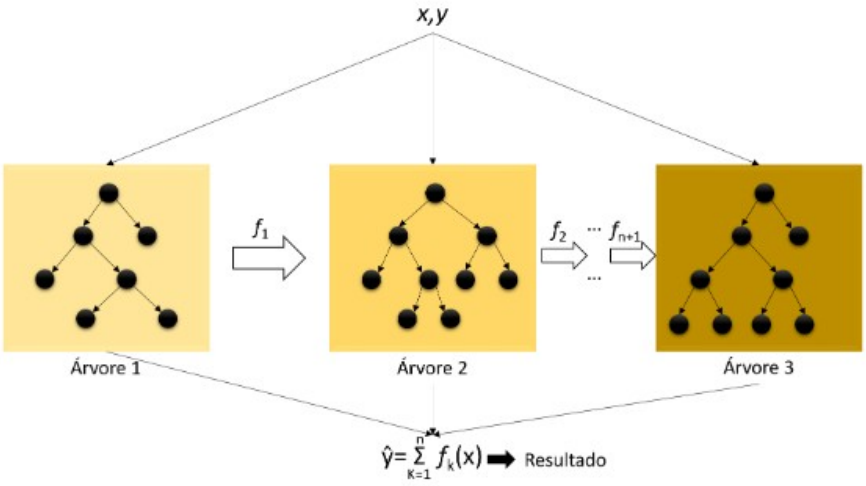


Fonte: SAXENA (2022)

O algoritmo *gradiente boosting* ao sofrer aumento do gradiente, cada preditor irá fazer uma correção do erro de seu predecessor, o que difere do *Adaboost* sendo que os pesos das instâncias de treinamento não são ajustados e cada preditor usa como rótulo os erros residuais, e assim cada preditor é treinado (SAXENA, 2022).

Em suma o XGBoost em suma seria uma implementação de árvores de decisão por *gradiente boosting*, onde as árvores de decisão são criadas de forma sequencial, a atribuição dos pesos é feita para todas as variáveis independentes inseridas na árvore de decisão que irá prever os resultados. Com o aumento do peso das variáveis previstas erroneamente pela árvore, as variáveis serão alimentadas por uma segunda árvore de decisão, dessa forma os preditores ou classificadores individuais irão se agrupar para apresentar um bom modelo, preciso e forte (SAXENA, 2022).

Figura 8: Modelo generalizado de XGBoost



Fonte: JUNIOR (2022)

2.3.4 *Mean Absolute Error* (MAE) e *Mean Squared Error* (MSE)

O *Mean Absolute Error (*MAE*) ou* Erro Médio Absoluto é uma métrica estatística que calcula o módulo da diferença entre o valor estimado e o valor real, podendo ser observado pela Equação 8.

$$MAE = \frac{|(y_r - y_e)|}{n} \qquad \text{(Equação 8)}$$

Em que

y_r é o valor real

y_e é o valor estimado

n é o número de observações ou colunas

O *Mean Squared Error* (MSE) é também um tipo de métrica estatística sendo esta definida como, a média dos quadrados das diferenças dos valores do modelo predito e dos dados reais elevados ao quadrado. Essas métricas são bastante utilizadas para analisar um modelo de regressão (ACHARYA, 2021). O MSE pode ser definido de acordo com a Equação 9.

$$MSE = \frac{\sqrt{(y_r - y_e)^2}}{n} \qquad \text{(Equação 9)}$$

Onde

y_r é o valor real

yp é o valor predito

n é o número de observações ou colunas

2.3.5 *Root Mean Squared Propagation* (RMSprop)

Propagação da raiz da média quadrática ou (RMSprop) é um tipo de algoritmo de aprendizagem semelhante ao algoritmo AdaGrad, que utiliza a soma dos gradientes ao quadrados para estimar A_i, porém usa-se a média exponencial. Como a média exponencial é usada para a normalização ao invés da agregação de valores, o processo não é retartado por um fator de escala A_i que é constantemente crescente.

A chave para isso é o fator de decaimento ρ, que é aplicado ao fator de escala e ao gradiente ao quadrado para o cálculo da média, fazendo que somente os gradientes recentes sejam acumulados (AGGARWAL, 2018). As equações que fazem a atualização do fator de escala e do peso estão nas Equações 10 e 11.

$$A_i \Leftarrow \rho A_i + (1-\rho)\left(\frac{\partial L}{\partial w_i}\right)^2 \qquad \text{(Equação 10)}$$

$$w_i \Leftarrow w_i - \frac{\alpha}{\sqrt{Ai}}\left(\frac{\partial L}{\partial w_i}\right) \qquad \text{(Equação 11)}$$

onde:

A_i representa o fator de escala

ρ representa o fator de decaimento, onde $\rho \in (0, 1)$

w_i representam os pesos

α é a taxa de aprendizado

$\partial L/\partial w_i$ é o gradiente

2.3.6 Seleção de características

Nesta subseção, são mostradas duas técnicas de seleção de características, a estatística F-test univariada (F-Value) e a informação mútua.

2.3.6.1 F-Value

Para se obter o F-Value, calcula-se o coeficiente de correlação de Pearson entre cada característica *X* e o alvo *Y* (PEDREGOSA *et al.*, 2011), conforme Equação 12 (WEISSTEIN, 2022).

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \qquad \text{(Equação 12)}$$

onde,

$\rho_{X,Y}$ é o coeficiente de correlação
μ_X é a média dos valores de *X*
μ_Y é a média dos valores de *Y*
σ_X é o desvio padrão de *X*
σ_Y é o desvio padrão de *Y.*

O $\rho_{X,Y}$ é convertido para um F-score e para um p-value.

As características com *k* maiores F-scores são selecionadas.

2.3.6.2 Informação mútua (*Mutual Information*)

Pode-se definir informação mútua entre uma característica e um alvo, em função da entropia (ZHANG *et al.*, 2022), como na Equação 13.

$$I(X,Y) = H(X,Y) - H(Y \vee X) - H(X \vee Y) \qquad \text{(Equação 13)}$$

onde,

$I(X,Y)$ é a informação mútua entre a característica *X* e o alvo *Y*
$H(X,Y)$ é a entropia conjunta entre a característica *X* e o alvo *Y*, a informação contida em *X* e *Y* juntas.
$H(Y \vee X)$ é a entropia condicional, a informação de *Y* dado *X*
$H(X \vee Y)$ é a entropia condicional, a informação de *X* dado *Y*

O que se quer é subtrair da informação contida em *X* e *Y* juntas, as informações contidas em X, mas não em Y, e as informações contidas em Y, mas não em X.

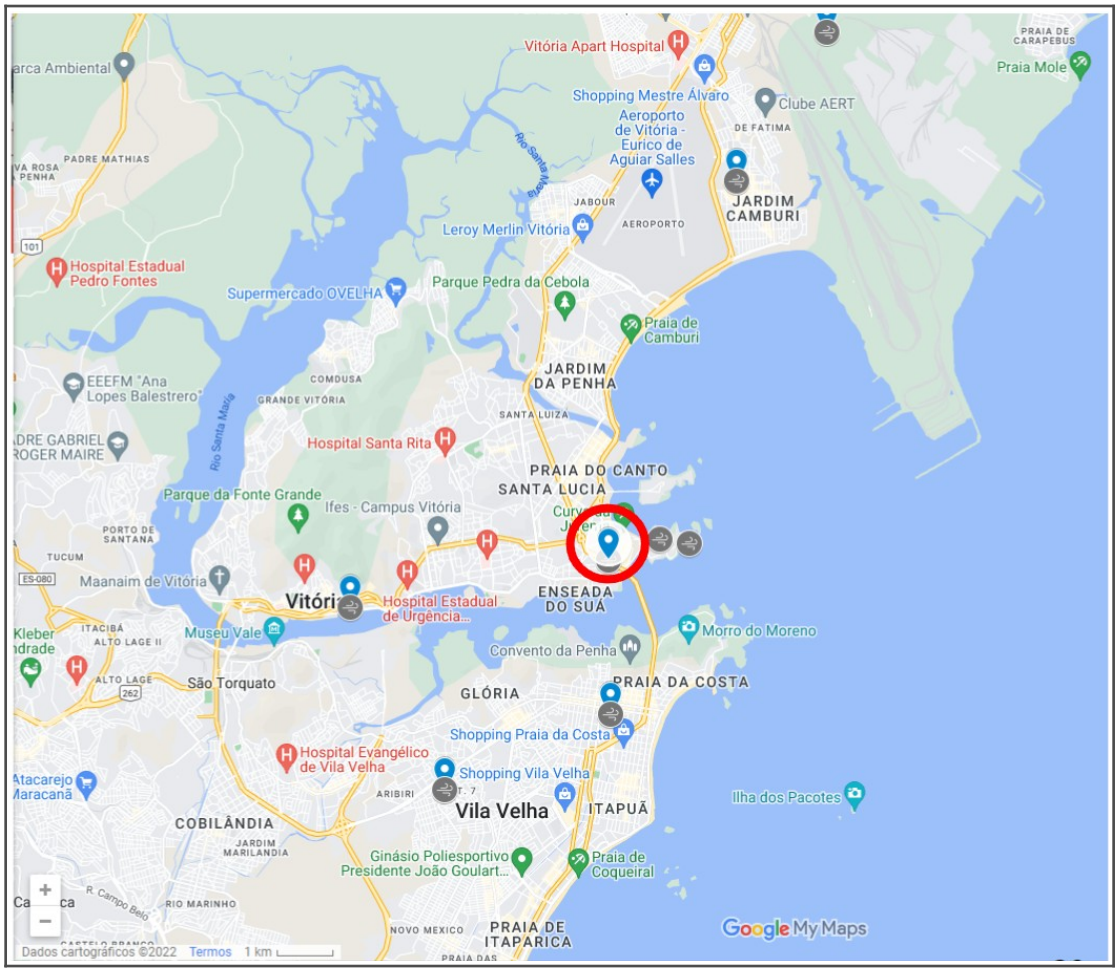
3. MATERIAIS E MÉTODOS

Nesta seção veremos o processo desde a obtenção de dados, onde e como foram executados os experimentos para a geração do modelo preditivo.

