UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RAFAEL ALESSANDRO RAMOS

COMPARAÇÃO ENTRE TÉCNICAS DE CLASSIFICAÇÃO APLICADAS EM SINAIS DE WI-FI PARA LOCALIZAÇÃO DE CÉLULAS NAS INSTALAÇÕES DA UTFPR-CM

CAMPO MOURÃO

2022

RAFAEL ALESSANDRO RAMOS

COMPARAÇÃO ENTRE TÉCNICAS DE CLASSIFICAÇÃO APLICADAS EM SINAIS DE WI-FI PARA LOCALIZAÇÃO DE CÉLULAS NAS INSTALAÇÕES DA UTFPR-CM

Comparison Between Classification Techniques Applied To Wi-Fi Signals, For Cell Location, In The UTFPR-CM Facilities

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Lucio Geronimo Valentin

Coorientador: Prof. Dr. Roberto Wilhelm Krauss Martinez

CAMPO MOURÃO

2022

RAFAEL ALESSANDRO RAMOS

COMPARAÇÃO ENTRE TÉCNICAS DE CLASSIFICAÇÃO APLICADAS EM SINAIS DE WI-FI PARA LOCALIZAÇÃO DE CÉLULAS NAS INSTALAÇÕES DA UTFPR-CM

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 01/julho/2022

Lucio Geronimo Valentin
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

---

Roberto Wilhelm Krauss Martinez
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

---

Juliano Henrique Foleis
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

---

André Luiz Satoshi Kawamoto
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

CAMPO MOURÃO

2022

RESUMO

Sistemas de localização interna e suas aplicabilidades estão em alta, no entanto, mitigar erros e interferências são pontos fundamentais para um bom resultado além do custo de aplica-la, seja por conta da aquisição de tecnologia ou pelos recursos disponíveis para tornar viável a execução da localização. Neste trabalho, realiza-se quatro experimentos baseados em aprendizado de máquina para resolver o problema de localização interna dentro da UTFPR-CM. O objetivo principal é identificar a melhor forma de realizar o mapeamento das células (salas) e além da comparação entre as acurácias obtidas pelos classificadores clássicos K-NN e SVM, para então definir qual o modelo mais eficiente para a localização interna. Os experimentos foram realizados coletando 1, 5, 6 ou mais pontos de mapeamento por célula, e posteriormente utilizado nos classificadores K-NN e SVM para obter a acurácia de acerto. Para construção dos modelos preditivos é utilizado o método de *GridSearch* através da biblioteca *Sklearn*. O método mais eficiente foi o cenário no qual foram coletados pontos a cada 1,5m variando de 14 a 36 pontos por célula e utilizado o classificador SVM com *Kernel* linear, *C* igual a 1 e *Gamma* igual a 0,002. Obtendo uma acurácia de 82% de acerto. Este resultado é promissor e abre campo para continuidade da pesquisa além de melhorias significantes tornando a solução mais eficiente.

Palavras-chave: localizao interna; rssi; k-nn; svm; utfpr-cm.

ABSTRACT

Internal location systems and their applicability are on the rise, however, mitigating errors and interference are fundamental points for a good result in addition to the cost of applying it, either due to the acquisition of technology or the resources available to make the execution of the localization. In this work, four experiments based on machine learning are carried out to solve the problem of indoor localization within the UTFPR-CM. The main objective is to identify the best way to carry out the mapping of cells (rooms) and in addition to comparing the accuracies obtained by the classical K-NN and SVM classifiers, to then define the most efficient model for the internal location. The experiments were carried out by collecting 1, 5, 6 or more mapping points per cell, and later used in the K-NN and SVM classifiers to obtain the accuracy. To build predictive models, the GridSearch method is used through the Sklearn library. The most efficient method was the scenario in which points were collected every 1.5m, ranging from 14 to 36 points per cell, and using the SVM classifier with linear Kernel, C equal to 1 and Gamma equal to 0.002. Obtaining an accuracy of 82%. This result is promising and opens the way for further research in addition to significant improvements making the solution more efficient.

Keywords: indoor localization; rssi; k-nn; svm; utfpr-cm.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Siglas

AP	Access Point
BD	Banco de Dados
CSV	Comma-Separated-Values
GPS	Sistema de Posicionamento Global
K-NN	K-Nearest Neighbors
LF	Impressão digital de localização
MAC	Controle de Acesso de Mídia
RSSI	Indicador de Intensidade do Sinal Recebido
SSID	identificador do conjunto de serviço
SVM	Support Vector Machine
UTFPR-CM	Universidade Tecnológica Federal do Paraná Campus de Campo Mourão
Wi-Fi	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO

A localização interna consiste em definir a posição de algum elemento dentro de um determinado espaço interno. Youssef (2008) define que, a localização interna, refere-se ao rastreamento de objetos em ambientes internos, podendo possuir 2, *2.5*[1] ou 3 dimensões, reportando-se à localização estimada como uma referência simbólica, pode-se basear em coordenadas. Já, segundo Sadowski e Spachos (2018) a localização interna é utilizada para localizar objetos ou dispositivos em ambientes onde os sistemas de localizações globais não conseguem ser utilizados. Ao referir-se a ambientes internos, engloba-se estruturas como: grandes edifícios, hospitais, aeroportos, shoppings, universidades, hotéis, entre outros. Todas essas estruturas são possíveis cenários para uma boa aplicação dos métodos de localização interna.

Existem várias fontes para a obtenção das informações que podem ser utilizadas na localização interna, o RSSI é uma delas, entretanto, o mesmo não exige hardware adicional, é propenso a perdas por multi-percurso, sombreamento, perda por distancia, efeitos de Caminhos Múltiplos (*multipath effect*) e interferências em ambientes internos complexos.

A importância de um sistema para localizar pessoas e equipamentos, se resume a uma questão de otimização de recursos e logística, seja em uma empresa, universidade, shoppings, presídios, hospital, entre outros. Quando se administra uma instituição, tudo se torna um recurso, inclusive as pessoas. Dessa forma, toda a gama de informações que podem ajudar a administrar tais recursos que visa a otimização dos processos é de suma importância, para torna-los o mais eficientes possível. Nesse processo, dados contendo informações referentes à localização de pessoas podem ser uma boa tática para tomada de decisões estratégicas para a instituição. Desta forma, saber quais salas de escritórios possuem um maior fluxo de pessoas, quais lojas são mais visitadas, entre outros, são exemplos de possíveis utilizações de uma localização interna e que podem ser a peça chave para tomada de decisões que a beneficiem. Logo, a localização interna, que utiliza recursos já disponíveis, como os sinais de rede sem fio Wireless Fidelity (Wi-Fi), é de fato uma ferramenta gerencial de otimização e qualidade da gestão de recursos que trazem benefícios aos envolvidos direta ou indiretamente.

