UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Ualison Rodrigo Ferreira Dias

Uma abordagem baseada em regras fuzzy auto-organizáveis para classificação de ambientes internos em aplicações de IoT

Juiz de Fora
2019

Ualison Rodrigo Ferreira Dias

Uma abordagem baseada em regras fuzzy auto-organizáveis para classificação de ambientes internos em aplicações de IoT

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Eduardo Pestana de Aguiar

Juiz de Fora

2019

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

Dias, Ualison Rodrigo Ferreira.
Uma abordagem baseada em regras fuzzy auto-organizáveis para classificação de ambientes internos em aplicações de IoT / Ualison Rodrigo Ferreira Dias. – 2019.
54 f. : il.

Orientador: Eduardo Pestana de Aguiar
Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2019.

1. Classificador Fuzzy Auto-organizável. 2. Sensores sem fio. 3. Sinal de Radio frequência. 4. Internet das Coisas. I. Aguiar, Eduardo Pestana, orient. II. Título.

Ualison Rodrigo Ferreira Dias

Uma Abordagem Baseada em Regras Fuzzy Auto-Organizáveis para Classificação de Ambientes Internos em Aplicações de IoT

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Aprovada em 18 de Dezembro de 2019

BANCA EXAMINADORA

Eduardo Pestana de Aguiar

Prof. Dr. Eduardo Pestana de Aguiar - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. José Manoel de Seixas
Universidade Federal do Rio de Janeiro

Luciano M. de A. Filho

Prof. Dr. Luciano Manhães de Andrade Filho
Universidade Federal de Juiz de Fora

Dedico está dissertação aos meus pais Márcio e Nilza, minha familia e minha namorada Rafaela.

AGRADECIMENTOS

Agradeço primeiramente Aquele que é o meu rochedo, e o meu lugar forte, e o meu libertador; o meu Deus, a minha fortaleza, em quem confio; o meu escudo, a força da minha salvação, e o meu alto refúgio.

Agradeço aos meus pais, Márcio e Nilza, que incentivaram sempre a busca pelo conhecimento, foram meus primeiros educadores e inspiração. Aos meus irmãos que estiveram sempre ao meu lado ensinando, educando e sendo mais que apenas irmãos, amigos. Agradeço a Rafaela, minha namorada, e toda sua família pelo acolhimento.

Agradeço a todos os meus amigos, principalmente aqueles que estiveram ao meu lado nessa jornada do mestrado.

Agradeço ao meu orientador, Eduardo Pestana de Aguiar, por todo ensinamento, atenção e confiança depositada no meu trabalho.

Agradeço também a todos os(as) professores(as), principalmente ao Prof. Álvaro Medeiros, que compartilharam comigo seus conhecimentos técnicos e ajudaram em meu processo de formação.

Agradeço também à banca por toda a disponibilidade e desde já agradeço a contribuição prestada no objetivo de engrandecer esse trabalho.

Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro dado a este trabalho.

"À medida que a complexidade aumenta, as declarações precisas perdem relevância e as declarações relevantes perdem precisão."

(LOTFI ZADEH, 1973, p. 28-44)

RESUMO

Atualmente, grande parte dos sensores utilizados em Internet das Coisas adota tecnologia sem fio, a fim de facilitar a construção de redes de sensoriamento. Neste sentido, a classificação do tipo de ambiente no qual estes sensores estão localizados exerce um importante papel no desempenho de tais redes de sensoriamento, uma vez que pode ser utilizada na determinação de níveis mais eficientes de consumo de energia dos sensores que as compõe. Assim, neste trabalho é apresentada a proposição de uma versão estendida do modelo classificador Fuzzy Auto-Organizável, que faz a classificação de ambientes internos a partir de medições do sinal de radiofrequência de uma rede de sensoriamento sem fio em um ambiente real. Foi realizada uma comparação do modelo de classificador original com o modelo proposto nesse trabalho, bem como outros métodos de aprendizado de máquina comuns na literatura. Como métricas foram avaliados: Acurácia média, F-Score, coeficiente Kappa e MSE. Os resultados experimentais mostram que a abordagem proposta obteve alto desempenho na solução do problema apresentado.

Palavras-chave: Internet das Coisas. Sensores sem fio. Classificador Fuzzy Auto-Organizável. Sinal de Radio frequência.

ABSTRACT

Nowadays, a great part of the sensors adopted in IoT use wireless technology to facilitate the construction of sensor networks. In this sense, the classification of the type of environment in which these sensors are located plays an important role in the performance of these sensor networks, since it leads to efficient power consumption when operating the deployed IoT sensors. Thus, this dissertation presents an enhancement in the Self-Organizing Fuzzy Classifier model, which makes the classification of indoor environments from real-time measurements of the radio-frequency signal of a real wireless sensor network. A comparison between the original classifier model and the model proposed in this dissertation was made, as well as other common machine learning methods literature. The evaluated metrics were Accuracy, F-Score, Kappa coefficient, and MSE. The experimental results show that the proposed approach obtained high performance in solving the presented problem.

Keywords: Internet of Things. Wireless Sensor. Self Organising Fuzzy Logic. Radio Frequency Signal.

LISTA DE ILUSTRAÇÕES

LISTA DE GRÁFICOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IoT	*Internet of Things*
RF	Radiofrequência
CTF	Channel Transfer Function
FCF	Frequency Coherence Function
SOF	*Self-Organising Fuzzy Logic Classifier*
SVM	Máquina de Vetor de Suporte
k-NN	k-*Nearest Neighbors*
MLP	*Multilayer Perceptron*
ANN	*Artificial Neural Networks*
FRB	*Fuzzy Rule-Based*
EDA	*Empirical Data Analysis*
PDF	*Probability Density Function*
RSSI	*Received Signal Strength Indicator*
FDP	Função Densidade de Probabilidade
CSI	*Channel State Information*
OISVM	*Online Independent Support Vector Machine*
FPSOGSA	Otimização de Enxame de Partículas e Algoritmo de Pesquisa Gravitacional
FFNT	Rede Neural de *Feed Forward*
WKNN	k-*Nearest Neighbor Algorithm*
IPS	Sistema de Posicionamento Interno
FK-NN	k-*Fuzzy Nearest Neighbors*
GRNN	Rede Neural de Regressão Generalizada
RBFN	Rede de Funções de Base Radial
RX	*Receiver*
TX	*Transmitter*
DUT	*Device Under Test*
VNA	*Vector Network Analyzer*

LISTA DE SÍMBOLOS

$\sim$	Similaridade
$\in$	Pertence
$\exists$	Existente
$\approx$	Aproximadamente
$\neq$	Diferente
$<$	Menor que
$>$	Maior que
$\leq$	Menor ou igual que
$\equiv$	Equivalente

SUMÁRIO

1 INTRODUÇÃO

O considerável aumento na aplicação de internet das coisas (do inglês *Internet of Things* - IoT) tem revolucionado o campo de telecomunicações. E, devido a isso, diversos trabalhos vêm sendo publicados com o uso de internet das coisas (2, 3, 4). Essas aplicações desempenham um papel cada vez mais importante em várias áreas, tais como: saúde, agricultura, monitoramento de ambiente, medição inteligente, dentre tantas outras. Utilizam-se sensores inteligentes (do inglês *smart sensors*) que, muitas vezes, são sem fio, como nos equipamentos de comunicações via infravermelho, bluetooth, comunicação via micro-ondas, comunicação via satélite e radiofrequência (RF).

Neste contexto, redes de sensores baseadas em comunicação via RF vêm sendo largamente utilizadas. Uma aplicação eficiente desses sensores permite um ajuste adequado do consumo de energia do sensor de acordo com o canal de propagação de radiofrequência (RF), possibilidade de miniaturização de componentes (como possibilitado por sensores MEMS) e tecnologias computacionais incorporadas (5).

Com a maior mobilidade trazida pela ausência de conexões físicas em redes de sensores sem fio, esses dispositivos podem ser utilizados em diversos ambientes. Em muitos casos, a determinação do tipo de ambiente onde o sensor está localizado desempenha um papel importante na eficiência da rede de sensores, uma vez que permite um ajuste mais adequado do consumo de energia dos sensores que a compõe.

Diversos trabalhos na literatura têm sido propostos com o intuito de se classificar, de forma automática, o tipo de ambiente que circunda um determinado sensor. Nela os ambientes são categorizados como: internos ou externos (6, 7, 8, 9, 10). No entanto, existem algumas variações em que os ambientes são categorizados como: internos, semi-externos e externos (11).

Existe na literatura diversos modelos de classificadores com arquiteturas bem diferentes. Em geral, as abordagens existentes podem ser categorizadas em dois tipos principais, sendo eles: *offline* (12, 13) e *online* (14, 15, 16). Dependendo da intensidade do sinal RF, o objeto pode se encontrar mais próximo de um ambiente do que de outro, o classificador Fuzzy Auto-Organizável (do inglês *Self-Organising Fuzzy Logic Classifier - SOF*) representa uma abordagem *online* promissora na solução do problema em questão. Isso porque, além de tratar tal variação, propicia uma adaptabilidade a problemas distintos, através da escolha de métricas de distância mais adequadas ao caso em estudo.

A aplicação de uma abordagem evolutiva para a solução desse problema, se deve pelo fato de o mesmo ser capaz de aprender continuamente, de modo a seguir as mudanças nos dados sem necessitar de um conhecimento prévio ou de qualquer especialista. Pois, os sistemas em evolução são inspirados na ideia de evolução do modelo, em um ambiente de mudança. Eles usam herança e mudança gradual com o objetivo de aprendizado e

adaptação, auto-organização, incluindo a evolução da estrutura do sistema, a fim de se adaptar ao ambiente atualizando sua estrutura e ajustando seus parâmetros.

1.1 OBJETIVOS

O presente trabalho propõe uma extensão do classificador SOF, com três novas métricas de distância, sendo elas distância de Hamming, Minkowski e Manhattan, além da aplicação do modelo para fazer a classificação de ambientes usando para tal uma base de dados composta por atributos de frequência e parâmetros S.

1.2 CONTRIBUIÇÕES

Podemos elencar como as principais contribuições dessa dissertação:

a) A proposição de uma versão estendida do SOF (17);

b) Utilização do modelo proposto em um problema de classificação de ambientes internos usando uma base de dados com atributos de frequência e parâmetros S do canal de comunicação RF;

c) Teste de métricas de distância com a finalidade de se determinar qual mais se adequa ao problema proposto;

d) Comparações com outros classificadores presentes na literatura, sendo eles: Fuzzy; *SVM linear*, *SVM RBF*; *Decision Tree*, *Random Forest*; *Nearest Neighbors*, *MLP-ANN*; *AdaBoost*, *Naive Bayes* e *QDA*.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho é organizado de modo que o capitulo 2 aborda a formulação do problema, de modo a tratar de classificação e da abordagem proposta e uma fundamentação teórica sobre ambiente *wireless* e parâmetros S. O capitulo 3 descreve o modelo utilizado, bem como sua fundamentação teórica, as distâncias usadas no modelo original e as propostas, os estágios do modelo, e uma breve discussão sobre a influência da granularidade para a abordagem. O capitulo 4 é descrito sobre a base de dados e o resultado das métricas propostas e as comparações com outros classificadores presentes na literatura. O último capitulo descreve as conclusões, observações finais do trabalho e a apresentação de trabalhos futuros.

2 SOF APLICADO PARA FAZER CLASSIFICAÇÃO DE AMBIENTES INTERNOS

Métodos de inteligência computacional vêm sendo amplamente empregados para tomada de decisão, reconhecimento de padrões, otimização, agrupamento e classificação (18). Os modelos de classificação supervisionada atribuem rótulos de classe a amostras de dados de entrada. O rótulo indica a qual classe pertence aquela amostra, dado um conjunto de classes (19).

Ao lidar com problemas de classificação, como exemplo de classificadores, temos: máquina de vetor de suporte (do inglês *Support Vector Machine* - SVM) (20), vizinhos mais próximos (do inglês k-*Nearest Neighbors* - k-NN) (21) e redes neurais multicamadas perceptron (do inglês *Multilayer Perceptron Artificial Neural Networks - MLP-ANN*) (22). Esses modelos podem ser de aprendizado supervisionado, semi-supervisionado e não supervisionado (23).

Pesquisas são publicadas atualmente com novos modelos, como, por exemplo: o classificador de aprendizado não supervisionado com Orientação de Direção Auto-Organizável (do inglês *Self-Organised Direction Aware - SODA* (24) e o classificador com aprendizado supervisionado SOF proposto em (17). Esses métodos novos surgem para suprir algumas deficiências trazidas nos modelos tradicionais de aprendizado. Como exemplo para tais deficiências, para aprendizado supervisionado, tem-se:

a) Os modelos se baseiam em suposições anteriores de especialistas e parâmetros predefinidos no modelo, para obter um bom desempenho;

b) Suas estruturas não têm capacidade de se auto-evoluir de acordo com as mudanças nos padrões dos dados.

No caso do classificador SOF, o mesmo se baseia na teoria dos conjuntos fuzzy, introduzida em 1965, por Zadeh (25). Esse classificador é não-paramétrico e baseado em regras (do inglês *Fuzzy Rule-Based - FRB*) que identificam protótipos de dados a partir de uma análise empírica da base de dados observada (do inglês *Empirical data analysis - EDA*) (26). O sistema fuzzy empírico é adequado para o processamento de dados de *streaming* (26).

2.1 DESAFIOS DOS CLASSIFICADORES NA ATUALIDADE

Com um crescimento exponencial na escala e na complexidade dos dados gerados por sensores, pessoas, sociedade, indústria e etc, classificadores baseados em sistemas de regras fuzzy são cada vez mais vistos como um recurso inexplorado, que oferece novas oportunidades para extrair informações agregadas e informar a um tomador de decisão.

De fato, os métodos tradicionais para projetar conjuntos fuzzy foram desenvolvidos na época em que os dados não estavam em tão grande escala. Desse modo, supunha-se que estivessem principalmente disponíveis *offline*, sem *streaming* e possivelmente estacionários (27). É um grande desafio projetar modelos fuzzy tradicionais a partir de uma grande quantidade de dados não identificadas ou, por exemplo, *big data*.

Percebe-se que, em telecomunicações, são gerados dados de alta dimensionalidade (28). Isso se dá pelo crescimento sem precedentes de dispositivos móveis, aplicativos e serviços na infraestrutura de rede móvel e sem fio. A pesquisa e desenvolvimento de modelos que dão suporte aos altos volumes de dados, à extração de informações e uma análise refinada, atrelada a um gerenciamento ágil, se faz muito necessária. Nesses modelos, deve-se levar em consideração as limitações computacionais e de energia, pois, muitas vezes eles são empregados em dispositivos móveis e compactos de IoT (29).

Com a mobilidade trazida pela ausência de conexões físicas em redes de sensores sem fio, esses dispositivos podem ser utilizados em diversos ambientes. Em muitos casos, a determinação do tipo de ambiente, onde o sensor está localizado desempenha um papel importante na eficiência da rede de sensores (30).