3.1 COLETA DE DADOS

A região de estudo é o município de Vitória que possui área de aproximadamente 97 km² com uma população estimada de 369.534 pessoas de acordo com o IBGE, um mapa de localização da estação trabalhada pode ser visto pela Figura 9. Os dados coletados sobre os poluentes atmosféricos disponibilizados pelo IEMA, que correspondem ao período de janeiro de 2010 até novembro de 2021, apresentavam muitas falhas e "buracos" sem a informação de valor do poluente. Devido a essas falhas, alguns sensores foram descartados, para que não houvesse uma predição instável e muito distante dos valores reais. Esses dados são registros horários, então precisou-se agrupar os valores para que se apresentassem em frequência por dia, agregando os valores fornecidos de horas para dia, a partir de uma média diária. Utilizou-se o Pandas, que é uma biblioteca desenvolvida para ser usada na linguagem de programação Python em que esta tem a finalidade de realizar a manipulação e análise de dados.

Figura 9: Mapa de localização da estação EMQAr - RGV4



Fonte: IEMA (2022)

Foram coletados registros de frota de veículos de 2010 até 2021. Esses dados foram extraídos do Instituto Brasileiro de Geografia e Estatística (IBGE), que inclui várias categorias como carros, ônibus, caminhões, utilitários entre outros. Os dados disponibilizados são mensais, logo precisou-se uma interpolação para obtenção de dados diários, para que assim se pudesse manipular os atributos de quantidade de veículos relacionando-os com as variáveis de poluentes atmosféricos para compor os exemplos de treinamento.

3.1.1 Tratamento de Arquivos

Após a manipulação dos dados de poluentes e frota de veículos (utilizando

regressão para agrupar as vairáveis de poluentes atmosféricos que se encontravam por hora e as variáveis de veículos que se apresentavam mensalmente), obteve-se os dados de maneira desejada se apresentando por dia e assim pudesse trabalhar e prosseguir para a próxima etapa que é a de seleção de características, onde foi possível selecionar as variáveis mais relevantes para o modelo preditivo. Após obtido os dados de entrada mais relevantes, por meio da seleção de características, todo o conjunto de dados foi exportado para texto delimitado por vírgula com a extensão *Comma Separated Values* (CSV).

3.1.2 Seleção de Características

No processo de seleção de características, devido a falta de dados de alguns dias e até meses das variáveis de poluentes atmosféricos, precisou-se excluir alguns desses poluentes por possuírem grandes buracos no conjunto de dados. Foi-se necessário descartar os seguintes poluentes: Partículas Inaláveis 2,5 μ (MP 2,5), Partículas Totais em Suspensão (PTS), Monóxido de Nitrogênio (NO), Óxido de Nitrogênio (NOx), Metano (CH_4), Hidrocarbonetos Não Metano (HCNM), Hidrocarbonetos Totais (HCT), e o parâmetro de meteorologia, Desvio Padrão da Direção do Vento. Após a retirada dessas variáveis, trabalhou-se apenas com os sensores de Partículas Inaláveis grossas (MP10), Dióxido de Enxofre (SO_2), Dióxido de Nitrogênio (NO_2), Monóxido de Carbono (CO), Ozônio (O_3) e por fim os sensores meteorológicos de Direção escalar do Vento e Velocidade Escalar do Vento.

Após a etapa de seleção de característica, é feita a normalização dos dados das características, ou seja, as 12 variáveis selecionadas anteriormente, para serem utilizadas no modelo de rede neural.

3.2 DESENVOLVIMENTO

O código foi implementado na linguagem Python 3.9, utilizando-se o ambiente de desenvolvimento Google Colab ou Colaboratory (ferramenta que permite realizar e executar códigos em Python no navegador sem a necessidade de uma configuração e com acesso grátis à GPUs.), e estruturado no formato Jupyter Notebook (ambiente de desenvolvimento de interface interativa em que é possível fazer uma organização da rotina e execução do código criado).

O desenvolvimento do código de modo geral se divide em cinco partes: a primeira parte contém o processamento do conjunto de dados, depois é criado um modelo de rede neural profunda, a rede neural é treinada e depois testada e por fim o modelo de rede neural obtido é comparado com dois outros algoritmos (XGBoost e RF).

3.2.1 Processamento do conjunto de dados

Foram importadas várias bibliotecas como *Pandas*, *Numpy*, *Matplot* entre outras. Após as importações feitas, partimos para o processamento do conjunto de dados, nesta etapa os dados são preparados para alimentar os modelos. Primeiramente é feito o carregamento dos dados de treino e de teste no *DataFrame* da biblioteca Pandas, e em seguida os dados de treino e teste são combinados e processados juntamente. Como já vimos na seção de Seleção de Características, as características de entrada foram pré-selecionados e esses dados foram exportados para a extensão CSV e assim carregados.

Foram utilizados os dados de treinamento dentro do intervalo de janeiro de 2010 e dezembro de 2018, e para teste o período de janeiro de 2019 até novembro de 2021. Após a combinação entre os dados de treino e teste, foi obtida a tabela de estatísticas descritivas de todos as variáveis de entrada.

Na sequência, uma função foi criada a fim de se conhecer as colunas não nulas e eliminando as colunas com valores vazios, o que resultará no número doze (12) de colunas que representam além das variáveis atmosféricas, categorias de veículos com os quais deseja-se trabalhar e em seguida foi obtido o histograma de frequência para cada sensor.

Após obtido os histogramas das variáveis, fez-se a correlação entre esses sensores resultando na Matriz de Correlação ou Mapa de Calor de Correlação.

Na seleção de características, para verificar quais foram as mais importantes, aplicou-se o método F-value e o método informação mútua, selecionando as características em cada método.

A etapa de normalização das características numéricas é feita pela média e desvio padrão em que as características devem estar em torno da média. Para que a execução da rede neural apresente uma melhor desempenho as características devem ser normalizadas, pois assim, os pesos não irão apresentar grandes diferenças e consequentemente não correm o risco de extrapolar.

3.3 EXPERIMENTOS

Nesta etapa do trabalho, fez-se alguns testes, para a obtenção do modelo preditivo usando o Google Colab, que permite a execução do código em Python através de um navegador. Repetiu-se a execução considerando inicialmente as variáveis de: automóveis, caminhões, motos, ônibus, utilitários, outros, MP2,5, MP10, PTS, dióxido de enxofre, monóxido de nitrogênio, dióxido de nitrogênio, óxido nitrogênio, monóxido de carbono, ozônio, metano, hidrocarboneto não metano, hidrocarboneto total, direção escalar do vento, desvio padrão de direção do vento e velocidade escalar do vento. Devido a grandes falhas existentes como já discutido em Coleta de Dados, seção 3.1, o número de variáveis utilizados reduziu-se para 12 (doze) restando somente as variáveis: Automóveis, Caminhões, Motos, Ônibus, Utilitários, Outros (esses advindos da tabela de frota de veículos do IBGE) e as variáveis de: Dióxido de Enxofre (SO_2), Dióxido de Nitrogênio (NO_2), Monóxido de Carbono (CO), Ozônio (O_3), os sensores meteorológicos de Direção escalar do Vento e Velocidade Escalar do Vento.

Após a seleção das doze variáveis acima, incluindo dados de frota de veículos, poluentes e dados meteorológicos, realizou-se cada etapa da execução do código e observou-se como o MAE e MSE se apresentavam. À medida que acontecia cada execução obtendo muitas vezes uma performance não satisfatória (analisando os valores dos MAE e MSE), testou-se a configuração da arquitetura usando camadas convolucionais e camadas totalmente conectadas, experimentando com camadas com dropout e max-pooling, variando o número de ‘nós’ e filtros, entre as camadas no modelo da rede neural em cada execução, até enfim chegar no modelo satisfatório pretendido. O quadro localizado no apêndice A, apresenta todas as configurações testadas para o modelo de redes neurais

profundas (DNN) e também para o modelo de rede multilayer perceptron (MLP), apesar de não apresentar resultados melhores que a rede DNN.

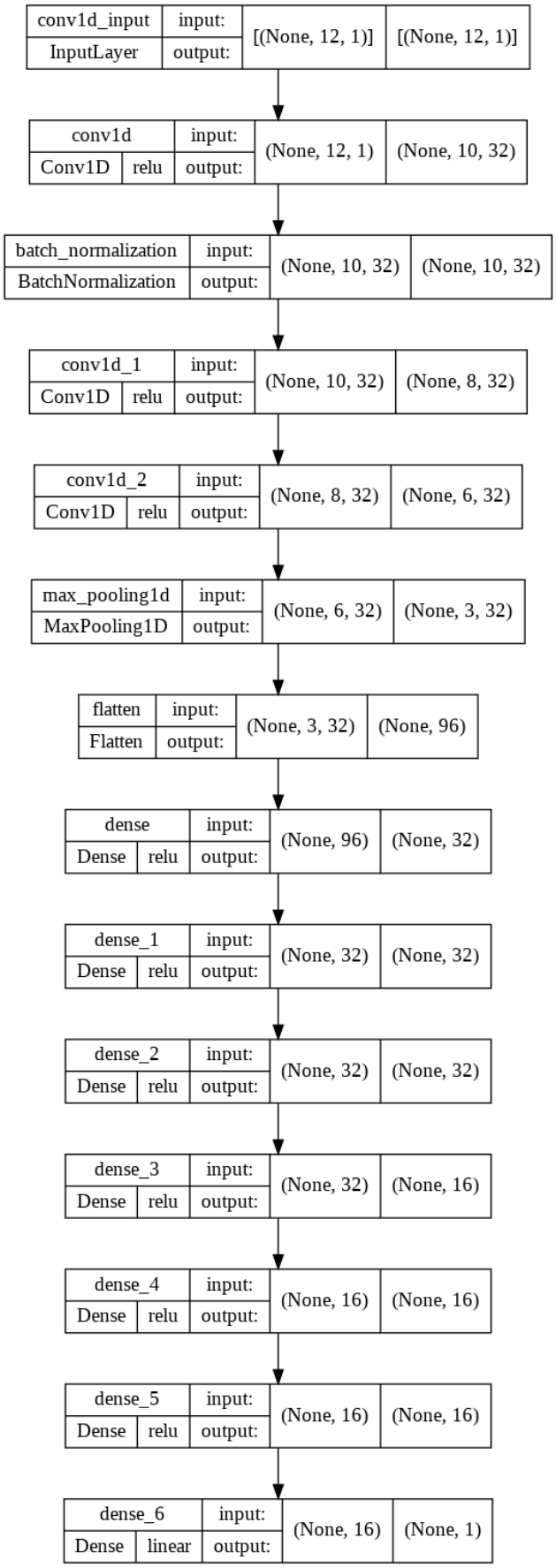
3.3.1 Criando a rede neural profunda

Para criar uma rede neural profunda é preciso definir antes um modelo sequencial e adicionar algumas camadas de neurônios. Foi utilizada a rede neural convolucional, em que a rede inicia com uma camada de entrada, em seguida, adiciona-se algumas camadas convolucionais e essa rede é terminada utilizando camadas densas, sendo que sua última camada densa possui a função de ativação linear com apenas uma saída.

