

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Lucas Rodrigues Frank

Aprendizado de Máquina Aplicado à Gerência de Redes Sem Fio

Juiz de Fora
2023

Lucas Rodrigues Frank

Aprendizado de Máquina Aplicado à Gerência de Redes Sem Fio

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Edelberto Franco Silva

Coorientador: Prof. Dr. Alex Borges Vieira

Juiz de Fora

2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Frank, Lucas Rodrigues.

Aprendizado de Máquina Aplicado à Gerência de Redes Sem Fio /
Lucas Rodrigues Frank. – 2023.
70 f. : il.

Orientador: Edelberto Franco Silva

Coorientador: Alex Borges Vieira

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto
de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computa-
ção, 2023.

1. Predição. 2. Redes sem fio. 3. Gerenciamento. 4. Recurso. 5.
Aprendizado de maquina. 6. Mobilidade. 7. Volume. 8. Handoff. I. Silva,
Edelberto Franco, orient. II. Vieira, Alex Borges, coorient. III. Título.

Lucas Rodrigues Frank

Aprendizado de Máquina Aplicado à Gerência de Redes Sem Fio

Dissertação
apresentada
ao Programa de Pós-
graduação em
Ciência da
Computação,
da Universidade
Federal de Juiz de
Fora, como requisito
parcial à obtenção do
título de Mestre em
Ciência da
Computação. Área de
concentração: Ciência
da Computação.

Aprovada em 18 de dezembro de 2023.

BANCA EXAMINADORA

Prof. Dr. Edelberto Franco Silva - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Alex Borges Vieira - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Heder Soares Bernardino
Universidade Federal de Juiz de Fora

Prof. Dr. Diogo Menezes Ferrazani Mattos
Universidade Federal Fluminense



Documento assinado eletronicamente por **Edelberto Franco Silva, Professor(a)**, em 18/12/2023, às 17:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Heder Soares Bernardino, Professor(a)**, em 28/12/2023, às 10:48, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Diogo Menezes Ferrazani Mattos, Usuário Externo**, em 28/12/2023, às 13:55, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Alex Borges Vieira, Coordenador(a)**, em 29/12/2023, às 07:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Lucas Rodrigues Frank, Usuário Externo**, em 05/01/2024, às 15:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1620090** e o código CRC **4D521909**.

AGRADECIMENTOS

Agradeço a todos os meus parentes, pelo encorajamento e apoio. Em especial, a minha mãezinha que se foi, mas se faz presente todos os dias em meu coração.

Aos professores Edelberto e Alex pelas orientações, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos amigos que fiz e também, aos que já fazem parte da minha vida e que de alguma forma me ajudou ao longo dessa jornada.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

Por fim, agradeço à Universidade Federal de Juiz de Fora (UFJF), pelo apoio financeiro sem o qual este trabalho não poderia ser realizado.

RESUMO

Com o rápido aumento de usuários móveis, a busca pela gestão de recursos em redes sem fio tornou-se um tópico importante. Por conta disso, os estudos de predição de mobilidade e volume de usuários conectados em rede mostram-se relevantes, principalmente considerando o avanço das técnicas de aprendizado de máquina recentemente. Para tanto, entender a mobilidade dos usuários, a fim de prever qual o próximo ponto de acesso para conexão gera benefícios no escopo do gerenciamento de recursos como a busca do tão desejado *handoff* transparente. Assim, prever o volume de usuários, e por volume, entende-se a quantidade de usuários conectados na rede, traz grandes oportunidades como gerir a banda de rede disponível e evitar o gasto de pontos de acessos ligados desnecessariamente. Logo, este trabalho utiliza-se de dois conjuntos de dados reais com origem de dois campos universitários (UFJF e KTH), para treinar modelos preditivos a partir de dados históricos e recentes. Neste trabalho, foram analisados os desempenhos desses modelos para as abordagens de predição de mobilidade e volume de usuários utilizando os seguintes algoritmos: Árvore de Decisão, rede neural Perceptron de Múltiplas Camadas e o processo de Aprendizado de Máquina Automatizado. Por fim, o melhor modelo de cada abordagem foi aplicado em um determinado caso de uso para analisar seu desempenho diante de uma possível aplicação real. Os resultados experimentais indicam que os modelos de mobilidade atingiram uma acurácia média de 91% e 74% para a UFJF e KTH, nessa ordem. Para a abordagem de predição de volume de usuários, o cenário da UFJF obteve coeficiente de ajuste de 0.96 e o cenário KTH de 0.94, o que pode ser classificado como um resultado eficaz dado a sua complexidade. Finalmente, ao analisar os casos de uso, os resultados demonstraram que o modelo de mobilidade da UFJF fornece uma acurácia média de 87.15%, e para o modelo de volume de usuários, a acurácia máxima chega próximo de 95.92% ao prever a quantidade de pontos de acessos necessário para atender os usuários conectados naquele instante. Analogamente, o cenário do campus KTH obteve uma acurácia média de 70.63% para mobilidade, e acurácia máxima de 97.28% para o modelo do volume de usuários conectados na rede.

Palavras-chave: Predição. Redes sem fio. Gerenciamento. Recurso. Aprendizado de máquina. Mobilidade. Volume. Handoff.

ABSTRACT

With the rapid increase of mobile users, the pursuit of resource management in wireless networks has become an important topic. Because of this, studies on predicting mobility and volume of users connected to a network are relevant, especially considering recent advances in machine learning techniques. To this end, understanding the mobility of users, in order to predict the next access point for connection, generates benefits in the scope of resource management such as the search for the much desired transparent *handoff*. Thus, predicting the number of users, and by volume, we mean the number of users connected to the network, brings great opportunities such as managing the available network bandwidth and avoiding the expense of unnecessarily connected access points. Therefore, this work uses two sets of real data originating from two university fields (UFJF and KTH), to train predictive models based on historical and recent data. In this work, the performance of these models was analyzed for mobility and number of users prediction approaches using the following algorithms: Decision Tree, Multi-Layer Perceptron neural network and the Automated Machine Learning process. Finally, the best model of each approach was applied to a given use case to analyze its performance in a possible real application. The experimental results indicate that the mobility models achieved an average accuracy of 91% and 74% for UFJF and KTH, in that order. For the number of users prediction approach, the UFJF scenario obtained an adjustment coefficient of 0.96 and the KTH scenario of 0.94, which can be classified as an effective result given its complexity. Finally, when analyzing the use cases, the results demonstrated that the UFJF mobility model provides an average accuracy of 87.15%, and for the number of users model, the maximum accuracy comes close to 95.92% when predicting the number of access points needed to serve the users connected at that moment. Similarly, the KTH campus scenario obtained an average accuracy of 70.63% for mobility, and a maximum accuracy of 97.28% for the model of the number of users connected to the network.

Keywords: Prediction. Wireless networks. Management. Resource. Machine learning. Mobility. Number. Handoff.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma de AutoML.	19
Figura 2 – Simples representação de uma árvore de decisão.	20
Figura 3 – Composição da estrutura de um EP.	23
Figura 4 – Exemplo - Arquitetura de uma rede neural <i>Feedforward</i> com 1 camada oculta e duas saídas.	25
Figura 5 – Exemplo - Arquitetura de uma rede neural <i>Feedforward</i> sem camada oculta e com 4 saídas.	26
Figura 6 – Gráfico da função logística.	27
Figura 7 – Gráfico da derivada da função logística.	27
Figura 8 – Gráfico da função ReLU.	28
Figura 9 – Gráfico da derivada da função ReLU.	28
Figura 10 – Representação do monitoramento de fluxo com NetFlow.	29
Figura 11 – Visão geral da proposta.	36
Figura 12 – Fluxograma que representa ambas as abordagens.	37
Figura 13 – Trajetória com 5 pontos de acesso.	37
Figura 14 – 4 trajetórias de tamanho $N = 5$	38
Figura 15 – Arquitetura MLP para a abordagem I.	41
Figura 16 – Arquitetura MLP para a abordagem II.	41
Figura 17 – Mapa do campus da UFJF.	42
Figura 18 – Matriz de confusão - Autosklearn - UFJF	49
Figura 19 – Matriz de confusão - Autosklearn - KTH	50
Figura 20 – Comparação entre valor predito e valor real - Autosklearn - UFJF	51
Figura 21 – Comparação entre valor predito e valor real - Autosklearn - UFJF	51
Figura 22 – Comparação entre valor predito e valor real - Autosklearn - KTH	52
Figura 23 – Comparação entre valor predito e valor real - Autosklearn - KTH	53
Figura 24 – Comparação do valor previsto e verdadeiro da quantidade de pontos de acesso necessários - Matriz de Confusão.	56
Figura 25 – Comparando o valor previsto e o valor real - a quantidade de pontos de acesso necessários.	57
Figura 26 – Comparação do valor previsto e verdadeiro da quantidade de pontos de acesso necessários - Matriz de Confusão.	58
Figura 27 – Comparando o valor previsto e o valor real - a quantidade de pontos de acesso necessários.	59

LISTA DE TABELAS

Tabela 1 – Resumo dos trabalhos relacionados escolhidos.	35
Tabela 2 – Exemplo do <i>dataframe</i> para a abordagem I.	38
Tabela 3 – Exemplo do <i>dataframe</i> para a abordagem II.	40
Tabela 4 – Exemplo da base de dados da UFJF.	43
Tabela 5 – Finalidade das colunas mais relevantes.	43
Tabela 6 – Resumo do conjunto de dados da UFJF.	44
Tabela 7 – Exemplo da base de dados da Suécia.	44
Tabela 8 – Resumo do conjunto de dados da KTH.	44
Tabela 9 – Resumo do desempenho dos modelos para predição de mobilidade no campus UFJF.	48
Tabela 10 – Resumo do desempenho dos modelos para predição de mobilidade no campus KTH.	49
Tabela 11 – Resumo do desempenho dos modelos para predição de volume de usuários no campus UFJF.	50
Tabela 12 – Resumo do desempenho dos modelos para predição de volume de usuários no campus KTH.	52
Tabela 13 – Resumo dos resultados para cinco usuários com maior ganho em <i>Mb</i>	53
Tabela 14 – Resumo geral dos resultados para o conjunto de dados da UFJF.	54
Tabela 15 – Resumo dos resultados para cinco usuários com maior ganho em <i>Mb</i>	55
Tabela 16 – Resumo geral dos resultados para o conjunto de dados do campus KTH.	55

LISTA DE ABREVIATURAS E SIGLAS

UFJF	Universidade Federal de Juiz de Fora
QoS	<i>Quality of Service</i>
QoE	<i>Quality of Service</i>
ML	<i>Machine Learning</i>
AP	<i>Access Point</i>
MLP	<i>Multilayer Perceptron</i>
DT	<i>Decision Tree</i>
AutoML	<i>Automated Machine Learning</i>
LSTM	<i>Long Short Term Memory</i>
KNN	<i>K-Nearest Neighbors</i>
SVM	<i>Support Vector Machine</i>
GRU	<i>Gated Recurrent Unit</i>
RF	<i>Random Forest</i>
BPNN	Rede Neural de <i>Backpropagation</i>
RNN	Rede Neural Recorrente
NB	<i>Naive Bayes</i>
ARMA	<i>Autoregressive Moving Average</i>
MTS	Séries Temporais Multivariadas
SARIMA	Média Móvel Integrada Sazonal Autorregressiv
(LA-ResNe	Rede Neural Recorrente
KTH	<i>KTH Royal Institute of Technology</i>
MSE	<i>Mean Squared Error</i>
MAE	<i>Mean Absolute Error</i>
SSL	<i>Squared Sum Left</i>
SSR	<i>Squared Sum Right</i>
RNA	Redes Neurais Artificiais
EP	Elementos de Processamento
ReLU	<i>Rectified Linear Unit</i>
SDN	<i>Software-Defined Networking</i>
HMM	<i>Hidden Markov Model</i>
CSPM	<i>Clustering and Sequential Pattern Mining</i>
SPMC	<i>Sequential Pattern Mining and Clustering</i>
IoT)	<i>Internet of Things</i>
PCA	<i>Principal Component Analysis</i>
IA	Inteligência Artificial
DPCCA	<i>Detrended Partial Cross Correlation Analysis</i>
GSTAR	<i>Generalized Spatio-Temporal Auto-Regressive</i>
ALQPSO	<i>Quantum Particle Swarm Optimization</i>
GA-BPNN	Rede Neural BP combinado à otimização por Algoritmo Genético

QPSO-BPNN	Rede Neural BP combinado à otimização por Enxame de Partículas Quânticas
IDS	Sistema de Detecção de Intrusão
PGA	Algoritmo Genético Paralelo
LGPD	Lei Geral de Proteção de Dados Pessoais
RAM	<i>Random Access Memory</i>

LISTA DE SÍMBOLOS

\cup	União
Σ	Somatório
$f(x)$	Função de x
$f'(x)$	Derivada da função de x
$ x $	Módulo de x
$>$	Maior que
\leq	Menor ou igual a
$\#$	Quantidade
\pm	Mais ou menos

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVO	15
1.2	CONTRIBUIÇÕES	16
1.3	ORGANIZAÇÃO DO TEXTO	16
2	FUNDAMENTOS	18
2.1	APRENDIZADO DE MÁQUINA AUTOMATIZADO	18
2.1.1	Ferramenta Auto-sklearn	19
2.2	ÁRVORE DE DECISÃO	20
2.2.1	Critério Índice de Gini	21
2.2.2	Critério Erro Quadrático Médio Friedman	22
2.3	REDES NEURAIS ARTIFICIAIS	22
2.3.1	Modelo <i>Feedforward</i>	24
2.3.2	Função de Ativação	25
<i>2.3.2.1</i>	<i>Logística</i>	25
<i>2.3.2.2</i>	<i>ReLU</i>	26
2.4	GERENCIA DE REDES	29
3	TRABALHOS RELACIONADOS	31
3.1	PREDIÇÃO DA MOBILIDADE DOS USUÁRIOS	31
3.2	PREDIÇÃO DO VOLUME DE USUÁRIOS	33
3.3	APRENDIZADO DE MÁQUINA AUTOMATIZADO	34
3.4	CONSIDERAÇÕES DO CAPÍTULO	35
4	MÉTODOS PROPOSTOS	36
4.1	ABORDAGEM I	37
4.1.1	Pré-processamento da Abordagem I	37
4.2	ABORDAGEM II	39
4.2.1	Pré-processamento da Abordagem II	39
4.3	MODELOS	40
4.3.1	MLP	40
4.3.2	Árvore de Decisão	40
4.3.3	Auto-sklearn	42
4.4	CENÁRIO	42
4.4.1	Conjunto de Dados - UFJF	42
4.4.2	Conjunto de Dados - KTH	44
5	EXPERIMENTOS E RESULTADOS	45
5.1	MÉTRICAS	45
5.1.1	Classificação	45
5.1.2	Regressão	47

5.2	RESULTADOS - AVALIAÇÃO DOS MODELOS	47
5.2.1	Modelos - Predição de Mobilidade - UFJF	48
5.2.2	Modelos - Predição de Mobilidade - KTH	48
5.2.3	Modelos - Predição de Volume - UFJF	49
5.2.4	Modelos - Predição de Volume - KTH	51
5.3	CASOS DE USO - GERÊNCIA DE REDES	52
5.3.1	Predição de Mobilidade - UFJF	52
5.3.2	Predição de Mobilidade - KTH	54
5.3.3	Predição do Volume - UFJF	55
5.3.4	Predição do Volume - KTH	57
5.4	CONSIDERAÇÕES DO CAPÍTULO	59
6	CONCLUSÃO	61
	REFERÊNCIAS	63

1 INTRODUÇÃO

Recentemente, são notáveis as mudanças de tecnologia, seja em relação aos equipamentos de uso pessoal conectados à Internet, à velocidade ofertada a esses usuários, ou ainda à experiência que pode-se obter considerando a cobertura de acesso à Internet. Dentre os principais avanços estão as redes sem fio em geral, e mais notavelmente as redes locais sem fio [1]. Seu crescimento é facilmente notado desde a casa dos usuários (*e.g.*, aumentando a cobertura), até espaços abertos e compartilhados, como aeroportos, *shoppings centers*, instituições e centros de ensino e pesquisa.

A distribuição de pontos de acessos (*Access Point* - AP) em redes estruturadas sem fio tem como um dos principais objetivos cobrir uma grande área, de modo a fornecer cobertura do sinal da rede para vários usuários. Tal característica é um dos principais motivos para o crescimento do número de usuários móveis, e por consequência, o aumento de dispositivos móveis com acesso à Internet. Porém, este aumento e suas consequências intrínsecas de requisitos fazem com que a rede precise lidar com diversas limitações de recursos, como a quantidade de banda larga disponível em uma determinada área [2], e o número de usuários simultâneos [3, 4].

Considerando essas limitações, mostra-se necessário gerenciar os recursos da rede da melhor forma possível. Tal gerência deve evitar desde a escassez dos recursos disponíveis quanto incrementar a qualidade de serviço (*Quality of Service* - QoS) e a qualidade de experiência (*Quality of Experience* - QoE) do usuário. Assim, é interessante considerar tomadas de decisão de forma inteligente de modo que não influencie negativamente na forma que os recursos são alocados, e por consequência, até mesmo afetar na QoS e QoE. Por conta disso, uma de muitas aplicações atuais são direcionadas a usuários móveis, onde um dos pontos interessantes a se estudar é a mobilidade desses usuários [5, 6]. Com o aumento do número desses usuários móveis e da cobertura das redes sem fio locais, a necessidade de associação do usuário nos APs durante seu deslocamento são frequentes [7]. Por exemplo, em um dos conjuntos de dados analisados neste trabalho, uma semana de dados possui mais de 500.000 associações a APs. Tais associações devem ser o mais transparente possível para o usuário, principalmente quando dentro de uma rede de mesmo domínio. Assim, considera-se tais transferências de associação, ou associação a APs distintos de maneira transparente utilizando seu termo comum, o *handoff* (horizontal). Quando um usuário experimenta um *handoff*, ele pode perceber desde o atraso nos pacotes enviados e recebidos até a queda de sua conexão. Assim, minimizar esse impacto em uma rede sem fio local (desde pequeno a grande porte) é de suma importância para que sejam alcançados os objetivos mencionados de QoS [8]. Além do mais, a partir dessa grande quantidade de associações de rede, tem-se o número de usuários conectados que influencia diretamente na necessidade de alocação de banda de rede disponível que possa atender a todos esses usuários, garantindo uma melhor QoE.

Já há algum tempo que as técnicas de Aprendizado de Máquina (*Machine Learning* - ML) são empregadas em redes sem fio possuindo diversas aplicações e melhorias [9]. Para alcançar tais melhorias na rede, diversas análises são necessárias, e é neste momento que ML possui uma enorme relevância. Com um grande sucesso, a cada dia as técnicas existentes e suas aplicações se expandem mais e mais [10]. Tanto na área acadêmica quanto empresarial, a aplicação de ML pode ser resumida em pequenas tarefas como busca por otimizações, extração de conhecimento de um conjunto de dados, automatizar ações a partir de previsões ou agrupamentos. Há alguns anos que diversas áreas vêm adotando ML no seu cotidiano, e fica cada vez mais evidente a sua eficácia, como exemplo, seja relacionado a um crescimento econômico na agricultura ou possuindo um papel importante na área da saúde, tal como uma análise detalhada da COVID-19 [11, 12].

Através de ML, a predição da mobilidade e volume/carga dos usuários conectados traz a possibilidade de uma melhora no gerenciamento de recursos para a rede sem fio, e proporciona uma melhor QoS para os próprios usuários. Uma preocupação que se destaca ao relacionar gerenciamento de recursos com QoS é a tentativa de otimizar os recursos utilizados pela rede, *i.e.*, alocação/balanceamento de usuários, APs, banda larga, etc, sem afetar os requisitos mínimos de qualidade, tal como em trabalhos que precisa-se otimizar a qualidade de transmissão sem que afete os requisitos específicos de QoS como atraso esperado [13].

Para este trabalho foram seguidas duas abordagens. Com a abordagem de predição de mobilidade é possível reservar recursos e migrar dados referentes à futura associação do usuário antes que ela aconteça. Desse modo, os *handoffs* do usuário acontecem de forma transparente, ou seja, sem que ele perceba que houve uma transferência de sua conexão de um AP para outro. Já para a abordagem de predição de volume de usuários é capaz de gerenciar recursos como: banda de rede alocada, ligar ou desligar APs de acordo com a quantidade de usuários previsto, entre outras funções. Assim, é viável gerenciar os dispositivos de rede (*e.g.*, APs) a fim de reduzir o consumo de energia, por exemplo, balanceando (forçando o *handoff* - a associação do usuário em um outro AP próximo a ele com a intenção de desligar aquele AP original que apresenta baixa carga de usuários.) a associação de usuários entre os APs.

1.1 OBJETIVO

O objetivo deste trabalho se resume em, primeiramente, utilizar de técnicas de aprendizado de máquina, como o Perceptron de Múltiplas Camadas (MLP), Árvore de Decisão (DT) e o processo de Aprendizado de Máquina Automatizado (AutoML) para estudar a mobilidade e volume de usuários conectados à rede sem fio. A partir disso, os modelos preditivos treinados para predição de mobilidade serão responsáveis por auxiliar uma técnica de *handoff* transparente, onde dado que o usuário esteja conectado em um

AP, deverá ser predito qual o próximo AP que este usuário irá se conectar. E os modelos treinados para predição de volume de usuários ficarão a cargo de prever o número de usuários conectados e também a aplicar os modelos para alocar apenas a quantidade necessária APs. Para avaliação, tanto o desempenho dos modelos quanto os casos de uso aplicados para gerência à redes serão avaliados em dois conjuntos de dados reais, sendo um conjunto da própria Universidade Federal de Juiz de Fora (UFJF) registrando as autenticações dos usuários no ano de 2019 e outro conjunto do Instituto Real de Tecnologia (KTH) situado na Suécia com registros entre os anos de 2014 e 2015 [14].

1.2 CONTRIBUIÇÕES

Em suma, nossas contribuições são três: (i) o uso de técnicas de pré-processamento em conjunto com modelos preditivos para predição da mobilidade e volume dos usuários em redes sem fio; (ii) a avaliação da proposta sobre dois conjunto de dados reais obtidos de redes sem fio de larga escala, e destacando que até o momento de estudo deste trabalho, o conjunto de dados advindo do campus KTH não possui trabalhos publicados que relacionam com este tema; e (iii) a análise do impacto dessa proposta no cenário de gerência à redes, *i.e.*, análises de casos de uso referente ao impacto do *handoff transparente* e alocação de APs de acordo com a demanda. Os resultados apresentados mostram que os modelos de mobilidade alcançaram uma acurácia média de 91% e 74% para o cenário da UFJF e do campus KTH, respectivamente. Destaca-se também que os resultados dos modelos de mobilidade já superaram resultados alcançados anteriormente onde o modelo atingiu uma acurácia de 45% [15]. Já os modelos de carga obtiveram um coeficiente de determinação de 0.96 e 0.94, nessa mesma ordem, o que pode ser classificado como um resultado satisfatório para a proposta desse trabalho a princípio dado a complexidade do problema. Além disso, quando analisado os casos de uso para a UFJF, o modelo de mobilidade chegou a 87.15% de acurácia média, e no melhor caso, uma acurácia máxima de 95.92% para o modelo de carga. Para o cenário do campus KTH, o modelo de mobilidade atingiu uma acurácia média de 70.63%, e 97.28% de acurácia máxima quando considerando o melhor caso para o modelo de carga.