Diversos trabalhos na literatura usam abordagens de classificação que caracterizam o ambiente como interno ou externo. Em (31), é apresentado um classificador binário, que utiliza o indicador de intensidade do sinal recebido (do inglês *Received Signal Strength Indicator - RSSI*) dos pontos de acesso. Em (32), os autores propõem utilizar a estimativa de Kernel, forma não-paramétrica para estimar a Função Densidade de Probabilidade (FDP), para reduzir as diferenças nas estimativas de probabilidade no algoritmo de localização bayesiana. Os autores em (33) apresentam o RADAR, um algoritmo simples de busca linear, baseado em radiofrequência para a classificação do ambiente. Por ultimo, os autores em (34) propõem a utilização de um algoritmo de navegação com base nos acelerômetros, giroscópios, posição inicial e os valores de direção para fazer classificação do ambiente onde o dispositivo se encontra.

Existem também trabalhos que fazem a classificação usando técnicas de aprendizado de máquina, como, por exemplo, em (35), em que os autores usam o *Naive Bayes*, *BayesNet*, aprendizagem ponderada localmente (do inglês *Locally Weighted Learning - LWL*) e otimização mínima sequencial (do inglês *Sequential Minimal Optimization - SMO*) (35). Em (36), usam métodos de análise de grande conjuntos de sinais WiFi, aplicando *BayesNet* para a classificação. Em (37), foi proposto uma nova abordagem de aprendizado profundo online, chamada de OSDELM, para fazer a classificação de um ambiente interno altamente dinâmico. Em (38), os autores propõem um novo sistema de posicionamento interno baseado em impressões digitais, usando aprendizado profundo, denominado DeepFi. Em (39), baseando-se no indicador de intensidade do sinal recebido (RSSI) dos sinais Wi-Fi, foi proposto um método de localização interna usando um

classificador *Online Independent Support Vector Machine - (OISVM)*. Em (40), os autores usam máquinas de aprendizado profundo (DNN, DBN e GB-DBN) para aumentar a precisão da estimativa e reduzir o erro de generalização em ambiente interno dinâmico. Em (41), com base na intensidade do sinal Wi-Fi, os autores propuseram um modelo híbrido Fuzzy de otimização de enxame de partículas e algoritmo de pesquisa gravitacional (FPSOGSA) com redes neurais. Em (42), a distância de Manhattan é introduzida no algoritmo K-nearest neighbor algorithm (WKNN) para distinguir a influência de diferentes nós de referência na classificação de ambientes interno com base em Wi-Fi. Em (43), é proposto um novo método baseado em uma abordagem de aprendizado de máquina de dois estágios em cascata para classificação de ambientes internos. Em (44), usa-se o Sistema de Posicionamento Interno (IPS) no classificador Fuzzy *K-Nearest Neighbor (FK-NN)*, que é uma combinação do algoritmo Fuzzy e K-NN, para aumentar a precisão da posição do objeto com base nos dados de aprendizagem. Em (45), é feito uma comparação do desempenho de 11 tipos diferentes de funções de treinamento usadas para treinar a Rede Neural de *Feed Forward* (FFNT) proposta para a localização de destino interno baseada em RSSI. Em (46), faz uma avaliação na comparação do desempenho da localização de várias arquiteturas de aprendizado supervisionado, como Rede Neural de Regressão Generalizada (GRNN), Perceptron de várias camadas (MLP), Rede de funções de base radial (RBFN) e Rede Neural de alimentação direta (FFNT) , baseado-se em um problema de localização interna. Por último, em (30), os autores propõem a utilização de *Decision Tree*, *Support Vector Machine* e *k-Nearest Neighbor*, baseando-se em medições do sinal de radiofrequência.

Porém, os modelos tradicionais sofrem com problemas de dimensionalidade e *streaming*. Enquanto os modelos supracitados de aprendizado de máquina sofrem com os problemas enumerados acima, além de sofrer, também, com a dimensionalidade e a classificação em tempo real. Por isso, nessa dissertação é proposta a utilização do SOF para a classificação de ambientes internos, visto que este lida muito bem com dados de alta dimensionalidade, classificação *online* e sobretudo é um modelo não paramétrico e auto-organizável, tornando-o assim uma abordagem completamente adaptável para diversos problemas.

2.1.1 Ambiente *wireless* e parâmetros S

Entre os grandes marcos da engenharia se encontra a comunicação sem fio (*wireless*), que impacta fortemente a sociedade e as pesquisas de telecomunicações. As comunicações sem fio, como são conhecidas, começaram apenas com o trabalho de Maxwell e Hertz, que lançaram as bases para o entendimento da transmissão de ondas eletromagnéticas. No entanto, não demorou muito para o trabalho inovador de Tesla demonstrar a transmissão de informações por essas ondas - em essência, o primeiro sistema de comunicações sem fio. No final de setembro de 1898, Marconi fez sua demonstração bem divulgada de tais comunicações de um barco para a Ilha de *Wight* no Canal da Mancha. Vale ressaltar

que, embora Tesla tenha sido o primeiro a ter sucesso nesse importante empreendimento, Marconi teve as melhores relações públicas e é amplamente citada como a inventora das comunicações sem fio, recebendo o prêmio Nobel em 1909. Na história, o termo comunicação sem fio foi utilizado inicialmente em meados de 1920 para definir o sistema que depois passou a ser chamado de rádio transmissor. Em torno de 1980, o termo ressurgiu sendo utilizado para qualquer tipo de tecnologia que fizesse comunicações sem o uso de fios, o que se mantém até hoje (47). A possibilidade de falar em qualquer lugar e a qualquer hora trouxe ao mundo mudanças de hábitos de trabalho, principalmente na maneira como os indivíduos têm se comunicado.

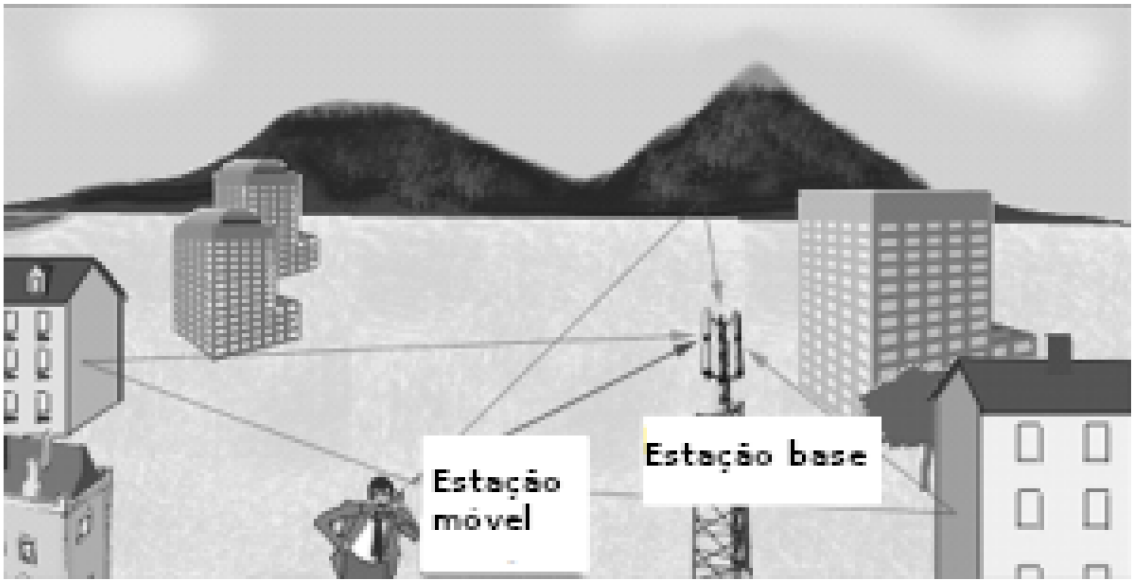
Por muito tempo, a comunicação sem fio foi associada a telefonia celular, pois este é um grande mercado do segmento e tem forte impacto no dia a dia das pessoas. Diante disso, tem-se como exemplo a possibilidade de uma pessoa trabalhar respondendo a um email sentada em um loja no *shopping* e tomando seu café. Porém, diversas outras aplicações utilizam dessa tecnologia, como o teclado e o *mouse* sem fio de computadores. Uma outra abordagem que vem se utilizando muito da comunicação sem fio são os sistemas de posicionamento, onde é monitorado a localização dos caminhões que possuem mercadorias identificadas por etiquetas de radiofrequência (RF) (48). Toda essa infinidade de aplicações aumentam a cada dia os desafios de técnicos e engenheiros de tornarem a comunicação mais estável e confiável.

2.1.2 Canal *wireless*

Para uma comunicação sem fio, o meio de transmissão entre o transmissor TX e o receptor RX é o canal de rádio. O sinal pode ir do TX ao RX através de vários caminhos de propagação, comumente chamado de multipercursos. Os vários objetos iterativos em um dado ambiente: externo (casas, prédios, montanhas e etc) ou internos (janelas, paredes e etc), fazem com que o sinal seja por muitas vezes refletido ou difratado. São essas iterações que geram os diferentes caminhos de propagação. Conforme mostrado na Figura 1, cada um desses caminhos têm características distintas, como: amplitude, atraso (tempo de execução do sinal), direção de partida do TX, direção de chegada do RX; mas dentre essas características, a mais importante é que os componentes têm mudanças de fase diferentes entre si.

A análise dos sinais de RF transmitidos no ambiente pode fornecer uma forte indicação para a classificação desse ambiente. Dado um sinal transmitido $X(f)$ em um canal RF interno, como ele se propaga com múltiplos caminhos (como mostrado na Figura 2, onde o sinal transmitido chega ao RX por dois caminhos de propagação diferente) criando réplicas no sinal receptor $Y(f)$. A função de transferência de canais (do inglês *Channel Tranfer Function - CTF*) é a resposta em frequência do canal de múltiplos caminhos *wireless* que afeta o sinal transmitido como $Y(f) = H(f)X(f)$ (30). $H(f)$

Figura 1 – Propagação com multipercursos



Fonte: (51).

pode ser representado como sendo a superposição dos ganhos associados aos diversos componentes (réplicas) do sinal de RF presentes no ambiente e pode ser expressa em (49):

$$H(f) = \sum_{l=1}^{L} a_l \exp[-j(2\pi f\tau - \theta_l)] \tag{2.1}$$

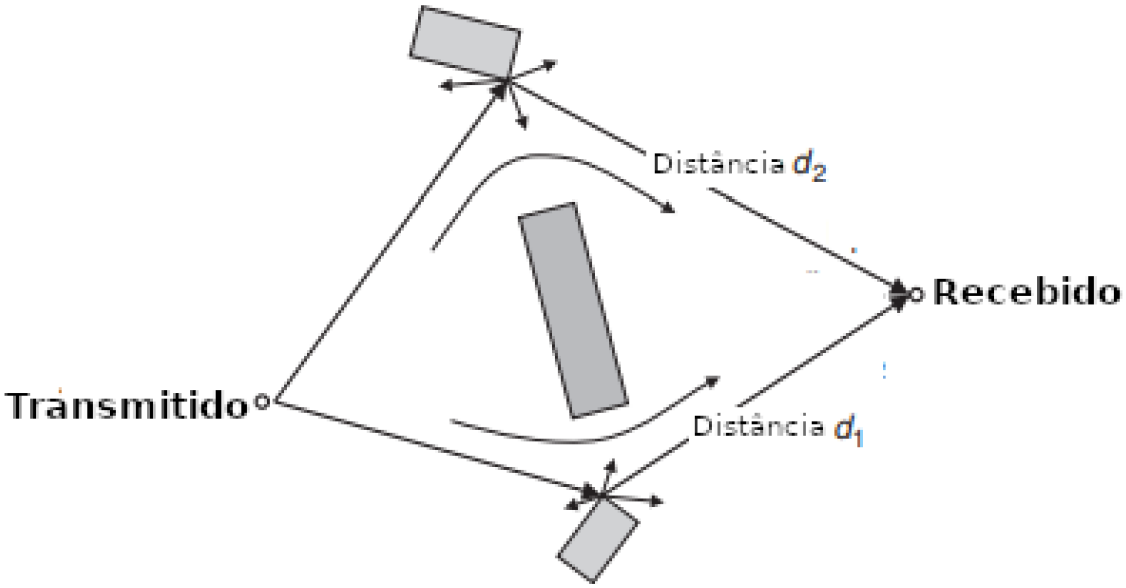
em que a_l, τ e θ_l são as componentes de amplitude, atraso e fase dos componentes do sinal, respectivamente. L é o número total de componentes dos vários caminhos, f é a frequência do canal classificado. A CTF $H(f)$ é considerada uma assinatura do sinal de RF, sendo única para cada posição no espaço do ambiente (50).

2.1.3 **Parâmetros** S

Em toda a literatura técnica sobre sistemas de RF, a representação do parâmetro S ou espalhamento (do inglês *scattering*) desempenha um papel fundamental. Isso se dá pelo fato de as características práticas do sistema não poderem mais ser analisadas por meio de medições simples de curto circuito ou circuito aberto, como costuma ser feito em aplicações de baixa frequência. Tendo isso em vista, deve-se lembrar que quando se curto-circuita um fio, é gerada uma indutância que pode ser de magnitude substancial em alta frequência. Por outro lado, um circuito aberto leva ao carregamento capacitivo do terminal. Tais características causarão reflexos indesejáveis de tensão e/ou corrente, levando a oscilações que podem resultar na destruição do dispositivo sob teste (do inglês *device under test* - DUT) (51).

Os parâmetros S são descritores de ondas de energia que permitem definir as relações entre a entrada e a saída de uma rede em termos das ondas (incidentes e refletidas)

Figura 2 – Modelo de dois caminhos invariante no tempo



Fonte: (51).

de energia normalizada. O instrumento capaz de medir tensão em termos da magnitude e da fase do sinal transmitido é chamado analisador de rede vetorial (do inglês *vector network analyzer* - VNA). Os dispositivos VNA de duas portas contam com 4 parâmetros S, sendo eles: S_{11}, S_{12}, S_{21} e S_{22}.

O parâmetro S_{11} é o coeficiente de reflexão da porta 1, ou seja, a relação entre a onda de saída b_1 e a onda incidente a_1 em uma medida direta com a porta 2. O parâmetro S_{21} é o coeficiente de transmissão para a frente, definido como a razão entre a onda de saída b_2 e a onda incidente a_1 em uma medição à frente com a porta 2. O parâmetro S_{12} é o coeficiente de transmissão reversa, definido como a razão entre a onda de saída b_1 e a onda incidente a_2 em uma medição à frente com a porta 1. O parâmetro S_{22} é o coeficiente de reflexão da porta 1, ou seja, a relação entre a onda de saída b_1 e a onda incidente a_1 em uma medida direta com a porta 1. Na Figura 3, pode-se verificar o fluxo de sinal para uma medição de 2 portas.