Na compilação da rede, o modelo será configurado para calcular a perda utilizando o MAE e o otimizador utilizado foi o RMSprop pois este apresenta melhor desempenho para modelos de regressão. É possível vermos o modelo sequencial, apresentado na Figura 10. A função de retorno de chamada de ponto de verificação, irá salvar os pesos da DNN com o melhor resultado para ser usado no teste.

Usou-se a função de ativação ReLu para as camadas ocultas e um inicializador 'normal' como *kernel_intializer*, que irá estabelecer como definir os pesos aleatórios iniciais da camada Keras (biblioteca escolhida para desenvolver a rede neural profunda). A configuração que apresentou o melhor resultado na rede DNN, foi a configuração em que a primeira camada é convolucional com 32 filtros, a segunda camada é a camada Batch Normalization, e em seguida mais 2 camadas convolucionais de 32 filtros, seguida de uma camada de max-pooling de tamanho 2. Na sequência do modelo, são utilizadas 6 camadas densas, essas possuindo 3 camadas de 32 neurônios e 3 camadas de 16 neurônios, respectivamente. O tamanho do filtro das camadas convolucionais é de 3. Ao final, é adicionada uma camada densa utilizando a função de ativação linear onde temos um (1) neurônio de saída que representa o valor do MP10 e todos os 16 neurônios da camada anterior são conectados a esse único neurônio.

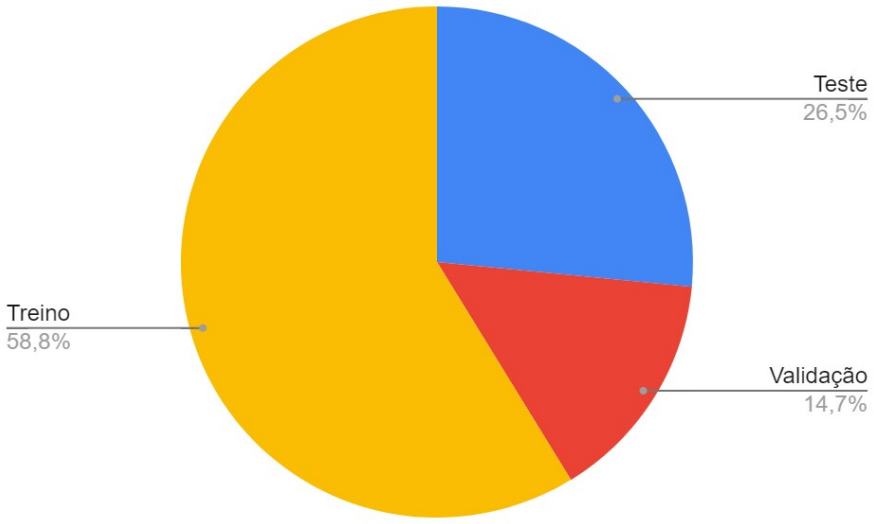
Figura 10: Esquema do modelo sequencial



3.3.2 Treinando a rede neural profunda

No treinamento do modelo, a rede foi treinada e testada. O parâmetro alvo de treino é o MP10 a partir de dados com dimensão de entrada de 12 (devido ao número de varáveis), executando por 800 épocas, onde cada época treina um tamanho de lote de 32 exemplos diferentes. A função de retorno de chamada de ponto de verificação será usada para armazenar para uso futuro os pesos com a perda de validação (val_loss) de melhor resultado (a mais baixa). Partindo de 1940 exemplos foi utilizado para teste os anos de 2019 e 2020 totalizando 26,54% dos exemplos (515 exemplos), apesar de terem dados faltantes de alguns dias desses anos. Dos 73,46% dos exemplos restantes, 20% foi retirado para validação (285 exemplos) e todo o resto foi utilizado para treinamento, um total de 1140 exemplos. Podemos ver a representação da partição dos exemplos na Figura 11.

Figura 11: Partição de exemplos para o treinamento, validação e teste da rede neural profunda



Com funções de plotagem serão produzidos os gráficos de MAE e MSE em relação ao treino e validação para apresentar em qual época o modelo obteve o melhor resultado. Executou-se o treinamento da rede neural profunda 10 vezes, e calculou-se a média do MAE de teste das execuções, para comparar com os modelos de RF e XGBoost.

3.3.4 Testando a rede neural profunda

Em seguida, depois da execução do treinamento, o arquivo de pesos do melhor modelo é carregado, ou seja, o modelo obtido que apresentou a menor perda na validação é escolhido. O modelo é então testado utilizando os 515 exemplos dos dados de teste, são calculados o MAE e o MSE de teste para o MP10.

3.3.5 Comparando o resultado da rede neural profunda com RF e XGBoost

Aqui utilizamos os algoritmos RF e XGBoost para que possamos comparar com os resultados do modelo da rede neural profunda anteriormente obtido. Foram realizadas 10 execuções nos dois modelos e na rede neural profunda, assim, usando a média das 10 execuções da rede neural foi possível comparar com a média das 10 execuções do modelo RF e do modelo XGBoost. Dividiu-se os dados de treino e validação para a execução dos dois algoritmos, obtendo o MAE de teste para o RF e para XGBOOST.

4. RESULTADO E DISCUSSÕES

Nesta etapa do trabalho, apresentaremos e analisaremos os resultados obtidos e discutiremos se o modelo preditivo gerado foi satisfatório.

4.1 PROCESSAMENTO DE CONJUNTO DE DADOS

Antes do treinamento e teste, utilizando os 1940 exemplos, foi obtida a Tabela 1 de medidas estatísticas descritivas para os atributos do conjunto de dados.

Tabela 1: Estatísticas descritivas para os atributos do conjunto de dados

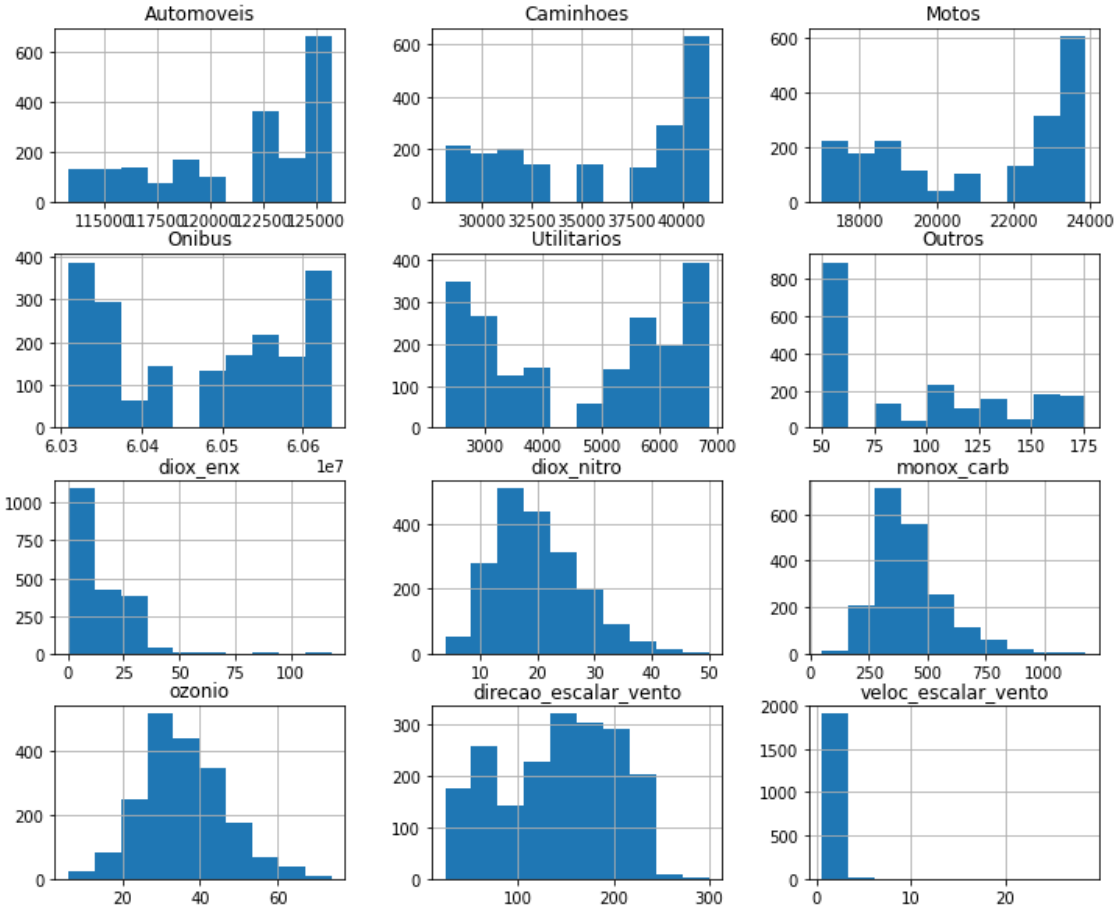
	média	desvio padrão	min	25%	50%	75%	max
Automoveis	121.490	3.949	113.301	118.339	122.899	125.115	125.715
Caminhoes	36.158	4.703	28.223	31.537	38.421	40.509	41.317
Motos	21.142	2.469	17.017	18.710	22.286	23.424	23.913
Onibus	6	1	6	6	6	6	6
Utilitarios	4.652	1.582	2.318	3.059	5.066	6.084	6.854
Outros	94	43	50	51	86	129	176
diox_enx	13,07	10,72	0,15	4,07	9,16	22,89	118,29
diox_nitro	19,94	7,32	3,85	14,43	18,88	24,62	50,00
monox_carb	422,16	144,66	49,52	321,50	393,38	492,28	1.179,51
ozonio	35,61	10,72	6,07	28,36	34,61	42,43	73,80
direcao_escalar_vento	141,50	60,06	23,46	87,95	147,51	191,70	299,76
veloc_escalar_vento	2,13	1,68	0,64	1,56	1,95	2,41	28,52

Observando o coeficiente de variação para cada variável da Tabela 1, foi possível observar que as maiores variações ocorreram nas variáveis de dióxido de enxofre e velocidade escalar do vento, já os menores coeficientes de variação se apresentaram nas variáveis: caminhões, motos e ônibus. A variável automóveis apresentou uma variação muito mais baixa dentre as variáveis classificados como pouco variáveis.

Após obter os dados de estatísticas descritivas para cada atributo, foi gerado

um histograma para cada variável apresentando o quantitativo de cada parâmetro como podemos ver na no conjunto de gráficos apresentados na Figura 12.