1.3 ORGANIZAÇÃO DO TEXTO

O restante deste trabalho está organizado da seguinte forma. No Capítulo 2 é apresentada a fundamentação teórica deste trabalho, onde são descritas cada uma das técnicas de ML utilizadas e uma breve descrição sobre gerência à redes. Logo após, no Capítulo 3, é realizada uma descrição e análise sobre os trabalhos relacionados que contribuíram para esta pesquisa. O Capítulo 4 descreve as propostas relacionadas a esta pesquisa, detalhando o pré-processamento empregado em cada conjunto de dados, assim como também especificações dos modelos preditivos usados nos treinamentos/testes e traz

uma breve descrição sobre cada um dos conjuntos de dados utilizados. No Capítulo 5, são apresentados e discutidos os resultados experimentais sobre os modelos e os casos de uso. Por fim, o Capítulo 6 conclui este trabalho e descreve as direções futuras.

2 FUNDAMENTOS

Neste capítulo são abordados conceitos e informações sobre a técnica de aprendizado de máquina automatizado, como também a estrutura e as funcionalidades dos modelos supervisionados árvore de decisão e rede neural. Além disso, são apresentados conceitos de gerencia de redes, principalmente ao que diz respeito à aplicação de técnicas de aprendizado de máquina nesta área.

2.1 APRENDIZADO DE MÁQUINA AUTOMATIZADO

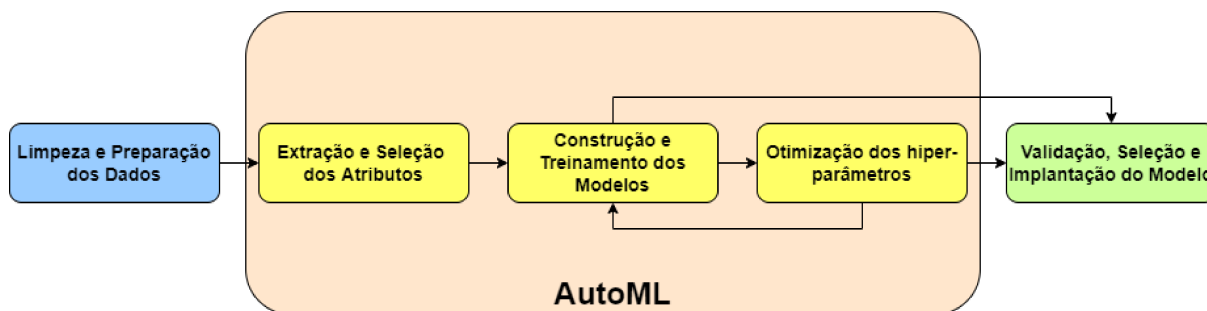
A área de aprendizado de máquina cresce cada vez mais, desde o número de técnicas disponíveis quanto o número de pessoas querendo aplicá-las. Um dos principais objetivos é a busca em automatizar processos e realizar predição de interesses dos clientes. Dentre as mais diversas aplicações existentes nos dias atuais, a utilização de aprendizado de máquina pode ser citada em: recomendações de conteúdo e de produtos, filtros de spam e respostas automáticas, detecção de fraudes, etc. [16]. Entretanto, não é uma tarefa fácil utilizar todo o conjunto de ferramentas fornecido pelo aprendizado de máquina. Mesmo um especialista na área teria que preparar o conjunto de dados passando por técnicas de pré-processamento, extração e seleção de atributos relevantes, e após isso, seleção do algoritmo mais adequado para o problema em questão, além de ter que otimizar seus hiperparâmetros [17]. Com isso, surge o conceito de Aprendizado de Máquina Automatizado, ou como é comumente denominado, *AutoML* que é a tentativa de deixar as técnicas de aprendizado de máquina mais acessíveis para todos, incluindo pessoas que não são especialistas da área [18].

Hoje em dia, já existem ferramentas que permitem trabalhar em nuvem que visam a ideia de automatizar o processo de treinamento e escolha dos modelos, ao mesmo tempo que reduzem o tempo necessário dessas ações. Grandes empresas como Google e Microsoft investiram no ramo de AutoML e disponibilizam a ferramenta em suas respectivas plataformas, Azure e Google Clouds, e ainda destacam para os futuros usuários a seguinte mensagem: "[...] modelos de *machine learning* personalizados e de alta qualidade com o mínimo de esforço e conhecimento sobre o assunto." [19, 20].

Na Figura 1 é ilustrado o fluxograma do funcionamento do processo de AutoML. Primeiramente, tem-se a limpeza e preparação dos dados que incluem a aplicação de transformações, normalizações, etc. Logo em seguida, a extração de novas características pode tomar-se necessária, e após isso, a seleção dos atributos do conjunto de dados para serem incluídos no sistema AutoML. Uma vez que os dados já foram pré-processados e incluídos no sistema autoML, um processo iterativo é iniciado em que é realizado a construção, treinamento e otimização dos modelos, e ao fim, o modelo pode ser validado e pronto para ser implantado [21].

Além dos serviços pagos que oferecem a ferramenta AutoML, existem diversas

Figura 1 – Fluxograma de AutoML.



Fonte: Adaptado de [21]. (2022)

bibliotecas gratuitas, até mesmo com códigos *open-source* que permitem a utilização dessa técnica. Bibliotecas essas como: AutoWEKA, Autosklearn, Auto-PyTorch, Auto-ViML, Auto-Keras, H2O AutoML, AutoGluon, etc. Para este trabalho, foi utilizado a biblioteca Autosklearn que já é largamente adotada na literatura e simples de usar, e por possuir o código aberto, facilitando na pesquisa [22].

2.1.1 Ferramenta Auto-sklearn

Como já mencionado, o Auto-sklearn faz parte de um conjunto de ferramentas que permite ao usuário criar modelos de aprendizado de máquina poderosos de forma automatizada. Para todo o processo de pré-processamento, treinamento e avaliação utiliza-se como base a biblioteca do *Scikit-learn*, que é considerada por muitos uma das mais utilizadas pela literatura [23].

Desenvolvido por Matthias Feurer et al em 2015 [24], o Auto-sklearn faz uso de uma otimização bayesiana eficiente para verificar quais modelos são possíveis e quais configurações funcionam melhor para um dado problema.

De acordo com Matthias Feurer et al em: "[...]based on scikit-learn (using 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods, giving rise to a structured hypothesis space with 110 hyperparameters)"[24], destaca-se que o Auto-sklearn inicialmente já possuía pelo menos 15 classificadores, e diversos métodos tanto para auxiliar na escolha dos atributos quanto para o pré-processamento dos dados.

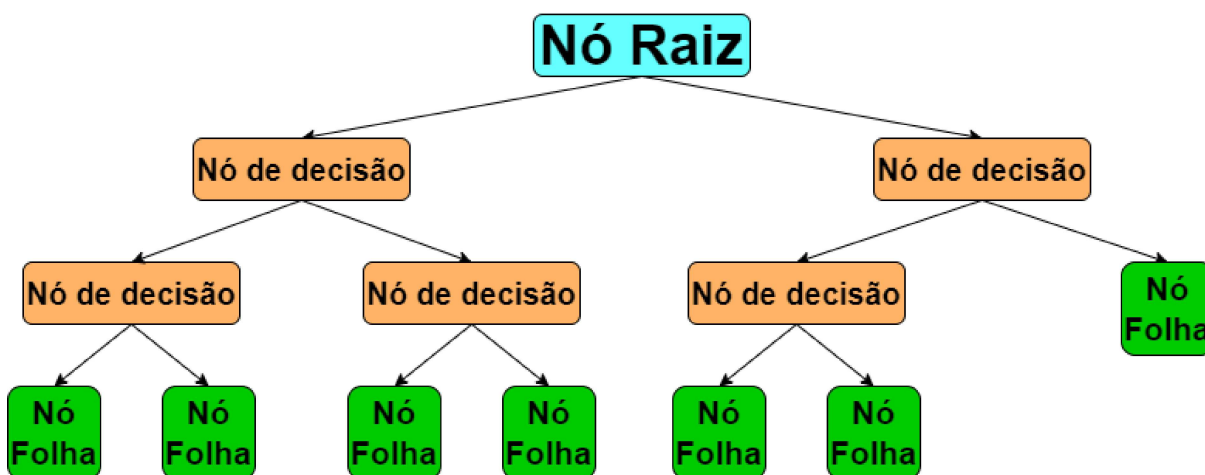
Nos dias atuais, a partir do código aberto que pode ser visualizado em [25, 26], o Auto-sklearn apresenta 16 e 12 algoritmos para modelos de classificação e regressão, respectivamente. Diversos modelos conhecidos podem ser encontrados dentre esses algoritmos, como por exemplo, máquinas de vetor suporte, K-vizinhos mais próximos, modelos baseado em árvore de decisão e redes neurais, métodos de *ensemble*, etc.

2.2 ÁRVORE DE DECISÃO

Uma árvore de decisão é uma poderosa ferramenta popular em Aprendizado de Máquina e é caracterizada por ser um modelo de predição simples e de fácil interpretação [27]. Definido por ser um modelo sequencial, que por sua vez, realiza uma sequência de diversos testes simples com objetivo de combiná-los [28]. Comumente comparado a construção de um fluxograma ou árvore binária, a árvore de decisão também é descrita como um conjunto de funções lógicas [29]. Na Figura 2 é exemplificado esse fluxograma, possuindo nós internos e externos, distribuídos hierarquicamente. A partir disso, os nós de uma árvore de decisão são classificados como:

- **Nó Raiz:** primeiro nó ou o nó mais superior na árvore de decisão, representando a característica que melhor divide o conjunto de dados,
- **Nó de decisão:** desde o nó raiz, os nós internos ou intermediários representam diferentes características que possam dividir o conjunto de dados até chegar a um nó folha.
- **Nó Folha:** localizado como último nó em um ramo da árvore, representa o valor a ser inferido.

Figura 2 – Simples representação de uma árvore de decisão.



Fonte: Elaborado pelo autor. (2022)

Por essa representação, pode-se extrair que uma árvore de decisão pode ser descrita por uma disjunção de conjunções, *i.e.*, para cada nó folha, temos um conjunto de nós que partem desde o nó raiz até esse nó folha formando uma conjunção [30].

Seguindo o aprendizado supervisionado (*i.e.*, utiliza-se de um conjunto de dados em que o valor a ser predito é conhecido), o treinamento de um modelo árvore de decisão

pode ser explicado de forma bem simples [31, 32]. A partir de uma abordagem gulosa *top-down*, primeiramente, um atributo é selecionado de modo que consiga separar os dados da melhor forma possível, gerando subconjuntos o mais heterogêneos entre si. E para que esse passo seja alcançado, é utilizado de um critério/métrica que ajude a calcular o ganho de informação de cada atributo, e assim é escolhido o atributo com maior ganho de informação [33]. Uma vez que esse passo é cumprido, esse atributo se torna a característica principal que vai condicionar na tomada de decisão desse nó. E posteriormente, novos atributos são selecionados de acordo com o critério utilizado para escolher os próximos nós de decisão. No momento em que o conjunto de dados utilizado estiver dividido o suficiente para que se possa inferir sobre o mesmo, a árvore de decisão terá seus nós folhas com o valor predito relacionado aos dados treinados [34].

A escolha do critério a ser utilizado no treinamento de uma árvore de decisão varia de problema a problema, e isso só destaca a sua importância para alcançar um bom modelo [35]. Dentre as variações possíveis na atualidade, temos os critérios mais comuns que são Entropia e Índice de Gini [36] para modelos de classificação (*i.e.*, predição do modelo é uma classe, *e.g.*, gato, cachorro, valores inteiros, etc...), e Erro Quadrático Médio e Erro Absoluto [37] para modelos de regressão (*i.e.*, predição do modelo é um valor real, *e.g.*, preços, volumes, etc...). Para este trabalho, o foco será nos seguintes critérios: Índice de Gini e uma variação do critério de Erro Quadrático Médio de Friedman, em que ambos resultaram em modelos com desempenhos melhores.

2.2.1 Critério Índice de Gini

Criado pelo estatístico Conrado Gini em 1912, o Índice de Gini é uma métrica bastante utilizada para medir desigualdade até os dias atuais, mais especificamente em áreas da economia, demografia, e até como ferramenta política [38]. Também usado para árvore de decisão, o objetivo dessa métrica é avaliar as características do conjunto de dados e escolher o atributo que fará o particionamento do mesmo de forma mais apropriada a cada nó através do cálculo de impureza desses atributos [39].

$$\text{Índice de Gini} = 1 - \sum_{i=1}^J p_i^2 \quad (2.1)$$

Esse índice de impureza pode ser calculado pela Equação 2.1, onde J é o número de classes e p_i descreve a frequência relativa de cada classe em cada nó. Dado que o intervalo dessa equação varia entre $[0, 1]$, um nó é definido puro quando seu valor é próximo de 0 (*i.e.*, a frequência de uma classe é majoritária), e impuro quando o valor se aproxima de 1 (*i.e.*, em um nó, as frequências das classes são distribuídas de forma uniforme). Levando isso em conta, o atributo com maior grau de impureza tem a maior probabilidade de ser escolhido como nó, e dessa forma, reduzindo o grau de impureza no próximo particionamento [40].

2.2.2 Critério Erro Quadrático Médio Friedman

Similar com o uso do Índice de Gini sendo critério de particionamento para árvores de decisão como modelo classificador, o critério de Erro Quadrático Médio, ou do inglês, *Mean Squared Error* (MSE) é comumente utilizado para problemas de regressão [41]. Mesmo a árvore de decisão agindo como um modelo regressor, o seu funcionamento segue a mesma proposta discutida anteriormente em que o conjunto de dados continua sendo particionado através da avaliação da impureza dos nós até alcançar um número mínimo de amostras.

Nesse caso, para calcular a impureza dos nós, o critério MSE é definido na Equação 2.2, sendo que SSL e SSR são a soma dos quadrados dos nós a esquerda e a direita do nó atual, respectivamente. Já as variáveis $w_{esquerda}$ e $w_{direita}$ são a soma dos pesos dos nós a esquerda e a direita, nessa ordem.

$$MSE_{impureza} = \frac{SSL}{w_{esquerda}} + \frac{SSR}{w_{direita}} \quad (2.2)$$

Considerando esse critério, em 1999, Jerome H. Friedman desenvolveu uma melhoria para essa métrica em que permite ao modelo não somente considerar o quanto próximo está do resultado esperado, como também se basear nas probabilidades de uma região (esquerda ou direita) [42]. Na Equação 2.3 é descrito o cálculo dessa melhoria de Friedman, em que $w_{esquerda}$ e $w_{direita}$ já foram mencionados, e $media_{esquerda}$ e $media_{direita}$ são simplesmente a média dos valores dos nós da esquerda e direita, nessa devida ordem.

$$Friedman_{melhoria} = \frac{w_{esquerda} \times w_{direita}}{w_{esquerda} + w_{direita}} \times (media_{esquerda} - media_{direita})^2 \quad (2.3)$$

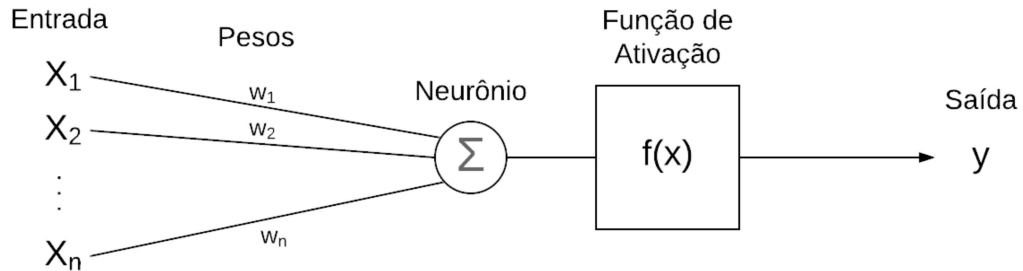
2.3 REDES NEURAIIS ARTIFICIAIS

Baseado em modelos matemáticos, as Redes Neurais Artificiais (RNA) simulam o funcionamento do sistema nervoso de um organismo vivo. Composta por diversos Elementos de Processamento (EP), as RNAs são comumente descritas como uma máquina de aprendizado adaptativa e distribuída não-linear [43]. Esses EPs, por sua vez, são caracterizados por poder se conectar a si próprio e a outros EPs onde carregam cálculos por toda sua estrutura. Também conhecidos como neurônios, são criados para entender e imitar o sistema nervoso através do paralelismo em uma rede densamente interconectada.

Em sua estrutura, as RNAs podem ser descritas por uma certa quantidade de neurônios distribuídos por um número de camadas onde é definido uma conexão entre esses EPs. Essa conexão é estabelecida ao associar um valor a mesma geralmente chamado de peso ou sinapse [44]. Grande parte do funcionamento satisfatório das RNAs dependem exclusivamente de como é determinado sua arquitetura (*e.g.* número de camadas, número

de neurônios em cada camada, definição dos pesos entre neurônios, etc), embora que cada EP realize um processamento simples, um conjunto de EPs permite solucionar problemas mais complexos, conforme o autor em [44].

Figura 3 – Composição da estrutura de um EP.



Fonte: Adaptado de [44]

A Figura 3 demonstra a estrutura de um simples EP podendo ser composta por n entradas X , sendo associado um peso w para cada uma dessas entradas. Para prosseguir com o processamento, o EP soma o valor dessas entradas multiplicadas ao valor do seu respectivo peso, e em seguida, essa soma é atribuída a uma função de ativação. Na Equação 2.4 é definido como é realizado o cálculo de cada neurônio, ou seja, para cada neurônio k , é executado o somatório da entrada x_i com o peso w_{ki} associado, ao fim somado a um bias b_k que, por padrão, é descrito como uma constante de excitação de valor igual 1 [44]. Logo após o resultado ser obtido dessa função, o mesmo é passado para os neurônios da camada seguinte para que o mesmo processo seja repetido até alcançar a última camada da arquitetura, assim obtendo a saída final da rede neural. Mas antes de obter qualquer resultado, é necessário um conjunto de informações ou vetor de características a ser fornecido para o treinamento para a RNA aprender sobre um determinado problema. Para que tal objetivo seja alcançado, durante o treinamento é importante que a RNA atualize os pesos que conectam seus elementos em função de minimizar uma determinada métrica de erro. Dentre as diversas técnicas presentes que possibilitam na atualização desses pesos, uma das mais utilizadas na literatura é o algoritmo de *Backpropagation*. Por sua vez, esse algoritmo utiliza-se da regra Delta que é caracterizado por um aprendizado gradiente descendente [45]. O cálculo da atualização dos pesos são feitos considerando a soma dos erros de cada entrada em conjunto com a derivada da função de ativação e da taxa de aprendizagem [46].

$$y_k = f \left(\left(\sum_{i=1}^m w_{ki} * x_i \right) + b_k \right) \quad (2.4)$$

Para um bom treinamento, as RNA possuem diversos hiper-parâmetros que influenciam diretamente no seu funcionamento. Dentre esses hiper-parâmetros, podem ser destacados alguns que já foram citados nessa seção tais como número de camadas e número de neurônios por camada que é evidenciado na arquitetura da RNA. Mas não menos importante, os hiper-parâmetros como taxa de aprendizagem e número de épocas, apesar de não aparecer na arquitetura, atuam durante o processo de treinamento da RNA. Um exemplo é o número de épocas que é destacado por determinar a quantidade de vezes que a RNA deve treinar sobre o conjunto de dados. A importância desse hiper-parâmetro relaciona-se tanto com a possível melhora do desempenho do modelo quanto também ocasionar outros problemas como *overfitting* (*i.e.* modelo se adaptar demais a um determinado conjunto, e não conseguir generalizar para novos dados).

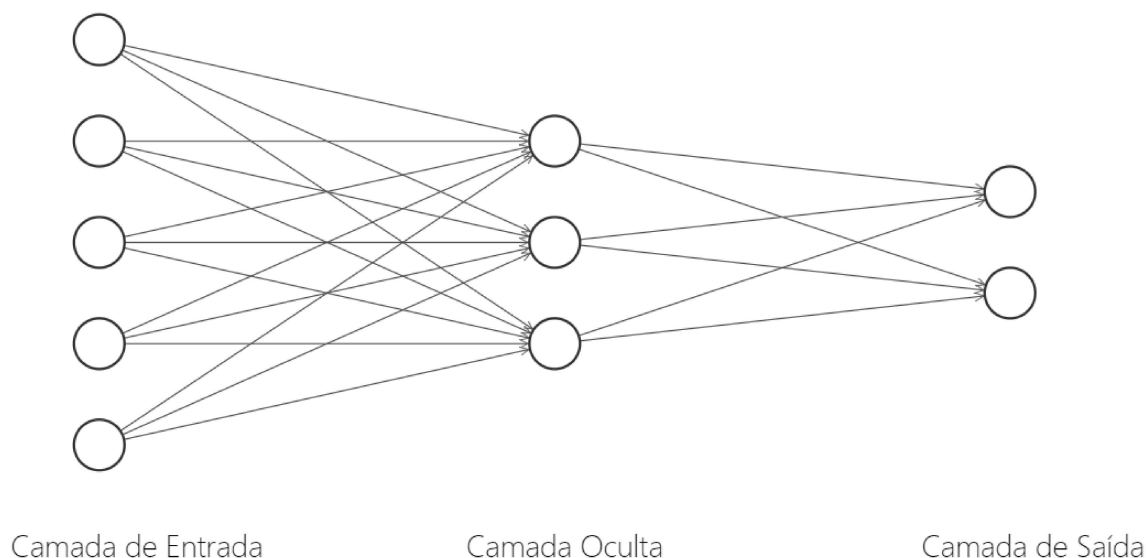
As RNAs podem ser divididas em várias categorias, estas sendo: redes neurais *feedforward*, recorrente e convolucionais. O modelo *feedforward* já teve grande parte do seu funcionamento já explicado por ser o algoritmo base para outras RNAs [43]. Já o modelo recorrente possui uma arquitetura geral parecida com a *feedforward*, mas diferencia na estrutura do neurônio caracterizando-se por executar uma tarefa pra cada elemento de uma sequência, com a sua saída sendo dependente de cálculos das entradas anteriores, sendo principalmente utilizado em aplicações para processamento de linguagem natural [47]. Com um destaque maior para processamento de imagens e vídeos, as redes neurais convolucionais diferenciam por aceitar uma imagem de entrada, podendo captar características únicas, e por assim ser possível diferenciá-las, frequentemente visto no ramo de aprendizado profundo [48]. A Seção **2.3.1** apresenta o embasamento teórico do modelo *feedforward*, uma vez que é o modelo adotado neste trabalho.