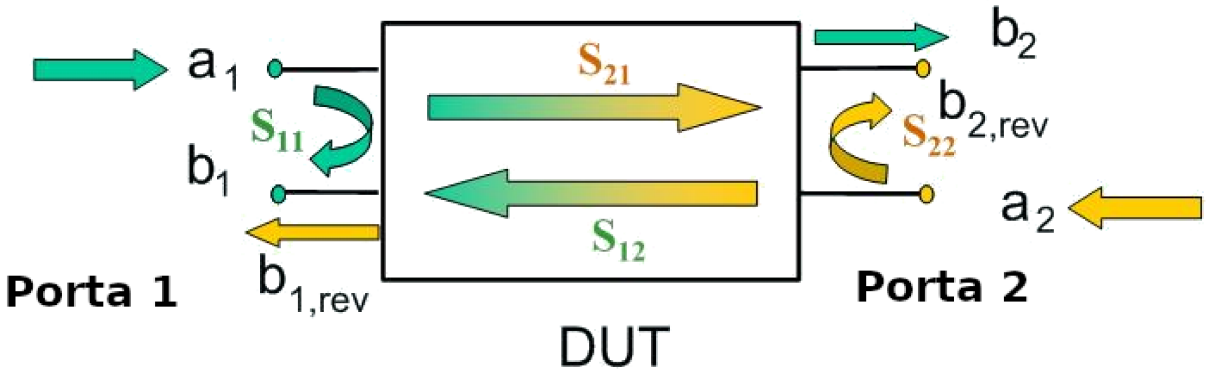
Para uma melhor compressão, encontram-se os termos abaixo:

$$S_{11} = \frac{b_1}{a_1}\Big|_{a_2=0} \equiv \frac{onda\ de\ energia\ refletida\ na\ porta\ 1}{onda\ de\ energia\ incidente\ na\ porta\ 1} \tag{2.2}$$

$$S_{21} = \frac{b_2}{a_1}\Big|_{a_2=0} \equiv \frac{onda\ de\ energia\ transmitida\ na\ porta\ 2}{onda\ de\ energia\ incidente\ na\ porta\ 1} \tag{2.3}$$

$$S_{22} = \frac{b_2}{a_2}\Big|_{a_1=0} \equiv \frac{onda\ de\ energia\ refletida\ na\ porta\ 2}{onda\ de\ energia\ incidente\ na\ porta\ 2} \tag{2.4}$$

Figura 3 – Parâmetros S do analisador de rede vetorial.



Fonte:`https://www.rohde-schwarz.com/webhelp/zvb_html_usermanual_en/system_overview/measurement_parameters/s-parameters.htm`.

$$S_{12} = \frac{b_1}{a_2}\Big|_{a_1=0} \equiv \frac{onda\ de\ energia\ transmitida\ na\ porta\ 1}{onda\ de\ energia\ incidente\ na\ porta\ 2} \tag{2.5}$$

Neste trabalho, além da frequência do sinal, serão usados também os parâmetros S_{11} e S_{21} para a classificação.

3 CLASSIFICADOR FUZZY AUTO-ORGANIZÁVEL

O classificador SOF é um modelo proposto em (17), não-paramétrico e é baseado em regras que identificam protótipos de dados a partir de uma base de dados observada. Utilizando-se, para tanto, um processo de treinamento *offline*, que emprega estes protótipos, o mesmo constrói um sistema de inferência fuzzy baseado em regras *AnYa* de ordem 0 (52). Uma vez preparado o treinamento *offline*, inicia-se o segundo estágio que é o *online*. Nesse estágio, o SOF é capaz de aprender continuamente a partir do fluxo de dados, de forma a seguir os padrões de mudança nos dados, atualizando a estrutura do sistema e os meta-parâmetros recursivamente.

Esses meta-parâmetros são derivados diretamente dos dados, sendo que ao se alterar o nível de granularidade do modelo, é possível fazer um *trade-off* entre desempenho e eficiência computacional. Essa possibilidade de alterar a granularidade torna o classificador capaz de lidar com uma ampla variedade de problemas com necessidades específicas (17).

Além disso, na abordagem, o SOF permite também a escolha da métrica de distância a ser empregada no algoritmo. Com isso, o modelo se torna altamente adaptável a diversas aplicações. No modelo original proposto em (17), são usadas as distâncias Euclideana, Cosseno e Mahalanobis.

3.1 SISTEMA FUZZY BASEADO EM REGRAS AnYa DE ORDEM 0

Sistemas do tipo Fuzzy Baseado em Regras (do inglês *Fuzzy Rule-Based - FRB*) foram introduzidos em (52) como uma abordagem alternativa para sistemas amplamente utilizados do tipo Takagi-Sugeno (53) ou Mamdani (54). Fazendo uma comparação com os predecessores, a parte antecedente das regras fuzzy do tipo *AnYa* é simplificada para uma forma vetorial mais compacta, objetiva e não paramétrica, não tendo a necessidade de definir funções de associação *ad hoc* (17). O formato de uma regra AnYa de ordem 0 é:

$$SE\ (x \sim p_1)\ OU\ (x \sim p_2)\ OU...OU\ (x \sim p_N)\quad ENTÂO\ (classe) \tag{3.1}$$

em que x é um vetor de entrada; "$\sim$" é a similaridade; p_i é o enésimo protótipo da classe; N é o número de protótipos identificados a partir das amostras de dados da classe. A rotulação de cada amostra de dado especifica pode ser gerada de diferentes maneiras de defuzzificação (55), como vencedor leva tudo (do inglês *winner-takes-all*), poucos vencedores levam tudo (do inglês *few-winners-take-all*), média ponderada (do inglês *fuzzily weighted average*), etc. No SOF é usado *winner-takes-all* (17).

3.2 OPERADORES EMPÍRICOS DE ANÁLISE DE DADOS

O SOF é um classificador não-paramétrico que emprega a análise empírica de dados (do inglês *Empirical data analysis - EDA*) (26) para conhecer as propriedades do conjunto e a distribuição mútua dos dados, bem como a proximidade relativa das amostras no espaço de dados.

EDA é uma análise de dados e seus fundamentos, com ele é possível estimar as propriedades do conjunto de dados, baseando-se inteiramente nas observações empíricas da amostras de dados e na proximidade desses pontos no espaço de dados (57). Uma característica distintiva da abordagem EDA é que ela não é limitada por suposições prévias sobre o modelo de geração de dados (26). As propriedades são consideradas informações importantes na análise de padrões e são derivadas dos dados diretamente, de forma discreta, em contraste com a abordagem tradicional, por exemplo a conhecida função de densidade de probabilidade (do inglês *Probability Density Function* - PDF), na qual é assumido previamente na forma contínua e estimada a partir dos dados posteriormente (57). No modelo, são descritos três quantificações de EDA: proximidade cumulativa, densidade unimodal e densidade multimodal.

Primeiramente, assume-se um conjunto/fluxo de dados dentro de um espaço de dados real $\mathbf{R}^N$ (sendo N a dimensionalidade do espaço) observado na K-ésima instância denotada por $\{\mathbf{x}\}_K = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K\}$ e pode ser apresentado na forma matricial em (3.2), em que $\mathbf{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,N}]^T \in \mathbf{R}^N$, o índice $i = 1, 2, ..., K$ denota a instância na qual a i-ésima amostra de dados x_i chegou. O conjunto de dados exclusivos que são classificados é indicado como: $\{\mathbf{u}\}_{U_K} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{U_K}\}$ ($\mathbf{u}_i = [u_{i,1}, u_{i,2}, ..., u_{i,N}]$, $\{\mathbf{u}\}_{U_K} \subseteq \{\mathbf{x}\}_K$, $U_K \leq K$ U_K é o número de amostras de dados exclusivas); e a frequência de ocorrência são: $\{f\}_{U_K} = \{f_1, f_2, ..., f_{U_K}\}$ ($\sum_{i=1}^{U_K} f_i = K$).

$$\mathbf{X}_K = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K] = \begin{bmatrix} x_{1,1} & x_{2,1} & ... & x_{K,1} \\ x_{1,2} & x_{2,2} & ... & x_{K,2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1,N} & x_{2,N} & ... & x_{K,N} \end{bmatrix} \tag{3.2}$$

3.2.1 **Proximidade cumulativa**

Proximidade cumulativa π é introduzida em (56). Em (26) é derivada empiricamente dos dados observados, sem conhecimento prévio ou suposições prévias. A proximidade cumulativa da amostra de dados $\mathbf{x}_i$ é expressa como:

$$\pi_k(\mathbf{x}_i) = \sum_{j=1}^{K} d^2(\mathbf{x}_i, \mathbf{x}_j);\ i = 1, 2, ..., K \tag{3.3}$$

em que $d(\mathbf{x}_i, \mathbf{x}_j)$ denota a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$, o tipo de medida de distância pode ser escolhido de acordo com os dados. É importante notar que a distância quadrada média entre as duas amostras de dados dentro de $\{\mathbf{x}\}_K$ pode ser expressada como $\overline{d}_k = \frac{1}{K^2} \sum_{i=1}^{K} \pi_K(\mathbf{x}_i)$.

3.2.2 Densidade unimodal

Densidade unimodal D, foi introduzida em (26). E é usada como um indicador do principal padrão de dados na estrutura EDA. A densidade unimodal em $\mathbf{x}_i$ é expressa como:

$$D_K(\mathbf{x}_i) = \frac{\sum_{l=1}^{K} \pi_K(\mathbf{x}_l)}{2K\pi_K(\mathbf{x}_i)} = \frac{\sum_{l=1}^{K}\sum_{j=1}^{K} d^2(\mathbf{x}_i, \mathbf{x}_j)}{2K \sum_{j=1}^{K} d^2(\mathbf{x}_i, \mathbf{x}_j)};\ i = 1, 2, ..., K \tag{3.4}$$

3.2.3 Densidade multimodal

Densidade multimodal D^{MM} (26, 57) é estimado na amostra de dados exclusiva $\mathbf{u}_i$ como a soma ponderada de sua densidade unimodal pelos tempos de ocorrência repetidos, expressos como:

$$D^{MM}(\mathbf{u}_i) = f_i D_K(\mathbf{u}_i) = f_i \frac{\sum_{l=1}^{K} \pi_K(\mathbf{x}_l)}{2K\pi_K(\mathbf{u}_i)};\ i = 1, 2, ..., U_K \tag{3.5}$$

3.2.4 Forma de cálculo recursivo

Quando estão sendo processados os dados de forma contínua (*streaming*), as formas de cálculo recursivo das quantidades não-paramétricas do EDA desempenham um papel importante, por garantirem que as técnicas sejam eficientes, tanto na memória, quanto no esforço computacional. São usadas para cálculo de distância as de Mahalanobis, euclideana ou cosseno, dentre outras, como em (26, 24). Abaixo será descrito o cálculo da expressão recursiva usando a distância Mahalanobis:

A expressão para o cálculo da distância Mahalanobis é (17):

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j) C_K^{-1} (\mathbf{x}_i - \mathbf{x}_j)^T};\ (\ i, j = 1, 2, ..., K) \tag{3.6}$$

para isso a expressão de cálculo recursivo é dada como:

$$\pi_k(\mathbf{x}_i) = K\left((\mathbf{x}_i - \mu_K) C_K^{-1} (\mathbf{x}_i - \mu_K)^T + X_K - \mu_K\, C_K^{-1}\, \mu_K^T\right) \tag{3.7}$$

em que C_K é matriz de covariância dada por (17):

$$C_K = \frac{1}{K-1}\sum_{l=1}^{K}(\mathbf{x}_i - \mu_K)^T(\mathbf{x}_i - \mu_K);\ X_K = \frac{1}{K}\sum_{l=1}^{K}\mathbf{x}_i\ C_K^{-1}\ \mathbf{x}_l^T;\ \mu_K = \frac{1}{K}\sum_{l=1}^{K}\mathbf{x}_l \tag{3.8}$$

A matriz de covariância C_K e a média global μ_K são atualizadas recursivamente como:

$$\mu_K = \frac{K-1}{K}\mu_{K-1} + \frac{1}{K}\ \mathbf{x}_K;\ \mu_1 = \mathbf{x}_1 \tag{3.9}$$

$$\mathbf{X}_K = \frac{K-1}{K}\mathbf{X}_{K-1} + \frac{1}{K}\ \mathbf{x}_K^T\mathbf{x}_K;\ \mathbf{X}_1 = \mathbf{x}_1^T\mathbf{x}_1 \tag{3.10}$$

$$C_K = \frac{K}{K-1}\left(\mathbf{X}_K - \mu_K^T\mu_K\right) \tag{3.11}$$

A soma de todas as proximidades cumulativas é dada por (58):

$$\sum_{l=1}^{K}\pi_K(\mathbf{x}_l) = 2K^2(X_K - \mu_K C_K^{-1}\mu_K^T) = 2K^2N \tag{3.12}$$

Consequentemente, a densidade unimodal em $\mathbf{x}_i$ é calculada recursivamente como:

$$D_K(\mathbf{x}_i) = \frac{2K^2(X_K - \mu_K C_K^{-1}\mu_K^T))}{2K.K\left((\mathbf{x}_i - \mu_K)C_K^{-}1(\mathbf{x}_i - \mu_K)^T + X_K - \mu_K C_K^{-1}\mu_K^T))\right)} \tag{3.13}$$

$$= \frac{1}{1 + \frac{(\mathbf{x}_i - \mu_K)C_K^{-1}(\mathbf{x}_i - \mu_K)^T}{N}} \tag{3.14}$$

Baseando-se nas equações (3.12) e (3.14), pode-se perceber que, se for usada a distância Mahalanobis, é possível calcular recursivamente a proximidade cumulativa e a densidade das novas amostras de dados mantendo apenas μ_K e $\mathbf{X}_K$ na memória.

Entretanto, tem-se que admitir que nem todos os tipos de medidas de distância permitem uma forma de cálculo recursivo e, para esses casos, as expressões gerais de cálculo recursivo são (59):

$$\pi_K(\mathbf{x}_i) = \pi_{K-1}(\mathbf{x}_i) + d^2(\mathbf{x}_i, \mathbf{x}_K) \tag{3.15}$$

$$\sum_{j=1}^{K}\pi_K(\mathbf{x}_j) = \sum_{j=1}^{K}\pi_{K-1}(\mathbf{x}_j) + 2\pi_K(\mathbf{x}_K) \tag{3.16}$$

$$D_K(\mathbf{x}_i) = \frac{\sum_{j=1}^{K} \pi_{K-1}(\mathbf{x}_j) + 2\pi_K(\mathbf{x}_K)}{2K\left(\pi_{K-1}(\mathbf{x}_i) + d^2(\mathbf{x}_i, \mathbf{x}_K)\right)} \tag{3.17}$$

Geralmente, a distância euclideana é a métrica de distância mais usada; sua eficácia e validade como medida de distância, na maioria dos casos, são garantidas. Entretanto, se os dados seguirem uma distribuição gaussiana, ou alguma distribuição semelhante, a distância mais adequada seria a Mahalanobis (26). Para problemas de alta dimensão, a distância mais eficaz e frequentemente usada é a cosseno, pois ela é livre de problemas com a dimensionalidade (quando a dimensionalidade aumenta e o volume do espaço aumenta tão rapidamente que os as distância disponíveis se tornam aproximadamente iguais, tornando-se difícil para o modelo agrupar significativamente os dados) (60, 61, 62).