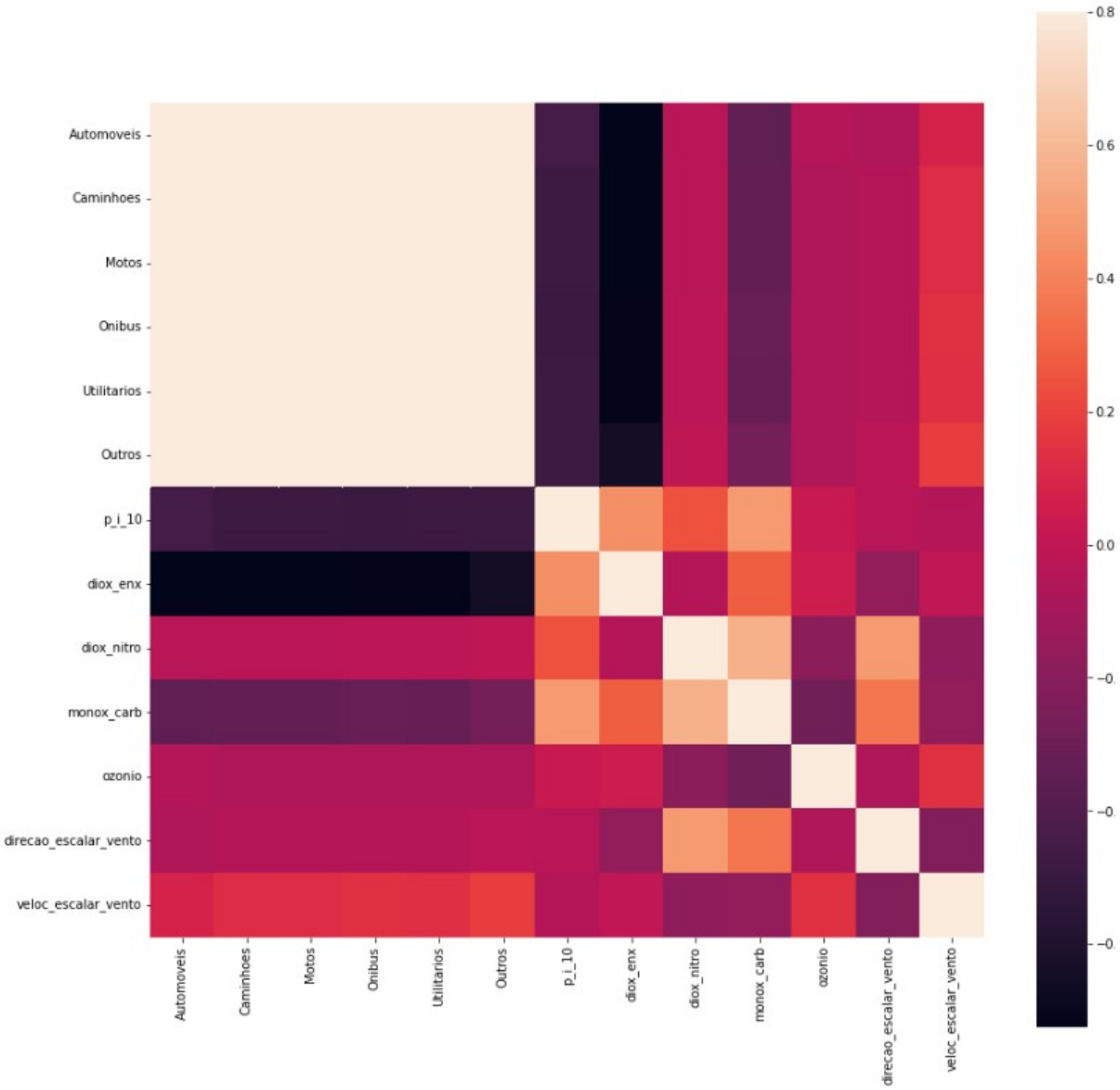
Figura 12: Histogramas do quantitativo das variáveis selecionadas



Pode-se observar que os histogramas das variáveis: dióxido de enxofre, monóxido de carbono e ozônio apresentam comportamento próximo a uma distribuição gaussiana.

Após a obtenção dos histogramas dos atributos, é feita a correlação entre as variáveis, apresentada pela matriz de correlação ou mapa de calor, como pode ser visto na Figura 13.

Figura 13: Matriz de correlação de características



Gerada a matriz de correlação, é possível notar o comportamento apresentado pelos diferentes tipos de veículos, em que esses possuem uma alta correlação entre si. Outro comportamento observado é o do parâmetro MP10 apresentando maiores correlações com as variáveis: monóxido de carbono, dióxido de enxofre e dióxido de nitrogênio.

Depois da etapa da geração da matriz de correlação, precisou-se selecionar as melhores características e para isso foram aplicados os métodos, F-value e a

informação mútua, em que foram selecionadas cinco (5) características em cada método. No método F-value as características selecionadas foram: motos, ônibus, utilitários, outros e monóxido de carbono e para o método informação mútua as melhores características selecionadas foram: caminhões, motos, ônibus, utilitários e outros. Podemos notar que o parâmetro monóxido de carbono selecionado pelo método F-value, possui uma boa correlação com o MP10 na matriz de correlação além de ter seu comportamento gaussiano no histograma. Uma outra observação é que a maioria das características selecionadas tanto no método F-value quanto no método informação mútua são variáveis de veículos e que estes possuem uma alta correlação linear entre si, como consta na matriz de correlação apresentada na Figura 12. Porém, não possuem uma boa correlação linear com o MP10, o que pode indicar que, ainda assim, a relação entre esses atributos e o MP10 seja não linear.

4.2 TREINANDO A REDE NEURAL PROFUNDA

Nesta etapa o treinamento da rede neural profunda que inclui tanto o conjunto de treinamento quanto de validação, gerou-se os gráficos de perdas por épocas do MAE e do MSE como podemos observar nas Figuras 14 e 15.

Figura 14: Perda por época do MAE na 9ª execução

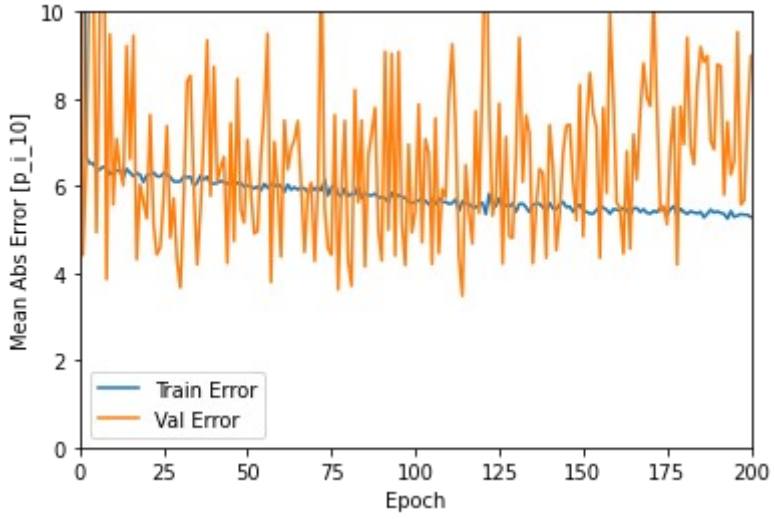
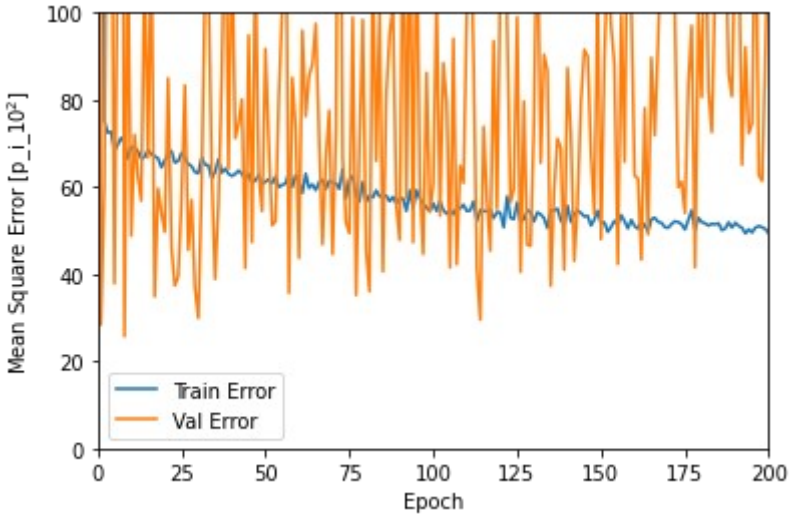


Figura 15: Perda por época do MSE na 9ª execução



Os gráficos acima apresentados foram gerados da nona execução onde apresentou um MAE de 4,09. Pode-se observar grandes amplitudes das oscilações nos gráficos acima, uma explicação seria devido ao número de camadas densas pelas quais os exemplos passam para treinar a rede, além da dificuldade de controlar os pesos da rede neural. Sem regularização a oscilação seria ainda mais caótica (é utilizada a regularização L2 em cada camada).

O gráfico de cor laranja que representa o MAE de validação está sobre o gráfico de cor azul que representa o MAE de treinamento, antes de alcançar a época 200, e isso nos diz que o modelo DNN foi treinado suficientemente e que a menor perda (val_loss) foi obtida. No exemplo dado, a menor perda alcançada foi de val_loss igual a 3,54936 com MAE de teste igual a 4,0953 na época 114.

4.3 TESTANDO A REDE NEURAL PROFUNDA

Foram executadas 10 vezes as melhores arquiteturas que foram a 76°, 78° e 79°. Dentre essas, a de número 79° apresentou melhor desempenho, podemos melhor ver essas execuções na Tabela 2.