1.1 Problema de Pesquisa

A localização em ambiente interno sempre foi um desafio, pois são locais que possuem muitas interferências físicas devido a existência de paredes, de outros equipamentos emissores de rádio frequência ou até mesmo á movimentação de pessoas e objetos, distribuídos em um espaço relativamente reduzido dificultando a propagação dos sinais de radio. O maior desafio da

[1] "2,5 Dimensões refere-se ao caso em que a posição do objeto é rastreada em planos discretos do espaço tridimensional, ao invés de todo o contínuo do espaço tridimensional. Por exemplo, rastrear uma pessoa em várias plantas baixas bidimensionais em um edifício tridimensional pode ser considerado um rastreamento 2,5-dimensional."(YOUSSEF, 2008)

localização interna é a margem de erro no momento de se localizar, pois como são ambientes muito bem delimitados e compactos, deve-se obter resultados na localização mais precisos.

Atualmente existem aplicações que realizam localização em ambientes internos, entretanto, muitos acabam por utilizar, além do RSSI, recursos como sensores infravermelhos espalhados, antenas direcionais, Sistema de Posicionamento Global (GPS), altímetros, giroscópios, entre outros, para identificar um transmissor (AP) ou receptor móvel do sinal. Utilizar uma tecnologia onde somente seja levado em conta os RSSI de cada AP é um atrativo principalmente voltado para a questão financeira, uma vez que o método não necessita de gastos com equipamentos extras em razão dos transmissores e receptores já existentes, necessitando somente da aplicação para tornar a localização efetiva.

Como cenário da pesquisa, será utilizada a planta baixa da UTFPR-CM, onde um problema muito comum em universidades é a localização dos professores por parte dos alunos. Muitas vezes, saber os horários dos professores não é o suficiente para poder encontrá-los, pois o tamanha físico da universidade e a quantidade de blocos podem se tornar um desafio. Imprevistos acontecem, a dinâmica de trocas de salas, atendimentos fora dos horários das aulas podem ser frequentes e saber exatamente onde o professor está poupa muito tempo e esforço. Em muitos casos os alunos acabam circulando pelas dependências da universidade por um tempo considerável, indo de sala em sala, perguntando para servidores e alunos pela localização do professor, apresentando grandes dificuldades para localiza-los.

1.2 Objetivos

O principal objetivo deste trabalho é comparar as técnicas de localização em ambientes internos que utilizam utilizam forças de sinais provenientes de APs fixos, através de modelos preditivos como K-NN e SVM, aplicadas no cenário das instalações da UTFPR-CM.

Como objetivos específicos, destaca-se:

- Mapear a planta baixa do campus da UTFPR-CM;
- Construir a base de dados populando-a com coletas de RSSI no cenário da pesquisa;
- Construir modelo preditivo para realizar a localização por meio de RSSI utilizando o classificador K-NN;
- Construir modelo preditivo para realizar a localização por meio de RSSI utilizando o classificador SVM;
- Comparar as acurácias obtidas a partir dos modelos preditivos, entre K-NN e SVM.

1.3 Estrutura do trabalho

No Capitulo 2 são apresentados conceitos, técnicas e metodologias importantes sobre o tema da pesquisa. No Capitulo 3 são apresentados trabalhos relacionados. No Capitulo 4 é apresentado a metodologia da pesquisa e os experimentos. O Capitulo 5 apresenta vários experimentos realizados para se chegar à uma conclusão no Capitulo 6.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados alguns conceitos importantes relacionados à ondas de rádio, métodos de posicionamento e medições de sinais.

2.1 Sinais de Radio e Sinais de Luz

Tanto sinais de rádio quanto sinais de luz por definição são classificadas pelos seus comprimentos de ondas. Segundo Kjærgaard (2018), para posicionamento, os tipos de ondas mais importantes são:

- Ondas de rádio com comprimento de onda em torno de 10^3 metros;
- Luz infravermelha com comprimento de onda em torno de 10^{-5} metros;
- Luz visível com comprimento de onda em torno de $0.5x10^{-6}$ metros.

No que se refere a posicionamento, mais importante que o tipo de onda e o comprimento da mesma, é a velocidade de propagação dos sinais, pois dependendo do meio em questão as velocidades das ondas podem variar drasticamente.

2.2 Indicador de Intensidade do Sinal Recebido (RSSI)

RSSI é um método simples que mede a intensidade do sinal dos pacotes que chegam ao receptor móvel, geralmente sendo utilizado para encontrar a distância entre o transmissor e receptor do sinal. A ampla utilização do método é acessível por não requerer hardware adicional, podendo ser encontrado em qualquer dispositivo que utilize rede sem fio. Entretanto, o método pode apresentar valores imprecisos, pois é suscetível a sofrer com ruídos do ambiente. (SADOWSKI; SPACHOS, 2018)

Como o RSSI é uma métrica que indica a potencia do sinal de rádio recebido, a mesma é encontrada na grande maioria dos equipamentos que utilizam transceptores de sinais de rede sem fio, tornando o método amplamente aplicável. (GUIDARA *et al.*, 2018)

Segundo Zegeye *et al.* (2016), as técnicas para localização por meio de RSSI podem ser categorizadas em dois tipos:

- Métodos de modelagens matemáticas para definir o posicionamento do receptor móvel;
- Métodos envolvendo mapeamentos de rádio do ambiente com base em impressão digital do RSSI.

2.3 Método de posicionamento

2.3.1 Proximidade

Neste método, estima-se posições através de sensores fixos e móveis por meio de proximidade, dessa forma estima-se a posição do sensor fixo que foi registrado por ultimo. (KJæRGAARD, 2018).