3.3 MÉTRICAS DE DISTÂNCIA USADAS NO MODELO ORIGINAL

Nesta seção, serão tratadas as métricas de distância do modelo original, que são Mahalanobis, Euclideana e Cosseno.

3.3.1 **Distância Euclideana**

A distância euclideana, nomeada em homenagem ao matemático grego Euclides (cerca de 325 aC a 265 aC), entre duas amostras de dados, $\mathbf{x}_i$, $\mathbf{x}_j \in \{\mathbf{x}\}_K$, é calculada com base na seguinte equação (55):

$$d_{euc}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j|| = \sqrt{\sum_{l=1}^{N}(x_{i,l} - x_{j,l})^2} \tag{3.18}$$

A medida Euclideana é a medida mais utilizada. A sua principal vantagem é a simplicidade e eficiência computacional e a principal desvantagem é o fato de atribuir igual peso/importância a cada uma das N dimensões. A distância euclideana é uma métrica completa.

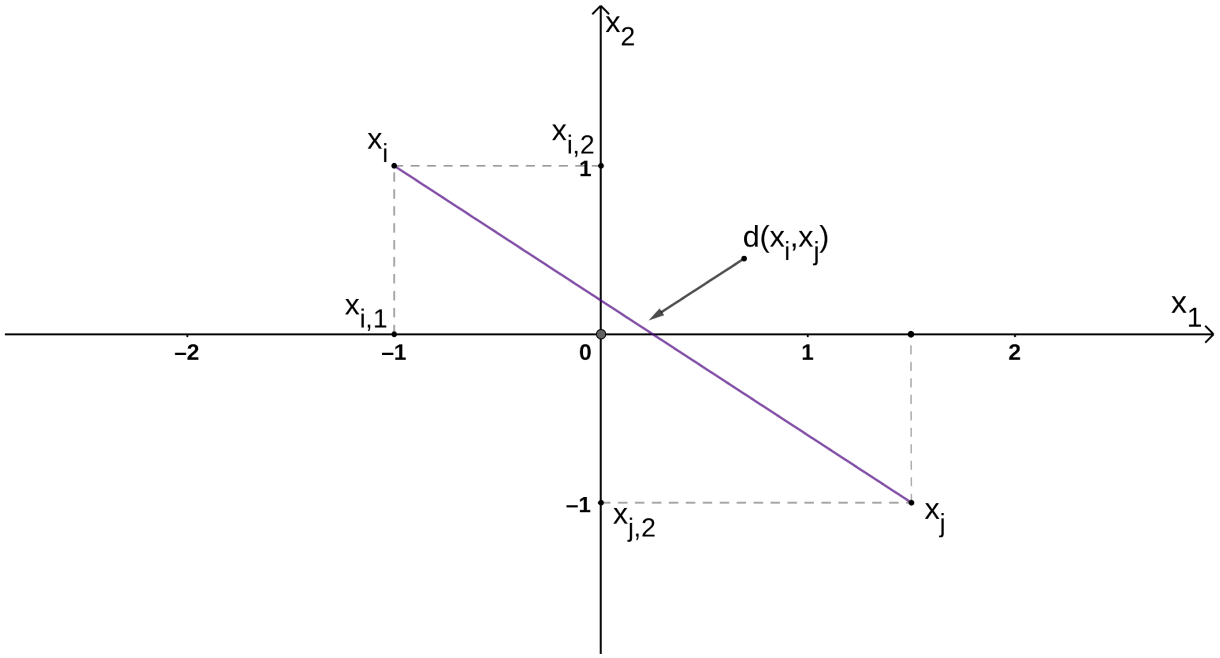
No Gráfico 1, temos um exemplo ilustrativo simples em um espaço 2D. As amostras são: $\mathbf{x}_i = [-1, 1]^T$ e $\mathbf{x}_j = [2, -1]^T$. Aplicando a equação (3.18), temos:

$$d_{euc}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2} \approx 3,2015$$

3.3.2 **Distância Mahalanobis**

Se os dados seguem uma distribuição gaussiana ou distribuições similares, é comum escolher a distância Mahalanobis, pois é a distância que considera os desvios padrão de $\mathbf{x}_i$ e $\mathbf{x}_j$ em uma forma vetorial, através da matriz de covariância C_K.

Gráfico 1 – Distância Euclideana



Fonte: (55).

A distância Mahalanobis pode ser vista como uma extensão da Euclideana. Para o caso especial em que $C_K = \mathbf{I}$, a distância Mahalanobis se torna Euclideana.

3.3.3 **Distância Cosseno**

Em problemas cujas as dimensões são elevadas, é muito comum utilizar a distância cosseno. Essa distância mede o cosseno do ângulo entre dois vetores diferentes de zero. Uma ilustração disso pode ser vista no Gráfico 2.

A distância cosseno entre $\mathbf{x}_i$ e $\mathbf{x}_j$ é formulada como:

$$d_{cos}(\mathbf{x}_i, \mathbf{x}_j) = \cos(\theta_x^{i,j}) \tag{3.19}$$

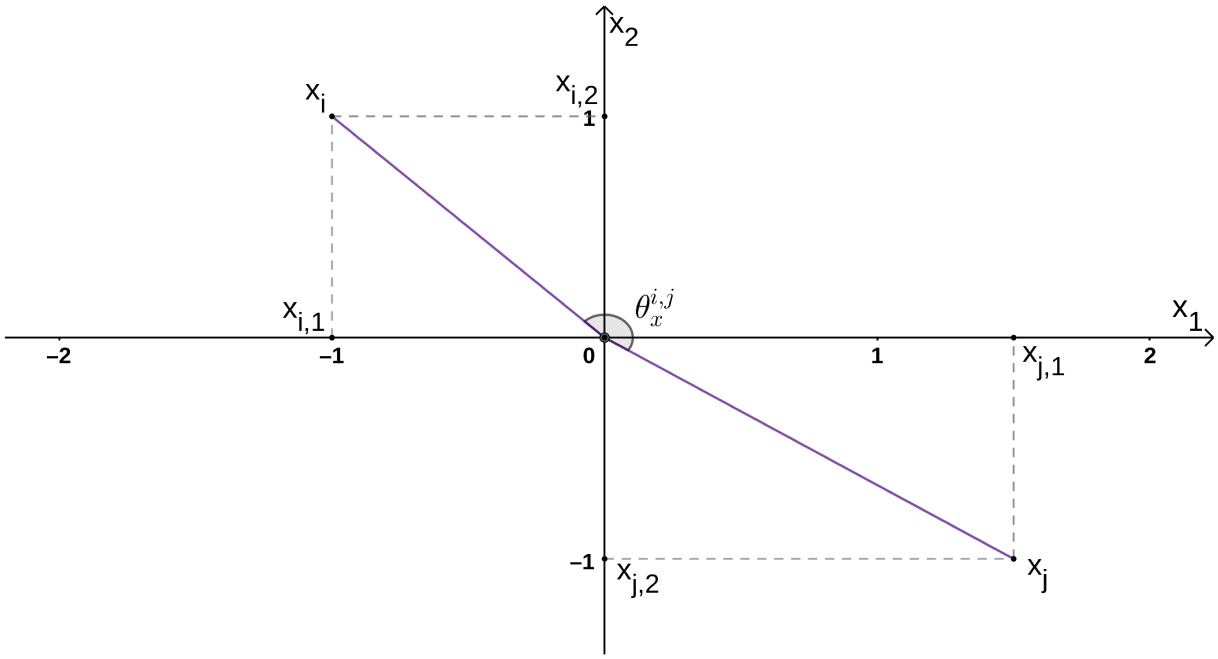
em que $\theta_x^{i,j}$ representa o angulo entre $\mathbf{x}_i$ e $\mathbf{x}_j$ em um espaço Euclideano. Com isso, é possível representar o produto interno dos dois vetores como:

$$d_{cos}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|} \tag{3.20}$$

em que $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{l=1}^{N} x_{i,l} x_{j,l}$ e $\|\mathbf{x}_i\| = \sqrt{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}$.

Após algumas manipulações, é possível chegar a uma formula de distância cosseno

Gráfico 2 – Distância Cosseno



Fonte: (55).

frequentemente usada (24):

$$d_{cos}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \cos\left(\theta_x^{i,j}\right) = \frac{1}{2}\left\| \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} - \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|} \right\|^2 \tag{3.21}$$

3.4 MÉTRICAS DE DISTÂNCIA PROPOSTAS

Uma das principais contribuições deste trabalho é a adição de mais 3 opções de distâncias, dando assim, ao usuário do modelo o poder de escolher além das distâncias do modelo original, Mahalanobis, Euclideana e Cosseno, as distâncias Hamming, Minkowski e Manhattan, visando ganhos de acurácia, pois o modelo se torna mais adaptável a diferentes problemas. No modelo original, o autor toma o cuidado de enfatizar sobre essa escolha mais adequada de distância, pois ela é específica do problema (dos dados). Pode-se usar o conhecimento atual no domínio do problema para escolher a medida desejada para uma aproximação razoável e um resultado de classificação desejado (17). Todas essas distâncias foram escolhidas para esse trabalho baseando-se em (42, 63, 64).

3.4.1 **Distância de Hamming**

A distância de Hamming é assim chamada em homenagem a Richard Hamming, que introduziu o conceito em um artigo fundamental sobre códigos de *Hamming Error*

detecting and error correcting codes em 1950 (65). Utiliza-se a distância de Hamming para representar estruturas semânticas complexas com alta fidelidade (66). E é dada por:

$$d_{ham}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{N} \delta(x_{i,l}, x_{j,l}) \tag{3.22}$$

em que $i, j \in 1, 2, ..., K$, N é a dimensionalidade do espaço, $\mathbf{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,N}]^T$ e $\delta(x_{i,l}, x_{j,l})$ como:

$$\delta(x_{i,l}, x_{j,l}) = \begin{cases} 1, se\ x_{i,l} \neq x_{j,l} \\ 0, se\ x_{i,l} = x_{j,l} \end{cases} \tag{3.23}$$

Foi demonstrado que, no espaço de hamming, é possível realizar a busca exata dos vizinhos mais próximos notavelmente mais rápida que a busca linear (67). A distância de Hamming foi escolhida para esse trabalho baseando-se em (63), em que a mesma se mostrou competitiva comparada a outras distâncias em um problema de classificação de ambiente interno.

3.4.2 **Distância de Minkowski**

A chamada distância de Minkowski, em homenagem ao matemático alemão Hermann Minkowski (1864-1909), é um espaço vetorial normalizado, que pode generalizar a distância Euclideana, a distância Manhattan e a distância Chebychev, e é dada por:

$$d_{min}(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{l=1}^{N} (|x_{i,l} - x_{j,l}|)^h \right)^{\frac{1}{h}} \tag{3.24}$$

em que $0 \leq h \leq 2$. Para $h = 1$ ou L_1 temos a distância Manhattan, $h = 2$ ou L_2 a distância Euclideana, para outros valores de h não existem nomes específicos. A distância de Minkowski é utilizada quando é necessário identificar e, de fato, ignorar, características irrelevantes e quando existe um número grande de aglomerados anômalos. Essa distância foi escolhida baseando-se nos resultados encontrados em (64), em que mostrou-se que abordagem é adequada para classificação de ambientes internos complexos usando sinais de Wi-Fi.

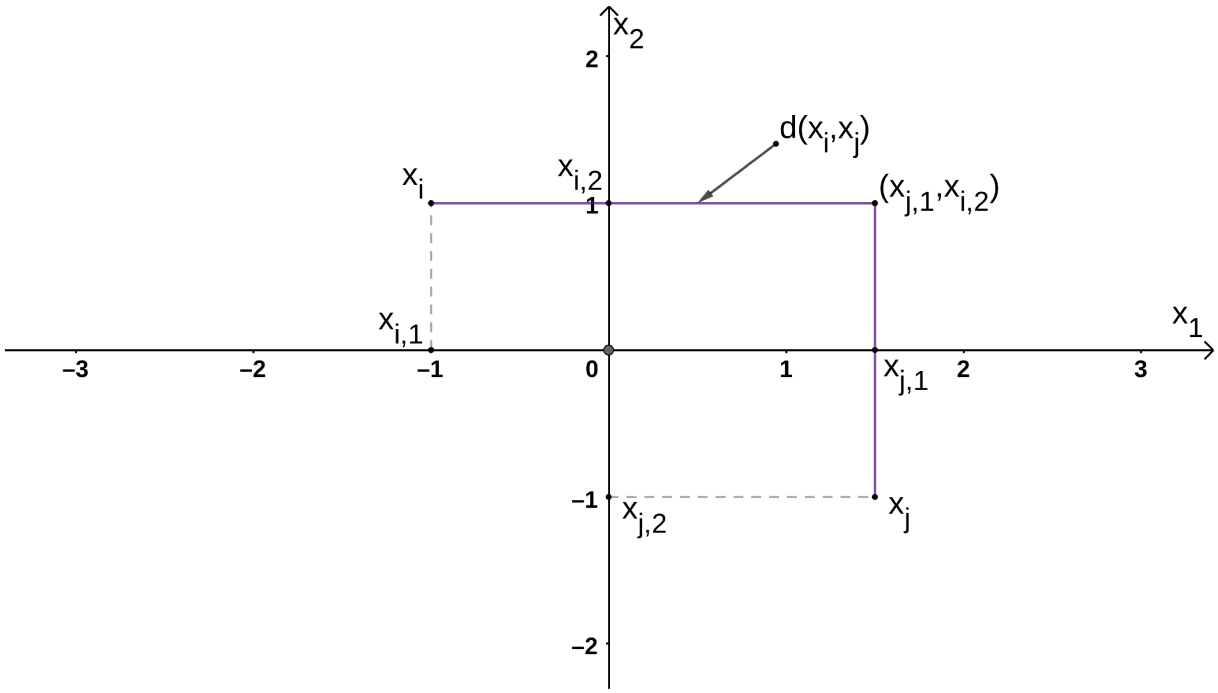
3.4.3 **Distância Manhattan**

A distância Manhattan (também conhecida como *City Block*) está incluída na mesma família de funções de distância, em que essa distância é um caso especial de Minkowski para h = 1. A expressão para calcular a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$ é:

$$d_{man}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{N} |x_{i,l} - x_{j,l}| \tag{3.25}$$

em que o módulo || denota valores absolutos. Podemos ver, analisando a equação acima, que a distância é a soma das diferenças absolutas de suas coordenadas cartesianas. No Gráfico 3, é possível verificar um exemplo da distância entre os pontos $\mathbf{x}_i$ e $\mathbf{x}_j$.