Tabela 2: As 10 melhores execuções da arquitetura 79°

# Execução	DNN	RF	XGB
1°	4,06	3,74	3,91
2°	4,04	3,77	3,91
3°	4,02	3,73	3,91
4°	4,01	3,85	3,91
5°	4,36	3,81	3,91
6°	4,27	3,77	3,91
7°	3,80	3,83	3,91
8°	3,94	3,77	3,91
9°	4,09	3,74	3,91
10°	4,12	3,77	3,91
Média	4,07	3,78	3,91
Desv.Pad.	0,16	0,04	0,00

4.4 COMPARANDO O RESULTADO DA REDE NEURAL PROFUNDA COM RF E XGBOOST

Os resultados obtidos gerados nos modelos de rede RF e XGBoost foram usados para comparar com a rede neural profunda. Nas Figuras 16 e 17, podemos ver árvores de decisão desses modelos sendo representadas. Na Figura 15, temos um exemplo de uma das árvores de decisão do modelo RF, mostrando apenas 2 níveis de profundidade. Em cada nó da árvore, ao ramo esquerdo, o fluxo segue a cada “sim” obtido e, ao ramo direito, a cada um “não” obtido após análise da condição especificada pelo nó. Dessa forma, a árvore vai resultando os valores para os exemplos, pelas ramificações até enfim chegar às folhas, que representam a saída com os valores de MP10. Podemos exemplificar pegando o exemplo do dia 26-03-2010 (caminhões = 28223, monox_carb = 1179,506, diox_enx = 28,321), partindo do nó com a condicional do parâmetro caminhões <= 39504,709, segue-se a ramificação para o próximo nó à esquerda, com a condicional parâmetro de monóxido de carbono <= 534,727 e na sequência, segue para o ramo à direita, ao nó com a condicional do parâmetro dióxido de enxofre <= 28,112. O mesmo

exemplo da rede RF pode ser aplicado a rede XGBoost (Figura 16).

Figura 16: Exemplo de uma árvore de decisão do modelo RF, mostrando somente profundidade de 2 níveis

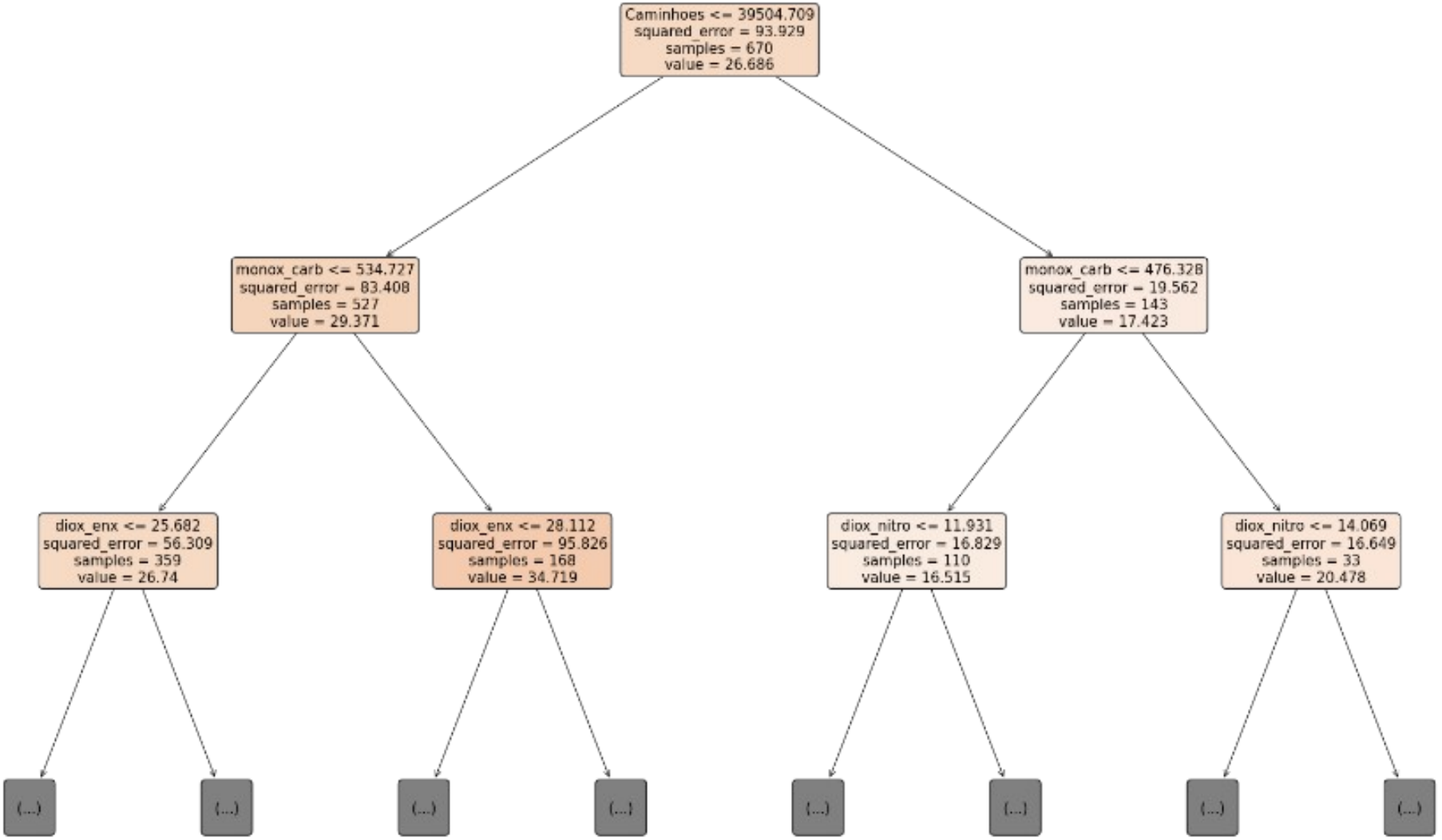
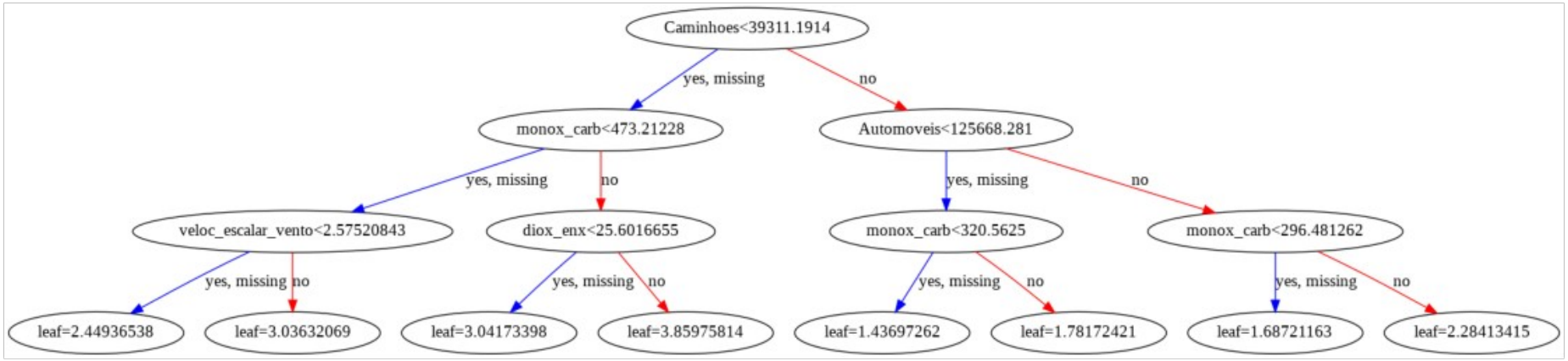


Figura 17: Árvore de decisão para o modelo XGBoost



As Figuras 18 a 23 representam a relação dos valores de MP10 reais e preditos ao longo do mês de abril de 2019 e 2020, para cada modelo. Observando cada gráfico é possível perceber proximidade entre os valores de MP10 reais e preditos. Além disso, nas curvas dos valores preditos, os picos são suavizados em relação aos picos das curvas dos valores reais, e isso demonstra a capacidade de generalização dos modelos que de certo modo ignoram as discrepâncias nos dados originais.

Figura 18: Valores reais e preditos de MP10 usando DNN para o mês de abril do ano de 2019

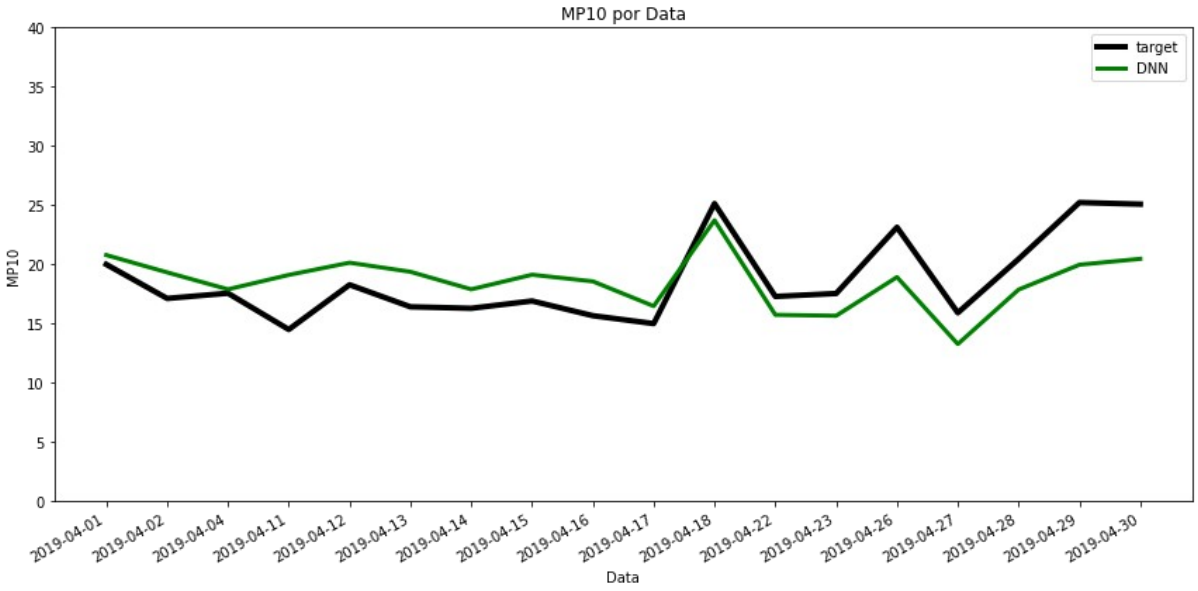


Figura 19: Valores reais e preditos de MP10 usando RF para o mês de abril do ano de 2019