- Vantagens:
  - Pode ser usado com quase todos os tipos de infraestruturas de rádio existentes;
  - Como os alvos só precisam emitir um código de identificação, eles podem ser projetados para ter um custo muito baixo, como no caso do RFID.
- Desvantagens:
  - A precisão é limitada pelo alcance dos sensores;
  - Os alvos só podem ser posicionados quando estão próximos;
  - A área onde os dispositivos estão dentro do alcance não é estática e, portanto, pode assumir formas arbitrárias. Isso significa que, se um sensor fixo for instalado em uma sala para registrar quais sensores estão na sala, é muito provável que ele também registre sensores no corredor adjacente ou perca sensores na sala.

2.3.2 Triangulação

A distância até pontos de referência conhecidos é frequentemente usada para determinar a posição, podendo ser calculada com câmeras, transmissores de infravermelho, sonar, entre outros. Um exemplo de triangulação ingênua é fazer três medições de distância e triangular a posição.

A triangulação funciona quando os sensores são confiáveis e relativamente livres de ruído. Quando os sensores possuem ruídos, os cálculos para triangulação tornam-se instáveis para posições e arranjos de pontos de referência, impactando em uma perda significativa de precisão. Várias medições podem ser combinadas ao longo do tempo para tentar para compensar as perdas, no entanto, alguns cuidados devem ser tomados na escolha do método de fusão ou resultados ruins serão obtidos.

Em casos em que os sensores são considerados confiáveis e possuem distribuições de ruído simples, a triangulação direta, ou triangulação com janelamento diferencial, pode produzir

bons resultados. Sensores com ruídos, complicam a triangulação, adicionando incerteza aos resultados. O GPS é uma tecnologia com base na triangulação.

2.3.3 Lateração

De acordo com Kjærgaard (2018), o método de lateração estima posições de medições relacionadas à distância para sensores fixos com posições conhecidas. Pode-se descrever dois tipos de lateração:

- **Lateração com distâncias absolutas**: Utiliza medidas que descrevem diretamente a distância entre um sensor móvel e vários sensores fixos. Cada medida de distância projetada formam um círculo de posições possíveis em torno dos sensores fixos. Dado um critério de erro específico à respeito aos círculos descritos é possível estimar a posição mais provável.
- **Lateração com distâncias relativas**: A lateração com distâncias relativas utiliza medidas que descrevem a relação entre as distâncias de um sensor móvel em relação a sensores fixos. Cada uma das relações entre os sensores formam uma hipérbole de posições possíveis relacionadas a pares de sensores fixos. Dado um critério de erro específico à respeito às hipérboles é possível estimar a posição mais provável.

Kjærgaard (2018) apresenta alguns prós e contras referente à utilização do método de posicionamento:

- Vantagens:
  - Pode ser usado para projetar sistemas com alta precisão;
  - Permite sistemas com grande cobertura porque as posições podem ser encontradas em todas as áreas cobertas por sensores.
- Desvantagens:
  - A maioria dos sistemas exige que sensores especiais sejam instalados na área coberta;
  - As posições dos sensores fixos devem ser estabelecidas, o que não é uma tarefa fácil em ambientes internos grandes e complexos;
  - Muitos sistemas de lateração dependem de alguma forma de sincronização de tempo que geralmente requer um cabeamento direto entre os sensores fixos;

  - A precisão pode ser severamente degradada por sinais de caminhos múltiplos. Sinais de caminhos múltiplos são sinais que não se propagam pelo caminho direto entre dois sensores. Esses sinais podem impactar as medições, de modo que os sensores parecem estar mais distantes do que realmente estão e, assim, degradam a precisão da estimativa da posição final.

2.3.4 Angulação

Para Kjærgaard (2018), o método de angulação estima posições de medições de ângulo a sensores fixos com localizações conhecidas. Cada uma das medidas de ângulo descreve uma linha de possíveis posições através das posições dos sensores fixos. Possibilitando assim, estimar a localização mais provável.

Assim como o método de Lateração o método de Angulação possui as mesmas vantagens e desvantagens, entretanto, é mais sensível aos sinais de múltiplos caminhos. A razão para tanto é que os sinais que se propagam pelo caminho direto sofrem interferências com os sinais que podem vir pela direção oposta, degradando de forma expressiva a precisão da estimativa da posição final.

2.3.5 Reconhecimento de Padrões

Conforme Kjærgaard (2018), através de reconhecimento de padrões relacionados à posições através das medições é possível estimar uma posição. Cada padrão deve estar disponível em alguma codificação, cada codificação por sua vez deve conter um mapeamento e para cada padrão uma posição relacionada. O método pode ser aplicado para padrões de reconhecimento de LF com a intensidade do sinal.

- Vantagens:
  - Pode oferecer suporte ao rastreamento de pessoas ou itens não marcados;
  - Pode ser aplicado a muitos tipos de medições.
- Desvantagens:
  - Os padrões devem ser registrados/codificados para que o método funcione;
  - No caso de sistemas de visão, uma infraestrutura de câmeras é necessária e as câmeras precisam de uma linha de visão direta para os objetos rastreados.

2.3.6 Impressão digital de localização (LF)

A impressão digital de localização (LF), codifica padrões de radio com base em impressões digitais e em cada mapa contem um mapeamento para cada padrão codificado relacionado à uma posição (KJæRGAARD, 2018).

- Vantagens:
  - Evita a necessidade de uma infraestrutura especialmente instalada, sendo possível utilizar as infraestruturas já disponíveis.
- Desvantagens:
  - **Heterogeneidade**: A heterogeneidade se da pois, mesmo se considerando clientes com a mesma tecnologia, pode-se ter diferenças entre os sinais de rádio, antenas e até mesmo *firmware*s[1] diferentes. Dessa forma quando se faz uma medição utilizando LF pode-se haver divergências tanto para mais quanto para menos em uma mesma posição ou até mesmo na sensibilidade do sensor de rádio, tornando assim as medições não comparáveis entre os clientes. O impacto que se resulta é a drástica diminuição da precisão utilizando o LF (KJæRGAARD, 2018);
  - **Escalabilidade**: Dimensionar o cálculo da estimativa, distribuição de medições e distribuição de estimativas de posição pode-se tornar um problema quando muitos clientes estão envolvidos, pois o cálculo de tais estimativas é muito exigente devido à quantidade de cálculos envolvidos, além disso, caso as medições de estimativas de posições não forem calculadas nas medições dos clientes, estas devem ser distribuídas tornando-se um verdadeiro desafio devido à frequência com que as medições são realizadas e da taxa de atualizações. (KJæRGAARD, 2018);
  - **Interferências**: O posicionamento utilizando LF requer a medição da intensidade do sinal dos APs próximos. No entanto, muitas tecnologias de comunicação sem fio separam a comunicação dividindo suas bandas de frequência em canais separados. As estações base (APs) para uma tecnologia normalmente operam apenas em um canal. Portanto, para medir todas as estações base próximas, os clientes precisam varrer todos os canais e bloquear a comunicação deixando o canal de comunicação atual. Este é um desafio porque não é desejável que o posicionamento LF quando em uso desabilite a comunicação.
  - **Necessidade de calibração**: Necessita calibração pois com a alteração dos APs e das redes é necessário refazer o mapeamento;

[1] Também conhecidos pela nomenclatura “Software Embarcado”, os *Firmware* são um conjunto de instruções operacionais que são programadas diretamente no hardware de equipamentos eletrônicos.