2.3.1 Modelo *Feedforward*

As RNAs surgiram na década de 50, e logo após, os primeiros *Perceptrons* fizeram sua aparição tendo o seu foco, a princípio, direcionado a solução de problemas para classificação binária. A sua estrutura fazia uso de um processamento com camada única com conjunto com uma função limiar [49]. Em função do que já foi visto anteriormente, o modelo *Feedforward*, como o próprio nome evidencia, traz o fluxo de processamento dos dados em um único sentido, levando em consideração a estrutura do neurônio e a equação que o descreve em Figura 3 e Equação 2.4, respectivamente. Uma vez que entra na camada de entrada, o treinamento ocorre passando por todas as camadas ocultas (caso exista) realizando os cálculos necessários, e finalmente, finaliza quando chega no neurônio da camada de saída [50].

As Figuras 4 e 5 exemplificam estruturas de redes neurais *feedforward*. Na Figura 4 destaca-se a arquitetura com 1 camada oculta, mas podendo ter diversas camadas ocultas a mais e 2 neurônios na camada de saída, e diferente dessa arquitetura, na Figura 5 temos

Figura 4 – Exemplo - Arquitetura de uma rede neural *Feedforward* com 1 camada oculta e duas saídas.



Fonte: Adaptado de [50]

uma rede neural sem camada oculta, porém possuindo 4 neurônios na camada de saída.

2.3.2 Função de Ativação

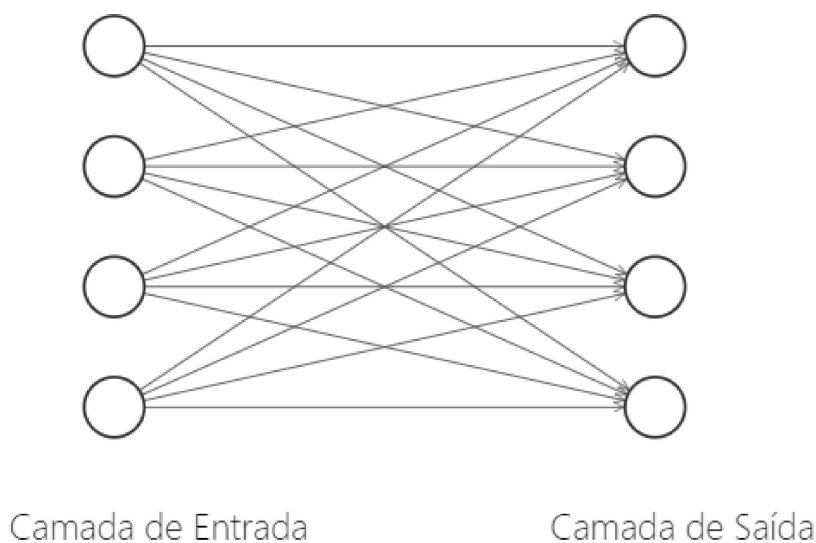
No contexto de redes neurais, a função de ativação possui um papel importante, além de indicar quando um neurônio está ativo, também influencia na convergência da rede neural [51]. Como pode ser visto na Equação 2.4, a função f tem o objetivo de transformar a saída de um neurônio para um novo intervalo, logo a escolha dessa função pode variar dependendo do tipo de problema a se lidar.

Para os experimentos deste trabalho foram selecionados duas funções de ativação com base no uso pelo estado da arte [52]. Tanto o comportamento dessas funções quanto de suas derivadas serão apresentadas nas Seções 2.3.2.1 e 2.3.2.2.

2.3.2.1 Logística

Também conhecida por função sigmóide, a função Logística possui dentre suas principais características a não-linearidade e destaca-se por sua curva assemelhar-se ao formato da letra S [53]. A utilização de funções lineares em RNA eram bem comuns no início, porém acabavam limitando o potencial de resolver problemas complexos. Com intuito de introduzir a não-linearidade as RNAs, o uso de funções não-lineares aumentou consideravelmente o seu poder de mapeamento em conjunto com seu processamento.

Figura 5 – Exemplo - Arquitetura de uma rede neural *Feedforward* sem camada oculta e com 4 saídas.



Fonte: Adaptado de [50]

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} \quad (2.6)$$

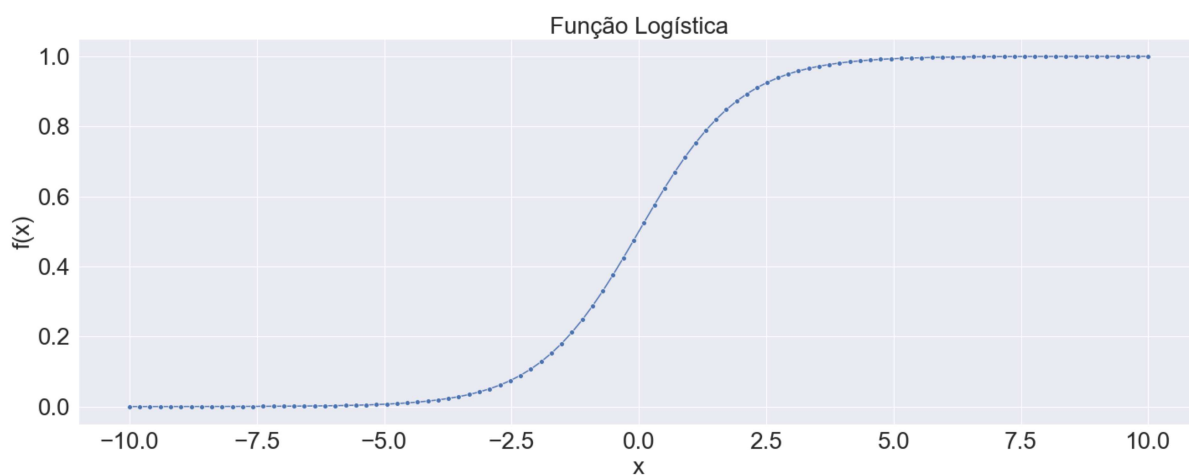
A definição da função logística e sua derivada podem ser visualizadas nas Equações 2.5 e 2.6, respectivamente. Pela Figura 6, destaca-se que para todo o domínio, a função é limitada no intervalo de $[0,1]$, e o valor de sua derivada não é constante como pode ser evidenciado na Figura 7.

Um dos problemas mais comuns que essa função apresenta é conhecido por desaparecimento do gradiente, ou do inglês, *Vanishing Gradient Problem*. Por utilizar a otimização *backpropagation*, o problema acontece pela condição da derivada poder alcançar valores cada vez menores, e por consequência dificultar no aprendizado dos neurônios [54].

2.3.2.2 ReLU

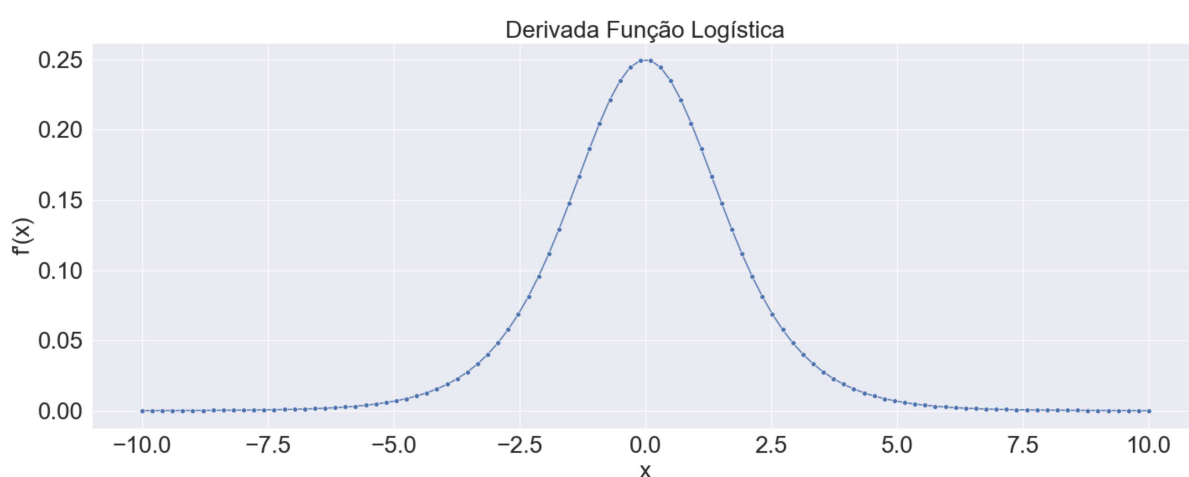
Apesar de parecer ter comportamento linear, a função *Rectified Linear Unit* (ReLU), por definição, é descrita como não-linear, e portanto também é utilizada como função de ativação em RNAs com objetivo de aumentar seu poder de processamento [55]. É caracterizada pelo seu simples cálculo como é destacado nas Equações 2.7 e 2.8, levando a

Figura 6 – Gráfico da função logística.



Fonte: Elaborado pelo autor. (2022)

Figura 7 – Gráfico da derivada da função logística.



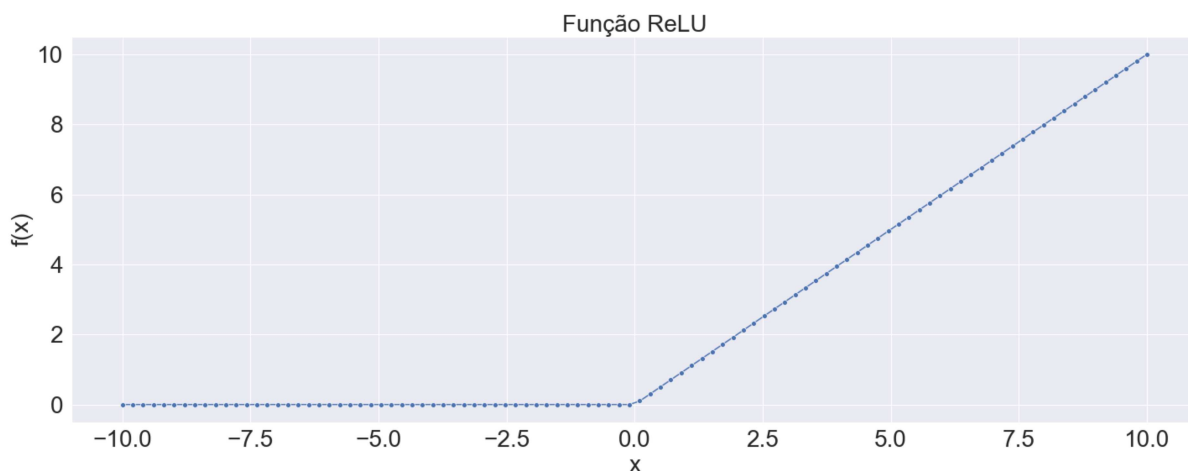
Fonte: Elaborado pelo autor. (2022)

rede neural a processar mais rapidamente. Na atualidade, é aplicada frequentemente e apresenta um bom desempenho em grandes áreas como é em *Deep Learning* [56].

$$f(x) = \max(0, x) \quad (2.7)$$

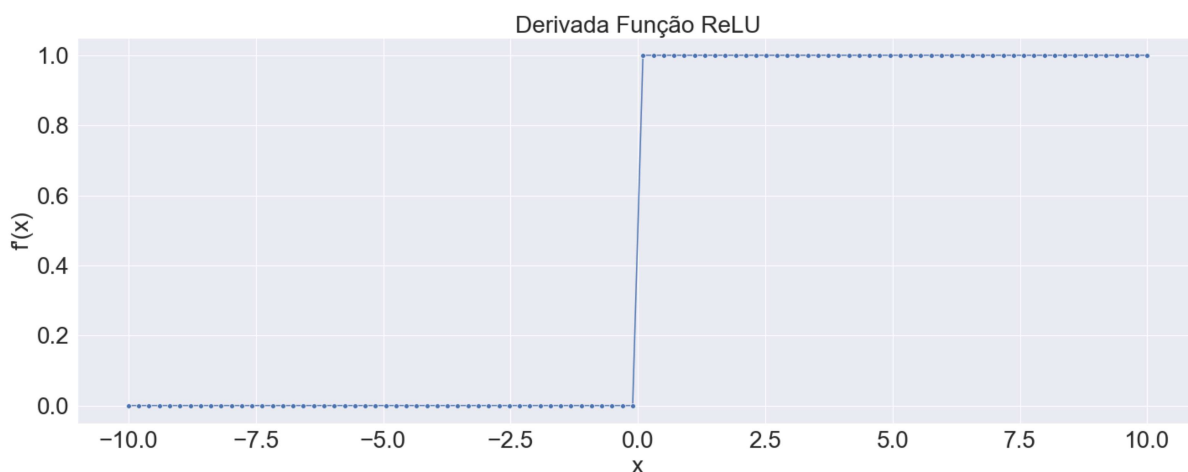
$$f'(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases} \quad (2.8)$$

Figura 8 – Gráfico da função ReLU.



Fonte: Elaborado pelo autor. (2022)

Figura 9 – Gráfico da derivada da função ReLU.



Fonte: Elaborado pelo autor. (2022)

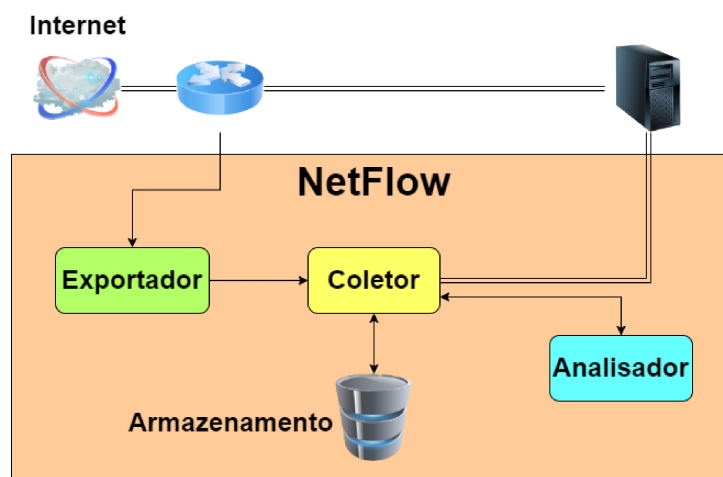
Diferente da função logística, a ReLU destaca-se por não apresentar o problema do desaparecimento do gradiente. Entretanto, surge um outro problema, para valores negativos, o resultado é sempre 0. Essa condição traz como consequência neurônios que ao alcançar esse valor dificilmente conseguirão se recuperar, esse problema é comumente denominado de *Dying ReLU*. Outro ponto a se evidenciar é que, por definição, $f'(0) = 0$. Tendo em vista isso, na Figura 9 o fator de continuidade pode ser visualizado.

2.4 GERENCIA DE REDES

O aumento de dispositivos com acesso a internet, e por consequência o acréscimo de usuários conectados na rede demarca a grande importância de se manter um ótimo gerenciamento de seus componentes. Entretanto, o gerenciamento de redes de forma geral engloba diversas dificuldades, e por muitas vezes, isso pode depender do formato de sua arquitetura (*i.e.*, centralizado, distribuído ou cooperativo) [57]. Por conta disso, obter um conhecimento prévio de toda a sua arquitetura é essencial para tomar decisões, e assim alcançar um gerenciamento satisfatório.

No presente momento, gerência à redes possui como objetivo garantir que a infraestrutura da rede funcione de forma eficiente e sem acarretar em problemas. E para garantir isso, vários procedimentos são necessários como: manutenção, automação, segurança, etc [58]. E por conta desses procedimentos que é necessário administrar os recursos disponíveis pela rede com eficácia. Dentre os desafios propostos nesse tema, uma das possíveis soluções que favoreceu essa área foi o uso de Redes Definidas por *Software* (SDN - *Software-Defined Networking*) para auxiliar na gerência de redes [59]. O paradigma SDN revolucionou a área de redes permitindo que fosse possível separar o plano de dados e o de controle, deste modo, o plano de dados fica responsável para lidar com o encaminhamento dos pacotes pelos *switches* da rede, e em formato centralizado, um *software* controla o comportamento da rede [60].

Figura 10 – Representação do monitoramento de fluxo com NetFlow.



Fonte: Adaptado de [3],

Outra forma que possibilita auxiliar no gerenciamento da rede é através de análises e monitoramentos. Para realizar tais ações, é necessário realizar a gestão de todo o tráfego da rede e do fluxo de dados. Essa gestão é importante para principalmente facilitar na extração das informações da rede de forma segura tanto para a sua arquitetura quanto para os usuários que as utilizam. Para assistir a coleta dessas informações, existem muitas

abordagens que tornam possíveis essas propostas [3]. Um exemplo é o protocolo de rede *NetFlow*, originado pela Cisco com intuito de proporcionar análises e coletas sobre o tráfego da rede [61]. Na Figura 10 é representado uma configuração do funcionamento do NetFlow para coletar, monitorar e analisar o fluxo. Intuitivamente, o Exportador fica responsável por agregar os dados e exporta os registros para os coletores. Já o Coletor realiza o pré-processamento e armazenamento dos dados, e por fim, o Analisador realiza ações baseado no contexto das informações que foram extraídas dos dados coletados.

Analogamente ao módulo Analisador, ao usar técnicas de aprendizado de máquina em conjunto com um Controlador que permita aplicar as decisões tomadas é uma outra maneira de otimizar o gerenciamento de recursos em rede. Através da grande quantidade de *logs* que são gerados diariamente devido a ampla distribuição de pontos de acesso (APs) em redes sem fio, é possível extrair informações úteis para análises. Como é evidenciado em [4], existem inúmeras aplicações que relacionam aprendizado de máquina com redes sem fio, tais quais envolvendo monitoramento e alocação de usuários em APs, estudo do comportamento do usuário, caracterização e análise das redes sem fio da próxima geração, e etc.

3 TRABALHOS RELACIONADOS

Neste capítulo são discutidos os trabalhos relacionados mais recentes com maior destaque na literatura. Na Seção 3.1 é focado na abordagem de predição da mobilidade dos usuários, e na Seção 3.2 é discutido sobre a abordagem de predição do volume de usuários conectados na rede sem fio.

3.1 PREDIÇÃO DA MOBILIDADE DOS USUÁRIOS

O aumento do volume de dados e dispositivos móveis trouxeram diversos desafios para o gerenciamento e otimização do tráfego de rede [62]. Uma das possíveis soluções é conseguir prever o comportamento da rede de modo a entender seus padrões, e assim tomar decisões importantes previamente. Com a ideia de tentar solucionar esses problemas, a aplicação de diferentes técnicas de aprendizado de máquina são propostas. Direcionado a abordagem de mobilidade dos usuários nas redes sem fio, os autores em [63] se utilizaram do Modelo de Markov Oculto (*Hidden Markov Model* - HMM) para predição, entretanto concluíram que a melhor escolha seria o AP com sinal mais forte, reduzindo o alto consumo de energia, porém aumentando o número médio de handoffs necessário, garantindo assim melhor QoS e QoE de acordo com os resultados apresentados. A aplicação do modelo HMM para esse tema em específico se estende a trabalhos a mais de uma década como em [2], onde o modelo apresentou bons resultados, variando de 60% a 90% de acurácia considerando 4 APs na trajetória, contudo o mesmo decaía em desempenho quando considerado uma trajetória do usuário com mais de 4 APs. Essa proposta de melhorar a experiência do usuário, e possivelmente reduzir a quantidade de *handovers* desnecessários segue em [64], onde novamente é aplicado a técnica HMM que alcançou uma acurácia de 80% nas simulações realizadas. Chegando a trabalhos mais recentes onde estudos demonstraram que métodos matemáticos desempenham melhor em ambientes abertos, e tendo resultado oposto em ambientes fechados [65]. Como proposta, o uso de HMM em conjunto com *K-means* para agrupar as trajetórias dos usuários foi sugerida, resultando em um método eficiente com acurácia melhor do que os modelos anteriores. Outro trabalho relevante é [66], onde é feito um estudo da previsibilidade do HMM de modo a analisar o seu limite teórico. Considerando um campus como cenário, com aproximadamente 7400 usuários, os resultados estimam que a previsibilidade máxima alcançada é de 82%, e o melhor modelo de Markov possui, em média, 67% de acurácia.

Ainda sobre predição de mobilidade dos usuários, em [67] foi proposto um modelo onde os autores baseiam-se em trabalhos anteriores sobre técnicas de agrupamento e sequência de padrões [68, 69, 70]. Sendo assim, dois modelos foram propostos. Um modelo *Clustering and Sequential Pattern Mining* (CSPM) que é para agrupar e remover movimentos aleatórios, e em seguida, ser feita a predição. E um segundo modelo,

denominado *Sequential Pattern Mining and Clustering* (SPMC), onde é feito a remoção destes movimentos aleatórios antes do agrupamento. Os resultados alcançados mostraram que o modelo CSPM atingiu melhor resultado, utilizando da métrica *Recall* (métrica que descreve dentro de uma classe, a porcentagem de predições corretas obtidas), atingiu 100% durante todos os testes realizados. De forma distinta, o modelo SPMC decaiu seu desempenho de acordo que uma maior mobilidade aleatória era percebida. Todavia vale ressaltar que a validação do modelo é realizada apenas com dados gerados por simulações, ao invés de dados reais.

O rápido crescimento do tráfego de dados implica importantes aplicações em redes 5G como alocação de recurso, gerenciamento da mobilidade e a alta frequência de *handovers*. Em [71], foi proposto o uso da Rede Neural Artificial (RNA) *Long Short Term Memory* (LSTM) para predizer a localização de acordo com estação base atual e a estação base vizinha. Os testes foram realizados em um conjunto de dados reais com diversas trajetórias de GPS. Os resultados mostraram que o modelo LSTM apresentou bom desempenho alcançando acurácia maior que 95%. Tendo como principal alvo serviços baseados em localização e sistemas de recomendação, é destacada a importância dessa aplicação em [72]. E para tal feito, foi comparado o desempenho de técnicas como RNAs, K-vizinhos mais próximos (*K-Nearest Neighbors* - KNN), Máquina de Vetor Suporte (*Support Vector Machine* - SVM) e Árvore de Decisão. Em geral, o modelo que se destacou foi o K-vizinhos mais próximos ponderado, alcançando uma acurácia de 91.98%, superando métodos similares que utilizaram a mesma base de dados em 2.72%.

Diferente dos trabalhos já mencionados, a predição de mobilidade aqui é destacada pela distância percorrida pelos usuários. Em trabalhos como [73], o modelo LSTM é usado para melhorar a alocação de recursos da rede. Baseado em dados simulados com 1000 trajetórias, o desempenho do LSTM obteve 90% dos erros de predições abaixo de 6 metros, superando técnicas similares. [74] segue essa mesma ideia de aplicação também utilizando o modelo LSTM. Através dos resultados simulados, foi possível perceber que o tempo de execução decaiu de 4616s para 932s no melhor caso, e alcançou como métrica um valor de perda de 0.10 em relação a distância. Considerando também o aumento de produtos direcionados a Internet das Coisas (*Internet of Things* - IoT), autores em [75] propuseram utilizar Análises de Componentes Principais (*Principal Component Analysis* - PCA) juntamente as redes neurais LSTM, *Gated Recurrent Unit* (GRU) e o Perceptron de Múltiplas Camadas (*MultiLayer Perceptron* - MLP). Foi destacado à partir dos resultados apresentados que a junção PCA-GRU obteve um desempenho melhor que técnicas similares como PCA-MLP e PCA-LSTM, alcançando uma média de erros menor que 20m.