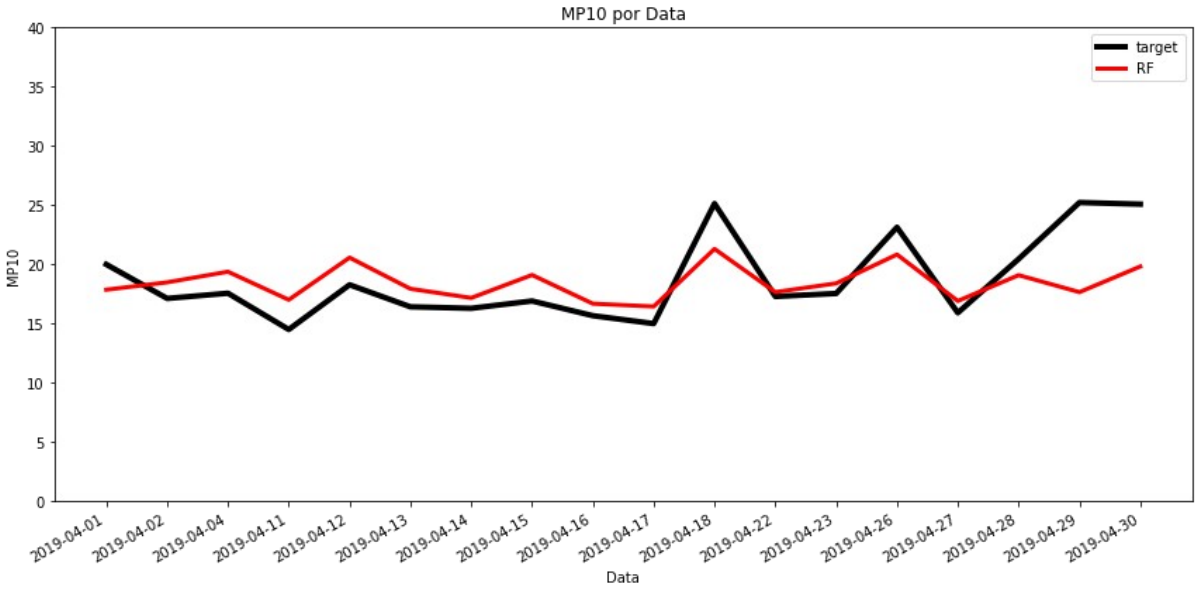


Figura 20: Valores reais e preditos de MP10 usando XGBoost para o mês de abril do ano de 2019

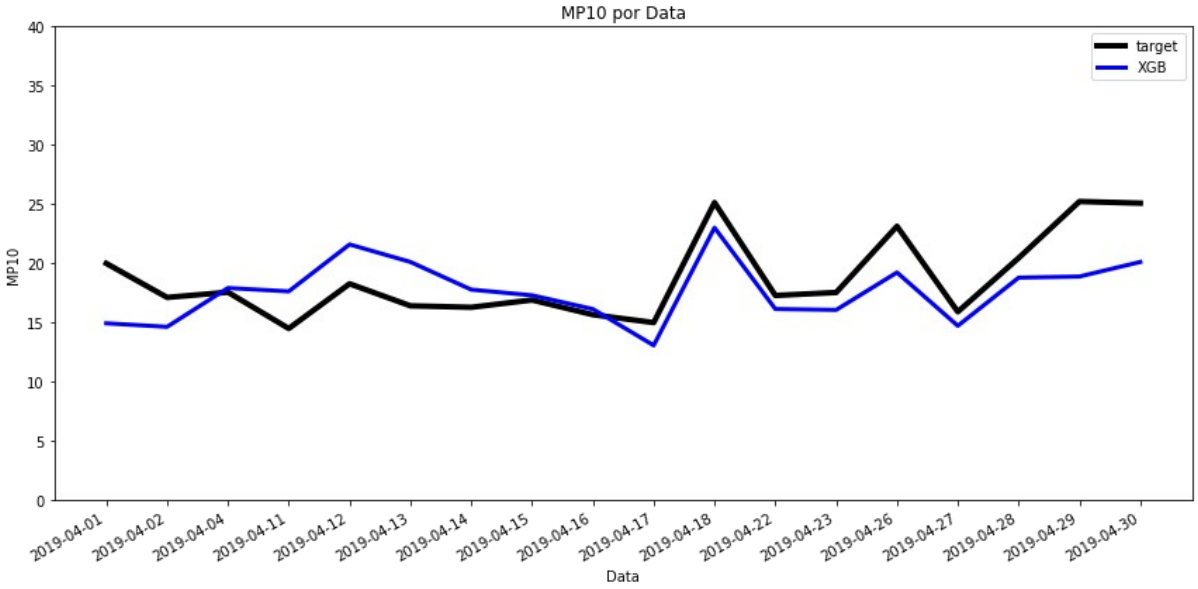


Figura 21: Valores reais e preditos de MP10 usando DNN para o mês de abril do ano de 2020

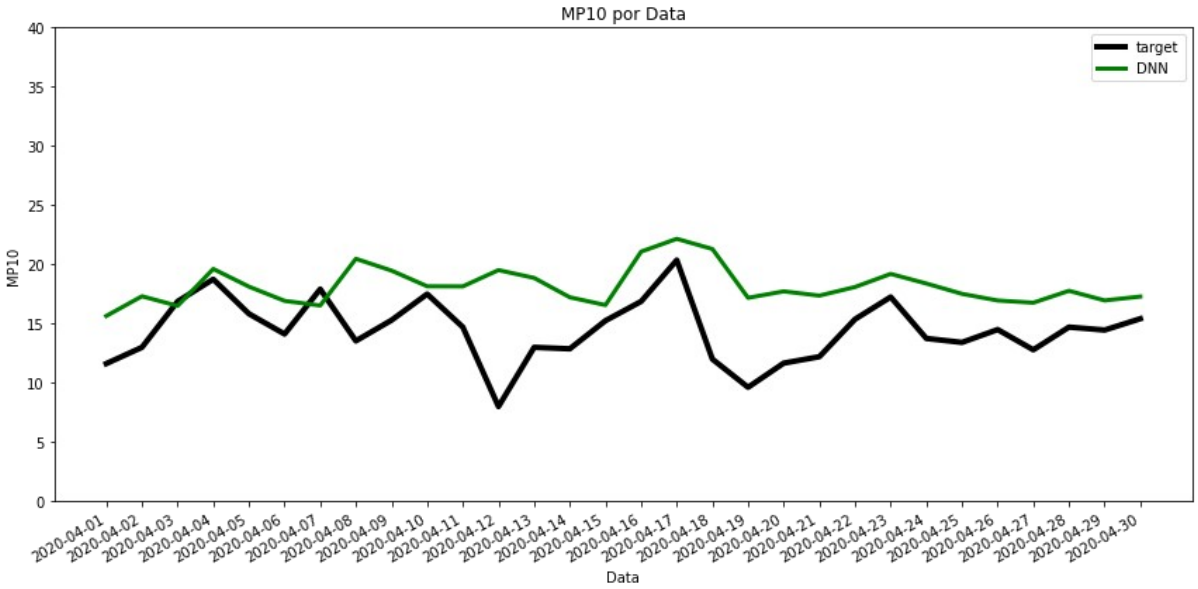


Figura 22: Valores reais e preditos de MP10 usando RF para o mês de abril do ano de 2020

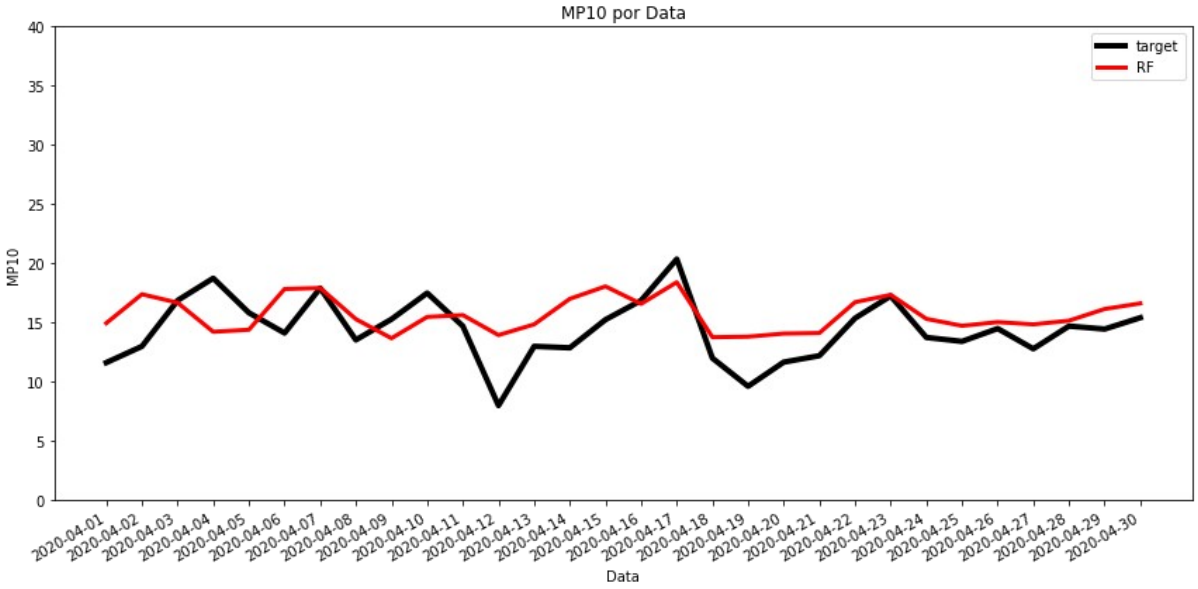
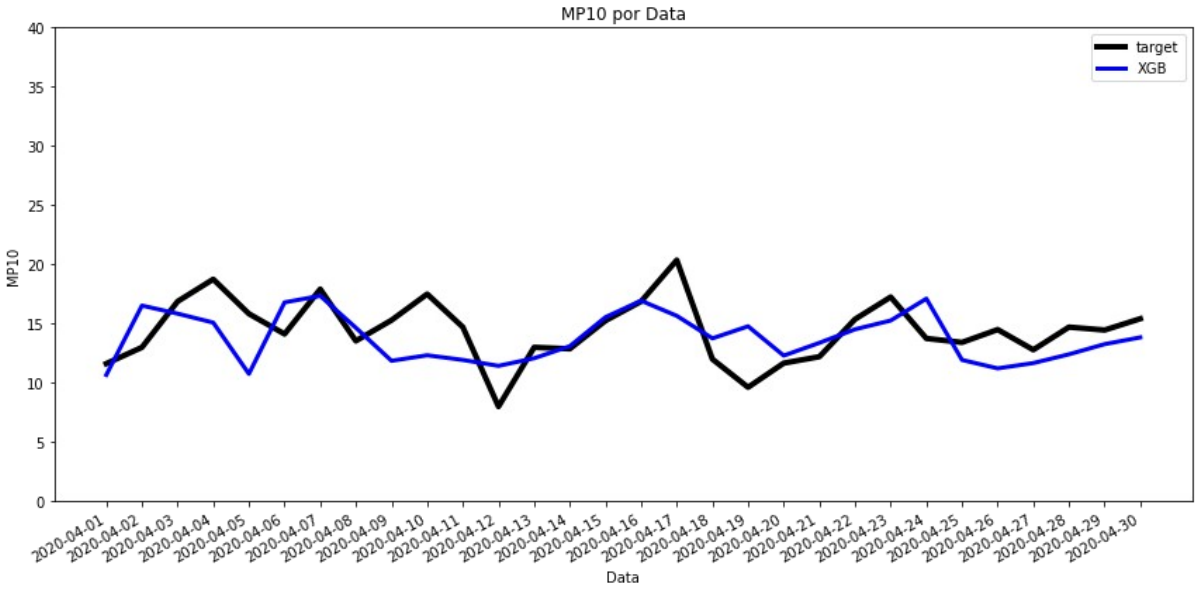


Figura 23: Valores reais e preditos de MP10 usando XGBoost para o mês de abril do ano de 2020



Os resultados obtidos da rede neural profunda não foram tão bons quanto os modelos de RF e XGBoost apesar de apresentarem uma boa aproximação. Em algumas execuções realizadas, obteve-se poucos resultados melhores que o modelo RF, apesar disso, os resultados apresentados foram esperados. Um dos motivos pelo qual o desempenho da DNN foi inferior aos outros modelos de aprendizagem de máquina, RF e XGBoost, é que a quantidade de dados usados para este trabalho foi insuficiente para que a rede neural profunda pudesse aprender e gerar um modelo melhor quando comparado às outras técnicas. Em suma para um conjunto de dados pequeno, o desempenho do RF será melhor se esses mesmos dados forem usados em uma rede neural profunda, pois o RF apresenta melhores resultados para dados tabulares.