Gráfico 3 – Distância Manhattan



Fonte: (55).

Aplicando os valores representados pela Figura 3 na equação (3.25) se obtém:

$$d_{man}(\mathbf{x}_i, \mathbf{x}_j) = |x_{i,1} - x_{j,1}| + |x_{i,2} - x_{j,2}| = 4,5$$

Analisando o Gráfico 4, nota-se que as distâncias Euclideana e Manhattan não são iguais, e essa última é sempre maior que a primeira. Isso devido a desigualdade do triângulo (55). Utiliza-se a distância Manhattan nas situações em que a importância relativa das características deve ser levada em consideração, sendo que ela é uma versão ponderada entre dois vetores normalizados. Essa distância foi uma das escolhidas por já ter demonstrado bons resultados em problemas de classificação de ambientes internos, quando substituída pela distância euclideana, visto que as distâncias euclidianas ignoram qualquer regularidade estatística que possa ser estimada a partir de um grande conjunto de dados de treinamento (42).

Gráfico 4 – Comparação entre distâncias Euclideana e Manhattan em uma dimensão.

$$d_{euc}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{j,1} - x_{i,1})^2 + (x_{j,2} - x_{i,2})^2} \approx 3,2015$$

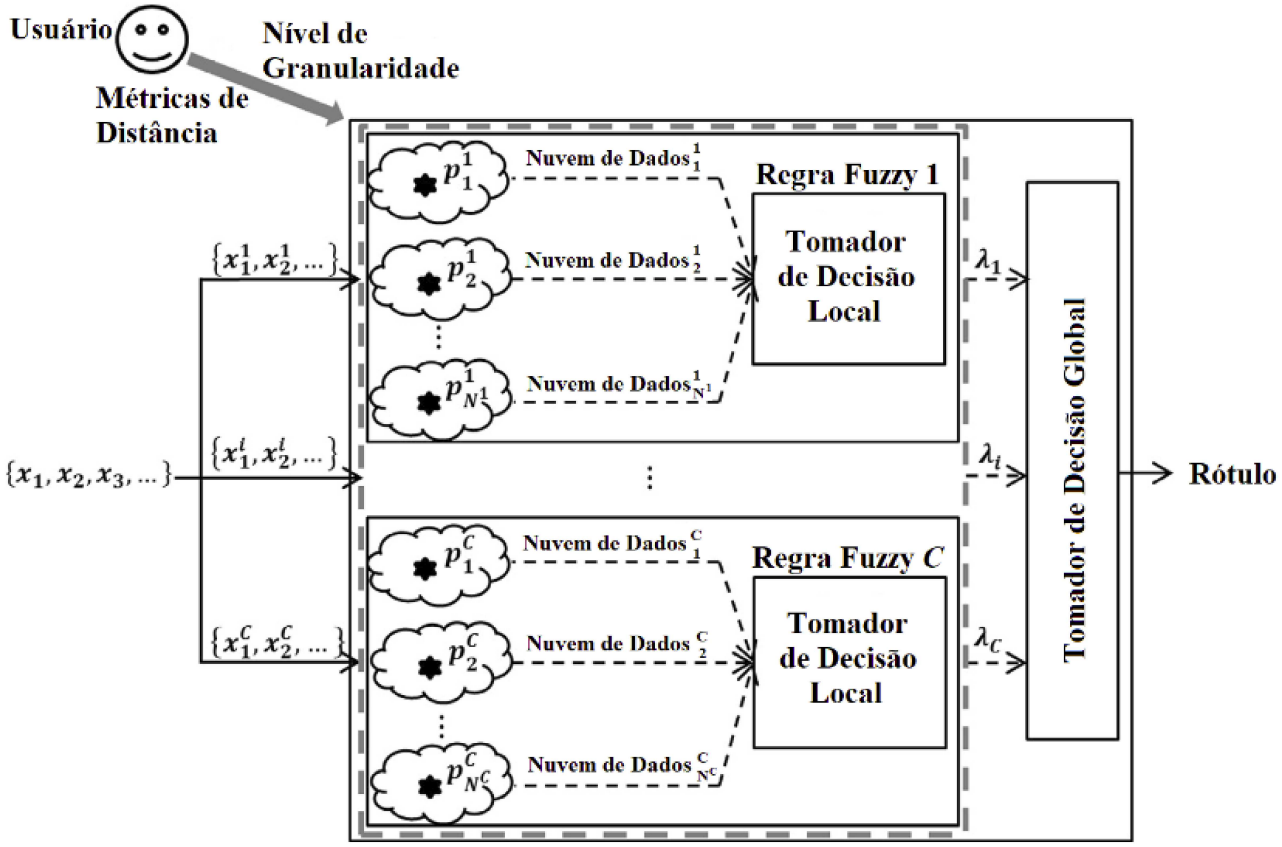
$$d_{man}(\mathbf{x}_i, \mathbf{x}_j) = |x_{j,1} - x_{i,1}| + |x_{j,2} - x_{i,2}| = 4,5$$

Fonte: (55)

3.5 ESTÁGIOS DO SOF

Nesta seção, os estágios de treinamento *offline*, *online* e de teste serão descritos com mais detalhes. A arquitetura do modelo é mostrada na Figura 4.

Figura 4 – Arquitetura do SOF.



Fonte: (17).

3.5.1 Treinamento *offline* do SOF

O processo de treinamento *offline* do classificador é mostrado na Figura 4, em que serão identificados protótipos de cada classe separadamente e sera formada uma regra do tipo AnYa com base nesses protótipos identificados por classe, como descrito na equação (3.1). Os processos de treinamento das regras de diferentes classes não se influenciam mutuamente.

Assumindo agora que o processo de treinamento é realizado em amostras de dados da classe $c = 1, 2, ..., C$, denotada por $\{\mathbf{x}\}^c_{K^c} = \{\mathbf{x}^c_1, \mathbf{x}^c_2, ..., \mathbf{x}^c_{K^c}\}$ ($\{\mathbf{x}\}^c_{K^c} \subset \{\mathbf{x}\}_K$), e o conjunto de amostras únicas de dados correspondentes e as frequências de ocorrência são denotadas, respectivamente, por $\{\mathbf{u}\}^c_{U^c_K} = \{\mathbf{u}^c_1, \mathbf{u}^c_2, ..., \mathbf{u}^c_{U^c_K}\}$ e $\{f\}^c_{U^c_K} = \{f^c_1, fu^c_2, ..., f^c_{U^c_K}\}$, em que K^c é o número de amostras de dados com $\{\mathbf{x}\}^c_{K^c}$, U^c_K é o número de amostras de dados exclusivas da classe c. Considerando todas as classes, tem-se $\sum\limits_{c=1}^{C} K^c = K$ e $\sum\limits_{c=1}^{C} U^c_K = U_K$ (17).

Os protótipos são identificados com base nas densidades e nas distribuições mútuas das amostras de dados. Primeiramente, densidades multimodais, equação (3.26), em todas as amostras de dados exclusivas dentro $\{\mathbf{u}\}^c_{U^c_K}$, são calculadas usando a equação (3.5). Em seguida, as amostras de dados são classificadas em uma lista indicada como $\{\mathbf{r}\}$ em termos de distâncias mútuas e valores de densidade multimodal.

$$D^{MM}_{K^c}(\mathbf{u}^c_i) = f^c_i \frac{\sum\limits_{l=1}^{K^c}\sum\limits_{j=1}^{K^c} d^2(\mathbf{x}^c_l, \mathbf{x}^c_j)}{2K^c \sum\limits_{j=1}^{K^c} d^2(\mathbf{u}^c_i, \mathbf{x}^c_j)} \tag{3.26}$$

em que $i = 1, 2, ..., U^c_K$.

Após descobrir a amostra de dados com a maior densidade multimodal, $\mathbf{r}_1 = arg \max\limits_{i=1,2,...,U^c_K} \left(D^{MM}_{K^c}(\mathbf{u}^c_i)\right)$, o primeiro elemento $\mathbf{r}_1$, da lista $\{\mathbf{r}\}$, é identificado. Então, o segundo elemento $\mathbf{r}_2$ é identificado como a amostra de dado com a distância mínima para $\mathbf{r}_1 : \mathbf{r}_2 = arg \min\limits_{i=1,2,...,U^c_K-1} (d(\mathbf{r}_1, \mathbf{u}^c_i))$. O terceiro elemento de $\{\mathbf{r}\}$, denotado por $\mathbf{r}_3$, é identificado baseando-se na distância mínima para $\mathbf{r}_2$. Repetindo o processo até que todas as amostras de dados tenham sido selecionadas, a lista completa $\{\mathbf{r}\}$ é construída e as densidades multimodais $\{\mathbf{u}\}^c_{U^c_K}$ são classificadas de acordo com a lista, indicada por $\{D^{MM}_{K^c}(\mathbf{r})\}$ (68). É necessário enfatizar que, quando uma amostra de dados é selecionada em $\{\mathbf{r}\}$, ela não pode ser selecionada pela segunda vez.

Protótipos, descritos como $\{\mathbf{p}\}_0$, são então identificados como os máximos locais das densidades multimodais classificadas, $\{D^{MM}_{K^c}(\mathbf{r})\}$, usando condição 1 (68):

$$\textbf{Condição 1} : SE \left(D^{MM}_{K^c}(\mathbf{r}_i) > D^{MM}_{K^c}(\mathbf{r}_{i+1})\right) E \left(D^{MM}_{K^c}(\mathbf{r}_i) > D^{MM}_{K^c}(\mathbf{r}_{i-1})\right) ENTÃO... \\ ...ENTÃO (\mathbf{r}_i \in \{\mathbf{p}\}_0) \tag{3.27}$$

Uma vez que todos os protótipos sejam identificados usando a equação (3.27), é possível notar alguns desses sendo menos representativos dentro de $\{\mathbf{p}\}_0$, portanto, é necessário realizar uma operação de filtragem para removê-los de $\{\mathbf{p}\}_0$.

Antes de iniciar a operação de filtragem, primeiramente usam-se protótipos para atrair amostras de dados próximas, para formar as nuvens de dados (52) que se assemelham ao mosaico de Voroni (69):

$$protótipo\ vencedor = arg \min_{\mathbf{p}\in\{\mathbf{p}\}_0} \left(d(\mathbf{x}_i, \mathbf{p})\right);\ \mathbf{x}_i \in \{\mathbf{x}\}^c_{K^c} \tag{3.28}$$

Depois que todas as nuvens de dados são formadas em torno dos protótipos existentes $\{\mathbf{p}\}_0$, podem-se obter os centros das nuvens de dados indicadas por $\{\phi\}_0$ e as densidades multimodais nos centros são calculadas usando a equação 3.5, como $D^{MM}_{K^c}(\phi_i) = S_i D_{K^c}(\phi_i)$, em que $\phi \in \{\phi\}_0$ e S_i é o suporte (número de membros) da i-ésima nuvem de dados.

Então, para cada nuvem de dados, assumindo o primeiro i ($\phi_i \in \{\phi\}_0$), os centros de nuvens de dados vizinhas, representado por $\{\phi\}^{vizinho}_i$, são identificados usando a seguinte condição:

$$\textbf{Condição 2} : SE\left(d^2(\phi_i, \phi_j) \leq G^{c,L}_{K^c}\right)\ ENTÃO\left(\phi_j \in \{\phi\}^{vizinho}_i\right) \tag{3.29}$$

em que $\phi_j \in \{\phi\}_0$, $\phi_j \neq \phi_i$; $G^{c,L}_{K^c}$ é definido como o raio médio da área influente local em torno de cada amostra de dados, que corresponde ao L-ésimo ($L = 1, 2, ...$) nível de granularidade. Esse raio é derivado dos dados da c-ésima classe, baseando-se na escolha do usuário.

Finalmente, o protótipo mais representativo da c-ésima classe, denotado por $\{\mathbf{p}\}^c$, é são selecionado entre os centros das nuvens de dados existentes que satisfazem a condição 3 (68):

$$\textbf{Condição 3} : SE\left(D^{MM}_{K^c}(\phi_i) > \max_{\phi\in\{\phi\}^{vizinho}_i}(D^{MM}_{K^c}(\phi))\right)\quad ENTÃO\left(\phi_i \in \{\mathbf{p}\}^c\right) \tag{3.30}$$

Depois que todos os protótipos representativos da c-ésima classe $\{\mathbf{p}\}^c$ são identificados, pode-se criar a regra do tipo AnYa da seguinte forma:

$$SE(\mathbf{x} \sim \mathbf{p}^c_1)OU(\mathbf{x} \sim \mathbf{p}^c_2)OU...OU(\mathbf{x} \sim \mathbf{p}^c_{N^c})\quad ENTÃO(classe\ c) \tag{3.31}$$

em que N^c é o número de protótipos em $\{\mathbf{p}^c\}$.

O procedimento principal do processo de treinamento *offline* do classificador SOF proposto é resumido no pseudo-código a seguir:

Algoritmo 1 Processo de treinamento *offline* do SOF

Calcula D^{MM} em $\{\mathbf{u}\}^c_{U^c_K}$;
Encontra $\mathbf{r}_1 = \arg\max(\mathrm{D}^{MM}_{K^c}(\mathbf{u}^c_i))$ e exclui $\mathbf{r}_1$ de $\{\mathbf{u}\}^c_{U^c_K}$;
$k \leftarrow 1$; $\{\mathbf{r}\} \leftarrow \mathbf{r}_1$; $\{D^{MM}_{K^c}(\mathbf{r})\} \leftarrow D^{MM}_{K^c}(\mathbf{r}_1)$;
Enquanto $U^c_K \neq 0$ **faça**
    $k \leftarrow k + 1$;
    Calcula $\mathbf{r}_k = \arg\min(d(\mathbf{r}_{k-1}), \mathbf{u}^c_i)$ e exclui $\mathbf{r}_k$ de $\{\mathbf{u}\}^c_{U^c_K}$;
    $\{\mathbf{r}\} \leftarrow \{\mathbf{r}\} + \mathbf{r}_k$; $\{D^{MM}_{K^c}(\mathbf{r})\} \leftarrow \{D^{MM}_{K^c}(\mathbf{r})\} + D^{MM}_{K^c}(\mathbf{r}_k)$;
Fim Enquanto
Identifica $\{\mathbf{p}_0\}$ usando **Condição 1**;
Organiza nuvens de dados vizinhos $\{\mathbf{p}\}_0$;
Identifica $\{\phi\}_0$ a partir da nuvem de dados;
Calcula D^{MM} em $\{\phi\}_0$;
Identifica $\{\phi\}^{vizinho}$ usando **Condição 2**;
Identifica $\{\mathbf{p}\}^c$ usando **Condição 3**;
Cria a c^{th} regra fuzzy com $\{\mathbf{p}\}^c$;

3.5.2 Treinamento *online* do SOF

Durante o estágio de treinamento *online*, o modelo SOF continua atualizando os parâmetros e a estrutura do sistema com a transmissão de dados amostra por amostra. Além disso, como as quantidades de EDA empregadas pelo classificador podem ser atualizadas recursivamente, ele pode ser do tipo passagem única, garantindo sua eficiência computacional e memória. Assumindo que o processo de treinamento do classificador com o conjunto de dados estáticos $\{\mathbf{x}\}_K$ tenha finalizado, novas amostras de dados começam a chegar em um formulário de fluxo de dados. De modo, similar ao treinamento *offline*, durante o treinamento *online*, as regras de diferentes classes são atualizadas separadamente. Durante o estágio *online*, expressões de cálculos recursivos das quantidades EDA com a distância escolhida pelo usuário são adotadas.