- **Reflexão**: Ocorre quando a onda incide em uma determinada superfície e retorna para o mesmo meio;
- **Refração**: Ocorre quando existe um desvio do percurso do sinal resultando em uma nova trajetória;
- **Difração**: Ocorre quando as ondas conseguem desviar obstáculos ou propagam-se por aberturas;
- ***Espelhamento***: Ocorre quando as ondas sofrem alterações no percurso original e são refletidas por múltiplas direções diferentes;
- **Caminhos Múltiplos**: Ocorre quando os pacotes acabam refletindo e retornando ao emissor de sinal, gerando perda significativas;
- **Imprecisão**: Não é capaz de fornecer a precisão centimétrica obtida com alguns dos outros métodos.

De modo geral métodos podem ser combinados para se obter a melhor precisão geral resultante, assim como o caso da combinação de Angulação e LF em (NICULESCU; NATH, 2004).

2.4 Medição de Sinais IEEE 802.11

O IEEE 802.11 subdivide o espectro de rádio usado em um conjunto de canais. Isso é importante para a varredura porque um cliente sem fio só pode ouvir um canal por vez. Portanto, durante a varredura, um cliente sem fio precisa sintonizar cada canal, um após o outro, para descobrir todas as estações base no alcance de comunicação.

Varredura Passiva Exige que o cliente sem fio apenas ouça a rede. O cliente sem fio permanece ouvindo os quadros de *beacon* em cada canal. Cada quadro de *beacon* é enviado por um AP periodicamente para manter a rede. Os quadros de *beacon* contém informações sobre a rede e são normalmente enviados a cada 100 ms, tempo este que pode ser configurado, (KJæRGAARD, 2018).

- Vantagens:
  - Nenhuma comunicação é necessária, a técnica é leve em termos de consumo de energia;
  - Preserva a privacidade do cliente porque a existência do cliente não é revelada. Portanto, o cliente sem fio pode se posicionar usando LF, mas permanece privado.
- Desvantagens:
  - Leva mais de um segundo para realizar cada varredura.

Varredura Ativa Exige que solicitações frequentes de um cliente sem fio às estações base que se identifiquem para o cliente sem fio. Em cada canal onde o cliente sem fio envia uma solicitação o mesmo aguarda as respostas dos APs no mesmo canal. O AP ao receber a solicitação do cliente sem fio responde o mais rápido possível com uma resposta de sondagem, onde conterá informações sobre a rede como: nome da rede e as taxas de dados. O receptor móvel aguarda uma resposta no canal ao enviar uma solicitação antes de ir para o próximo canal, o tempo de espera pode ser configurável. No máximo 20 milissegundos de espera pelo receptor móvel são o suficientes para cada canal. Levando em consideração uma varredura por todos os canais pode levar menos de 260 milissegundos.(KJæRGAARD, 2018)

- Vantagens:
  - Exige menos de 260 milissegundos para suportar uma frequência de amostragem de quase 4 Hz.
- Desvantagens:
  - Os clientes precisam enviar ativamente solicitações que revelam a existência do cliente e consomem energia.

2.5 K-Nearest Neighbors - K-NN

O K-NN, (FUKUNAGA; NARENDRA, 1975), é um dos mais triviais algoritmos de aprendizado de máquina baseado na técnica de aprendizado supervisionado. O algoritmo armazena os dados disponíveis e classifica o novo ponto com base na similaridade. O algoritmo é utilizado principalmente para problemas de classificação.

É um algoritmo não paramétrico, ou algoritmo de aprendizado preguiçoso pois, sua etapa de treinamento não é utilizada para aprendizado mas sim para armazenamento, para que no momento da classificação realizar a classificação do novo ponto de dados com alguma das classificações mais semelhantes contida no conjunto de dados armazenados.

O K-NN funciona basicamente seguindo os seguintes passos:

- Selecionar o número de *K* vizinhos;
- Calcular a distância de *K* com o número de vizinhos;
- Compara o número de *K* vizinhos mais próximos considerando as distâncias calculadas;
- Retorna a categoria para o qual o ponto de dados possui a mais probabilidade de pertencer.

A escolha do valor de *K* interfere diretamente na resultante do algoritmo, escolher um valor muito baixo para *K* pode ser ruidoso e levar aos efeitos de *outliers*[2] no modelo, por outro lado se escolher valores muito grandes, apesar de bons, podem dificultar no momento da decisão. Deve-se encontrar um valor ajustado, dando preferência à números ímpares, caso contrário há o risco de acontecer empates dificultando a decisão. Comumente o valor de *K* utilizado é 5.

- Vantagens:
  - Algoritmo simples de ser implementado;
  - É robusto para dados de treinamento com muitos ruídos;
  - Eficaz para dados de treinamentos grandes.
- Desvantagens:
  - Determinar o valor de *K* ideal é complexo;
  - Alto custo computacional.

2.6 Support Vector Machine - SVM

O Support Vector Machine (SVM) é uma técnica de aprendizado embasada pela teoria de aprendizado estatístico. Como meio para uma boa generalização esta teoria estabelece alguns princípios a serem seguidos. Desta forma, dada uma nova entrada existente no domínio utilizado na etapa de aprendizagem, o classificador é capaz de inferir a qual classe as novas entradas pertencem. (CORTES; VAPNIK, 1995).