3.2 PREDIÇÃO DO VOLUME DE USUÁRIOS

Para a abordagem de predição de carga dos usuários na rede sem fio, uma das formas de lidar com esse desafio é encontrar uma forma autônoma para gerenciar o congestionamento da rede. Uma solução interessante é aplicar os conceitos de *Software-Defined Networking* (SDN), que isolam naturalmente os planos de controle e dados, a fim de oferecer suporte ao provisionamento rápido de serviços, e minimizar as restrições de *hardware* [62]. Por exemplo, a proposta de balanceamento de carga apresentado por [76], determina, com base nas informações de carga da rede e topologia, quais roteadores devem ser mantidos ligados ou não, melhorando assim a gestão dos recursos.

Aplicar Inteligência Artificial (IA) é uma forma promissora para gerir a carga de redes sem fio [77]. Tanto redes neurais (NN) quanto modelos baseado em árvore de decisão fornecem melhores resultados que os modelos *Naive Bayes* (NB), *Support Vector Machines* (SVM), *Random Forest* (RF), e *k Nearest Neighbors* (kNN) em termos de precisão e predição, conforme [78]. Em particular, os modelos baseados em árvore de decisão representam uma solução atraente para análise de rede sem fio, principalmente, devido ao seu baixo custo computacional. Nesta linha, [79] propõem o uso de uma Rede Neural de *Backpropagation* (BPNN) após uma fase de pré-processamento otimizada aplicando *denoising wavelet*. O erro médio absoluto (MAE) alcançado foi menor do que o modelo *Autoregressive-moving-average* (ARMA). Por sua vez, o Aprendizado de Máquina (*Machine Learning* ou ML) oferece suporte mesmo quando é necessário caracterizar ou inferir os estados do protocolo da camada de aplicação por meio de fluxos diversos de tráfego de rede sem fio intercalados como pode ser visto em [80]. Os trabalhos citados deixam claro os benefícios da utilização de técnicas de IA como auxiliar na gestão da rede de comunicação.

Outros resultados relevantes também são encontrados na literatura ao aplicar-se *Deep Learning* na predição de tráfego de rede sem fio. Em [81] foi proposta uma rede convolucional espaço-temporal (LA-ResNet), combinando rede residual, Rede Neural Recorrente (RNN) e um mecanismo de atenção. A rede residual é usada para modelar as características espaciais dos dados de sequência. Em contraste, a RNN e um mecanismo de atenção são combinados para resolver a modelagem espaço-temporal e prever o tráfego de rede sem fio. O RNN também foi aplicado junto com a decomposição do tensor Bayesiano Gaussiano para construir um modelo de predição de tráfego de rede sem fio. O algoritmo, chamado BGCP-RNN-ReLU, pode prever efetivamente as mudanças no tráfego de rede tanto *upstream*, quanto *downstream*, para um futuro próximo [82]. Por si só, destaca-se que uma RNN simples pode obter melhor desempenho de predição do que outras NN complexas. Mas, ainda assim, a RNN *Long-Short Term Memory* (LSTM) foi comparada com o modelo *Gated Recurrent Unit* (GRU) para prever o tráfego de rede sem fio e os dados de utilização do canal [83]. Como resultado, concluiu-se que modelos de ML são adequados para previsões de média escala. Já para prever dados de larga escala, *HoltWinters* e a

Média Móvel Integrada Sazonal Autorregressiva (SARIMA) se destacam, principalmente por conta da sua insensibilidade a valores discrepantes do conjunto de dados.

Em [84] foi proposta uma solução para o problema de séries temporais multivariadas (MTS) também para prever o tráfego de rede sem fio. Em primeiro lugar, foi construída uma abordagem de correlação espaço-temporal por meio do método *Detrended Partial Cross Correlation Analysis* (DPCCA) para extração de características, e após isso, aplicado um modelo RNN para prever o tráfego de redes sem fio. A abordagem oferece melhor desempenho do que um modelo *Generalized Spatio-Temporal Auto-Regressive* (GSTAR), uma vez que não envolve preditores externos. Por fim, uma BPNN também foi avaliada como arquitetura básica de um modelo de predição de tráfego de rede sem fio. Na qual uma análise, aplicando o método *wavelet* de multi-resolução, compreendeu o pré-processamento de dados, e um algoritmo aprimorado *Quantum Particle Swarm Optimization* (ALQPSO) como otimização [85]. Comparado com uma Rede Neural BP combinado à otimização por Algoritmo Genético (GA-BPNN) e uma Rede Neural BP combinado à otimização por Enxame de Partículas Quânticas (QPSO-BPNN), seu valor de erro é menor e, portanto, apresenta uma predição mais precisa.

3.3 APRENDIZADO DE MÁQUINA AUTOMATIZADO

Ao tentar relacionar o processo de AutoML com os temas abordados nesse trabalho, não foi encontrado no estado da arte trabalhos que o abordam até o presente momento dessa pesquisa. E ao estender a busca aos que relacionam AutoML com aplicações em redes sem fio, poucos trabalhos são retornados.

Dentre as possíveis aplicações encontradas na literatura, destaca-se o trabalho de [86] que propuseram usar AutoML para auxiliar em um sistema de detecção de intrusão (IDS) em redes sem fio. Com intuito de minimizar a taxa de falso alarme, um *framework* de AutoML foi utilizado para otimizar, sendo validado em 2 conjuntos de dados para *benchmark*. Quando comparado com modelos convencionais, o modelo do AutoML produziu uma melhor taxa de detecção e maior acurácia. Neste mesmo tema, foi proposto em [87] utilizar o processo de AutoML para agilizar o processo de otimização dos modelos com intuito de prevenir e detectar intrusão em uma rede de sensores sem fio de forma rápida. Por esse estudo foi identificado que o modelo obtido pelo AutoML superou diversos outros modelos baseados em aprendizado profundo de trabalhos anteriores.

Outro exemplo de aplicação é em redes IoT (*Internet of Things*), os autores em [88] propuseram avaliar a robustez do modelo AutoML quando aplicado aos dados desses sensores, que por sua vez, são classificados como dados sensíveis. O principal objetivo desse estudo é avaliar como os modelos desempenham quando os dados possuem alguns problemas. Os resultados demonstram um bom desempenho de 3 sistemas de AutoML utilizando dados reais de temperatura. Continuando em redes IoT, mas utilizando modelos de

aprendizado profundo, é proposto o uso de um otimizador de hiper-parâmetros, nesse caso, o Algoritmo Genético Paralelo (PGA) junto ao processo de AutoML [89]. Comparando o método proposto com outros métodos convencionais, foi provado que a utilização da otimização PGA com AutoML obteve melhores resultados em relação a tempo e predição dos resultados. Ainda em aprendizado profundo, mas aplicando AutoML para classificação de sinal sem fio e analisando suas vulnerabilidades, o trabalho proposto em [90] possui como objetivo reduzir o tempo gasto para treinar e otimizar os hiper-parâmetros. Ao fim é concluído que a abordagem foi eficaz para o dado problema.

3.4 CONSIDERAÇÕES DO CAPÍTULO

Diferentemente dos trabalhos destacados na Tabela 1, nesta pesquisa é proposto uma comparação entre modelos convencionais (rede neural e árvore de decisão otimizados por *grid-search*) e o processo de AutoML. Por fim, os 3 modelos foram aplicados na predição do próximo AP que o usuário irá se conectar e o volume de usuários conectados na rede sem fio. Outro diferencial é a análise dentro de um cenário de dados reais para verificação da qualidade de experiência do usuário em relação ao número de desconexões desnecessárias, como também da economia na alocação de largura de banda disponível e do espectro sem fio, de forma que mantenha um SLA contratado, mas reduza o número de recursos/APs necessários para isso.

Tabela 1 – Resumo dos trabalhos relacionados escolhidos.

Trabalho	Ano	Tema	Técnica	Base de Dados
[63]	2017	Mobilidade do Usuário	HMM	Dados sintéticos
[64]	2018	Mobilidade do Usuário	HMM	Dados sintéticos
[65]	2022	Mobilidade do Usuário	HMM	Dados reais
[66]	2018	Mobilidade do Usuário	HMM	Dados reais
[67]	2015	Mobilidade do Usuário	CSPM e SPMC	Dados sintéticos
[71]	2021	Mobilidade do Usuário	LSTM	Dados reais
[72]	2021	Mobilidade do Usuário	NN, KNN, SVM e DT	Dados reais
[73]	2018	Mobilidade do Usuário	LSTM	Dados sintéticos
[74]	2020	Mobilidade do Usuário	LSTM	Dados sintéticos
[75]	2019	Mobilidade do Usuário	MLP, LSTM e GRU	Dados reais
[79]	2020	Tráfego de Rede sem fio	BPNN	Dados reais
[81]	2020	Tráfego de Rede sem fio	LA-ResNet	Dados reais
[82]	2021	Tráfego de Rede sem fio	BGCP-RNN-ReLU	Dados reais
[83]	2021	Tráfego de Rede sem fio	LSTM e GRU	Dados reais
[84]	2020	Tráfego de Rede sem fio	RNN	Dados reais
[85]	2020	Tráfego de Rede sem fio	BPNN	Dados sintéticos
[86]	2023	Detecção de Intrusão	AutoML	Dados híbridos
[88]	2021	Predição de Séries Temporais	AutoML	Dados reais
[90]	2022	Classificação de Sinal Sem Fio	AutoML	Dados reais
[89]	2022	Predição de Séries Temporais	AutoML	Dados reais
[87]	2022	Detecção de Intrusão	AutoML	Dados híbridos
Frank <i>et al</i>	2023	Mobilidade do Usuário, Tráfego de Rede sem fio	MLP, DT e AutoML	Dois conjuntos de dados reais

Fonte: Elaborado pelo autor (2023).

4 MÉTODOS PROPOSTOS

Como já mencionado durante o Capítulo 2, onde foi destacada a estrutura que foi considerada para este trabalho. Na Figura 11 é mostrada parte dessa estrutura destacando-se o módulo Analisador (indicado pela seta laranja) que é o foco de estudo desta pesquisa onde a proposta que segue faz parte desse módulo.

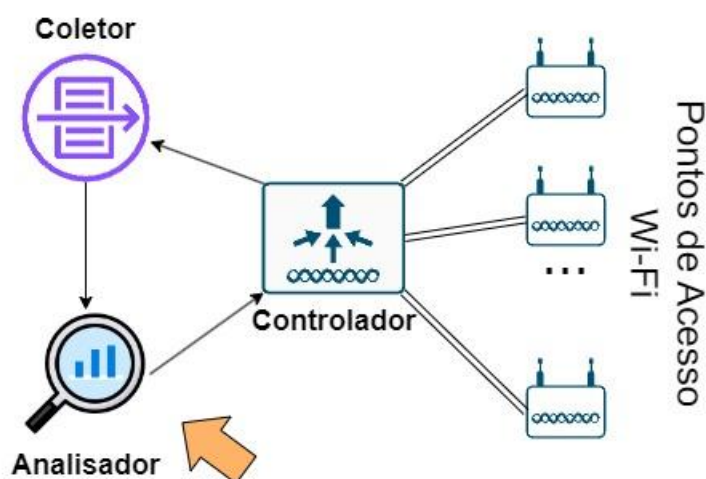


Figura 11 – Visão geral da proposta.

Fonte: Elaborado pelo autor (2022).

Neste capítulo é detalhada a proposta deste trabalho. Com objetivo de aplicar técnicas de aprendizado de máquina à área de gerência de redes sem fio, foram propostas duas abordagens que tem como intuito utilizar modelos preditivos supervisionados para prever:

- I. a mobilidade dos usuários conectados; e
- II. o volume de usuários conectados em um intervalo de tempo.

A Figura 12 demonstra todo o processo desde o pré-processamento até a avaliação dos modelos preditivos. A etapa de pré-processamento dessas abordagens engloba técnicas de seleção, remoção e criação de atributos para melhorar o modelo, já o treinamento do modelo será feito a partir da utilização da técnica de validação cruzada, e por fim, os modelos serão devidamente avaliados de acordo com as métricas selecionadas. De agora em diante, os itens I e II serão mencionados como abordagens I e II, nesta ordem.

Nas Seções 4.1 e 4.2 referente às abordagens I e II, respectivamente, serão fornecidos mais detalhes sobre o pré-processamento utilizado em cada uma dessas.

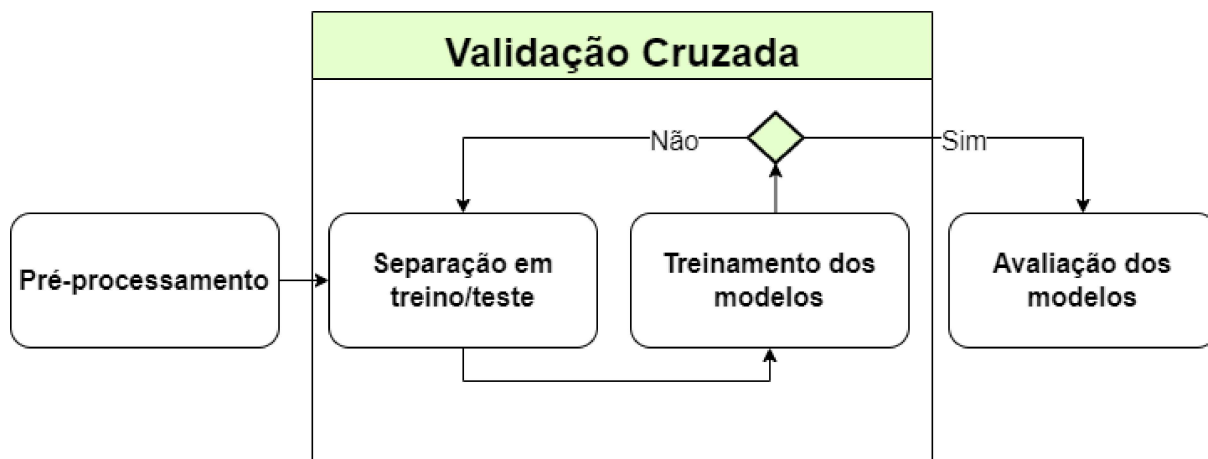


Figura 12 – Fluxograma que representa ambas as abordagens.

Fonte: Elaborado pelo autor (2022).

4.1 ABORDAGEM I

Uma das abordagens deste trabalho é analisar e avaliar o ganho em relação ao *handoff* para usuários em um cenário real quando aplicada a predição de mobilidade (próximo AP ao qual o usuário irá se conectar). Para isso, será feito o uso de 3 estratégias para que possa ser comparado o desempenho dessa predição. Com o foco em classificação, as técnicas escolhidas foram um modelo de rede neural MLP, uma árvore de decisão e o processo AutoML.

4.1.1 Pré-processamento da Abordagem I

O pré-processamento se destaca por ser a etapa onde a devida seleção das características se mostra importante para obter um modelo preditivo com boas métricas. E para selecionarmos essas trajetórias de forma eficaz, foi necessário definir o tamanho mínimo de uma trajetória, nesse caso esse tamanho mínimo foi definido como N . E um outro fator foi definir um tempo adequado no qual seria o valor limitante para ainda considerar que um usuário está se movimentando em uma mesma trajetória. Nesse caso, foi definido esse tempo limite de $t = 60$ em segundos baseado em testes iniciais.

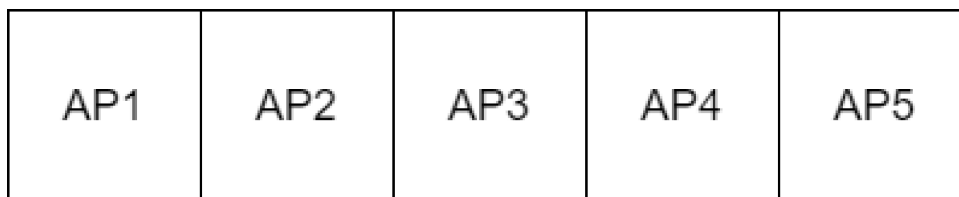


Figura 13 – Trajetória com 5 pontos de acesso.

Fonte: Elaborado pelo autor (2022).

Figura 13 mostra como seria a trajetória de um usuário qualquer, considerando $N = 5$. Vale ressaltar que como estamos interessados na mobilidade do usuário, então um mesmo AP não pode aparecer duas ou mais vezes consecutivamente. Outro ponto importante é que ao selecionar um determinado valor para N , filtramos a base de dados de forma a remover dados não muito significativos como associações em um único AP, associações repetidas e dados faltantes.

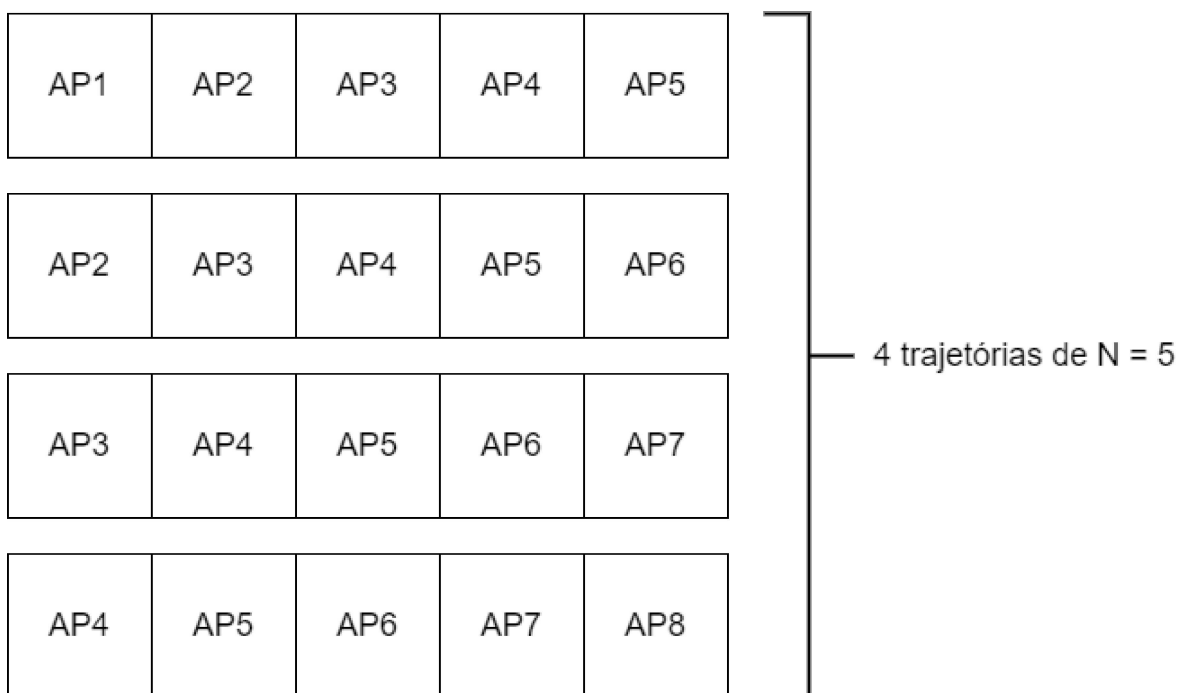


Figura 14 – 4 trajetórias de tamanho $N = 5$.

Fonte: Elaborado pelo autor (2022).

Uma outra consideração foi com as trajetórias existentes com tamanho maior que N . Nesse caso, essas trajetórias são divididas para também serem consideradas no treinamento, mas sempre respeitando a ordem que os APs foram colocados. Definindo M como sendo a quantidade de APs em uma trajetória onde que $M > N$. Exemplificando, se possui $M = 8$ e $N = 5$, dentro dessa trajetória, tem-se $M - N + 1$ trajetórias de tamanho N . A Figura 14 demonstra o resultado após extrair 4 trajetórias de uma trajetória.

Tabela 2 – Exemplo do *dataframe* para a abordagem I.

AP1_feature	AP2_feature	AP3_feature	AP4_feature	AP5_feature
AP1	AP2	AP3	AP4	AP5
AP2	AP3	AP4	AP5	AP6
AP3	AP4	AP5	AP6	AP7
AP4	AP5	AP6	AP7	AP8

Fonte: Elaborado pelo autor (2022).

A Tabela 2 exemplifica como ficaria o *dataframe* de acordo com a Figura 14. A nomenclatura de cada coluna apenas indica a ordem dos APs considerados na trajetória. O último AP da trajetória é o AP alvo que queremos prever, neste caso, a coluna *AP5_feature*. Para este trabalho, o valor de N que resultou nos melhores resultados foi de 12 para os dois conjuntos de dados após testes iniciais.

4.2 ABORDAGEM II

A segunda abordagem deste trabalho tem como objetivo prever o volume, isto é, a quantidade de usuários conectados na rede sem fio dado um determinado intervalo de tempo. Com um modelo de confiança e a predição realizada, alocações de largura de banda e pontos de acesso podem ser feitas para gerenciar os recursos da melhor forma possível. Com uma estratégia bem similar a abordagem I, as técnicas MLP, árvore de decisão e o processo autoML foram adotadas, o que difere nesse caso é por se tratar de um problema de regressão.

4.2.1 Pré-processamento da Abordagem II

Para lidar com a predição do volume de usuários conectados em um determinado intervalo de tempo é necessário, primeiramente, agrupar os dados em janelas de t minutos, e para este trabalho foi escolhido o valor de t igual 5. Uma vez que tem-se os registros agrupados nesses intervalos, é possível saber a quantidade de usuários que se conectou. Porém, prever o volume de usuários no futuro utilizando somente o volume atual não demonstra ser muito útil. Portanto, a criação de novos atributos é importante para aperfeiçoar o modelo. A partir disso, duas variáveis são definidas: P que representa o número de observações recentes e Q que representa o número de observações históricas. Nesse caso, entende-se observações sendo cada janela contendo o número de usuários conectados.

$$x_1 = \{V_t, V_{t-1}, \dots, V_{t-P}\} \cup \{V_{t+1-(1 \text{ dia})}, V_{t+1-(2 \text{ dias})}, \dots, V_{t+1-(Q \text{ dias})}\} \quad (4.1)$$

A Equação 4.1 descreve o vetor de características utilizado nessa proposta, onde V_t simboliza o volume de usuários no instante t , e V_{t-1} a V_{t-P} caracteriza o volume de usuários em P intervalos recentes anterior ao instante t . Já $V_{t+1-(1 \text{ dia})}$ a $V_{t+1-(Q \text{ dias})}$ retrata o volume dos usuários no instante $t + 1$, porém até Q dias anteriores.