Assumindo na instância $K+1$ uma nova amostra de dados da c-ésima classe, denotada como $\mathbf{x}^c_{K^c+1}$, chega, o classificador SOF em primeiro lugar, atualiza os meta-parâmetros $\mu^c_{K^c}$, $\mathbf{X}^c_{K^c}$ e $\mathbf{C}^c_{K^c}$ para $\mu^c_{K^c+1}$, $\mathbf{X}^c_{K^c+1}$ e $C^c_{K^c+1}$ usando as equações (3.9), (3.10) e (3.11). O raio médio da área local de influência $G^{c,L}_{K^c}$ é atualizado posteriormente de forma recursiva, baseando-se em $\overline{d}^c_{K^c}$ e $G^{c,L}_{K^c}$:

$$G^{c,L}_{K^c} = \frac{\overline{d}^c_{K^c+1}}{\overline{d}^c_{K^c}} G^{c,L}_{K^c} = \frac{\frac{1}{(K^c+1)^2}\sum_{l=1}^{K^c+1}\pi_{K^c+1}(\mathbf{x}^c_l)}{\frac{1}{(K^c)^2}\sum_{l=1}^{K^c}\pi_{K^c}(\mathbf{x}^c_l)} \tag{3.32}$$

em que $\overline{d}^c_{K^c}$ e $\overline{d}^c_{K^c+1}$ denotam as distâncias quadradas médias entre quaisquer duas amostras de dados com $\{\mathbf{x}\}^c_{K^c}$ e $\{\mathbf{x}\}^c_{K^c+1}$, respectivamente.

Como um caso especial, para a distância Mahalanobis, $G^{c,L}_{K^c+1} = G^{c,L}_{K^c}$. Analisando

a equação (3.32), percebe-se que, em vez de derivar $G^{c,L}_{K^c+1}$ de maneira *offline*, a equação (3.32) reduz amplamente a complexidade computacional e memória requerida e melhora a eficiência do classificador SOF.

Então, a amostra $\mathbf{x}^c_{K^c+1}$ é checada para avaliar seu potencial de ser um novo protótipo, baseando-se nas seguintes informações (56, 70):

$$\begin{aligned}\textbf{Condição 4: } SE \left(D_{K^c+1}(\mathbf{x}^c_{K^c+1}) > \max_{\mathbf{p}\in\{\mathbf{p}\}^c}(D_{K^c+1}(\mathbf{p})) \right) OU... \\ ...OU \left(D_{K^c+1}(\mathbf{x}^c_{K^c+1}) < \min_{\mathbf{p}\in\{\mathbf{p}\}^c}(D_{K^c+1}(\mathbf{p})) \right) ENTÃO... \\ ...ENTÃO \left(\mathbf{x}^c_{K^c+1} \in \{\mathbf{p}\} \right)\end{aligned} \tag{3.33}$$

em que a equação (3.14) é usada para calcular $D_{K^c+1}(\mathbf{x}^c_{K^c+1})$ e $D_{K^c+1}(\mathbf{p})(\mathbf{p} \in \{\mathbf{p}\}^c)$. Se $\mathbf{x}^c_{K^c+1}$ atender a condição 4, um novo protótipo é adicionado à regra fuzzy da c-ésima classe (equação (3.30)) e os meta-parâmetros do SOF são atualizados como a seguir:

$$N^c \leftarrow N^c + 1;\ \mathbf{p}^c_{N^c} \leftarrow \mathbf{x}^c_{K^c+1};\ S^c_{N^c} \leftarrow 1;\ \{\mathbf{p}\}^c \leftarrow \{\mathbf{p}\}^c + \mathbf{p}^c_{N^c} \tag{3.34}$$

Se a condição 4 não for satisfeita, faz-se necessário verificar se $\mathbf{x}^c_{K^c+1}$ está muito perto de um protótipo existente, usando a Condição 5 (68).

$$\textbf{Condição 5} : SE \left(\min_{\mathbf{p}\in\{\mathbf{p}\}^c}(d^2(\mathbf{x}^c_{K^c+1}, \mathbf{p})) > G^{c,L}_{K^c+1} \right) ENTÃO \left(\mathbf{x}^c_{K^c+1} \in \{\mathbf{p}\}^c \right) \tag{3.35}$$

Se a Condição 5 for atendida, um novo protótipo é adicionado à regra fuzzy da c-ésima classe ($\{\mathbf{p}\}^c \leftarrow \{\mathbf{p}\}^c + \mathbf{p}^c_{N^c}$) e a nova nuvem de dados, correspondente com meta-parâmetros inicializados pela equação (3.34), é adicionada ao classificador.

Se as Condições 4 e 5 não forem satisfeitas, $\mathbf{x}^c_{K^c+1}$ é atribuído a um protótipo vizinho mais próximo $\mathbf{p}^c_{n*} = arg \min_{\mathbf{p}\in\{\mathbf{p}\}^c} \left(d(\mathbf{x}^c_{K^c+1}, \mathbf{p}) \right)$ e os meta-parâmetros da nuvem de dados correspondente são atualizados seguindo (56):

$$\mathbf{p}^c_{n^*} \leftarrow \frac{S^c_{n^*}}{S^c_{n^*}+1}\mathbf{p}^c_{n^*} + \frac{1}{S^c_{n^*}+1}\mathbf{x}^c_{K^c+1}; S^c_{n^*} \leftarrow S^c_{n^*} + 1 \tag{3.36}$$

Depois que os meta-parâmetros são atualizados, a regra fuzzy do tipo AnYa é atualizada e o SOF estará pronto para processar a próxima amostra de dados ou realizar a classificação. A seguir está o pseudo-código do principal procedimento do processo de treinamento *online*.

Algoritmo 2 Processo de treinamento *online* do SOF

Enquanto uma nova amostra de dado da c^{th} classe $\mathbf{x}^c_{K^c+1}$ está disponível (ou até interrupção) **faça**
    Atualiza $\mu^c_{K^c}$, $\mathbf{X}^c_{K^c}$, $\mathbf{C}^c_{K^c}$, $G^{c,L}_{K^c}$ para $\mu^c_{K^c+1}$, $\mathbf{X}^c_{K^c+1}$, $\mathbf{C}^c_{K^c+1}$, $G^{c,L}_{K^c+1}$;
    Calcula D em $\mathbf{x}^c_{K^c+1}$ e $\{\mathbf{p}\}^c$;
    se (**Condição 4**) *ou* (**Condição 5 então**
        $N^c \leftarrow \mathrm{N}^c + 1$; $\mathbf{p}^c_{N^c} \leftarrow \mathrm{x}^c_{K^c+1}$; $S^c_{N^c} \leftarrow 1$; $\{\mathbf{p}\}^c \leftarrow \{\mathbf{p}\}^c + \mathbf{p}^c_{N^c}$
    senão
        Encontra $\mathbf{p}^c_{n^*}$;
        $\mathbf{p}^c_{n^*} \leftarrow \frac{S^c_{n^*}}{S^c_{n^*}+1}\mathrm{p}^c_{n^*} + \frac{1}{S^c_{n^*}+1}x^c_{K^c+1}$; $S^c_{n^*} \leftarrow \mathrm{S}^c_{n^*} + 1$;
    Fim se
    $K^c \leftarrow \mathrm{K}^c + 1$;
    Atualiza a regra fuzzy
Fim Enquanto

3.5.3 **Fase de teste do SOF**

Nesta subseção, é descrito o procedimento do classificador SOF para tomada de decisão. Como mostrado na Figura 4, durante o estágio de teste, para um teste particular da amostra de dados, representada por $\mathbf{x}$, cada regra do tipo AnYa terá um poder de disparo dado pelo tomador de decisão local, denotado por $\lambda^c(\mathbf{x})(c = 1, 2, ..., C)$, que é determinado como:

$$\lambda^c(\mathbf{x}) = \max_{\mathbf{p}\in\{\mathbf{p}\}^c} \left(e^{-d^2(\mathbf{x},\mathbf{p})}\right);\ c = 1, 2, ..., C \tag{3.37}$$

Com base nos pontos focais do nível de disparo das regras fuzzy C correspondentemente (um por regra), o rotulo de $\mathbf{x}$ é decidido pelo tomador de decisão geral usando o princípio vencedor leva tudo da seguinte forma:

$$rotulo = arg \max_{c=1,2,...,C} (\lambda^c(\mathbf{x})) \tag{3.38}$$

3.6 CLASSIFICAÇÃO EM DIFERENTES NÍVEIS DE GRANULARIDADE

Como o SOF é um uma abordagem baseada em protótipo, é importante definir uma área de influência local adequada para cada protótipo, com o intuito de aumentar a capacidade descritiva das regras fuzzy e, ao mesmo tempo, evitar sobreposições. Existem duas maneiras mais comuns de definir isso: a primeira consiste em ajustar um raio baseado nos conhecimentos prévios (71) e a segunda deriva de acordo com os princípios codificados (72, 73). Entretanto, em muitos casos, o conhecimento prévio não está disponível, enquanto os princípios codificados são muito sensíveis à natureza dos dados. Então, para a solução do problema entra a computação granular, que a grosso modo, explora vários níveis de granularidade na solução de problemas. Os níveis de granularidade podem ser interpretados como os níveis de abstração, detalhe, complexidade e controle em contextos específicos. Os objetos da computação granular são famílias de grânulos que representam um problema

em vários níveis (74). Desse modo, a seguir, será descrito como definir as áreas locais em torno do protótipo baseado-se nos dados e no nível de granularidade.

Sob o primeiro nível de granularidade ($L = 1$), o raio médio da área de influência local ao redor de cada protótipo da c-ésima classe, denotada por $G_{K^c}^{c,1}$, é definido como:

$$G_{K^c}^{c,1} = \frac{\sum_{\mathbf{x},\mathbf{y}\in\{\mathbf{x}_K^c{}^c\},\mathbf{x}\neq\mathbf{y},d^2(\mathbf{x},\mathbf{y})\leq\overline{d}_K^c{}^c} d^2(\mathbf{x},\mathbf{y})}{Q_{K^c}^{c,1}} \tag{3.39}$$

em que $Q_{K^c}^{c,1}$ é o número de pares das amostras de dados com $\{\mathbf{x}\}_{K^c}^c$ entre qualquer distância que seja menor que a média de distâncias $\overline{d}_{K^c}^c$.

A partir do nível 2, para um nível arbitrário de granularidade ($L = 2, 3, ...,$), pode-se calcular o raio médio iterativamente usando a seguinte equação:

$$G_{K^c}^{c,1} = \frac{\sum_{\mathbf{x},\mathbf{y}\in\{\mathbf{x}_K^c{}^c\},\mathbf{x}\neq\mathbf{y},d^2(\mathbf{x},\mathbf{y})\leq G_{K^c}^{c,L-1}} d^2(\mathbf{x},\mathbf{y})}{Q_{K^c}^{c,L}} \tag{3.40}$$

em que $G_{K^c}^{c,L-1}$ é o raio médio correspondente ao $(L-1)$-ésimo nível de granularidade; $Q_{K^c}^{c,L}$ é o número de pares de amostra de dados entre os quais a distância é menor que $G_{K^c}^{c,L-1}$.

Comparando com abordagens tradicionais, há vantagens em obter informações locais desta maneira. Primeiramente, $G_{K^c}^{c,L-1}$ é garantido ser válido o tempo todo. A definição antecipada do limite ou dos princípios matemáticos codificados pode sofrer de vários problemas, como foi discutido no inicio desta seção. Enquanto, se $G_{K^c}^{c,L-1}$ é derivado a partir dos dados diretos e é sempre significativo, esse não necessita de conhecimento prévio do conjunto de dados e o nível de granularidade escolhido para o SOF pode ser decidido baseando-se na preferência do usuário. Além disso, os usuários são livres para fazer escolhas, podendo sempre adaptar o classificador fazendo mudanças no nível de granularidade, baseando-se nas necessidades específicas.

Em geral, quando a alta granularidade é escolhida, chamada granularidade fina, o SOF faz maior extração de mais detalhes (gera mais protótipos) dos dados. Porém, o classificador consome mais recursos computacionais e memória, e pode ocorrer *overfitting*. De modo contrario, quando escolhido um baixo nível de granularidade, chamada granularidade grossa, o classificador aprende apenas informações pouco especificas do treinamento. Alguns problemas requerem mais detalhamentos, enquanto outros podem precisar ser mais generalistas.

4 RESULTADOS EXPERIMENTAIS DO SOF

Neste capitulo, será apresentada a base de dados e a forma que como foi coletada. Além disso, será também discutido os resultados obtidos comparando o modelo proposto com o original, e outras abordagens de *machine learning* para a base de dados utilizada.

4.1 BASE DE DADOS

A base de dados utilizada neste trabalho pode ser encontrada no *website* `https://archive.ics.uci.edu/ml/datasets/2.4+GHZ+Indoor+Channel+Measurements`, e foi publicada em (30, 50). Essa base foi produzida no *campus* da Universidade de Khalifa, Emirados Árabes Unidos. Na Figura 5, encontra-se a planta que contém os quatro ambientes analisados, que são o laboratório, salão de esporte, corredor estreito e salão de espera. Os triângulos vermelhos são os transmissores e os asteriscos pretos representam os receptores.

Figura 5 – Planta do ambiente interno de múltiplos caminhos

Fonte: (30)

Cada ambiente foi dividido uniformemente com espaços de comprimento de onda ($\lambda = 12,5cm$) e frequência de $f = 2,4Ghz$ selecionadas para examinar as bandas Wi-Fi associadas ao padrão *IEEE 802.11g*. Essas divisões resultaram em um total de 196 posições, sendo que as pequenas variações de escala podem ser melhor capturadas em (50). As medições foram realizadas no ambiente estático, não havendo movimento entre o transmissor e o receptor a fim de não se aumentar o erro associado ao sistema.

O sistema de medição foi constituído por: cabos RF de baixa perda, analisador de rede vetorial (VNA) ZVB14 (usado para medir o coeficiente de transmissão), antenas omnidirecionais de altura igual a 1,5m nas extremidades do receptor e transmissor, sendo que o VNA fazia 10 varreduras consecutivas; cada varredura cobrindo uma faixa de 100 MHz usando 601 pontos de frequência, variando a frequência em 0,167 MHz. O conjunto de dados gerado para cada ambiente consiste em 196 amostras para cada medição, totalizando 1960 amostras.