Um bom classificador deve garantir o melhor desempenho de generalização possível. A teoria do aprendizado estatístico (VAPNIK, 1995), da qual os SVMs se originaram, estabelece condições matemáticas que auxiliam na escolha do classificador. O SVM é responsável por encontrar o hiperplano que separa, de acordo com os rótulos/classes, os dados de treinamento que estão maximamente distante da fronteira de separação (hiperplano de margem máxima), ou seja, o SVM busca encontrar o melhor hiperplano para um dado data set cujas classes são linearmente separáveis. Quando nenhuma separação linear é possível, um mapeamento não linear em um espaço de recursos de dimensão mais alta é realizado. O hiperplano encontrado no espaço de características corresponde a um limite de decisão não linear no espaço de entrada. (CRISTIANINI; RICCI, 2008).

[2] Dados discrepantes; Dados que diferem drasticamente dos demais.

3 TRABALHOS RELACIONADOS

Nesta sessão será apresentado trabalhos com alguma ligação a esta pesquisa que apresentam fatores que favorecem o entendimento do assunto de forma geral.

O estudo de Zhu e Feng (2013), aborda que a localização do nó sem fio é uma das tecnologias-chave para redes de sensores sem fio. Como a localização interna requer maior precisão, o uso de GPS ou para localização interna não era viável na visão da época. O algoritmo de localização trilateral baseado em RSSI se torna convencional em redes de sensores sem fio, devido ao seu baixo custo, sem suporte de hardware adicional e fácil compreensão. O problema principal é que o valor RSSI é relativamente vulnerável à influência do ambiente físico, causando grande erro de cálculo no algoritmo de localização baseado em RSSI. O artigo propõe um algoritmo baseado em RSSI melhorado, os resultados experimentais mostram que, em comparação com os algoritmos de localização originais baseados em RSSI, o algoritmo melhora a precisão da localização e reduz o desvio.

O *RADAR*, Bahl e Padmanabhan (2000), foi o primeiro sistema de localização interna baseado em radio frequência apresentado Trata-se de um sistema baseado em rádio frequência para localizar e rastrear usuários dentro dos edifícios. O sistema utiliza informações de intensidade do sinal reunidas para triangular as coordenadas do receptor móvel. A triangulação é feito usando informações de intensidade de sinal empíricas determinadas e teoricamente calculadas. o *RADAR* é capaz de estimar a localização de um usuário a poucos metros da sua localização real. Isso sugere que uma grande classe de serviços com localização de localização pode ser construída em uma rede de dados sem fio. Este algoritmo é utilizado como base para vários estudos ao longo do tempo justamente por sem um dos pontos de partida do tema.

O algoritmo do *Centroide* posiciona um ponto central de todos os sinais ouvidos durante uma varredura. Calcula-se uma média das posições estimadas, com base na intensidade das redes, de cada um dos APs ouvidos, resultando no ponto central geográfico (CHENG *et al.*, 2005). Por outro lado, o algoritmo de *Centroide Ponderado* é similar ao do *Centroide*, entretanto, não resulta no centro geográfico dos sinais ouvidos, mas sim atribui pesos para cada sinal, para então, calcular a posição estimada (ZHAO *et al.*, 2014).

Como relatado no artigo de Zhao *et al.* (2014), há uma proposta de um novo algoritmo de localização AP baseado em gradiente SSI. Consiste nas seguintes três etapas principais: em primeiro lugar, usa as variações de intensidade de sinal recebidas locais para estimar a direção (menos gradiente) de AP e, em seguida, emprega um método de agrupamento de direção para identificar e filtrar medições de valores aberrantes e, finalmente, adota o método de triangulação para localizar AP com as direções de gradiente selecionadas. Os resultados experimentais demonstram que o erro de localização médio do algoritmo proposto é inferior a 2 metros, superando em grande parte o da abordagem centroide ponderada.

Guidara *et al.* (2018) em sua pesquisa exploraram os impactos da temperatura e umidade relativa do ar sobre o RSSI em ambiente controlado. Foi apresentado uma forte relação

negativa entre a temperatura e o RSSI para distâncias entre transmissor e receptor do sinal de pelo menos 5 metros. Para distâncias menores que 5 metros os impactos negativos diminuem. Outra resultante do experimento foi uma forte relação positiva entre a umidade relativa do ar e o RSSI, quanto menor a distância entre transmissor e receptor do sinal menor o impacto positivo entre a umidade e o RSSI.

Han *et al.* (2009) propôs um algoritmo de localiza denominado *Gradient* para localizar APs. No artigo os dados foram obtidos através da técnica de busca *Wardriving*[1]. A ideia principal do algoritmo é apontar setas em direção à fonte fixa do sinal, estimando a direção do AP a cada medição. Após a combinação das setas resultantes estima-se a localização do AP. O algoritmo *Gradient*, durante a pesquisa, reduziu o erro máximo em 33%, se comparado com a abordagem do Centroide Ponderado que foi a motivação do autor.

O artigo de Bahl (BAHL; PADMANABHAN, 2000), aborda a viabilidade de construir um sistema de posicionamento baseado em Wi-Fi 802.11 de área ampla. Ele compara um conjunto de algoritmos de posicionamento baseados em rádio sem fio para entender como eles podem ser adaptados para essa implantação ubíqua com uma calibração mínima. Em particular, ele estuda o impacto dessa calibração limitada na precisão dos algoritmos de posicionamento. Os experimentos mostram que é possível estimar a posição de um usuário com um erro de posicionamento médio de 13 a 40 metros (dependendo das características do ambiente). Embora essa precisão seja menor do que os sistemas de posicionamento existentes, ele requer uma sobrecarga de calibração substancialmente mais baixa e fornece fácil implantação e cobertura em grandes áreas metropolitanas.

Assim como os trabalhos citados acima, este aborda uma metodologia de posicionamento baseado nas forças dos sinais Wi-Fi para localizar células dentro da UTFPR-CM. Diferente dos demais citados será utilizado classificadores para inferir qual a célula mais provável do receptor móvel estar localizado, dessa forma, não é necessário uma precisão milimétrica uma vês que não se deseja a posição exata dentro da célula mas apenas qual a célula que o receptor está. Existem várias pesquisas na área, entretanto é a primeira vez que será estudado no cenário da UTFPR-CM.

[1] Prática de procurar por redes sem fio dirigindo um automóvel.

4 METODOLOGIA

A proposta da pesquisa é viabilizar a utilização do método de localização interna baseado em RSSI, para identificar em qual célula o receptor móvel está. São utilizados classificadores clássicos (K-NN e SVM) para inferir qual a célula mais provável e posteriormente comparar qual classificador retorna melhor acurácia. Um ponto fundamental para a pesquisa é identificação em qual célula o receptor móvel está e não em que local da célula o receptor móvel está.