Apenas para exemplificação dos atributos utilizados, a Tabela 3 demonstra como fica o *dataframe* após os dados serem pré-processados. Nessa situação, cada linha da tabela representa um dia, sendo então uma sequência de 3 dias, e os valores de P e Q são iguais a 1 e 2, respectivamente. Desse modo, durante o pré-processamento, o valor da coluna V_{t+1} é distribuído para as colunas $V_{t+1-(1 \text{ dia})}$ e $V_{t+1-(2 \text{ dias})}$, sempre respeitando

Tabela 3 – Exemplo do *dataframe* para a abordagem II.

	V_t	V_{t-1}	$V_{t+1-(1 \text{ dia})}$	$V_{t+1-(2 \text{ dias})}$	V_{t+1}
linha 1	a	d	-	-	y
linha 2	b	e	y	-	x
linha 3	c	f	x	y	z

Fonte: Elaborado pelo autor (2022).

a ordem dos dias. Ao fim, somente as linhas completas, ou seja, que possuem todas as colunas preenchidas permanecem para treinar o modelo, diante disso, somente a linha 3 permanece. Para essa abordagem, a coluna V_{t+1} corresponde à variável alvo a ser predita pelo modelo, e os valores de P e Q escolhidos que melhor se adaptaram após testes iniciais foram de 11 e 3, nessa ordem.

4.3 MODELOS

Nessa seção, é detalhada a arquitetura dos modelos escolhidos para esse trabalho. A seleção desses foi motivada pela sua técnica de predição simples, porém eficiente. Como já mencionado, algumas dessas técnicas são largamente utilizadas pela literatura, e por isso, é interessante realizar uma comparação entre elas.

4.3.1 MLP

O MLP é uma famosa RNA que já demonstrou sucesso na predição de cenários cíclicos no passado [91, 92]. Sua arquitetura contém uma camada de entrada, uma ou mais camadas ocultas¹, e uma camada de saída. Pela predição ser uma resposta unidimensional, ou seja, representa a predição do ponto de acesso de conexão do usuário ou a predição do volume do usuário, a camada de saída contém apenas um neurônio.

As Figuras 15 e 16 ilustram as arquiteturas dos modelos utilizados na abordagens I e II, nessa ordem. A arquitetura é composta por: N sendo o número de neurônios na camada oculta e M sendo o número de neurônios na camada de entrada (M depende do número de atributos, logo dependerá de qual abordagem está sendo aplicado). Os hiperparâmetros N e taxa de aprendizagem são otimizados utilizando *grid-search*.

4.3.2 Árvore de Decisão

A árvore de decisão é um modelo de predição simples e facilmente interpretável [93]. Por conta disso, é considerado um dos algoritmos mais comuns usados entre todas as técnicas de ML. Para esse trabalho, o modelo é utilizado para ambas propostas. O que difere um do outro é a configuração inicial de alguns hiperparâmetros. No caso do modelo

¹ Para a arquitetura do nosso modelo consideramos três camadas ocultas para a abordagem I, e uma para abordagem II.

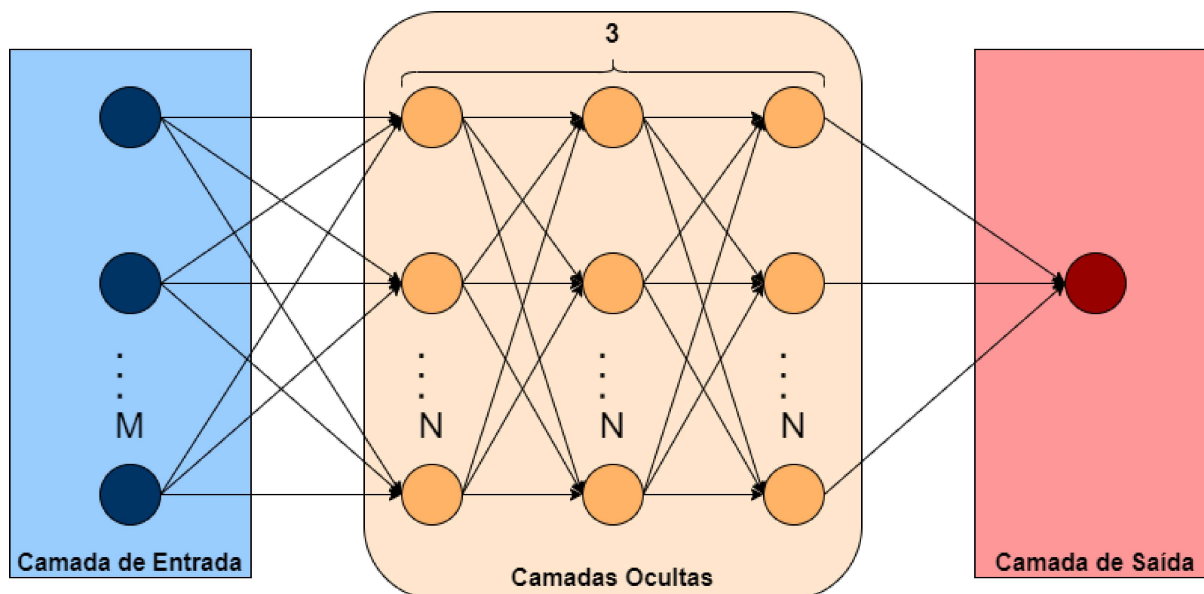


Figura 15 – Arquitetura MLP para a abordagem I.

Fonte: Elaborado pelo autor (2022).

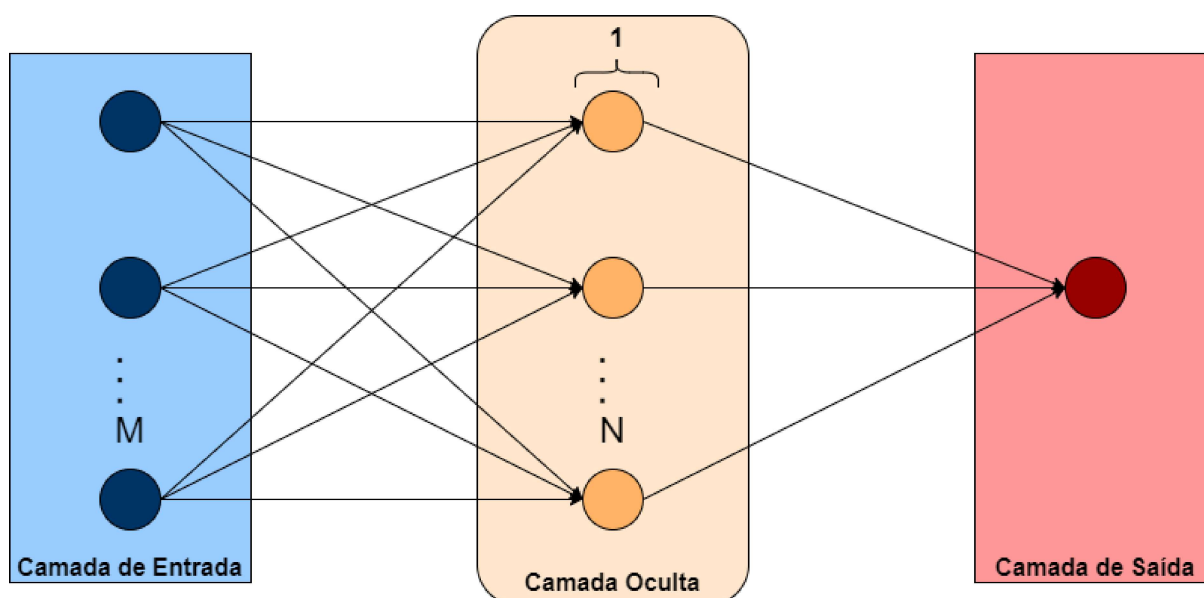


Figura 16 – Arquitetura MLP para a abordagem II.

Fonte: Elaborado pelo autor (2022).

de classificação, a escolha do hiperparâmetro de critério usado foi o Índice de Gini, e Erro Quadrático Médio de Friedman para o modelo de regressão. Semelhante ao que é feito para alguns hiperparâmetros do modelo MLP, a técnica de *grid-search* foi aproveitada para otimizar os hiperparâmetros profundidade máxima e número mínimo de amostras para cada nó.

4.3.3 Auto-sklearn

Para o ajuste dos modelos, o auto-sklearn se assemelha aos modelos já mencionados em termos de treinamento e avaliação, utilizando de um conjunto de dados X e Y para treino e teste. Evidentemente, a diferença, se deve pelo fato de que o sistema AutoML engloba vários modelos, e realiza vários treinamentos com objetivo de otimizar tanto os hiperparâmetros usados como também na escolha do algoritmo que melhor se encaixa ao problema em questão.

Por ser um técnica que geralmente consome bastante memória e leva um tempo considerável para executar, alguns parâmetros foram necessários ser manipulados para ser possível o treinamento. Um dos parâmetros importantes que possibilitou os testes foi aumentar o limite de memória para 16GB para que alguns algoritmos possam ser treinados. E também na tentativa de deixar a comparação mais justa, apenas modelos únicos foram usados.

4.4 CENÁRIO

Nesta seção é descrito os conjuntos de dados usados nos testes deste trabalho. Nas Seções 4.4.1 e 4.4.2 é exemplificado uma amostra dos dados, e destacado alguns fatores importantes sobre a caracterização de cada cenário.

4.4.1 Conjunto de Dados - UFJF

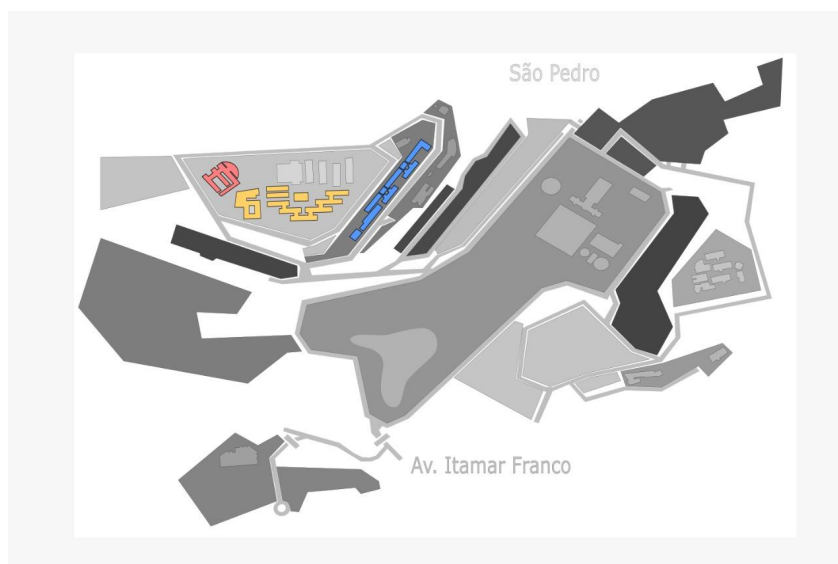


Figura 17 – Mapa do campus da UFJF.

Fonte: Elaborado pelo autor (2022).

O campus da Universidade Federal de Juiz de Fora (UFJF)², mostrado na Figura

² <https://www2.ufjf.br/ufjf/sobre/mapadocampus/>

17, possui diversos APs espalhados por toda sua extensão, fazendo a cobertura das áreas de diversos departamentos. Através destes APs, uma grande quantidade de dados (*logs*) são capturados pela controladora. A base de dados utilizada neste trabalho foi extraída destes *logs*. Considerando a LGPD (Lei Geral de Proteção de Dados Pessoais)³ e a possibilidade de haver diversas informações sensíveis que possibilitem correlacionar os dados aos usuários, foram ofuscados.

Tabela 4 – Exemplo da base de dados da UFJF.

timestamp	timet	client	AP
2019-09-02 10:56:01	1.0	1	27
2019-09-02 10:56:54	2.0	1	38
2019-09-02 11:07:37	2.0	1	35
2019-09-02 11:09:38	5.0	1	59
2019-09-02 11:23:26	3.0	1	21
2019-09-02 11:25:56	5.0	1	7
2019-09-02 11:26:30	2.0	1	11

Fonte: Elaborado pelo autor (2022).

Na Tabela 4, pode ser visto um fragmento da base de dados com as informações extraídas dos *logs* fornecidos. Na Tabela 5, estão descritas as colunas da base de dados do fragmento. As colunas suprimidas são referentes às ações que ocorreram durante a associação do usuário, o que não traz informações relevantes para o estudo feito. A Tabela 6 descreve o número de meses, de registros, de usuários e de pontos de acessos que o conjunto de dados possui. Os dados fornecidos são do mês de Junho a Setembro de 2019. Para a abordagem I, todo o conjunto de dados foi utilizado, porém para a abordagem II com intuito de facilitar os testes, o conjunto de dados foi filtrado pelos APs que se situam na região do restaurante universitário que somam um total de 7 APs.

Tabela 5 – Finalidade das colunas mais relevantes.

Atributo	Finalidade
timestamp	Indica o instante de tempo que ocorreu a associação
timet	Indica o intervalo de tempo que o usuário passou associado
client	Id do usuário
AP	Id do ponto de acesso

Fonte: Elaborado pelo autor (2022).

³ http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709compilado.htm

Tabela 6 – Resumo do conjunto de dados da UFJF.

# Meses	# Registros	# Usuários Únicos	# APs Únicos
4	16020374	36191	108

Fonte: Elaborado pelo autor (2022).

4.4.2 Conjunto de Dados - KTH

O conjunto de dados de redes sem fio do campus KTH, localizado na Suécia, obteve os registros no intervalo de 2014 a 2015 [94]. A Tabela 7 demonstra os dados extraídos. Diferente do conjunto de dados da UFJF, esse não oferece o atributo mostrando o intervalo de tempo que o usuário passou associado ao ponto de acesso. Portanto, exceto pela coluna *timet*, as outras colunas possuem as mesmas finalidades definidas da Tabela 5 referente a UFJF.

Tabela 7 – Exemplo da base de dados da Suécia.

timestamp	client	AP
2014-01-01 00:00:24	b7f22f3abd9f367af65bd70b4b328e0c35f7cfac	Bldg11AP21
2014-01-01 00:00:30	336501aeaedff0462f3cd55089fa433dbbfff493	Bldg44AP3
2014-01-01 00:00:35	4b912f490ddc1f289f35def68a1662bcc5089a1a	Bldg48AP65
2014-01-01 00:00:40	8e10f9d5aa9c229baac2f745abd25b9c9522ed64	Bldg25AP19
2014-01-01 00:00:51	680fc4c2bd6b252fd6e1a67ad7aa69cbbd3d6e0f	Bldg25AP5
2014-01-01 00:00:59	d87196dde7bad58154db0c0a70cfcb3ef143a774	Bldg14AP12
2014-01-01 00:01:05	980b004cef94929b5c0d7dde384fe3dc274f5437	Bldg16AP22

Fonte: Elaborado pelo autor (2022).

Tabela 8 – Resumo do conjunto de dados da KTH.

# Meses	# Registros	# Usuários Únicos	# APs Únicos
16	93310771	109197	1120

Fonte: Elaborado pelo autor (2022).

Com um total de 93310771 de registros, a Tabela 8 traz um resumo geral sobre a base de dados do campus KTH. Com 16 meses, tem-se dados de janeiro de 2014 até abril de 2015, e nesse intervalo foi encontrado um total de 109197 usuários e 1120 APs conhecidos distribuídos pelo campus. Para ambas as abordagens, devido a falta de conhecimento da região que situa-se o campus e da disposição geográfica dos APs, foi tomada a decisão de filtrar o conjunto de dados pelo termo "Bldg6A" que indica pelo prefixo "Bldg6" uma região de APs do edifício 6. Com isso, somou-se um total de 15 APs inicialmente. Ambos os conjuntos de dados foram divididos em 70% para treinamento e teste dos modelos utilizando a validação cruzada, e o restante para os testes realizados nos casos de uso.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo serão avaliados os modelos preditivos escolhidos de acordo com as métricas de avaliação selecionadas. Os testes foram executados utilizando a linguagem Python na versão 3.9.5 com o auxílio de bibliotecas como *scikit-learn* na versão 0.24.2, *auto-sklearn* versão 0.15.0 para os modelos e métricas, *pandas* na versão 1.5.3 para auxiliar na manipulação dos dados e *matplotlib* versão 3.6.3 para visualização. Para a validação cruzada foi utilizado o valor padrão de $k = 5$ [95]. Os testes foram realizados no sistema Ubuntu 22.04.

5.1 MÉTRICAS

Nesta seção são definidas as métricas utilizadas tanto para avaliar um problema de classificação (*i.e.*, acurácia, precisão, matriz de confusão, etc) quanto um problema de regressão (*i.e.*, coeficiente de determinação, erro absoluto médio, raiz do erro quadrático médio, etc).

5.1.1 Classificação

Além de optar por métricas convencionais quando um modelo de classificação é abordado, a escolha dessas métricas também se deve ao fato da base de dados possuir diversas classes, e isso gerar um desbalanceamento. Desse modo, é necessário utilizar métricas que considerem esse desbalanceamento ou auxiliem no cálculo do acerto das predições de cada classe.

Precisão: Usada para avaliar o desempenho de modelos de classificação, a precisão mede a proporção de predições positivas verdadeiras entre todas as predições positivas. Em outras palavras, ela nos diz quantos dos resultados positivos preditos são realmente positivos verdadeiros.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (5.1)$$

O cálculo desta métrica é descrita na Equação 5.1, onde VP são os Verdadeiros Positivos e FP os Falsos Positivos. Um alto valor na precisão indica que o modelo está fazendo poucas predições falsas positivas e está identificando corretamente a maioria das instâncias positivas. E o contrário pode ser dito quando a precisão é baixa, significando que o modelo está fazendo um grande número de predições falsas positivas e está classificando incorretamente muitas instâncias negativas como positivas.

Recall: Também comumente utilizada para avaliar modelos de classificação, seu objetivo é medir a proporção de predições positivas verdadeiras entre todas as instâncias positivas

reais. Ou seja, esta métrica informa quantas instâncias positivas no conjunto de dados o modelo é capaz de identificar corretamente.

$$\text{Recall} = \frac{VP}{VP + FN} \quad (5.2)$$

A Equação 5.2 descreve como é computado a métrica *recall*. Diferente da métrica precisão, tem-se a quantidade de Falsos Negativos (FN) no lugar dos falsos positivos. Portanto, um alto *recall* aponta que muitas instâncias positivas estão sendo preditas corretamente. E o oposto também vale para um baixo valor de *recall*.

F1: Possuindo uma grande utilidade quando o conjunto de dados está desequilibrado (número de instâncias de uma classe é muito maior que o da outra), a métrica F1 é aplicada para avaliar o desempenho do modelo realizando uma média harmônica entre precisão e *recall*. Na Equação 5.3 é definido como é calculado essa métrica.

$$\text{F1 score} = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (5.3)$$

Uma das vantagens dessa métrica é considerar de igual importância tanto a precisão quanto o *recall*, assim fornecendo uma medida equilibrada do desempenho do classificador. Modelos com alta pontuação de F1 indicam que possuem alta precisão e alto *recall*, sendo assim, capaz de identificar corretamente a maioria das instâncias positivas no conjunto de dados, reduzindo falsos positivos e falsos negativos.

Acurácia Ponderada: Usualmente é definida como a média da métrica *Recall* obtida em cada classe. Assim considerando um desempenho melhor quanto mais próximo de 100% e pior quanto mais próximo de 0%. Para o cálculo final da acurácia ponderada, uma média desse valor é realizado para cada classe.

AUC ROC: Sendo uma métrica de desempenho popular usada em tarefas de classificação para avaliar o poder preditivo de um modelo, a curva de Característica de Operação do Receptor (ROC - *Receiver Operating Characteristic*) demonstra a taxa de verdadeiro positivo e falso positivo. De modo a facilitar a interpretação dessa curva, é realizado o cálculo da área sob a curva (AUC - *Area Under the Curve*), assim resumindo a curva em um único valor agregando todos os seus limiares.

Por se tratar de um valor único, quanto mais próximo de 1, melhor é o desempenho do modelo preditivo, Por outro lado, um modelo que classifica de forma aleatória possui uma pontuação próxima de 0.5.

Matriz de Confusão: A Matriz de Confusão é uma matriz que permite visualizar o desempenho do modelo mostrando a frequência de predição para cada classe do conjunto de dados. Para esta métrica, quanto mais os valores estiverem concentrados na diagonal principal é um indicativo de que o modelo está realizando boas predições.

5.1.2 Regressão

Coefficiente de Determinação (R^2): A métrica R^2 indica o quanto das variáveis dependentes podem ser explicadas pelo modelo [96]. Seu cálculo é definido pela Equação 5.4, onde y_t é o valor observado, \hat{y}_t é o valor previsto pelo modelo, \bar{y}_t é o valor médio observado e n o número de amostras, e quando R^2 está mais próximo do valor 1, melhor é o desempenho do modelo.

$$R^2 = 1 - \frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{\sum_{t=1}^n (y_t - \bar{y}_t)^2} \quad (5.4)$$

Raiz do Erro Quadrático Médio (RMSE): A métrica RMSE é a raiz quadrada da métrica Erro quadrático médio (MSE), que indica a variação dos erros, também chamada de desvio padrão dos erros [97]. Seu cálculo pode ser visto na Equação 5.5. Seu valor indica um modelo melhor quanto mais próximo de 0 estiver.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (5.5)$$

Erro Absoluto Mediano (MAE): O erro absoluto mediano caracteriza-se por medir a diferença média entre os valores reais e preditos. Mesmo existindo outra métrica que possui o mesmo acrônimo, o erro absoluto mediano se diferencia do erro absoluto médio por calcular a mediana entre os valores preditos e reais ao invés da média. Destaca-se por ser uma métrica popular que auxilia na análise de desempenho de modelos de forma robusta não sendo influenciada por *outliers*.

$$\text{MAE} = \text{median}(|y_t - \hat{y}_t|) \quad (5.6)$$

A Equação 5.6 descreve que a mediana deve ser calculada no valor absoluto entre a diferença do valor real pelo valor predito. Dessa forma, essa diferença será sempre positiva.

Erro Percentual Médio Absoluto (MAPE): Frequentemente utilizada em problemas de regressão, o erro percentual médio absoluto possui como objetivo medir a diferença em percentual entre os valores reais e preditos. Sendo expresso pela Equação 5.7, seu cálculo considera o somatório dos erros relativos absolutos (módulo da diferença do valor real e predito dividido pelo valor real) de cada observação. Por fim, é multiplicado por 100 para ser representado em porcentagem, e dividido por n para obter-se a média.

$$\text{MAPE} = \frac{1}{n} \left(\sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \cdot 100 \right) \quad (5.7)$$

5.2 RESULTADOS - AVALIAÇÃO DOS MODELOS

Nesta seção o desempenho dos modelos escolhidos é apresentado e uma breve análise é abordada. Para simplificar, somente os resultados dos modelos com as melhores

métricas são explicitados aqui, o restante estará disponível em <https://github.com/LucasFrank/Mestrado-testes> para visualização.