A abordagem SOF possui dois estágios distintos. No primeiro deles, denominado de estágio *offline*, sumarizado no Algoritmo 1, os protótipos dos dados são definidos e utilizados na determinação da base de regras de um modelo *AnYa* estável de ordem 0. Já no segundo estágio, denominado de estágio *online*, as regras identificadas são atualizadas de acordo com os dados de fluxo contínuo, acompanhando assim as possíveis alterações nestes dados. Este estágio é sumarizado no Algoritmo 2.

A base de dados utilizada é composta por 4 classes (laboratório, salão de esporte, corredor estreito e salão de espera) que representam os ambientes analisados, os 5 atributos (frequência do sinal, parâmetros S_{11} real e imaginário, parâmetros S_{21} real e imaginário). A base foi reduzida para 100.000 amostras e depois dividida, como mostrado na Tabela 1. Essa divisão totaliza 50% de treinamento e 50% teste, sendo que para treinamento destina-se 35% para treinamento *offline* e 15% para treinamento *online*.

Tabela 1 – Base de dados.

Base de Dados		Número de Classes	Número de Amostras	Número de Atributos
Radiofrequência	Conjunto de Treinamento *offline*	4	42500	5+1 rótulo
	Conjunto de Treinamento *online*		7500	
	Conjunto de Teste		50000	

Fonte: Elaborado pelo autor.

4.2 ANÁLISE DE PERFORMANCE

Neste trabalho, as métricas de distância apresentadas: distância de Hamming, distância de Minkowski e distância Manhattan foram testadas a fim de se verificar qual é a mais recomendada para o problema proposto. Diversos valores de granularidade foram heuristicamente testadas. As simulações foram conduzidas com a granularidade igual a 13 com o intuito de se obter o melhor desempenho para o modelo proposto.

As especificações do computador usado para gerar os modelos de aprendizado são fornecidas da seguinte forma: CPU Intel Core i7-5500U (2 cores and 4 Threads de 2.4GHz e memória cache de 4MB), memória RAM de 8GB e sistema operacional Linux Fedora 29.

Os classificadores adotados para fins comparativos foram implementados de acordo com as configurações apresentada em (75). São eles: *SVM linear*, *SVM RBF*, *Decision Tree*, *Random Forest*, *Nearest Neighbors*, *MLP-ANN*, *AdaBoost*, *Naive Bayes* e *QDA*. Adicionalmente, foi implementado um classificador Fuzzy, baseado no projeto apresentado em (76). Maiores informações sobre os códigos desenvolvidos podem ser encontrados em `https://github.com/ualisondias/new_sof/`.

4.2.1 **Validação cruzada**

A técnica de validação cruzada foi sugerida em (77, 78) e tem considerável importância no processo de aprendizado considerando modelos de inteligencia computacional. Uma das técnicas de validação cruzada é a *k-fold* (79). Essa técnica consiste em dividir a base de dados de forma aleatória em k subconjuntos (em que k é definido previamente) com aproximadamente a mesma quantidade de amostras em cada um deles. A cada iteração, treino e teste, um conjunto formado por $k-1$ subconjuntos são utilizados para treinamento e o subconjunto restante será utilizado para teste gerando um resultado de métrica para avaliação, como pode ser visto na Tabela 2, em que é calculado a acurácia média e as métricas F-Score, coeficiente Kappa e MSE para treinamento e teste. Esse processo garante que cada subconjunto será utilizado para teste em algum momento da avaliação do modelo.

Na Tabela 2, são apresentados os resultados obtidos através do *software MATLAB*, pela aplicação do SOF ao problema de classificação de ambientes internos, considerando as métricas de distância de Hamming, Minkowski e Manhattan, originando os classificadores SOF-Hamming, SOF-Minkowski e SOF-Manhattan, respectivamente. Para fins de comparação, são considerados os seguintes classificadores, também implementados em *software MATLAB*: SOF-Cosseno, SOF-Euclideana e SOF-Mahalanobis, todos apresentados em (17) e que adotam como métricas de distância Cosseno, Euclideana e Mahalanobis, respectivamente. Os outros classificadores comparados neste trabalho são: Um classificador Fuzzy baseado em (81), *SVM linear*, *SVM RBF*, *Decision Tree*, *Random Forest*, *Nearest Neighbors*, *MLP-ANN*, *AdaBoost*, *Naive Bayes* e *QDA*. Eles são baseados no *scikit-learn*,

uma biblioteca de aprendizado de máquina de código aberto em Python (82).

Os códigos foram executados 33 vezes, de modo a avaliar a aproximação da curva a uma distribuição normal (83) e se existe alguma variação na acurácia.

Aplicando *k-fold*, com $k = 5$, foram obtidos os resultados mostrados na Tabela 2. O desvio padrão oriundo das simulações se devem às inicializações distintas para cada execução.

Analisando os dados da Tabela 2, vemos que dentre os modelos de classificadores, o que obteve maior acurácia, durante a fase de treinamento, foi o SOF-Mahalanobis, seguido por SOF-Minkowski e SOF-Euclideana, quando comparado aos demais. Contudo, na fase de teste, o que obteve maior valor de acurácia média foi o SOF-Manhattan, seguido de SOF-Minkowski e SOF-Euclideana, respectivamente. Esses valores maiores, na fase de teste para a abordagem SOF-Manhattan, se DÃO porque a atualização das regras e parâmetros foram melhor estruturadas nesse modelo. Nos outros modelos de aprendizado de máquina ocorreu o mesmo com *Nearest Neighbors* e com *RBF SVM*, em que no primeiro se obtém resultados maiores de acurácia média para a fase de treinamento e no segundo, maiores para a fase de teste. Os valores de F-Score seguem o mesmo padrão dos resultados de acurácia média. Os valores de Kappa sofrem algumas alterações bruscas, como por exemplo: o modelo SOF-Mahalanobis, em que existe uma queda de treinamento para teste. O que obteve menor diferença do valor de Kappa e maior valor para treinamento e teste, foi a abordagem SOF-Manhattan. O classificador Fuzzy foi o que obteve o menor valor de acurácia, F-Score e Kappa, e maior valor de erro médio quadrático dentre os modelos Fuzzy. O menor valor de MSE no treinamento foi obtido pelo SOF-Mahalanobis e o menor para teste, foi obtido pelo SOF-Manhattan.

Para a distância de Hamming a acurácia média do modelo atinge 76,19% $\pm$ 2,54, sendo esse o menor valor encontrado na fase de teste entre as abordagens propostas nesse trabalho. Já para a distância Manhattan, obteve-se acurácia média de 98,66%, para a fase de teste, sendo esse o melhor resultado dentre todos, em todos os casos esse tipo de abordagem ofereceu os maiores valores de Kappa e F-score e os menores valores de MSE. O modelo SOF-Minkowski obteve o segundo maior resultado de acurácia média para a fase de teste, entre todos. Com isso, percebe-se que as propostas são atraentes ao lidar com o problema de classificação de ambientes internos.

Tabela 2 – Performance em função da média e desvio padrão.

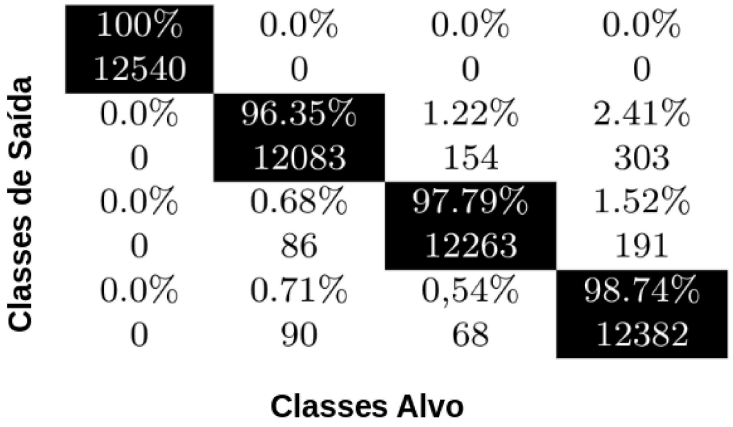
Abordagem	Treinamento acurácia média (%)	Teste acurácia média (%)	Treinamento F-Score	Teste F-Score	Treinamento Kappa	Teste Kappa	Treinamento MSE	Teste MSE
SOF-Cosseno	92,31±0,15	87,37±0,10	0,9230±0,0015	0,8735±0,0016	0,7951±0,0040	0,6632±0,0043	0,2600±0,0074	0,4272±0,0068
SOF-Euclideana	99,81±0,02	97,26±0,10	0,9981±0,0002	0,9726±0,0011	0,9951±0,0007	0,9269±0,0029	0,0047±0,0009	0,0567±0,0021
SOF-Minkowski	99,81±0,02	97,26±0,10	0,9981±0,0002	0,9726±0,0011	0,9951±0,0007	0,9269±0,0029	0,0047±0,0009	0,0567±0,0021
SOF-Mahalanobis	99,99±0,00	88,84±0,14	0,9999±6,7E-05	0,8883±0,0014	0,9997±0,0001	0,7025±0,0038	0,0003±0,0002	0,2559±0,0042
SOF-Hamming	82,99±5,01	76,19±2,54	0,8211±0,0532	0,7555±0,0260	0,5464±0,1336	0,3650±0,0678	0,3849±0,1210	0,5389±0,0579
SOF-Manhattan	99,75±0,02	98,66±0,05	0,9975±0,0002	0,9866±0,0005	0,9935±0,0007	0,9644±0,0014	0,0060±0,0008	0,0311±0,0015
Fuzzy	45,52±0,00	45,49±0,00	0,4290±0,0004	0,4285±0,0017	0,3116±0,0005	0,3120±0,0023	0,5805±0,0005	0,5803±0,0021
Nearest Neighbors	93,18±0,04	83,88±0,17	0,9318±0,0004	0,8387±0,0016	0,9091±0,0006	0,7850±0,0022	0,1662±0,0012	0,3956±0,0053
Linear SVM	41,65±0,13	41,60±0,19	0,3945±0,0010	0,3942±0,0023	0,2221±0,0018	0,2214±0,0026	1,5784±0,0068	1,5790±0,01301
RBF SVM	86,95±0,03	86,15±0,15	0,8694±0,0003	0,8617±0,0015	0,8260±0,0005	0,8154±0,0020	0,2999±0,0012	0,3176±0,0046
Decision Tree	42,17±0,13	42,11±0,26	0,3768±0,0013	0,3756±0,0033	0,2296±0,0017	0,2274±0,0032	2,3531±0,0097	2,3533±0,01816
Random Forest	50,47±1,38	50,41±1,39	0,4842±0,0160	0,4802±0,0175	0,3398±0,0185	0,3392±0,0185	1,8193±0,0891	1,8308±0,0893
MLP-ANN	71,04±0,41	70,83±0,48	0,7060±0,0047	0,7029±0,0051	0,6138±0,0054	0,6111±0,0066	0,7092±0,0126	0,7174±0,0171

Fonte: Elaborado pelo autor.

As propostas SOF-Euclideana e SOF-Minkowski apresentaram os mesmos valores, comprovando que a distância de Minkowski com o $h = 2$ é uma generalização da distância Euclideana.

Na Figura 6 são mostradas as classes de saída para os quatro ambientes diferentes denotados pela primeira linha/coluna (corredor estreito), segunda linha/coluna (laboratório), terceira linha/coluna (sala de espera) e quarta linha/coluna (salão de esporte), em função das classes alvo na vertical. A matriz de confusão apresenta os resultados do estágio de teste, para a distância Manhattan com granularidade igual a 13.

Figura 6 – Matriz de confusão do modelo SOF-Manhattan utilizado para classificação de ambientes internos



Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Neste trabalho foram apresentadas abordagens aprimoradas baseadas em um classificador Fuzzy Auto-Organizável, aplicado à análise de dados de sinais de RF em redes de sensores sem fio, a fim de classificar o ambiente onde esses sensores estão localizados. O classificadores SOF-Minkowski, SOF-Hamming e SOF-Manhattan mostraram-se independentes de parâmetros predefinidos ou suposições anteriores, pois são conduzidos apenas por dados empiricamente observados. Os classificadores identificam protótipos dos dados de treinamento e continuam aprendendo com o fluxo de dados.

Para fins de comparações dos modelos, várias métricas de distâncias (Cosseno, Euclideana, Minkowski, Mahalanobis, Hamming e Manhattan) foram adotadas, além de outras abordagens tradicionais como: Fuzzy, *Nearest Neighbors*, *Linear SVM*, *RBF SVM*, *Decision Tree*, *Random Forest* e *MLP-ANN*.

Resultados experimentais mostraram que o SOF-Manhattan apresentou maior valor de acurácia média , maior F-Score e coeficiente Kappa e menor erro médio quadrático quando aplicado a um banco de dados composto por medições de sinais de RF em tempo real de ambientes internos. Avaliando o desvio padrão do modelo, é possível perceber que o mesmo não apresenta resultados distintos em mais de uma execução.

Como indicação para trabalhos futuros, fazer adaptações no modelo de modo a calcular de forma autônoma o melhor nível de granularidade, pretende-se aplicar o SOF estendido em *streaming* de dados relacionados à classificação de ambientes internos. Além disso, os modelos discutidos neste trabalho serão apresentados em outras aplicações para solucionar diferentes problemas de engenharia (por exemplo, detecção de eventos, diagnóstico de falhas em equipamentos e outros), com o intuito de auxiliar os profissionais não apenas nos processos de tomada de decisão, mas também na elaboração de estratégias no setor.

REFERÊNCIAS

1 ZADEH, Lotfi A. Outline of a new approach to the analysis of complex systems and decision processes. **IEEE Transactions on Systems, Man, and Cybernetics**, n. 1, p. 28-44, 1973.

2 ZANELLA, Andrea et al. Internet of things for smart cities. **IEEE Internet of Things Journal**, v. 1, n. 1, p. 22-32, 2014.

3 AL-FUQAHA, Ala et al. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys & Tutorials**, v. 17, n. 4, p. 2347-2376, 2015.

4 DA XU, Li; HE, Wu; LI, Shancang. Internet of things in industries: A survey. **IEEE Transactions on Industrial Informatics**, v. 10, n. 4, p. 2233-2243, 2014.

5 TIWARI, Prashant et al. A survey of localization methods and techniques in wireless sensor networks. **HCTL Open International Journal of Technology Innovations and Research (IJTIR)**, v. 14, p. 2321-1814, 2015.

6 LIU, Zhongwei et al. An energy-efficient and robust indoor-outdoor detection method based on cell identity map. **Procedia Computer Science**, v. 56, p. 189-195, 2015.

7 SUNG, Rakmin; JUNG, Suk-hoon; HAN, Dongsoo. Sound based indoor and outdoor environment detection for seamless positioning handover. **ICT Express**, v. 1, n. 3, p. 106-109, 2015.

8 CANOVAS, Oscar; LOPEZ-DE-TERUEL, Pedro E.; RUIZ, Alberto. Detecting indoor/outdoor places using WiFi signals and *AdaBoost*. **IEEE Sensors Journal**, v. 17, n. 5, p. 1443-1453, 2016.