5. CONCLUSÃO

O objetivo do presente trabalho foi desenvolver um modelo de predição do quantitativo de material particulado de 10 μ (MP10) baseado nas variáveis de qualidade do ar e dados meteorológicos advindos dos dados da estação de monitoramento da qualidade do ar (EMQAr - RGV 4) e número de veículos, utilizando aprendizagem de máquina. Foram construídas, treinadas e testadas redes neurais profundas de arquitetura convolucional, experimentando-se diversas configurações. Obteve-se um modelo de rede profunda de melhor resultado com 12 camadas (79° do apêndice A), sendo 3 camadas convolucionais, uma camada de *batch normalization*, uma camada de max-pooling, 6 camadas totalmente conectadas, além da camada de saída. Além de redes neurais profundas, foram treinados e testados modelos RF e XGBoost para que posteriormente pudéssemos fazer uma comparação de desempenho entre os modelos gerados.

A rede neural profunda construída alcançou desempenho médio próximo aos dos modelos RF e XGBoost, mas não melhor. Isso pode ser explicado pela pouca quantidade de dados trabalhados para se treinar uma rede neural profunda, que depende de muitos dados de treinamento, e pela melhor adequação a dados tabulares por parte dos modelos de aprendizagem de máquina comparados. O modelo RF alcançou o melhor resultado também pois é um *ensemble* de árvores de decisão, e da mesma forma o XGBoost, portanto o resultado foi como esperado. Além disso, a rede neural profunda possui muitos hiperparâmetros para ajustar, até chegar a uma configuração satisfatória. Apesar da rede profunda não apresentar um modelo de predição do MP10 melhor do que os outros dois modelos de aprendizagem de máquina (salvo em algumas execuções em que obteve um valor de MAE menor que do modelo RF), ainda sim, obtivemos os modelos RF e XGBoost que podem ser usados na tarefa de predição do MP10.

Alinhado com o objetivo do trabalho da criação de uma rede neural profunda a ser treinada e testada, para todo o estudo feito, não se preocupou em julgar se o valor predito do erro (MAE) está baixo, alto ou aceitável para o MP10.

Independentemente, como já discutido, para um bom desempenho em uma rede neural profunda ter uma grande quantidade de dados é primordial, logo, estudos futuros podem ser feitos a fim de investigar se esses valores preditos no modelo são aceitáveis, partindo de que, com mais dados históricos gerados de poluentes, frota veicular e dados meteorológicos, a rede pode ser melhorada com a adição desses novos dados e consequentemente tende a gerar um modelo de melhor performance para o MP10.

Com um bom modelo predito de poluentes, no nosso caso, mais especificamente o poluente atmosférico MP10, é possível auxiliar no controle deste poluente em ambientes urbanos, onde se tem uma grande concentração de veículos e indústrias, e consequentemente, com o controle da qualidade do ar, os problemas referentes aos poluentes atmosféricos que afetam a saúde humana, (principalmente problemas cardiorespiratórios) poderão ser quantificados antecipadamente e assim melhor controlados.

REFERÊNCIAS

ACHARYA, S. What are RMSE and MAE? **Towards Data Science**, 2021. Disponível em: https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383. Acesso em: 19, julho, 2022.

AGGARWAL, C. C. Neural Networks Deep Learning**.** Cham: **Springer International Publishing**, 2018. Disponível em: http://link.springer.com/10.1007/978-3-319-94463-0.

ALVES G. Entendendo Redes Convolucionais (CNNs). **neurônio BR**. Disponível em: https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184. Acessado em 11 de agosto de 2022.

BUDALAKOTI, S.; SRIVASTAVA, A. N.; OTEY, M. E. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. **IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, v. 39, n. 1, p. 101–113, 2009.

BRASIL. Decreto-lei nº8.468, de 8 de setembro de 1976. Aprova o Regulamento da Lei n.° 997, de 31 de maio de 1976, que dispõe sobre a prevenção e o controle da poluição do meio ambiente. **Coletânea de legislação: edição federal**, São Paulo, 1976.

BRASIL. Ministério do Meio Ambiente (MMA). Conselho Nacional do Meio Ambiente (CONAMA). **Resolução CONAMA Nº 003, de 28 de junho de 1990**. Dispõe sobre os padrões de qualidade do ar. Disponível em: http://www.ibama.gov.br/sophia/cnia/legislacao/MMA/RE0003-280690.PDF. Acessado em 12 de junho de 2022.

BRASIL. Ministério do Meio Ambiente (MMA). Conselho Nacional do Meio Ambiente (CONAMA). **Resolução CONAMA Nº 491, de 19 de novembro de 2018**. Dispõe sobre os padrões de qualidade do ar. Disponível em: https://www.in.gov.br/web/guest/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/51058895/do1-2018-11-21-resolucao-n-491-de-19-de-novembro-de-2018-

51058603. Acessado em 12 de junho de 2022.

BREIMAN, L. Random Forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001. Disponível em: http://www.springerlink.com/content/u0p06167n6173512/fulltext.pdf.

BROWNLEE, J. A Gentle Introduction to Batch Normalization for Deep Neural Networks. **Machine Learning Mastery**. 2019. Disponível em: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/. Acessado em 11 de Julho de 2022.

BROWNLEE, J. **Deep Learning for Computer Vision:** Image Classification, Object Detection and Face Recognition in Python. Machine Learning Mastery. 563p. 2019.

CARDERELLI, L. D. O que é Regularização - L1 e L2. **Data Hackers**. 2021. Disponível em: https://medium.com/data-hackers/o-que-%C3%A9-regulariza%C3%A7%C3%A3o-l1-l2-6697ada36a51. Acessado em 11 de julho de 2022.lima

CETESB. Companhia de Tecnologia de Saneamento Ambiental. Disponível em https://cetesb.sp.gov.br/ar/padroes-de-qualidade-do-ar/. Acessado em março de 2022. **CETESB**, São Paulo.2021.

CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. KDD '16, 2016, New York, NY, USA: **Association for Computing Machinery**, 2016. p. 785–794. Disponível em: <https://doi.org/10.1145/2939672.2939785>.

DATA Science Academy. **Deep Learning Book**, 2022. Disponível em: https://www.deeplearningbook.com.br. Acessado em: 11 Agosto. 2022.

DELAVAR, M. R. et al. A novel method for improving air pollution prediction based on machine learning approaches: A case study applied to the capital city of Tehran. **ISPRS International Journal of Geo-Information**, v. 8, n. 2, 2019.

DESNISKO, D., HOFFMAN, M. M. Classification and interaction in random forests. **Proceedings of the National Academy of Sciences**, v. 115, n.8, p.1690–1692. 2018. Disponívem em: https://doi.org/10.1073/pnas.1800256115.

GÉRON, A. **Hands-on Machine Learning whith Scikit-Learing, Keras and Tensorfow**. [S.l: s.n.], 2019.

IBM Cloud Education. What are Convolutional Neural Networks? **IBM Cloud Learn Hub,** 20 de outubro de 2020. Disponível em: https://www.ibm.com/cloud/learn/convolutional-neural-networks. Acesso em: 10 de agosto de 2022.

IEMA. Instituto Estadual de Meio Ambiente e Recursos Hídricos. Disponível em: https://iema.es.gov.br/qualidadedoar/historico#:~:text=O%20monitoramento%20da%20qualidade%20do,poluentes%20atmosf%C3%A9ricos%20em%20tempo%20real.

JUNIOR, O. O Guia do XGBoost com Python. **Dados ao cubo**. 2022. Disponível em: https://dadosaocubo.com/o-guia-do-xgboost-com-python/. Acessado em 10 de Agosto de 2022.

KIM, P. MATLAB Deep learning. **With Machine Learning, Neural Networks and Artificial Intelligence.** Ed. Apress. 2017.

LECUN, Y. *et al.* Gradient-Based Learning Applied to Document Recognition. **Proceedings of the IEEE**, n. November, p. 1–46, 1998.

LIRA, T. S. de. **Modelagem e Previsão da Qualidade do Ar na Cidade de Uberlândia-MG**. 2009. 152 f. 2009.

MONTANTES, J. 3 Reasons to Use Random Forest Over a Neural Network–Comparing Machine Learning versus Deep Learning. **Towards Data Science**, 4 de fevereiro de 2020. Disponível em: https://towardsdatascience.com/3-reasons-to-use-random-forest-over-a-neural-network-comparing-machine-learning-versus-deep-f9d65a154d89. Acessado em: 10 de agosto de 2022.

NOOR, N., *et al*. **Filling the Missing Data of Air Pollutant Concentration using Single Imputation Methods**. 2015. Disponível em: DOI: 10.4028/www.scientific.net/AMM.754-755.923. Acessado em: 24 de agosto de 2022.

NIU, M. et al. A novel hybrid decomposition-and-ensemble model based on CEEMD and GWO for short-term PM2.5 concentration forecasting. **Atmospheric Environment**, v. 134, p. 168–180, 2016. Disponível em: http://dx.doi.org/10.1016/j.atmosenv.2016.03.056.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., and DUCHESNAY, E. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p.2825–2830. 2011.