5.2.1 Modelos - Predição de Mobilidade - UFJF

Para resumir os resultados das métricas para cada modelo, a Tabela 9 apresenta o desempenho dos modelo mostrando cada uma das métricas escolhidas em conjunto com seu desvio padrão. Como pode ser visto, o modelo treinado pelo Auto-sklearn se destaca alcançando a acurácia ponderada média de 91.75% dentre todos os modelos, 4% acima do segundo colocado que foi o modelo MLP. E até quando observamos outras métricas como F1, *recall* e precisão, por mais que a diferença do Auto-sklearn seja menor quando comparado com os outros modelos, o seu desempenho continua sendo o melhor.

Tabela 9 – Resumo do desempenho dos modelos para predição de mobilidade no campus UFJF.

Modelos	Acurácia Ponderada	F1	Recall	Precisão	AUC-ROC
MLP	0.8775 ± 0.0088	0.9094 ± 0.0029	0.9102 ± 0.0031	0.9114 ± 0.0027	0.9892 ± 0.0005
DT	0.8643 ± 0.0090	0.9054 ± 0.0028	0.9062 ± 0.0026	0.9074 ± 0.0027	0.9831 ± 0.0015
Auto-sklearn	0.9194 ± 0.0122	0.9220 ± 0.0026	0.9225 ± 0.0025	0.9235 ± 0.0023	0.9894 ± 0.0007

Fonte: Elaborado pelo autor (2022).

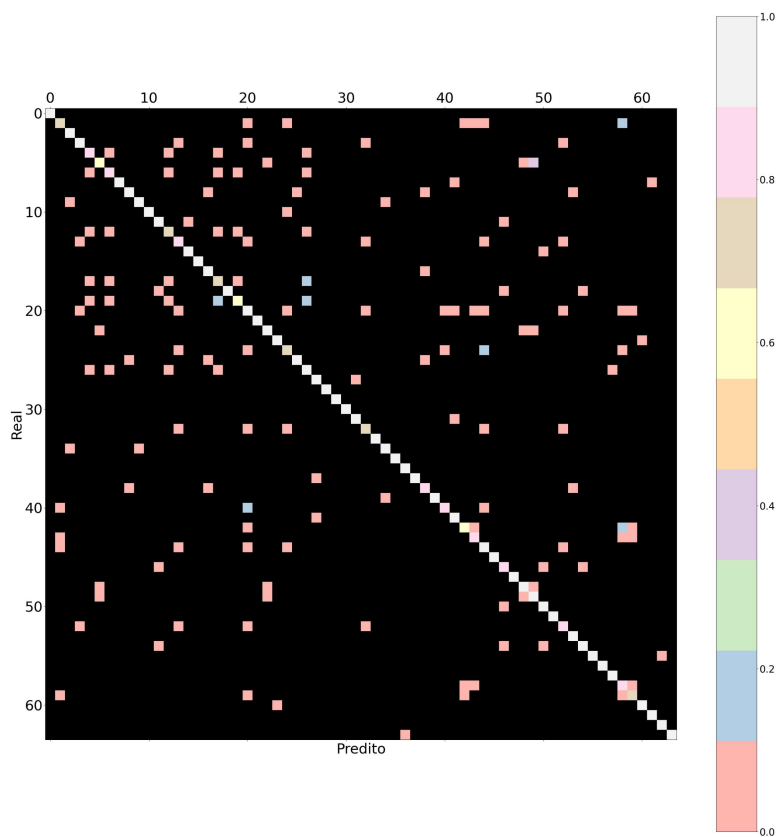
Para analisar com mais detalhes o modelo que mais se destacou, a Figura 18 traz uma matriz de confusão para visualizar como está o comportamento das predições do modelo. Para facilitar na visualização, os dados foram normalizados entre 0 e 1. Como fica evidente, a maior parte das predições está situada na diagonal principal, e uma boa parte dessas predições se encontra na faixa de 0.6 até 1.0, o que indica que o modelo está conseguindo predizer todas as classes. Para as regiões fora da diagonal principal, é possível ver que varia do intervalo de 0.4 até 0 (valor 0 recebeu a cor preta), com grande concentração para valores abaixo de aproximadamente 0.1 quando ocorre erro de predição.

5.2.2 Modelos - Predição de Mobilidade - KTH

A Tabela 10 traz destacado em negrito o desempenho do melhor modelo. Novamente o modelo do Auto-sklearn superou o modelo MLP e DT por aproximadamente 6% e 10%, respectivamente, atingindo um pico médio de 72.01% de acurácia ponderada. Ao analisar as outras métricas, é evidente a superioridade do modelo Auto-sklearn alcançando uma diferença de pelo menos 1% em todas as métricas utilizadas.

Na Figura 19 pode ser visto a matriz de confusão referente ao modelo Auto-sklearn gerado para o campus KTH. Novamente, quando verifica-se a diagonal principal, nota-se que que das 12 classes disponíveis, 11 delas se encontram no intervalo de 0.5 até 1.0, com exceção da classe 0 que está localizada na faixa de 0 até 0.1. E para os valores fora da diagonal principal, salienta-se que grande parte desses valores encontram-se no intervalo de

Figura 18 – Matriz de confusão - Autosklearn - UFJF



Fonte: Elaborado pelo autor (2022).

Tabela 10 – Resumo do desempenho dos modelos para predição de mobilidade no campus KTH.

Modelos	Acurácia Ponderada	F1	Recall	Precisão	AUC-ROC
MLP	0.6594 ± 0.0259	0.7979 ± 0.0180	0.8044 ± 0.0195	0.8106 ± 0.0218	0.9791 ± 0.0039
DT	0.6197 ± 0.0406	0.7668 ± 0.0232	0.7721 ± 0.0209	0.7748 ± 0.0248	0.9525 ± 0.0085
Auto-sklearn	0.7402 ± 0.0273	0.8395 ± 0.0114	0.8428 ± 0.0115	0.8474 ± 0.0107	0.9853 ± 0.0016

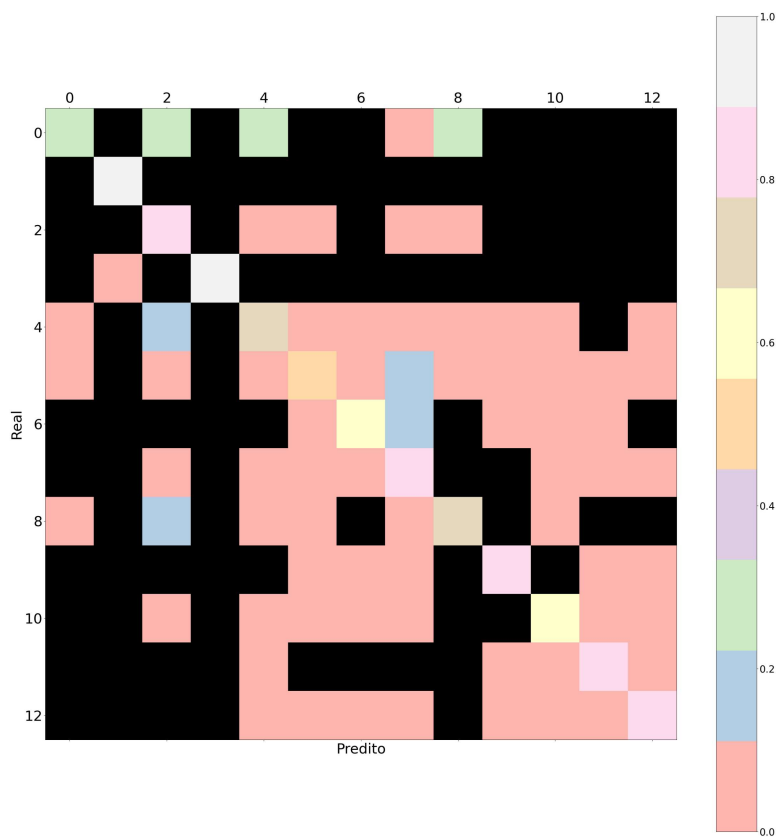
Fonte: Elaborado pelo autor (2022).

0 até 0.4. Ao considerar que as classes com maior concentração de valores está localizada na diagonal principal, pode-se afirmar que o modelo preditivo obteve um bom desempenho dado sua complexidade.

5.2.3 Modelos - Predição de Volume - UFJF

Para o desempenho dos modelos para predição de volume de usuários conectados em uma determinada janela de tempo, a Tabela 11 resume o resultado das métricas usadas para cada um. Evidencia-se que o modelo obtido pelo Auto-sklearn obteve o melhor desempenho superando os modelos MLP e DT nas métricas R^2 (maior valor), RMSE, MAE e MAPE (menores valores).

Figura 19 – Matriz de confusão - Autosklearn - KTH



Fonte: Elaborado pelo autor (2022).

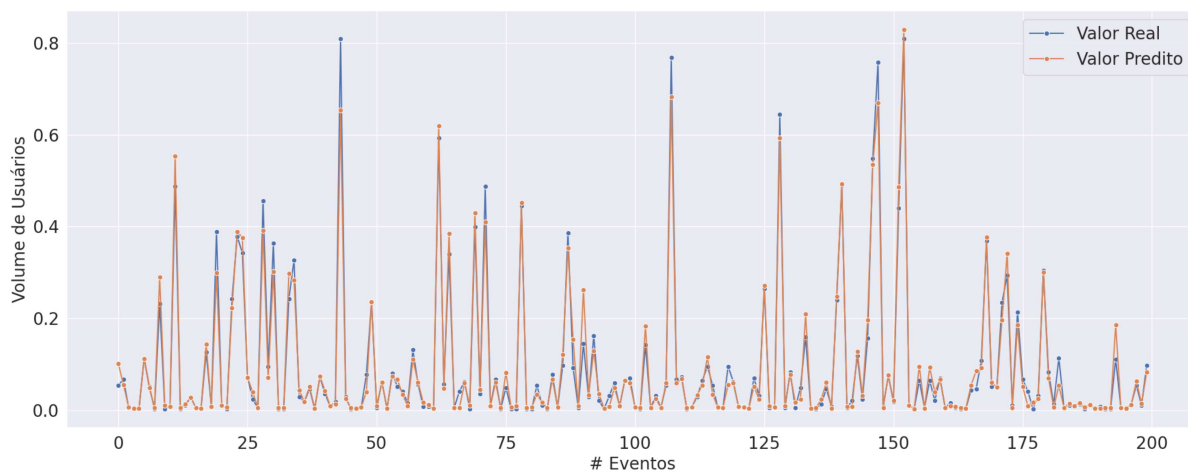
Tabela 11 – Resumo do desempenho dos modelos para predição de volume de usuários no campus UFJF.

Modelos	R^2	RMSE	MAE	MAPE
MLP	0.9587 ± 0.0028	0.0332 ± 0.0007	0.0089 ± 0.0004	0.8499 ± 0.0670
DT	0.9374 ± 0.0060	0.0410 ± 0.0023	0.0094 ± 0.0002	0.5617 ± 0.0489
Auto-sklearn	0.9606 ± 0.0031	0.0324 ± 0.0005	0.0078 ± 0.0002	0.5296 ± 0.039

Fonte: Elaborado pelo autor (2022).

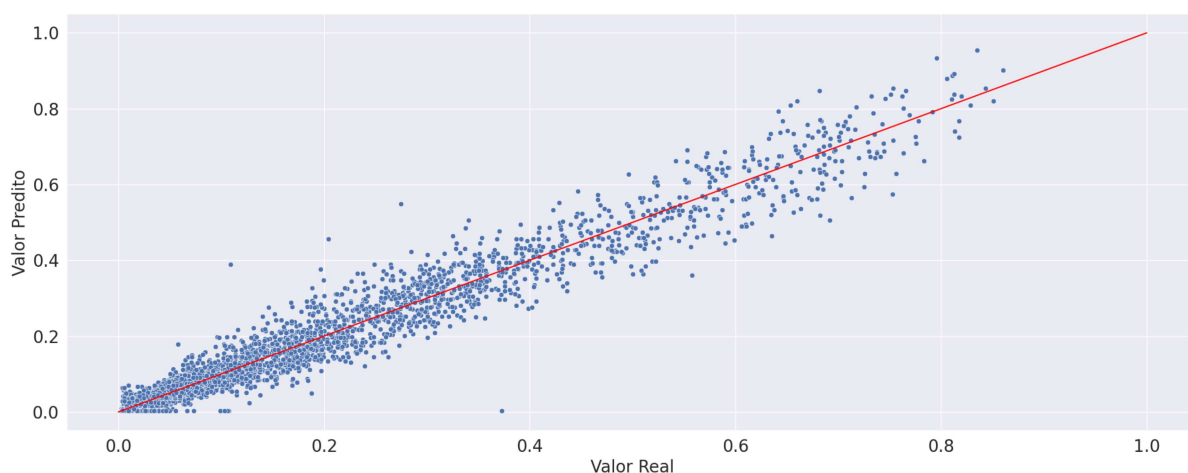
Para corroborar com o bom desempenho do modelo Auto-sklearn, as Figuras 20 e 21 são apresentadas. A Figura 20 compara o valor predito (em laranja) com o valor real (em azul). Dessa forma, é possível ver que as predições do modelo seguem um comportamento bem próximo dos valores reais. E na Figura 21 também é feita uma comparação de valor predito com valor real, mas analisando em uma gráfico de dispersão, onde quanto mais próximo da reta vermelha, melhor para o modelo. Sendo assim, destaca-se a concentração de pontos próximo a reta vermelha, principalmente no intervalo de 0 até aproximadamente 0.5.

Figura 20 – Comparação entre valor predito e valor real - Autoklearn - UFJF



Fonte: Elaborado pelo autor (2022).

Figura 21 – Comparação entre valor predito e valor real - Autoklearn - UFJF



Fonte: Elaborado pelo autor (2022).

5.2.4 Modelos - Predição de Volume - KTH

No cenário do campus KTH, a Tabela 12 traz o resultado do desempenho dos modelos utilizados para predição de volume. Novamente o modelo Auto-sklearn demonstra superioridade com as métricas R^2 , RMSE e MAE, mas desta vez é o modelo MLP que leva vantagem na métrica MAPE, alcançando uma menor porcentagem de erro médio.

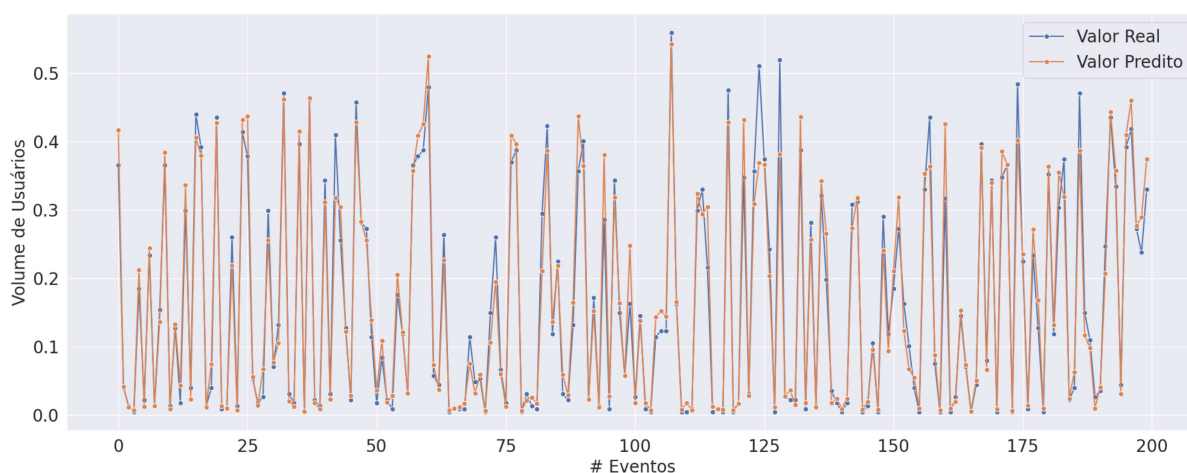
Com a Figura 22, evidencia-se que os valores desse conjunto de dados são maiores que do cenário da UFJF, e isto explica a métrica MAPE possuir um resultado menor para este cenário já que essa métrica se adapta melhor para valores altos. Outro ponto é que a predição fica bem mais próximo dos valores reais para valores menores, já para valores

Tabela 12 – Resumo do desempenho dos modelos para predição de volume de usuários no campus KTH.

Modelos	R ²	RMSE	MAE	MAPE
MLP	0.9435 ± 0.0015	0.0386 ± 0.0005	0.0137 ± 0.0004	0.3172 ± 0.0059
DT	0.9269 ± 0.0017	0.0439 ± 0.0006	0.0153 ± 0.0002	0.3551 ± 0.0074
Auto-sklearn	0.9481 ± 0.0015	0.0370 ± 0.0005	0.0131 ± 0.0003	0.3292 ± 0.0068

Fonte: Elaborado pelo autor (2022).

Figura 22 – Comparação entre valor predito e valor real - Autosklearn - KTH



Fonte: Elaborado pelo autor (2022).

mais altos, essa distância já aumenta. E isso fica enfatizado quando visualiza-se a Figura 23, onde no intervalo de 0 até aproximadamente 0.3, tem-se os pontos bem mais próximo da reta vermelha, e acima desse intervalo, os pontos começam a se dispersar, apesar de que uma boa concentração deles ainda seguem próximo da reta vermelha.

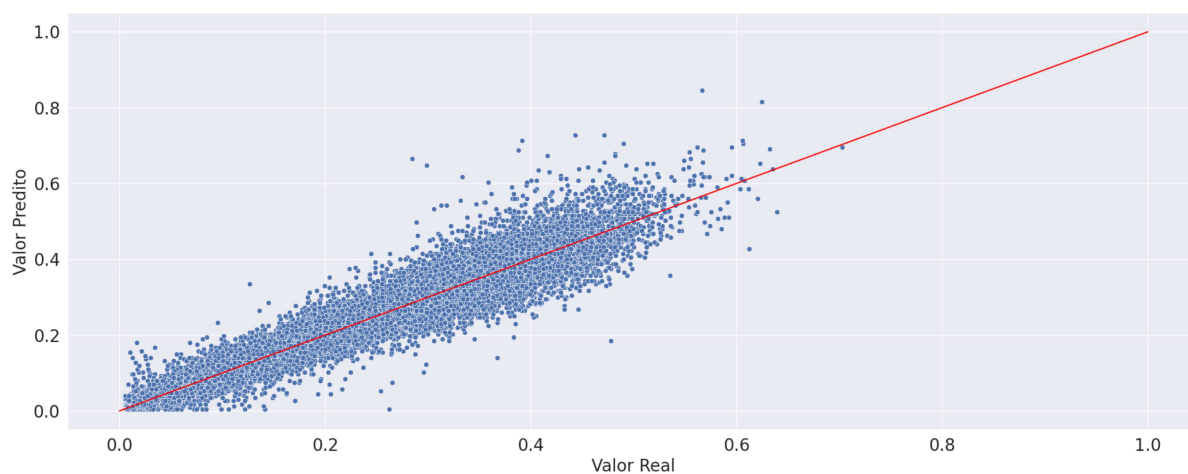
5.3 CASOS DE USO - GERÊNCIA DE REDES

Nesta seção são apresentados os casos de uso tanto para o modelo de predição de mobilidade quanto para o de predição de volume para ambos os cenários. E para facilitar, o modelo com as melhores métricas foi selecionado para cada cenário. Desta forma, uma breve demonstração de aplicações dos modelos treinados são apresentadas.

5.3.1 Predição de Mobilidade - UFJF

Nesta seção analisa-se o ganho em *Megabits (Mb)* do usuário a partir da utilização dos modelos preditivos treinados. E para tal feito, foi necessário saber em média o tempo de atraso dos *handoffs* da rede. Trabalhos como em [98], estipula-se que o tempo médio estaria entre 1s a 2s, portanto, o tempo médio entre a desassociação do usuário com o AP

Figura 23 – Comparação entre valor predito e valor real - Autosklearn - KTH



Fonte: Elaborado pelo autor (2022).

atual para com a associação desse mesmo com o AP seguinte foi calculada realizando a média de todos esses intervalos do conjunto de dados. O valor de tempo médio encontrado foi de 2.18s, sendo esse o tempo médio utilizado nesse trabalho para demonstrar o ganho dos usuários. Adicionalmente, foi definido uma taxa de dados constante (CBR - *Constant Bit Rate*) de 256Kbps para ser usado na exemplificação [99].

Para a análise do benefício da utilização da técnica em uma rede real com predição de mobilidade no auxílio ao *handoff* transparente, foram escolhidos 5 usuários que obtiveram os melhores desempenhos em relação ao ganho em *Mb* para serem analisados. A simulação é realizada utilizando o cálculo de ganho de dados trafegados realizando a multiplicação entre o tempo médio e a CBR definida a cada vez que o modelo acerta, baseando-se que a cada predição correta, o usuário realizaria o *handoff* de forma transparente.

Tabela 13 – Resumo dos resultados para cinco usuários com maior ganho em *Mb*.

Usuário	Aproveitamento (%)	Ganho em <i>Mb</i>
49	92.31%	36598.38 <i>Mb</i>
2958	99.23%	3787.75 <i>Mb</i>
9319	97.56%	1166.84 <i>Mb</i>
20058	75.52%	1115.61 <i>Mb</i>
11934	99.61%	439.27 <i>Mb</i>

Fonte: Elaborado pelo autor (2022).

Assim, resume-se os ganhos de dados trafegados para cada um dos usuários avaliados na Tabela 13. Fica evidente que o usuário 49, além de obter o maior ganho em *Mb*, alcançando o valor de 36598.38*Mb*, é classificado como um dos usuários mais frequentes do conjunto de dados, e ainda assim alcançou um ótimo aproveitamento de 92.31%. Já os

usuários 2958 e 11934 foram ambos que atingiram o mais alto aproveitamento com um ganho em *Mb* expressivo, chegando bem próximo de 100%, o aproveitamento foi de 99.23% e 99.61%, respectivamente. O ganho em *Mb* desses dois usuários se diferenciam por conta da frequência dos mesmos, apesar do aproveitamento ser próximo, o usuário 2958 realiza muito mais *handoffs* que o usuário 11934, logo o seu ganho em *Mb* é maior. Quanto aos usuários 9319 e 20058, ambos obtiveram um ganho em *Mb* próximo, sendo 1166.84*Mb* para o primeiro e 1115.61*Mb* para o segundo, mas distinguem-se no aproveitamento com uma diferença de aproximadamente 22%, o que fica claro que as predições corretas entre esses dois usuários foram semelhantes, mas o modelo errou aproximadamente 25% dos *handoffs* do usuário 20058.

Tabela 14 – Resumo geral dos resultados para o conjunto de dados da UFJF.

Geral	Média	Desvio Padrão	Intervalo de Confiança NC 95%
Aproveitamento	87.15%	25.45%	[85.01%, 89.29%]
Ganho em <i>Mb</i>	95.06	1579.86	[-37.94, 228.06]

Fonte: Elaborado pelo autor (2022).

A Tabela 14 apresenta um resumo geral dos resultados alcançados a partir da utilização da técnica proposta. Considerando todos os usuários, tem-se em média 87.15% de aproveitamento, com um desvio padrão de 25.45% devido a uma grande variação do aproveitamento desses usuários. Mas considerando um nível de confiança de 95%, tem-se que esse aproveitamento varia no intervalo de 85.01% a 89.29%. Analisando o ganho em *Mb*, a média é de 95.06*Mb* e o desvio padrão de 1579.86*Mb* que novamente demonstra uma grande variação nas amostras (alguns usuários com ganhos elevados e outros com ganho baixo), porém com o nível de confiança de 95%, tem-se o intervalo de -37.94 *Mb* a 228.06*Mb*.