4.1 Reconhecimento do cenário

Como cenário para a pesquisa, foi escolhido o campus da UTFPR-CM, onde a mesma apresenta uma estrutura física propícia para realizar a aplicação dos estudos apresentados pela pesquisa referente à localização interna.

A UTFPR possui, até então, um bloco térreo administrativo e oito blocos alinhado contendo dois andares cada, como mostra a Figura 1. Cada andar possui um corredor central e salas situadas tanto à direita quanto à esquerda de cada pavimento. A disposição dos APs, dentro do bloco é, em sua maioria, seguindo a extensão de cada corredor de cada bloco, organizados de forma alinhada, logo, se o bloco possui dois andares teremos uma fila de APs no térreo e outra no primeiro andar.

Vale frisar que, para validar o cenário é necessário saber se existe uma quantidade mínima de APs distribuídos pelo local. Suas localizações não necessariamente precisam ser conhecidas. A Figura 1 mostra os APs com intuito de validar a quantidade distribuída pelo cenário e não sua localização exatamente. Este ponto é crucial de se entender, pois no momento do mapeamento das células não é necessário conhecer suas localizações exatas.

Além de toda a estrutura física característica da UTFPR-CM somada com a quantidade de APs, a universidade apresenta um grande número de pessoas circulando o ambiente, onde cada um é um possível receptor móvel por possuir um aparelho celular ou notebook. Um dos recursos mais importantes na universidade é o professor, e saber onde localizá-lo é imprescindível, tanto para a instituição quanto para os alunos. Pois, para a instituição que possui o professor como recurso próprio, ter acesso a um histórico de rastreio pode ser muito importante para tomadas de decisões estratégicas, como saber quais as salas mais movimentadas e em quais horários, torna-se possível a identificação de tais padrões resultando em possíveis remanejamentos de turmas melhorando a logística do ambiente... O fato é, que obter a localização de recursos em ambientes internos dentro da universidade torna-se um beneficio, possibilitando a criação de aplicações para diversos fins, utilizando-se dos dados capturados. Tais características somam positivamente para a escolha do cenário propício para a aplicação dos métodos de localização interna.

As Figura 1 apresenta o cenário da pesquisa, a estrutura do campus UTFPR-CM, em formato de mapa e vista por satélite. Nota-se a disposição das estruturas pelo campus universitário, com foco nos blocos agrupados de forma ordenada. Destaca-se em vermelho e amarelo os pontos onde estão localizados APs inferiores e superiores respectivamente, tornando visível as possibilidades de triangularização entre os mesmos, possibilitando a localização de um ponto móvel.

Figura 1 – Posicionamento dos APs no Campus UTFPR-CM: (a) Mapa da estrutura, (b) Imagem de satélite

(a) Mapa da estrutura **(b) Imagem de satélite**

Fonte: Fonte Modificada: Google Maps - Imagens 2021 Maxar Technologies.

Considerando que cada bloco possui dois andares, e que em ambos apresentam APs instalados, vide Figura 2, o cenário apresenta não somente uma estrutura composta pelos eixos X e Y, como também por um eixo Z compondo um cenário tridimensional.

4.2 Mapeamento das Células

Uma vez reconhecido o cenário da pesquisa, é necessário realizar a coleta dos dados para o mapeamento da intensidade dos sinais RSSI provenientes dos APs distribuídos em cada célula pelos blocos da UTFPR-CM.Será utilizado o método de impressão digital de sinais, LF através de um receptor móvel de sinal.

Uma característica importante para a pesquisa é que, tanto no momento do mapeamento quanto no momento da localização, a posição física dos APs não são conhecidas. As Figura 1 e 2, apresentam as localização dos APs no ambiente físico, apenas para evidenciar que o cenário contem os parâmetros para a aplicação da pesquisa, entretanto, no momento da execução da mesma, as posições dos APs não são importantes, justificando assim o uso do método de LF. Pois com este método mediremos apenas as forças dos sinais que chegam até o receptor móvel independente de onde os APs estejam.

Figura 2 – Posicionamento dos APs nas plantas baixas do Bloco-E UTFPR-CM: (a) Andar Térreo, (b) Primeiro Andar

(a) Andar Térreo **(b) Primeiro Andar**

Fonte: Autoria própria (2022).

Uma célula é definida como um espaço fisicamente delimitado, onde qualquer ponto que atenda os parâmetros predefinidos será considerado pertencente à célula. No caso desta pesquisa, será considerada como célula cada sala pertencente a cada bloco da UTFPR-CM, realizando coletas de RSSI, através do método de LF, dentro de cada célula e armazenando as intensidades dos sinais em um Banco de Dados (BD) para, posteriormente, realizar a construção do modelo preditivo.

A base de Dados para esta pesquisa foi construída manualmente, ou seja, os registros dos pontos dentro da célula foram coletados pessoalmente, ponto a ponto. A quantidade de pontos varia, como esta é uma pesquisa inicial foram criadas bases distintas simulando padrões de coletas diferentes, em pontos diferentes no interior de cada célula.

Na tentativa de prever a melhor forma de realizar o mapeamento manualmente, foram criadas três bases para o mapeamento, como exemplificado na Figura 3. A primeira contendo apenas um ponto central, outra contendo cinco pontos com foco nas extremidades da célula, por fim a terceira contendo seis pontos por célula. As bases foram utilizadas nos experimentos posteriores onde seus resultados serão analisados. Foi coletada também uma base contendo pontos diferentes dos coletados na base de treino constituindo assim a base utilizada para teste no modelo preditivo. Dessa forma possui-se uma base para treino e uma base para teste. Além das três bases citadas anteriormente, foi mapeada uma quarta base de dados contendo 205 medições no total com um espaçamento de aproximadamente 1,5m de distância entre um ponto e outro dentro da célula.

As coletas dos dados foram realizadas através de um único *notebook* simples utilizando o sistema operacional *Windows 10* como receptor móvel apenas no *Bloco E* do campus. O intervalo entre a coleta de um ponto e outro foi de aproximadamente dois minutos ou até que os valores se estabilizassem. Este processo de espera para convergência dos valores dos sinais é necessária por conta da taxa de atualização dos sinais que chegam nas placas de rede dos receptores móveis. Como se trata de uma coleta manual a estratégia empregada é o de aguardar um tempo mínimo, caso contrário, a chance de registros errôneos ou até mesmo idênticos à outros pontos são grandes, podendo induzir os modelos preditivos ao erro no momento da predição, diminuindo assim a acurácia.