POLAMURI, S. How the Random Forest algorithm works in Machine Learning. **Dataaspirant**, 22 de maio 2017. Disponível em: https://dataaspirant.com/random-forest-algorithm-machine-learing/. Acessado em: 10 de agosto de 2022.

REIZEN, V. *et al*. **Modeling and forecasting daily average PM10 concentrations by a seasonal long-memory model with volatility**. ELSELVIER. 2014. Disponível em: https://doi.org/10.1016/j.envsoft.2013.09.027. Acessado em 24 de agosto de 2022.

RAMSUNDAR B., ZADER R. B. O'REILLY. **Fully Connected Deep Networks. Capítulo 4**. 2022. Disponível em: https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html.

SAXENA, P. XGBoost. **GeeksforGeeks**, 2022. Disponível em: https://auth.geeksforgeeks.org/user/pawangfg/articles. Acessado em: 11 de julho de 2022.

SHANG, Z. *et al*. A novel model for hourly PM2.5 concentration prediction based on CART and EELM. **Science of the Total Environment**, v. 651, p. 3043–3052, 2019. Disponível em: https://doi.org/10.1016/j.scitotenv.2018.10.193.

SOARES, L. G. *et al*. Simulação da concentração de material particulado inalável de

origem veicular em uma interseção sinalizada de Uberlândia-Mg. *In:* XI Congresso Brasileiro de Engenharia Química em Iniciação Científica, 2015, Unicamp - Campinas - SP. **Anais [...]**, Campinas: Unicamp. 2015. p. 399–404.

TZANIS C. *et al.* Addressing Missing Environmental Data via a Machine Learning Scheme. ***Atmosphere.*** 2021. Disponível em: https://doi.org/10.3390/atmos12040499.

VOULODIMOS, A. *et al.* Deep Learning for Computer Vision: A Brief Review. **Computational Intelligence and Neuroscience**, v. 2018, 2018.

WANG, F., ROSS, C. L. Machine Learning Travel Mode Choices: Comparing the Performance of an Extreme Gradient Boosting Model with a Multinomial Logit Model. **Transportation Research Record**, v. 2672, n. 47, p.35 – 45, 2018.

WEISSTEIN, E.W. **Statistical Correlation**. Disponível em: mathworld.wolfram.com. Acessado em: 11 agosto de 2022.

WU, Q.; LIN, H. A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors. **Science of the Total Environment**, v. 683, p. 808–821, 2019. Disponível em: https://doi.org/10.1016/j.scitotenv.2019.05.288.

ZHANG, A. *et al*. **Dive into Deep Learning**. [S.l.]: arXiv, 2022. Disponível em: https://arxiv.org/abs/2106.11342. Acessado em: 11 de Agosto de 2022.

Apêndice A - Configurações dos modelos de redes DNN e MLP experimentadas

Sequência	Modelo de rede	Dropout	N° de épocas	N° de camadas convolucionais	Tamanho do filtro	Nº de camadas densas	N° neurônios camadas densas	Ativação de camadas densas
1°	DNN	-	800	-	3x3	6	128,64,64,64,32,32	ReLU
2°	MLP	-	1200	16,32,64	3x3	1	15	ReLU
3°	MLP	-	1200	16,32,64	3x3	1	30	ReLU
4°	MLP	-	1200	16,32,64	3x3	1	74	ReLU
5°	MLP	-	1200	16,32,64	3x3	1	148	ReLU
6°	MLP	-	1200	16,32,64	3x3	1	296	ReLU
7°	DNN	-	1200	16,32,64	3x3	1	512	ReLU
8°	MLP	-	1200	16,32,64	3x3	1	592	ReLU
9°	MLP	-	1200	16,32,64	3x3	1	1184	ReLU
10°	MLP	-	1200	16,32,64	3x3	1	2368	ReLU
11°	DNN	-	1200	16,32,64	3x3	2	64,64	ReLU
12°	DNN	-	1200	16,32,64	3x3	2	128,128	ReLU
13°	DNN	-	1200	16,32,64	3x3	4	32,32,16,16	ReLU
14°	DNN	-	1200	32,32,32	3x3	1	8	ReLU
15°	DNN	-	1200	32,32,32	3x3	1	12	ReLU
16°	DNN	-	500	32,32,32	3x3	1	16	ReLU
17°	DNN	-	1200	32,32,32	3x3	1	16	ReLU
18°	DNN	-	1200	32,32,32	3x3	1	20	ReLU
19°	DNN	-	800	32,32,32	3x3	1	24	ReLU
20°	DNN	-	800	32,32,32	3x3	1	32	ReLU
21°	DNN	-	1200	32,32,32	3x3	1	32	ReLU
22°	DNN	-	800	32,32,32	3x3	1	32	Sigmóide
23°	DNN	-	1200	32,32,32	3x3	1	1024	ReLU
24°	DNN	-	1200	32,32,32	3x3	1	2048	ReLU
25°	DNN	-	800	32,32,32	3x3	2	16,16	ReLU
26°	DNN	-	500	32,32,32	3x3	2	32,16	ReLU
27°	DNN	-	800	32,32,32	3x3	2	128,128	ReLU
28°	DNN	-	800	32,32,32	3x3	2	256,256	ReLU
29°	DNN	-	1200	32,32,32	3x3	2	512,512	ReLU
30°	DNN	-	1200	32,32,32	3x3	2	1024,1024	ReLU
31°	DNN	-	800	32,32,32	3x3	3	128,64,32	ReLU
32°	DNN	-	500	32,32,32	3x3	3	16,16,16	ReLU
33°	DNN	-	500	32,32,32	3x3	3	32,32,16	ReLU

34°	DNN	-	800	32,32,32	3x3	4	128,64,32,16	ReLU
35°	DNN	-	800	32,32,32	3x3	4	32,16,16,16	ReLU
36°	DNN	-	800	32,32,32	3x3	4	32,32,16,16	ReLU
37°	DNN	-	1200	32,32,32	3x3	4	32,32,16,16	ReLU
38°	DNN	-	800	32,32,32	3x3	4	32,32,32,16	ReLU
39°	DNN	-	1200	32,32,32	3x3	4	32,32,32,16	ReLU
40°	DNN	-	800	32,32,32	3x3	4	32,32,32,32	ReLU
41°	DNN	-	800	32,32,32	3x3	4	64,32,16,16	ReLU
42°	DNN	-	800	32,32,32	3x3	4	64,32,16,8	ReLU
43°	DNN	-	800	32,32,32	3x3	4	64,32,32,16	ReLU
44°	DNN	-	800	32,32,32	3x3	4	64,64,64,64	ReLU
45°	DNN	-	800	32,32,32	3x3	4	8,8,8,8	ReLU
46°	DNN	-	800	32,32,32	3x3	5	128,64,32,16,8	ReLU
47°	DNN	-	800	32,32,32	3x3	5	128,64,64,32,32	ReLU
48°	DNN	-	800	32,32,32	3x3	6	128,128,64,32,32,32	ReLU
49°	DNN	-	800	32,32,32	3x3	6	128,64,64,32,32,16	ReLU
50°	DNN	1x0.1	800	32,32,32	3x3	6	128,64,64,32,32,16	ReLU
51°	DNN	-	800	32,32,32	3x3	6	128,64,64,32,32,32	ReLU
52°	DNN	-	800	32,32,32	3x3	6	128,64,64,64,32,32	ReLU
53°	DNN	1x0.1	800	32,32,32	3x3	6	32,16,16,16,16,16	ReLU
54°	DNN	1x0.1	800	32,32,32	3x3	6	32,32,16,16,16,16	ReLU
55°	DNN	-	800	32,32,32	3x3	6	32,32,32,16,16,16	Linear
56°	DNN	6x0.3	800	32,32,32	3x3	6	32,32,32,16,16,16	ReLU
57°	DNN	1x0.1	800	32,32,32	3x3	6	32,32,32,16,16,16	ReLU
58°	DNN	2x0.1	800	32,32,32	3x3	6	32,32,32,16,16,16	ReLU
59°	DNN	1x0.1	800	32,32,32	3x3	6	32,32,32,16,16,16	ReLU
60°	DNN	2x0.1	800	32,32,32	3x3	6	32,32,32,16,16,16	ReLU
61°	DNN	1x0.1	800	32,32,32	3x3	6	32,32,32,32,16,16	ReLU
62°	DNN	1x0.1	800	32,32,32	3x3	6	32,32,32,32,32,16	ReLU
63°	DNN	1x0.2	800	32,32,32	3x3	6	32,32,32,32,32,16	ReLU
64°	DNN	-	800	32,32,32	3x3	6	32,32,32,32,32,32	Linear
65°	DNN	1x0.1	800	32,32,32	3x3	6	64,32,32,32,32,16	ReLU
66°	DNN	-	800	32,32,32	3x3	6	64,32,32,32,32,16	ReLU
67°	DNN	6x0.3	800	32,32,32	3x3	6	64,64,64,64,64,64	ReLU
68°	DNN	-	800	32,32,32	3x3	6	64,64,64,64,64,64	ReLU
69°	DNN	1x0.3	800	32,32,32	3x3	6	64,64,64,64,64,64	ReLU
70°	DNN	1x0.1	800	32,32,32	3x3	6	64,64,64,64,64,64	ReLU

71°	DNN	-	800	32,32,32	3x3	7	128,64,64,32,32,32,16	ReLU
72°	DNN	-	800	32,32,32	3x3	7	128,64,64,32,32,32,32	ReLU
73°	DNN	-	800	32,32,32	3x3	9	128,64,64,32,32,32,16,16,16	ReLU
74°	DNN	-	800	32,32,32	3x3	9	128,64,64,32,32,32,32,32,32	ReLU
75°	DNN	-	500	32,BN,32,32	3x3	2	32,16	ReLU
76°	DNN	-	800	32,BN,32,32	2x2	2	32,16	ReLU
77°	DNN	-	800	32,BN,32,32	3x3	2	32L,16L	Linear
78°	DNN	-	800	32,BN,32,32	3x3	4	32,32,16,16	ReLU
79°	DNN		800	32,BN,32,32	3x3	6	32,32,32,16,16,16	ReLU
80°	DNN	-	1200	64,64,64	3x3	1	16	ReLU
81°	DNN	-	1200	64,64,64	3x3	1	32	ReLU
82°	DNN	-	1200	64,64,64	3x3	1	64	ReLU
83°	DNN	-	1200	64,64,64	3x3	1	64	ReLU
84°	DNN	-	1200	64,64,64	3x3	1	128	ReLU
85°	DNN	-	1200	64,64,64	3x3	1	256	ReLU