5.3.2 Predição de Mobilidade - KTH

Parecido ao que foi feito para a base de dados da UFJF, um teste análogo foi aplicado para a base de dados do campus KTH com os mesmos parâmetros de tempo de *handoff* médio e CBR já definidos anteriormente.

Destaca-se na Tabela 15 o resultado dos 5 usuários que obtiveram o maior ganho em *Mb*. Com o maior ganho em *Mb* e um aproveitamento de 99.82%, o usuário 23067 se sobressai. Semelhante a esse usuário, tem-se o usuário 28777 que atingiu 100% de aproveitamento, mas com uma quantidade de *handoffs* menor, teve um ganho quase 4x menor, atingindo 354.80*Mb*. Os resultados obtidos pelos usuários 2112, 1946 e 51 equiparam-se quando analisamos seus aproveitamentos de 72.63%, 76.39% e 74.53%, nessa ordem. E mesmo com uma diferença de até 4 pontos entre si, nota-se que o número de *handoffs* diminui do usuário 2112 para o usuário 51.

Tabela 15 – Resumo dos resultados para cinco usuários com maior ganho em *Mb*.

Usuário	Aproveitamento (%)	Ganho em <i>Mb</i>
23067	99.82%	1230.61 <i>Mb</i>
2112	72.63%	365.15 <i>Mb</i>
28777	100%	354.80 <i>Mb</i>
1946	76.39%	170.04 <i>Mb</i>
51	74.53%	87.20 <i>Mb</i>

Fonte: Elaborado pelo autor (2022).

Tabela 16 – Resumo geral dos resultados para o conjunto de dados do campus KTH.

Geral	Média	Desvio Padrão	Intervalo de Confiança NC 95%
Aproveitamento	70.63%	30.19%	[66.72%, 74.54%]
Ganho em <i>Mb</i>	15.82	88.46	[4.36, 27.28]

Fonte: Elaborado pelo autor (2022).

Similar à Tabela 14, a Tabela 16 exibe um resumo geral dos resultados para o cenário do campus KTH. O aproveitamento médio de todos os usuários alcançou o valor de 70.63% com um desvio padrão de 30.19%. Além disso, com um nível de confiança de 95%, o intervalo de aproveitamento situa-se de 66.72% a 74.54%. Em relação ao ganho em *Mb* médio, foi de 15.82*Mb* com desvio padrão de 88.46*Mb*, e com o nível de confiança de 95%, atingiu o intervalo de ganho em *Mb* de 4.36*Mb* a 27.28*Mb*. Como já esperado, esses valores de desvio padrão são consequência da alta variância dos valores da amostra. Apesar dos valores reduzidos quando comparado com os resultados da base de dados da UFJF, pode-se dizer que o comportamento assemelha-se, porém em uma escala menor pela frequência dos usuários também ser menor.

5.3.3 Predição do Volume - UFJF

Nesta seção verifica-se a economia de recursos ao manter-se somente APs realmente necessários ligados em um determinado momento em uma região específica. Através do modelo treinado, é possível desativar APs em momentos em que o número de usuários conectados não exige todos os recursos ativos.

Tendo em vista um ambiente de rede sem fio atual de larga escala, e a tendência de uso de recursos por aplicações muito famosas, como *streaming* de vídeo e *cloud-gaming* [100], foi definida a taxa base de transferência de dados analisada de acordo com [101]. Segundo [101], um único usuário para desfrutar de um serviço de mídia com qualidade mínima de 720p, é necessário uma taxa média de dados de 10*Mbps*. A partir disto, tipicamente um AP corporativo Cisco (*e.g.* 1200 Series), pode suportar de 150 a 200 usuários simultaneamente. Entretanto, foi visto que essa quantidade varia em testes reais em um intervalo bem menor,

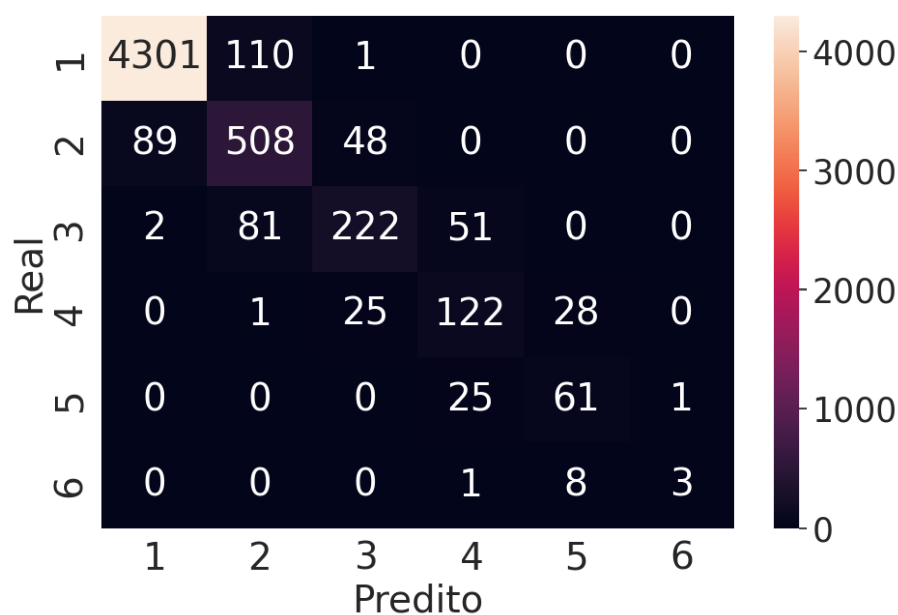
de 20 a 70 usuários [102].

Sendo assim, considera-se um cenário de 1 AP com *link* Ethernet de 1Gbps, e um limite de 65 usuários por AP, onde cada usuário tem tráfego médio de 10Mbps a 15Mbps. Para ilustrar o cenário real, a região do campus universitário selecionado para análise possui cerca de 7 APs. Permitir que todos esses APs estejam ativos em momentos de pouco tráfego de rede, ou mesmo sem usuários conectados, leva ao desperdício de recursos.

$$\# \text{ APs} = \text{round} \left(\frac{\# \text{ Usuários}}{65} \right) \quad (5.8)$$

Para auxiliar nesse processo, a Equação 5.8 foi definida para calcular quantos APs são necessários para cobrir a demanda de usuários em um determinado instante. Logo, o objetivo é realizar o cálculo dividindo a predição do volume de usuários ($\#$ Usuários) pelo limite de usuários permitidos que foi considerado para esse estudo (nesse caso, 65 usuários). Ao fim do cálculo, o resultado é sempre arredondado para cima para ter-se o número correto de APs ($\#$ APs) para atender a quantidade predita.

Figura 24 – Comparação do valor previsto e verdadeiro da quantidade de pontos de acesso necessários - Matriz de Confusão.

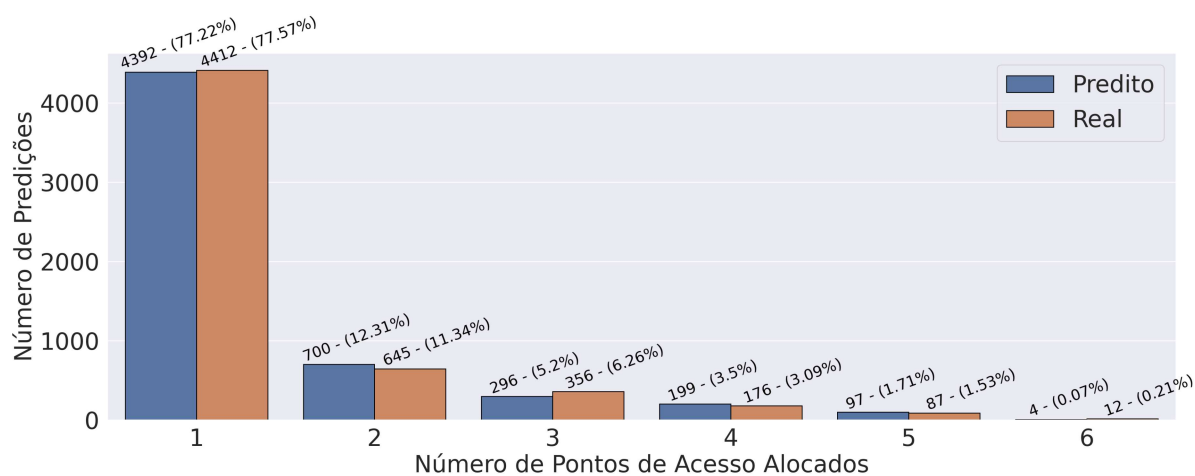


Fonte: Elaborado pelo autor (2022).

Mesmo que inicialmente o problema seja de regressão, por lidar-se com quantidade de APs limitados, o problema pode facilmente ser convertido em classificação (cada classe indica a quantidade de APs a ser utilizado). A partir disto, a Figura 24 mostra a matriz de confusão que compara a quantidade de APs preditos em relação ao valor real necessário. De acordo com a figura, o modelo acertou 97.48% (4301) das predições ao alocar um AP.

Quando são necessários dois APs, a acurácia é de 78.75% (508). Quando se trata de três APs, a acurácia é de 62.35% (222). Já quando são alocados quatro APs, a acurácia chega próximo a 69.31% (122). Para cinco APs, a acurácia é de 70.11% (61), e finalmente, para seis APs, acurácia de 25% (3). Em geral, o modelo possui uma acurácia de 91.72% ao prever o número de APs necessários. Adicionalmente, se considerar como um problema que comprometa a QoS somente quando predizer uma quantidade de APs menor que a quantidade real necessária, nesse caso, a acurácia da técnica alcançaria 95.92%.

Figura 25 – Comparando o valor previsto e o valor real - a quantidade de pontos de acesso necessários.



Fonte: Elaborado pelo autor (2022).

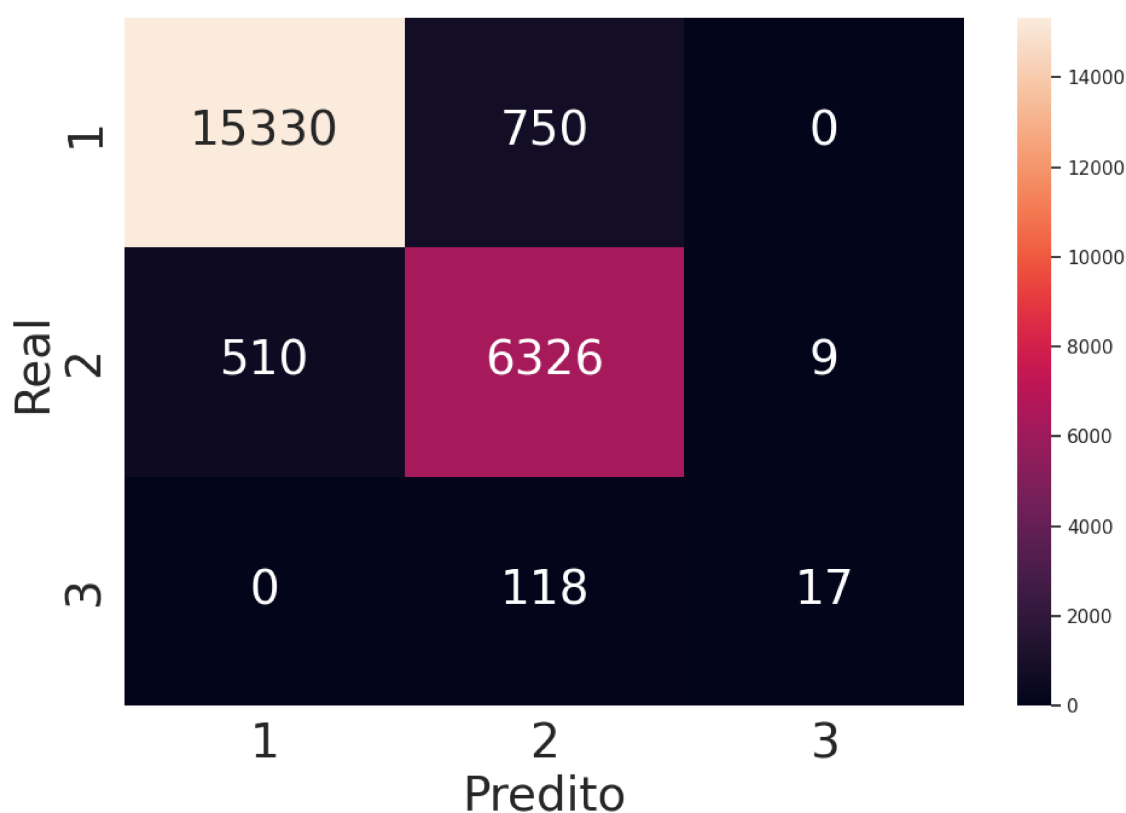
Levando em consideração a possibilidade de ligar e desligar APs, a Figura 25 demonstra a distribuição da quantidade de APs a serem alocados, comparando o valor predito e o real de usuários conectados em determinado instante. Os resultados mostram o quão equiparados estão as predições com o valor real para cada quantidade de APs alocados. Como pode ser evidenciado, a diferença é de aproximadamente 1% entre o número de predições e valor real em cada classe de alocação. Assim, é possível validarmos que, numericamente, nosso modelo atende às expectativas de redução da alocação de recursos mantendo a QoE para o usuário final.

5.3.4 Predição do Volume - KTH

Semelhante ao que foi feito para o conjunto de dados da UFJF, o mesmo teste foi realizado para o conjunto de dados do campus KTH utilizando os mesmos parâmetros do teste anterior de taxa média de dados e configurações dos APs (link ethernet e limite de usuários).

Considerando um cenário com 15 APs disponíveis para alocação, a Figura 26 traz a comparação entre valor predito e valor real da quantidade de pontos de acessos a serem

Figura 26 – Comparação do valor previsto e verdadeiro da quantidade de pontos de acesso necessários - Matriz de Confusão.

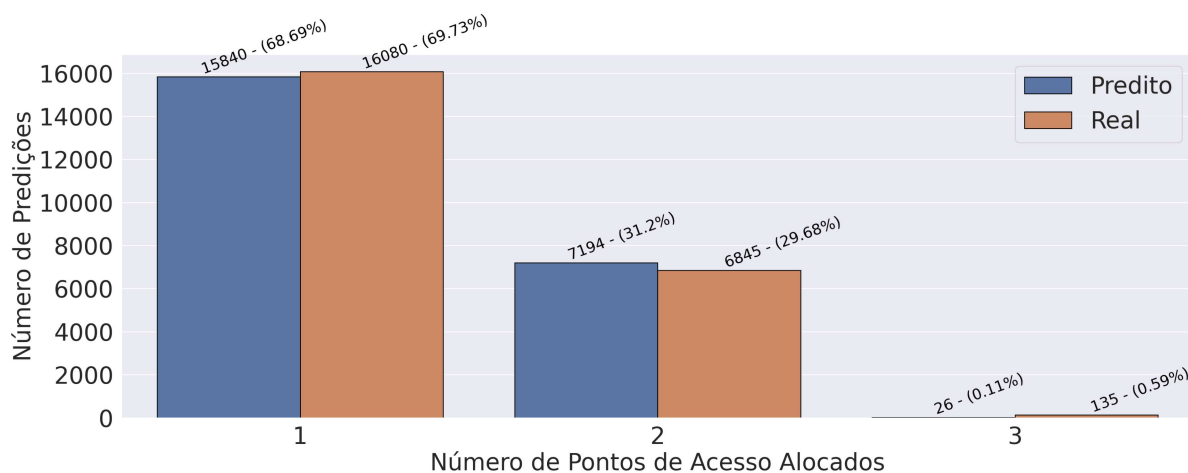


Fonte: Elaborado pelo autor (2022).

alocados. Como pode ser evidenciado, a alocação necessária foi de até três APs para cobrir todo o volume em cada momento. Individualmente, quando foi necessário alocar um AP, a técnica acertou 95.33% (15330) das vezes. Para alocação de dois APs, a acurácia foi de 92.42% (6326). Porém, quando foi necessário alocar três APs, a acurácia foi de apenas 12.59% (17). Como um todo, para predição da quantidade de APs a serem alocados, a técnica atingiu uma acurácia de 93.99%, tendo boa parte de suas falhas para alocação de 3 APs. E analogamente, se considerar da mesma forma que não é problema para técnica quando os casos em que a alocação de APs predita é maior que a alocação necessária, a acurácia aumenta para 97.28%.

Para mostrar o equilíbrio entre a quantidade de predições e valor real, a Figura 27 apresenta a distribuição comparando número de predições com a quantidade real de alocações necessárias para cada classe de alocação de APs. Fica claro que para alocação de um AP, a diferença é de aproximadamente 1%. Já para alocação de dois APs, essa diferença aumenta para aproximadamente 1.5%. Apesar da porcentagem ser baixa quando é para alocar três APs, a diferença de número de predições para quantidade real de alocações necessárias é de quase 5 vezes o próprio valor. Para casos ainda mais específicos

Figura 27 – Comparando o valor previsto e o valor real - a quantidade de pontos de acesso necessários.



Fonte: Elaborado pelo autor (2022).

assim, é de extrema importância que ao acoplar essa técnica a uma rede real, regras sejam definidas para poder garantir cobertura e QoE a todos os usuários que se conectem à rede.

5.4 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo foram apresentados os resultados referentes aos modelos preditivos propostos com objetivo de comparação para cada um dos cenários (UFJF e KTH). Além disso, casos de uso foram demonstrados para as propostas de predição de mobilidade e predição de volume de usuários de modo a apontar possíveis aplicações dos modelos em cenários reais.

De forma geral, como foi evidenciado nesse capítulo, o modelo que mais se destacou foi o modelo gerado pelo AutoML. Tanto na proposta de predição de mobilidade quanto na de predição de volume de usuários, as métricas alcançadas destacam esse modelo como o melhor. Entretanto, com um destaque maior para a primeira proposta onde a diferença nas métricas são mais significativas. Para a segunda proposta, apesar das métricas apontarem o AutoML ainda como o melhor modelo, a diferença aqui é bem menor quando comparado com o modelo MLP. Desta forma, tanto o modelo MLP quanto o AutoML possuem um desempenho aceitável para esse problema. A escolha entre esses modelos diferencia em o modelo MLP, por ser tratar de um modelo único, possui seu treinamento bem mais rápido, enquanto o AutoML ainda precisa treinar diversos modelos, sendo muito mais custoso em relação ao tempo de treinamento e consumo de memória (RAM e disco rígido).

A respeito dos casos de uso percorridos para ambas as propostas, é importante salientar sua eficácia mostrando sua devida importância na utilização dos modelos. Logo, a partir de um modelo bem configurado, a aplicação para predição de mobilidade dos

usuários poderá trazer ótimos benefícios para área de gerência à redes, principalmente destacando um grande problema que é a busca pelo *handoff* transparente, dessa forma, evidencia-se uma melhor qualidade de experiência para os usuários.

Para a aplicação direcionada à predição de volume de usuários, considera-se que ao saber a quantidade de usuários conectados em um determinado ambiente, é possível tomar decisões importantes como a alocação de banda larga e de pontos de acesso necessários para atender todos os usuários sem desperdiçar recursos. Nesse caso foi escolhido apresentar o desempenho do modelo para alocar a quantidade de pontos de acessos disponíveis para atender uma determinada região sem que o cliente perca qualidade no acesso. Tendo isso em vista, em ambos os cenários, evidencia-se que em boa parte dos testes não mostrou necessário que todos os APs estejam ligados, logo se em determinados momentos que a quantidade de usuários conectados for menor que o limite médio por AP, então seria possível economizar recursos como energia e banda larga disponível. Com isso, conclui-se que essa aplicação possui uma eficiência significativa dentro dos testes efetuados e destaca-se que esse estudo foi realizado considerando que os APs estão distribuídos de modo a cobrir toda uma região, já em uma aplicação real torna-se importante que seja feito um estudo do posicionamento dos APs para que seja otimizado sua cobertura para que as possíveis tomadas de decisão não afetem a qualidade da conexão.

6 CONCLUSÃO

O conhecimento da importância da gestão de recursos e o aumento progressivo do número de usuários conectados às redes completam parte da motivação que realça a importância de estudos aprofundados sobre a gestão de redes sem fios. Juntamente com a evolução de diversas técnicas poderosas de aprendizado de máquina, esse gerenciamento pode ser feito de forma eficiente por meio de modelos preditivos.

Este trabalho propôs e avaliou o uso de modelos preditivos para auxiliar no gerenciamento de recursos em redes sem fio. Para tal feito, duas abordagens foram empregadas, sendo a primeira abordagem a predição da mobilidade do usuário, isto é, prever qual próximo AP o usuário irá se conectar, e a segunda sendo a predição do volume de usuários, ou seja, prever a quantidade de usuários conectados na rede em um determinado instante. Para treinar e avaliar tanto os modelos quanto os casos de uso, dois conjuntos de dados reais foram utilizados. Um dos conjuntos advém da UFJF e o outro do campus KTH na Suécia. Antes de mencionar os modelos utilizados, uma série de pré-processamentos foram usados para cada uma das abordagens para preparar os conjuntos de dados. No pré-processamento da primeira abordagem foi importante definir um tempo limite, neste trabalho foi de 60 segundos, onde considera os APs dentro da mesma trajetória dado que o usuário desconectou de um AP e conectou em outro AP. E também foi importante para escolher o número de APs que seriam considerados em uma trajetória, o número de APs limite foi de 12, de modo a facilitar na seleção de todas as trajetórias possíveis que os usuários realizaram. Já no pré-processamento da segunda abordagem, foi necessário definir o tempo referente ao tamanho da janela que agruparia os usuários para realizar a contagem de quantos estariam conectados no intervalo de tempo definido, para isso o valor de 5 minutos foi escolhido. Além disso, teve-se que definir os parâmetros para a quantidade de dados históricos e dados recentes, onde nesse caso foram escolhidos os valores 3 e 11, respectivamente. Com intuito de minimizar a quantidade de testes ao todo, testes iniciais foram realizados para definir esses parâmetros de pré-processamento. Os modelos preditivos escolhidos para testes foram os algoritmos árvore de decisão, a rede neural Perceptron de Múltiplas Camadas e a técnica de aprendizado de máquina automatizado. Os testes realizados usaram da técnica de validação cruzada, onde 70% foi para treino e teste, e o restante para os casos de uso.