O mapeamento das células e a quantidade de pontos a ser colhida em cada uma pode influenciar no resultado final da localização, podendo ter impactos diretamente na acurácia final. A opção de escolher poucos pontos é que o mapeamento se trata de um processo manual e humanamente possível. As coletas foram realizadas em um dia com temperaturas próximas do 22ºC, seco e com o Campus vazio, sem alunos.

Figura 3 – Posicionamento dos pontos de medição de intensidades de sinais no Primeiro Andar do Bloco-E: (a) 1 ponto por célula, (b) 5 pontos por célula, (c) 6 pontos por célula

(a) 1 ponto por célula **(b) 5 pontos por célula** **(c) 6 pontos por célula**

Fonte: Autoria própria (2022).

4.3 Estrutura e Armazenamento dos Dados da Coleta

Cada ponto coletado em uma célula é armazená-los em um BD, para isso escolheu-se o *MongoDB*, pois se trata de um banco *NoSQL*[1] orientado a documentos e totalmente livre de

[1] O termo NoSQL é referente a ausência do SQL.

schemas[2]. Como em cada medição a quantidade de redes é variável, no momento da preparação dos dados, um modelo NoSQL se torna mais interessante para trabalhar os dados. Em sua estrutura baseada em documentos, o MongoDB armazena os dados da seguinte forma:

- *ID* do objeto no banco;
- Nome da célula;
- Array de APs onde os sinais chegaram ao receptor móvel, contendo:
  - **SSID**: Nome da rede;
  - **Endereço MAC**: Endereço único do aparelho emissor do sinal;
  - ***Qualidade* (RSSI)**: Sinal que chega até o receptor móvel.

4.4 Construção do Modelo Preditivo

Para a construção do modelo preditivo foi utilizado a base de dados coletadas e adequando-os. Inicialmente foi desenvolvido uma aplicação onde são realizadas as leituras para a construção da base de dados. Através da base utilizada para mapear as células, foi possível identificar quais colunas apresentou valores de leitura de sinais, sendo assim, possível grava-los em um arquivo do tipo Comma-Separated-Values (CSV). Conforme cada registro da base de mapeamento é lido, as leituras de cada AP se transformam em uma coluna no arquivo CSV de treino, respeitando a ordenação na qual elas são lidas.

Identificadas todas as redes, e sua ordenação, contidas na base de mapeamento das células, são adicionados os valores das medições considerando todas as redes. Caso a rede não exista na leitura obtida é adicionado o valor zero. Ao final de cada leitura é adicionado o nome da célula. Este arquivo é utilizado como base de treino para o classificador.

Uma vez obtidas todas as redes contidas no arquivo de treino, serão elas que devem ser consideradas no modelo preditivo. Para tanto, no momento de obter os dados contidos no banco para teste, o sistema verifica as mesmas colunas que foram consideradas para construir o arquivo de treino, garantindo assim que todas as colunas estejam contidas e na mesma ordenação no arquivo CSV de treino. As redes "extras"contidas na base de teste são desconsideradas. Dessa forma, as colunas da base de teste são diretamente ligadas à base de treino.

A opção de criar bases diferentes se da principalmente pois é uma forma simples e eficiente de garantir que as quantidades e os locais de cada medição serão iguais para cada célula, possibilitando uma base bem balanceada. Outro fator, é que assim é garantido que a base não possuirá medições idênticas, pois respeitando as regras para o mapeamento do ponto respeita o tempo de atualização da placa de rede e convergência dos valores que chegam até o receptor móvel.

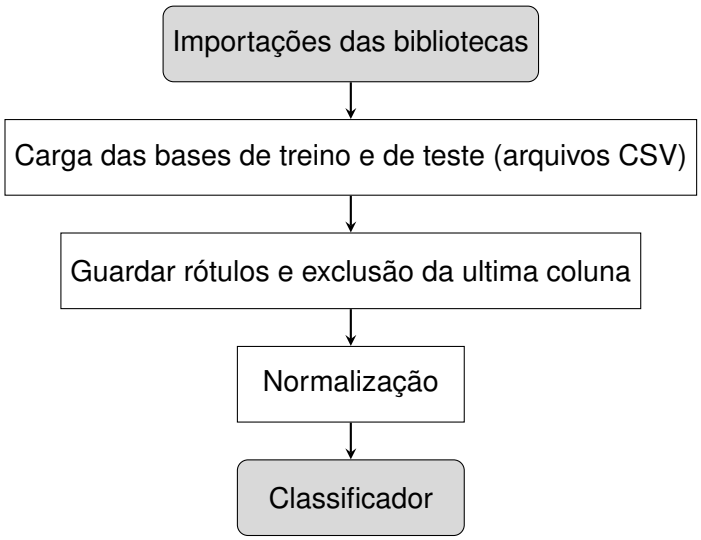
[2] Schemas são uma coleção de objetos dentro de um determinado banco de dados, servem para agrupar objetos no nível de aplicação.

4.5 Modelos preditivos

Os modelos preditivos são construídos na linguagem Python, utilizando a biblioteca Sklearn (PEDREGOSA *et al.*, 2011), por conta da sua facilidade de implementação e a farta disponibilidade de documentação.

Para a construção dos modelos preditivos inicialmente foram seguidos o fluxograma comum tanto para o classificador K-NN quanto paro classificador SVM, representado pela Figura 4. O item "*Classificador*"será esmiuçado individualmente.

Figura 4 – Diagrama com o fluxo de criação do modelo preditivo.



Fonte: Autoria própria (2022).

O primeiro ponto para o modelo preditivo é adicionar as importações necessárias para a utilização das ferramentas necessárias para a construção (pandas, numpy, matplotlib e sklearn).

Através da biblioteca *pandas* é possível abrir os arquivos CSVs construídos na etapa de pré-processamento, tanto para o arquivo de treino quanto o arquivo de teste.