9 ALI, Mohsen; ELBATT, Tamer; YOUSSEF, Moustafa. SenseIO: Realistic ubiquitous indoor outdoor detection system using smartphones. **IEEE Sensors Journal**, v. 18, n. 9, p. 3684-3693, 2018.

10 ZENG, Qinghua et al. Seamless pedestrian navigation methodology optimized for indoor/outdoor detection. **IEEE Sensors Journal**, v. 18, n. 1, p. 363-374, 2017.

11 ZHOU, Pengfei et al. Iodetector: A generic service for indoor outdoor detection. **ACM Conference on Embedded Network Sensor Systems**. ACM, 2012. p. 113-126.

12 LU, Lizhen; DI, Liping; YE, Yanmei. A decision-tree classifier for extracting transparent plastic-mulched landcover from Landsat-5 TM images. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 7, n. 11, p. 4548-4558, 2014.

13 FERREIRA, V. H. et al. A survey on intelligent system application to fault diagnosis in electric power system transmission lines. **Electric Power Systems Research**, v. 136, p. 135-153, 2016.

14 ANGELOV, Plamen et al. Symbol recognition with a new autonomously evolving classifier autoclass. *In*: **IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)**. IEEE, 2014. p. 1-7.

15 NOORBEHBAHANI, Fakhroddin et al. An incremental intrusion detection system using a new semi-supervised stream classification method. **International Journal of Communication Systems**, v. 30, n. 4, p. e3002, 2017.

16 SUÁREZ-CETRULO, Andrés L.; CERVANTES, Alejandro. An online classification algorithm for large scale data streams: iGNGSVM. **Neurocomputing**, v. 262, p. 67-76, 2017.

17 GU, Xiaowei; ANGELOV, Plamen P. Self-organising fuzzy logic classifier. **Information Sciences**, v. 447, p. 36-51, 2018.

18 ÖRKÇÜ, H. Hasan; DOĞAN, Mustafa; ÖRKÇÜ, Mediha. A Hybrid Applied Optimization Algorithm for Training Multi-Layer Neural Networks in the Data Classification. **Gazi University Journal of Science**, v. 28, n. 1, p. 115-132, 2015.

19 GU, Xiaowei. **Self-organising transparent learning system**. 2018. Thesis (Doctor of Philosophy). School of Computing and Communications, Lancaster University, Lancaster, 2018.

20 CRISTIANINI, Nello et al. **An introduction to support vector machines and other kernel-based learning methods**. Cambridge university press, 2000.

21 MAILLO, Jesus et al. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. **Knowledge-Based Systems**, v. 117, p. 3-15, 2017.

22 PRASAD, Sharat C.; PRASAD, Piyush. Deep recurrent *Neural Net*works for time series prediction. **arXiv preprint arXiv:1407.5949**, 2014.

23 NARASIMHA, Murty M. et al. **Introduction to pattern recognition and machine learning**. World Scientific, 2015.

24 GU, Xiaowei et al. Self-organised direction aware data partitioning algorithm. **Information Sciences**, v. 423, p. 80-95, 2018.

25 ZADEH, Lotfi A. Fuzzy sets. **Information and Control**, v. 8, n. 3, p. 338-353, 1965.

26 ANGELOV, Plamen; GU, Xiaowei; KANGIN, Dmitry. Empirical data analytics. **International Journal of Intelligent Systems**, v. 32, n. 12, p. 1261-1284, 2017.

27 ANGELOV, Plamen P.; GU, Xiaowei. **Empirical approach to machine learning. Springer**, 2019.

28 IDRIS, Adnan; RIZWAN, Muhammad; KHAN, Asifullah. Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. **Computers & Electrical Engineering**, v. 38, n. 6, p. 1808-1819, 2012.

29 ALHAJRI, Mohamed I.; ALI, Nazar T.; SHUBAIR, Raed M. Classification of indoor environments for IoT applications: A machine learning approach. **IEEE Antennas and Wireless Propagation Letters**, v. 17, n. 12, p. 2164-2168, 2018.

30 ALHAJRI, Mohamed I.; ALI, Nazar T.; SHUBAIR, Raed M. A machine learning approach for the classification of indoor environments using RF signatures. **IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. IEEE, 2018. p. 1060-1062.

31 CANOVAS, Oscar; LOPEZ-DE-TERUEL, Pedro E.; RUIZ, Alberto. WiFiBoost: A terminal-based method for detection of indoor/outdoor places. **Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services**. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014. p. 352-353.

32 PARK, Jun-geun et al. Implications of device diversity for organic localization. **Proceedings IEEE INFOCOM**. IEEE, 2011. p. 3182-3190.

33 BAHL, Paramvir et al. RADAR: An in-building RF-based user location and tracking system. **Proceedings IEEE INFOCOM**. IEEE, 2000.

34 HSU, Hui-Huang et al. Smartphone indoor localization with accelerometer and gyroscope. **International Conference on Network-Based Information Systems**. IEEE, 2014. p. 465-469.

35 RADU, Valentin et al. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. **Proceedings of the ACM Conference on Embedded Network Sensor Systems**. ACM, 2014. p. 280-294.

36 RUIZ-RUIZ, Antonio J. et al. Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. **IEEE International Conference on Pervasive Computing and Communications (PerCom)**. IEEE, 2014. p. 130-138.

37 GU, Yang et al. Online deep intelligence for wi-fi indoor localization. **ACM International Symposium on Wearable Computers**. 2015. p. 29-32.

38 WANG, Xuyu et al. DeepFi: Deep learning for indoor fingerprinting using channel state information. **IEEE wireless communications and networking conference (WCNC)**. IEEE, 2015. p. 1666-1671.

39 WU, Zheng et al. A fast and resource efficient method for indoor positioning using received signal strength. **IEEE Transactions on Vehicular Technology**, v. 65, n. 12, p. 9747-9758, 2016.

40 FÉLIX, Gibrán; SILLER, Mario; ALVAREZ, Ernesto Navarro. A fingerprinting indoor localization algorithm based deep learning. **Eighth International Conference on Ubiquitous and Future Networks (ICUFN)**. IEEE, 2016. p. 1006-1011.

41 ROHRA, Jayant G. et al. User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks. **Proceedings of Sixth International Conference on Soft Computing for Problem Solving**. Springer, Singapore, 2017. p. 286-295.

42 LI, Changgeng; QIU, Zhengyang; LIU, Changtong. An improved weighted k-nearest neighbor algorithm for indoor positioning. **Wireless Personal Communications**, v. 96, n. 2, p. 2239-2251, 2017.

43 ALHAJRI, Mohamed Ibrahim; ALI, Nazar T.; SHUBAIR, Raed M. Indoor Localization for IoT Using Adaptive Feature Selection: A Cascaded Machine Learning Approach. **IEEE Antennas and Wireless Propagation Letters**, v. 18, n. 11, p. 2306-2310, 2019.

44 MALIK, R. F. et al. WLAN Based Position Estimation System Using Classification Fuzzy K-Nearest Neighbor (FK-NN). **IOP Conference Series:** Earth and Environmental Science. IOP Publishing, 2019. p. 012003.

45 JONDHALE, Satish R. et al. Comparison of Neural Network Training Functions for RSSI Based Indoor Localization Problem in WSN. **Handbook of Wireless Sensor Networks:** Issues and Challenges in Current Scenario's. Springer, Cham, 2020. p. 112-133.

46 JONDHALE, Satish R. et al. Application of Supervised Learning Approach for Target Localization in Wireless Sensor Network. **Handbook of Wireless Sensor Networks:** Issues and Challenges in Current Scenario's. Springer, Cham, 2020. p. 493-519.

47 MEDEIROS, Julio Cesar de O. **Princípios de telecomunicações–teoria e prática**. Saraiva Educação SA, 2018.

48 GOREN, David; BRIDGELALL, Raj; WILLINS, Bruce. **Multimode wireless local area network/radio frequency identification asset tag**. U.S. Patent Application n. 10/827,935, 10 fev. 2005.

49 NERGUIZIAN, Chahé; DESPINS, Charles; AFFÈS, Sofiène. Geolocation in mines with an impulse response fingerprinting technique and *Neural Net*works. **IEEE Transactions on Wireless Communications**, v. 5, n. 3, p. 603-611, 2006.

50 ALHAJRI, M. I. et al. Classification of indoor environments based on spatial correlation of RF channel fingerprints. **IEEE international symposium on antennas and propagation (APSURSI)**. IEEE, 2016. p. 1447-1448.

51 MOLISCH, Andreas F. **Wireless communications**. John Wiley Sons, 2012.

52 ANGELOV, Plamen; YAGER, Ronald. A new type of simplified fuzzy rule-based system. **International Journal of General Systems**, v. 41, n. 2, p. 163-185, 2012.

53 TAKAGI, Tomohiro; SUGENO, Michio. Fuzzy identification of systems and its applications to modeling and control. **IEEE transactions on systems, man, and cybernetics**, n. 1, p. 116-132, 1985.

54 MAMDANI, Ebrahim H.; ASSILIAN, Sedrak. An experiment in linguistic synthesis with a fuzzy logic controller. **International journal of man-machine studies**, v. 7, n. 1, p. 1-13, 1975.

55 ANGELOV, Plamen P.; GU, Xiaowei. Applications of autonomous anomaly detection. *In*: **Empirical Approach to Machine Learning**. Springer, Cham, 2019. p. 249-259.

56 ANGELOV, Plamen. **Autonomous learning systems: from data streams to knowledge in real-time**. John Wiley & Sons, 2012.

57 ANGELOV, Plamen P.; GU, Xiaowei; PRÍNCIPE, José C. A generalized methodology for data analysis. **IEEE transactions on cybernetics**, v. 48, n. 10, p. 2981-2993, 2017.

58 KANGIN, Dmitry; ANGELOV, Plamen; IGLESIAS, José Antonio. Autonomously evolving classifier TEDAClass. **Information Sciences**, v. 366, p. 1-11, 2016.

59 ANGELOV, Plamen. **Autonomous learning systems: from data streams to knowledge in real-time**. John Wiley & Sons, 2012.

60 SENOUSSAOUI, Mohammed et al. Efficient iterative mean shift based cosine dissimilarity for multi-recording speaker clustering. **IEEE International Conference on Acoustics, Speech and Signal Processing**. IEEE, 2013. p. 7712-7715.

61 AGGARWAL, Charu C.; HINNEBURG, Alexander; KEIM, Daniel A. On the surprising behavior of distance metrics in high dimensional space. **International conference on database theory**. Springer, Berlin, Heidelberg, 2001. p. 420-434.

62 BEYER, Kevin et al. When is "nearest neighbor" meaningful?. **International conference on database theory**. Springer, Berlin, Heidelberg, 1999. p. 217-235.

63 MACHAJ, Juraj; BRIDA, Peter; PICHÉ, Robert. Rank based fingerprinting algorithm for indoor positioning. **International Conference on Indoor Positioning and Indoor Navigation**. IEEE, 2011. p. 1-6.

64 LI, Qiyue et al. Fingerprint and assistant nodes based Wi-Fi localization in complex indoor environment. **IEEE Access**, v. 4, p. 2993-3004, 2016.

65 HAMMING, Richard W. Error detecting and error correcting codes. **The Bell system technical journal**, v. 29, n. 2, p. 147-160, 1950.

66 JIA, Hong; CHEUNG, Yiu-ming; LIU, Jiming. A new distance metric for unsupervised learning of categorical data. **IEEE transactions on *Neural Net*works and learning systems**, v. 27, n. 5, p. 1065-1079, 2015.

67 NOROUZI, Mohammad; PUNJANI, Ali; FLEET, David J. Fast search in hamming space with multi-index hashing. **IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2012. p. 3108-3115.

68 GU, Xiaowei; ANGELOV, Plamen P.; PRÍNCIPE, José C. A method for autonomous data partitioning. **Information Sciences**, v. 460, p. 65-82, 2018.

69 OKABE, Atsuyuki et al. **Spatial Tessellations: Concepts and Applications of Voronoi Diagrams**. John Wiley & Sons, 2009.

70 ANGELOV, Plamen P.; ZHOU, Xiaowei. Evolving fuzzy-rule-based classifiers from data streams. **IEEE Transactions on Fuzzy Systems**, v. 16, n. 6, p. 1462-1475, 2008.

71 COMANICIU, Dorin; MEER, Peter. Mean shift: A robust approach toward feature space analysis. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, n. 5, p. 603-619, 2002.

72 LUGHOFER, Edwin et al. Explaining classifier decisions linguistically for stimulating and improving operators labeling behavior. **Information Sciences**, v. 420, p. 16-36, 2017.

73 PRATAMA, Mahardhika et al. PANFIS: A novel incremental learning machine. **IEEE Transactions on *Neural Net*works and Learning Systems**, v. 25, n. 1, p. 55-68, 2013.

74 PEDRYCZ, Witold. **Granular Computing: Analysis and Design of Intelligent Systems**. CRC press, 2016.

75 Classifier comparison. **Scikit-learn Machine Learning in Python**, 2014. Disponível em: `https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`. Acesso em: 28 de nov. de 2019.

76 SIVANANDAM, S. N. et al. **Introduction to fuzzy logic using MATLAB**. Berlin: Springer, 2007.

77 RUDEMO, Mats. Empirical choice of histograms and kernel density estimators. **Scandinavian Journal of Statistics**, p. 65-78, 1982.

78 BOWMAN, Adrian W. An alternative method of cross-validation for the smoothing of density estimates. **Biometrika**, v. 71, n. 2, p. 353-360, 1984.

79 STONE, Mervyn. Cross-validatory choice and assessment of statistical predictions. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 36, n. 2, p. 111-133, 1974.

80 RODRIGUEZ, Juan D.; PEREZ, Aritz; LOZANO, Jose A. Sensitivity analysis of k-fold cross validation in prediction error estimation. **IEEE transactions on pattern analysis and machine intelligence**, v. 32, n. 3, p. 569-575, 2009.

81 EVSUKOFF, Alexandre G.; COSTA, Myrian CA; EBECKEN, Nelson FF. Parallel implementation of a fuzzy rule based classifier. **International Conference on High Performance Computing for Computational Science**. Springer, Berlin, Heidelberg, 2004. p. 184-193.

82 PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. **Journal of machine learning research**, v. 12, n. Oct, p. 2825-2830, 2011.

83 HAIR, Joseph F. et al. **Multivariate data analysis**. Upper Saddle River, NJ: Prentice hall, 1998.

APÊNDICE – Publicação cientifica

Trabalho publicado em anais de congresso:

a) DIAS, Ualison *et al.* Self-Organizing Fuzzy Rule-Based Approach for Dealing with the Classification of Indoor Environments for IoT Applications. *In:* ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL (ENIAC), 16. , 2019, Salvador. **Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional**. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. 1044-1055. DOI: https://doi.org/10.5753/eniac.2019.9356.