Grande parte dos estudos aqui produzidos foram originados em artigos publicados pelo autor deste trabalho. Os trabalhos [92] e [91] que foram publicados no evento *International Conference on Computational Science and Its Applications* (ICCSA) no ano de 2019 já relatava bons resultados utilizando as técnicas de pré-processamento similares ao que foi mencionado, porém para a abordagem de predição de volume de veículos. Posteriormente, em [103], publicado em 2021 no evento *IEEE Symposium on Computers and Communications* (ISCC), foi descrito o uso dessa mesma técnica, mas agora aplicado

à abordagem de predição de volume de usuários e o custo da alocação de largura de banda necessária para atendê-los. Após isso, o artigo [104] traz uma extensão do trabalho anterior e foi publicado nos Anais do Workshop de Computação Urbana (CoURB) do SBRC 2022, e expõe uma breve caracterização dos dados, realizando uma comparação/otimização dos modelos preditivos utilizados.

Os resultados experimentais obtidos em ambas abordagens foram satisfatórios, principalmente para o conjunto de dados da UFJF, onde alcançou uma acurácia ponderada média de mais de 90% para a primeira abordagem mesmo possuindo um certo nível de complexidade devido à grande quantidade de pontos de acessos considerados. Atingindo uma acurácia ponderada média de 74%, os resultados obtidos através do conjunto de dados do campus KTH superou as expectativas, justamente por não possuir um conhecimento geográfico prévio dos APs e suas identificações. Ao analisar a segunda abordagem, os modelos tiveram desempenho satisfatório para ambos conjuntos de dados, apresentando um coeficiente de ajuste bem próximo entre si, em uma faixa de 0.94 a 0.96. Tendo em vista o modelo mais bem ajustado para a primeira abordagem, os casos de uso demonstraram um aproveitamento médio de 87.15% e 70.63% para os usuários na predição de mobilidade para o cenário da UFJF e para o campus KTH, respectivamente. Por fim, os casos de uso para a segunda abordagem apontam uma acurácia média de 91.72% para UFJF e 93.99% para o campus KTH ao sugerir o número de APs necessários de acordo com o volume de usuários conectados predito. Com isso, conclui-se que as duas abordagens são importantes e devem ser consideradas nas implementações de soluções em redes de médio e grande porte. Com ambas as abordagens há o aumento tanto na qualidade de serviço, quanto no gerenciamento de recursos, e na qualidade de experiência do usuário com a possibilidade do *handoff* transparente, evitando esperas ou interrupção de serviços e conexões, além de otimizar o uso da banda larga total ofertada e evitar o gasto de energia com equipamentos ligados desnecessariamente.

Como trabalhos futuros, espera-se a expansão da base de dados para mais registros, assim como, otimizar os modelos e seus parâmetros. Além disso, pretende-se avaliar a seleção de características relevantes dos atributos da base de dados. Estudar de forma aprofundada os parâmetros referentes ao pré-processamento das características. E por fim, unir ambas as abordagens em uma única aplicação e aplicá-las em um cenário de tempo real de modo a analisar a viabilidade da solução.

REFERÊNCIAS

- TORRES, R. L. S.; VISVANATHAN, R.; ABBOTT, D.; HILL, K. D.; RANASINGHE, D. C. A battery-less and wireless wearable sensor system for identifying bed and chair exits in a pilot trial in hospitalized older people. **PloS one**, Public Library of Science San Francisco, CA USA, v. 12, n. 10, p. e0185670, 2017.
- PRASAD, P. S.; AGRAWAL, P. "Movement prediction in wireless networks using mobility traces". 2010. 1–5 p.
- MEDEIROS, D.; NETO, H. C.; LOPEZ, M. A.; MAGALHAES, L.; SILVA, E.; VIEIRA, A.; FERNANDES, N.; MATTOS, D. Análise de dados em redes sem fio de grande porte: Processamento em fluxo em tempo real, tendências e desafios. **Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC**, v. 2019, p. 142–195, 2019.
- MEDEIROS, D. S.; NETO, H. N. C.; LOPEZ, M. A.; MAGALHÃES, L. C. S.; FERNANDES, N. C.; VIEIRA, A. B.; SILVA, E. F.; MATTOS, D. M. A survey on data analysis on large-scale wireless networks: online stream processing, trends, and challenges. **Journal of Internet Services and Applications**, SpringerOpen, v. 11, n. 1, p. 1–48, 2020.
- SUMATHI, R.; POORNIMA, M.; SUGANTHI, M. User aware mobility management in cognitive radio cellular network. In: **2014 International Conference on Electronics and Communication Systems (ICECS)**, 2014. p. 1–5.
- OJIMA, T.; FUJII, T. Resource management for mobile edge computing using user mobility prediction. In: **2018 International Conference on Information Networking (ICOIN)**, 2018. p. 718–720.
- KIM, D. U.; PARK, S. B.; HONG, C. S.; HUH, E. N. Resource allocation and user association using reinforcement learning via curriculum in a wireless network with high user mobility. In: **2023 International Conference on Information Networking (ICOIN)**, 2023. p. 382–386.
- BRITO, I. V. S.; FIGUEIREDO, G. B. Improving qos and qoe through seamless handoff in software-defined ieee 802.11 mesh networks. **IEEE Communications Letters**, IEEE, v. 21, n. 11, p. 2484–2487, 2017.
- ZHANG, C.; PATRAS, P.; HADDADI, H. Deep learning in mobile and wireless networking: A survey. **IEEE Communications surveys & tutorials**, IEEE, v. 21, n. 3, p. 2224–2287, 2019.
- HOLZINGER, A.; KIESEBERG, P.; WEIPPL, E.; TJOA, A. M. Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable ai. In: SPRINGER. **Machine Learning and Knowledge Extraction: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings 2**, 2018. p. 1–8.
- SHARMA, A.; JAIN, A.; GUPTA, P.; CHOWDARY, V. Machine learning applications for precision agriculture: A comprehensive review. **IEEE Access**, IEEE, v. 9, p. 4843–4873, 2020.

KWEKHA-RASHID, A. S.; ABDULJABBAR, H. N.; ALHAYANI, B. Coronavirus disease (covid-19) cases analysis using machine-learning applications. **Applied Nanoscience**, Springer, v. 13, n. 3, p. 2013–2025, 2023.

CHAARI, H.; MNIF, K.; ZARAI, F.; KAMOUN, L. Qos enhancement based on resource allocation in wireless networks. In: **2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)**, 2017. p. 1407–1412.

PAJEVIC, L.; KARLSSON, G.; FODOR, V. **CRAWDAD kth/campus**, 2022. Disponível em: <<https://dx.doi.org/10.15783/c7-5r6x-4b46>>.

CREMONEZI, B.; FILHO, A. R. G.; SILVA, E. F.; NACIF, J. A. M.; VIEIRA, A. B.; NOGUEIRA, M. Improving the attribute retrieval on abac using opportunistic caches for fog-based iot networks. **Computer Networks**, Elsevier, v. 213, p. 109000, 2022.

APRENDIZADO de Máquina. <https://www.consumidormoderno.com.br/2021/03/17/evolucao-aprendizado-de-maquina-impacto-mercado/>. Acessado em: 15 de Novembro de 2022.

AUTOML Definition.

https://en.wikipedia.org/wiki/Automated_machine_learning. Acessado em: 15 de Novembro de 2022.

AUTOML. <https://www.automl.org>. Acessado em: 15 de Novembro de 2022.

AUTOML Google. <https://cloud.google.com/automl>. Acessado em: 15 de Novembro de 2022.

AUTOML Microsoft. <https://learn.microsoft.com/en-us/azure/machine-learning/concept-automated-ml>.

Acessado em: 15 de Novembro de 2022.

WARING, J.; LINDVALL, C.; UMETON, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. **Artificial intelligence in medicine**, Elsevier, v. 104, p. 101822, 2020.

AUTOSKLEARN. <https://automl.github.io/auto-sklearn/master/>. Acessado em: 15 de Novembro de 2022.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. et al. Scikit-learn: Machine learning in python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.

FEURER, M.; KLEIN, A.; EGGENSBERGER, K.; SPRINGENBERG, J.; BLUM, M.; HUTTER, F. Efficient and robust automated machine learning. **Advances in neural information processing systems**, v. 28, 2015.

AUTOSKLEARN Classification. <https://github.com/automl/auto-sklearn/tree/master/autosklearn/pipeline/components/classification>. Acessado em: 17 de Novembro de 2022.

AUTOSKLEARN Regression. <https://github.com/automl/auto-sklearn/tree/master/autosklearn/pipeline/components/regression>. Acessado em: 17 de Novembro de 2022.

DECISION tree. https://en.wikipedia.org/wiki/Decision_tree. Acessado em: 07 de Novembro de 2022.

KOTSIANTIS, S. B. Decision trees: a recent overview. **Artificial Intelligence Review**, Springer, v. 39, n. 4, p. 261–283, 2013.

EHRENFEUCHT, A.; HAUSSLER, D. Learning decision trees from random examples. **Information and Computation**, Elsevier, v. 82, n. 3, p. 231–246, 1989.

FREUND, Y.; MASON, L. The alternating decision tree learning algorithm. In: **icml**, 1999. v. 99, p. 124–133.

CUNNINGHAM, P.; CORD, M.; DELANY, S. J. Supervised learning. In: **Machine learning techniques for multimedia**, 2008. p. 21–49.

SUTHAHARAN, S. Decision tree learning. In: **Machine Learning Models and Algorithms for Big Data Classification**, 2016. p. 237–269.

GAMA, J. Árvores de decisão. **Palestra ministrada no Núcleo da Ciência de Computação da Universidade do Porto, Porto**, 2002.

LOH, W.-Y. Classification and regression trees. **Wiley interdisciplinary reviews: data mining and knowledge discovery**, Wiley Online Library, v. 1, n. 1, p. 14–23, 2011.

WANG, X. Decision-tree-based relay selection in dualhop wireless communications. **IEEE Transactions on Vehicular Technology**, IEEE, v. 68, n. 6, p. 6212–6216, 2019.

ONODA, M.; EBECKEN, N. F. Implementação em java de um algoritmo de árvore de decisão acoplado a um sgbd relacional. In: **SBBD**, 2001. p. 55–64.

PEKEL, E. Estimation of soil moisture using decision tree regression. **Theoretical and Applied Climatology**, Springer, v. 139, n. 3, p. 1111–1119, 2020.

LANGEL, M.; TILÉ, Y. Corrado gini, a pioneer in balanced sampling and inequality theory. **Metron**, Springer, v. 69, n. 1, p. 45–65, 2011.

CHANDRA, B.; VARGHESE, P. P. Fuzzifying gini index based decision trees. **Expert Systems with Applications**, Elsevier, v. 36, n. 4, p. 8549–8559, 2009.

LIU, H.; ZHOU, M.; LU, X. S.; YAO, C. Weighted gini index feature selection method for imbalanced data. In: IEEE. **2018 IEEE 15th international conference on networking, sensing and control (ICNSC)**, 2018. p. 1–6.

GUPTA, A.; BANSAL, A.; ROY, K. et al. Solar energy prediction using decision tree regressor. In: IEEE. **2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)**, 2021. p. 489–495.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, JSTOR, p. 1189–1232, 2001.

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. **Neural and adaptive systems: fundamentals through simulations**, 2000.

FURTADO, M. I. V. **Redes Neurais Artificiais: Uma Abordagem Para Sala de Aula**, 2019.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

RUSSELL, I. The delta rule. **University of Hartford, West Hartford**. Accessed, v. 5, 2012.

SUNDERMEYER, M.; NEY, H.; SCHLÜTER, R. From feedforward to recurrent lstm neural networks for language modeling. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, IEEE, v. 23, n. 3, p. 517–529, 2015.

XU, L.; REN, J. S.; LIU, C.; JIA, J. Deep convolutional neural network for image deconvolution. **Advances in neural information processing systems**, v. 27, 2014.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

SAZLI, M. H. A brief review of feed-forward neural networks. **Communications, Faculty of Science, University of Ankara**, v. 50, n. 1, p. 11–17, 2006.

GOMES, G. S. d. S.; LUDERMIR, T. B.; LIMA, L. M. Comparison of new activation functions in neural network for forecasting financial time series. **Neural Computing and Applications**, Springer, v. 20, n. 3, p. 417–439, 2011.

BIRCANOĞLU, C.; ARICA, N. A comparison of activation functions in artificial neural networks. In: IEEE. **2018 26th Signal Processing and Communications Applications Conference (SIU)**, 2018. p. 1–4.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Mathematics of control, signals and systems**, Springer, v. 2, n. 4, p. 303–314, 1989.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**, 2010. p. 807–814.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: **Proc. icml**, 2013. p. 3.

PRAS, A.; SCHONWALDER, J.; BURGESS, M.; FESTOR, O.; PEREZ, G. M.; STADLER, R.; STILLER, B. Key research challenges in network management. **IEEE Communications Magazine**, v. 45, n. 10, p. 104–110, 2007.

WHAT is Network Management?

<https://www.microfocus.com/en-us/what-is/network-management>. Acessado em: 21 de Novembro de 2022.

- KIM, H.; FEAMSTER, N. Improving network management with software defined networking. **IEEE Communications Magazine**, v. 51, n. 2, p. 114–119, 2013.
- KREUTZ, D.; RAMOS, F. M. V.; VERÍSSIMO, P. E.; ROTHENBERG, C. E.; AZODOLMOLKY, S.; UHLIG, S. Software-defined networking: A comprehensive survey. **Proceedings of the IEEE**, v. 103, n. 1, p. 14–76, 2015.
- CLAISE, B. **Cisco systems netflow services export version 9**, 2004.
- FU, Y.; WANG, S.; WANG, C.-X.; HONG, X.; MCLAUGHLIN, S. Artificial intelligence to manage network traffic of 5g wireless networks. **IEEE Network**, v. 32, n. 6, p. 58–64, 2018.
- YAP, K.-L.; CHONG, Y.-W. "optimized access point selection with mobility prediction using hidden markov model for wireless network". In: IEEE. **2017 ninth international conference on ubiquitous and future networks (ICUFN)**, 2017. p. 38–42.
- FU, W.; LI, X.; JI, H.; ZHANG, H. Optimal access points selection based on mobility prediction in heterogeneous small cell networks. In: IEEE. **2018 IEEE/CIC International Conference on Communications in China (ICCC)**, 2018. p. 421–425.
- WANG, P.; YANG, J.; ZHANG, J. Indoor trajectory prediction for shopping mall via sequential similarity. **Information**, MDPI, v. 13, n. 3, p. 158, 2022.
- PAJEVIC, L.; FODOR, V.; KARLSSON, G. Predicting the users' next location from wlan mobility data. In: IEEE. **2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)**, 2018. p. 61–66.
- DUONG, T.-V. T.; TRAN, D. Q. **Mobility prediction based on collective movement behaviors in public WLANs**. 2015. 1003–1010 p.
- DUONG, T. V. T.; TRAN, D. Q. **An effective approach for mobility prediction in wireless network based on temporal weighted mobility rule**. 2012. 29–36 p.
- DUONG, T. V. T.; TRAN, D. Q.; TRAN, C. H. **A weighted combination similarity measure for mobility patterns in wireless networks**. 2012.
- DUONG, T.; TRAN, D. **Clustering mobility patterns in wireless networks**. 2013. 4263–4284 p.
- GAO, Y.; CHEN, J.; LIU, Z.; LIU, L.; HU, N. Deep learning based location prediction with multiple features in communication network. In: IEEE. **2021 IEEE Wireless Communications and Networking Conference (WCNC)**, 2021. p. 1–5.
- HAMOUDZADEH, A.; BEHZADI, S. Predicting user's next location using machine learning algorithms. **Spatial Information Research**, Springer, v. 29, n. 3, p. 379–387, 2021.
- ZHANG, W.; LIU, Y.; LIU, T.; YANG, C. Trajectory prediction with recurrent neural networks for predictive resource allocation. In: IEEE. **2018 14th IEEE International Conference on Signal Processing (ICSP)**, 2018. p. 634–639.

- BAHRA, N.; PIERRE, S. Rnn-based user trajectory prediction using a preprocessed dataset. In: **IEEE. 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)**, 2020. p. 1–6.
- ADEGE, A. B.; LIN, H.-P.; WANG, L.-C. Mobility predictions for iot devices using gated recurrent unit network. **IEEE Internet of Things journal**, IEEE, v. 7, n. 1, p. 505–517, 2019.
- VIEIRA, A. B.; PARAIZO, W. N.; CHAVES, L. J.; CORREIA, L. H.; SILVA, E. F. An sdn-based energy-aware traffic management mechanism. **Annals of Telecommunications**, Springer, p. 1–12, 2021.
- CHIH-LIN, I.; SUN, Q.; LIU, Z.; ZHANG, S.; HAN, S. The big-data-driven intelligent wireless network: architecture, use cases, solutions, and future trends. **IEEE vehicular technology magazine**, IEEE, v. 12, n. 4, p. 20–29, 2017.
- CASAS, P.; SEUFERT, M.; WEHNER, N.; SCHWIND, A.; WAMSER, F. Enhancing machine learning based qoe prediction by ensemble models. In: **IEEE ICDCS**, 2018.
- LI, Y.; HUANG, J.; CHEN, H. Time series prediction of wireless network traffic flow based on wavelet analysis and BP neural network. **Journal of Physics: Conference Series**, IOP Publishing, v. 1533, n. 3, p. 032098, apr 2020. Disponível em: <<https://doi.org/10.1088/1742-6596/1533/3/032098>>.
- FORD, G.; CARGAN, R.; AHMED, A.; RIGNEY, K.; BERRY, C.; BUCCI, D.; KAM, M. Wireless network traffic disaggregation using bayesian nonparametric techniques. In: **CISS**, 2018. p. 1–6.
- LI, M.; WANG, Y.; WANG, Z.; ZHENG, H. A deep learning method based on an attention mechanism for wireless network traffic prediction. **Ad Hoc Networks**, v. 107, p. 102258, 2020. ISSN 1570-8705. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1570870519310923>>.
- DENG, T.; WAN, M.; SHI, K.; ZHU, L.; WANG, X.; JIANG, X. Short term prediction of wireless traffic based on tensor decomposition and recurrent neural network. **SN Applied Sciences**, Springer Science and Business Media LLC, v. 3, n. 9, ago. 2021. Disponível em: <<https://doi.org/10.1007/s42452-021-04761-8>>.
- SONE, S. P.; LEHTOMÄKI, J.; KHAN, Z.; UMEBAYASHI, K. Forecasting wireless network traffic and channel utilization using real network/physical layer data. In: **EuCNC/6G Summit**, 2021. p. 31–36.
- SAIKHU, A.; SETYADI, A. T.; PURWANANTO, Y.; WIJAYA, A. Y. Spatio-temporal recurrent neural networks modeling for number of users prediction on wireless traffic networks. In: **ICICoS**, 2020. p. 1–6.
- YU, Y.; LIU, S.; WANG, L.; TENG, F.; LI, S. Traffic prediction model based on improved quantum particle swarm algorithm in wireless network. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 768, n. 6, p. 062110, mar 2020. Disponível em: <<https://doi.org/10.1088/1757-899x/768/6/062110>>.

- KHAN, M. A.; IQBAL, N.; JAMIL, H.; KIM, D.-H. et al. An optimized ensemble prediction model using automl based on soft voting classifier for network intrusion detection. **Journal of Network and Computer Applications**, Elsevier, v. 212, p. 103560, 2023.
- SINGH, A.; AMUTHA, J.; NAGAR, J.; SHARMA, S.; LEE, C.-C. Automl-id: Automated machine learning model for intrusion detection using wireless sensor network. **Scientific Reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 9074, 2022.
- HALVARI, T.; NURMINEN, J. K.; MIKKONEN, T. Robustness of automl for time series forecasting in sensor networks. In: IEEE. **2021 IFIP Networking Conference (IFIP Networking)**, 2021. p. 1–3.
- WU, D.; GUAN, Q.; FAN, Z.; DENG, H.; WU, T. Automl with parallel genetic algorithm for fast hyperparameters optimization in efficient iot time series prediction. **IEEE Transactions on Industrial Informatics**, IEEE, 2022.
- DURBHA, K. S.; AMURU, S. Automl models for wireless signals classification and their effectiveness against adversarial attacks. In: IEEE. **2022 14th International Conference on COMMunication Systems & NETWORKS (COMSNETS)**, 2022. p. 265–269.
- FRANK, L. R.; FERREIRA, Y. M.; JULIO, E. P.; FERREIRA, F. H. C.; DEMBOGURSKI, B. J.; SILVA, E. F. Multilayer perceptron and particle swarm optimization applied to traffic flow prediction on smart cities. In: SPRINGER. **International Conference on Computational Science and Its Applications**, 2019. p. 35–47.
- FERREIRA, Y. M.; FRANK, L. R.; JULIO, E. P.; FERREIRA, F. H. C.; DEMBOGURSKI, B. J.; SILVA, E. F. Applying a multilayer perceptron for traffic flow prediction to empower a smart ecosystem. In: SPRINGER. **International Conference on Computational Science and Its Applications**, 2019. p. 633–648.
- QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, n. 1, p. 81–106, 1986.
- PAJEVIC, L.; KARLSSON, G.; FODOR, V. **CRAWDAD dataset kth/campus (v. 2019-07-01)**. jul. 2019. Downloaded from <https://crawdad.org/kth/campus/20190701>.
- WITTEN, D.; JAMES, G. **An introduction to statistical learning with applications in R**, 2013.
- GLANTZ, S.; SLINKER, B. **Primer of Applied Regression & Analysis of Variance**, ed, 2001.
- LEHMANN, E. L.; CASELLA, G. **Theory of point estimation**, 2006.
- MARTINEZ, S.; CARDONA, N.; BOTERO, J. F. Seamless handoff management in iee 802.11 networks using sdn. In: IEEE. **2018 IEEE Colombian Conference on Communications and Computing (COLCOM)**, 2018. p. 1–6.

BANDUNG, Y.; SUBEKTI, L. B.; TANJUNG, D.; CHRYSOSTOMOU, C. Qos analysis for webrtc videoconference on bandwidth-limited network. In: **2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)**, 2017. p. 547–553.

CAI, W.; SHEA, R.; HUANG, C.-Y.; CHEN, K.-T.; LIU, J.; LEUNG, V. C.; HSU, C.-H. A survey on cloud gaming: Future of computer games. **IEEE Access**, IEEE, v. 4, p. 7605–7620, 2016.

CARRASCOSA, M.; BELLALTA, B. Cloud-gaming: Analysis of google stadia traffic. **arXiv preprint arXiv:2009.09786**, 2020.

KAFI, M. A.; MOURADIAN, A.; VEQUE, V. Offline qos association scheme based on clients priorities and demands in wlan networks. In: **IEEE GLOBECOM**, 2019. p. 1–6.

FRANK, L. R.; OLIVEIRA, R. M. D.; VIEIRA, A. B.; SILVA, E. F. Improving a smart environment with wireless network user load prediction. In: **2021 IEEE Symposium on Computers and Communications (ISCC)**, 2021. p. 1–6.

FRANK, L.; CARNEVALE, L.; GALLETTA, A.; VILLARI, M.; VIEIRA, A.; SILVA, E. Alocação de recursos em redes sem fio de grande porte por meio de meta-heurísticas e predição da carga de usuários. In: **Anais do VI Workshop de Computação Urbana**, 2022. p. 29–42. ISSN 2595-2706. Disponível em: <<https://sol.sbc.org.br/index.php/courb/article/view/21442>>.