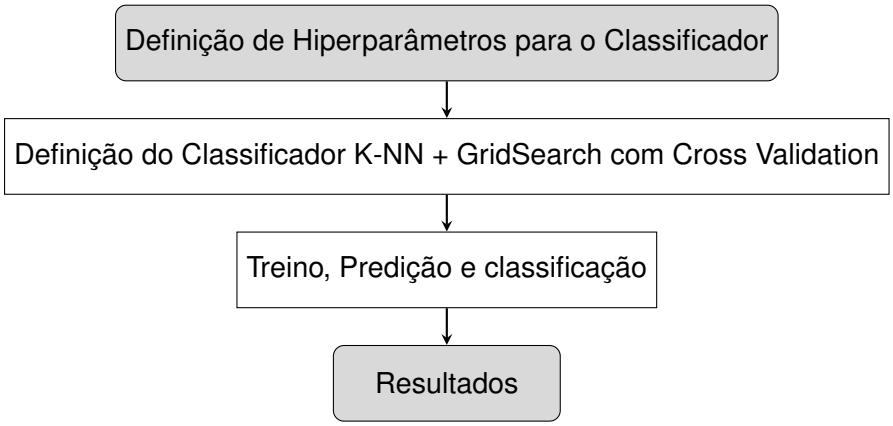
Após o carregamento dos arquivos, deve-se criar uma lista com os rótulos de cada célula contida na base de dados. Após armazenar as classes é necessário remover a coluna para que as taxas sejam fidedignas. Este processo é realizado em ambas as bases.

Deve-se então normalizar os dados, esta ação é necessária pois dessa forma remove-se as divergências entre os valores obtidos com os dados brutos. Esta ação é realizada através da função *"MinMaxScaler()"*, contida na biblioteca Sklearn (PEDREGOSA *et al.*, 2011). Uma vez que o *MinMax* considera valores não negativos, apresenta uma representação dos dados mais compreensível com os sinais reais emitidos pelos s. A normalização é construída com base no treino e aplicada na base de treino e teste, garantindo assim que ambas as bases estejam na mesma faixa de valores.

4.5.1 Classificador K-NN

Para a construção do modelo preditivo foi seguido o fluxograma representado pela Figura 5

Figura 5 – Diagrama com o fluxo de criação Classificador K-NN do modelo preditivo.



Fonte: Autoria própria (2022).

Uma vez com ambas as bases preparadas, deve-se configurar o classificador a ser utilizado no modelo preditivo, K-NN no caso. Para tanto, foi utilizado o classificador K-NN na implementação da biblioteca Sklearn. Para abranger uma melhor gama de resultados e conseguir avaliar maiores possibilidades é utilizado a busca exaustiva por hiperparâmetros, onde a própria biblioteca consegue alterar entre os valores dos vizinhos (*K*) próximos e as métricas pré-definidas.

As listas contendo os valores de *K* e as métricas desejadas devem ser passadas como parâmetro para o classificador. Utilizar a função "*GridSearchCV()*"passando como parâmetro o classificador ("*KNeighborsClassifier()*"), os hiperparâmetros e o valor do *CrossValidation* (PEDREGOSA *et al.*, 2011).

Segue valores para os hiperparâmetros:

- **Número de Vizinhos (K)**: 3, 5, 7, 9, 11, 13, 15, 17, 19, 21;
- **Métricas**: "euclidean", "manhattan", "chebyshev", "minkowski", "braycurtis";
- ***Cross Validation* (CV)**: 5.

O *CrossValidation* particiona a base de dados de forma aleatória e com mesma quantidade de amostras conforme valor setado. Cada iteração entre os conjuntos treino e teste, um novo conjunto contendo *k-1* subconjuntos é criado para ser utilizado como treino e o restante utilizado para teste. Cada resultado obtido em cada parte são então utilizados para calcular um resultado médio entre todos os valores para então definir como resultado final (PEDREGOSA *et al.*, 2011). Os valores para o *CrosValidation* deve variar conforme a quantidade de classes disponíveis no treinamento.

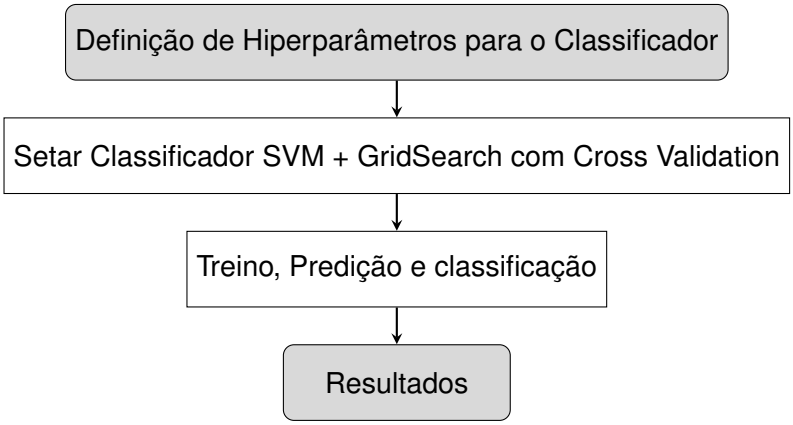
Após o treinamento, é realizada a predição utilizando a base teste e por fim a classificação. Dessa forma possui-se um modelo criado, treinado e testado, podendo apresentar suas taxas de acurácia. Pode-se também, verificar quais foram os melhores parâmetros no classificador utilizados para obter a melhor acurácia com a menor taxa de erro entre todos os que o *GridSearch* testou.

Por fim, é possível criar visualizações, como a matriz de confusão, gráficos, entre outros. Uma das opções para tais representações visuais é utilizar a biblioteca "*matplotlib*".

4.5.2 Classificador SVM

Assim como no classificador anterior, segue-se passos similares para a criação do modelo preditivo, alterando detalhes específicos na implementação. Segue-se o fluxo da Figura 6

Figura 6 – Diagrama com o fluxo de criação do Classificador SVM do modelo preditivo.



Fonte: Autoria própria (2022).

Com as bases de treino e de teste prontas, deve-se configurar os parâmetros do classificador SVM. Para isso, foi utilizado o classificador SVM na implementação da biblioteca *Sklearn*. Para abranger uma melhor gama de resultados e conseguir avaliar melhor será utilizado o método de busca exaustiva por hiperparâmetro, onde a própria biblioteca consegue alterar entre os valores de (*C*), *gama* e *Kernels* pré-definidas e devolver o melhor resultado encontrado.

As listas contendo os valores de *C*, *Gamma* e os *kernels* que se deseja avaliar devem ser passadas como parâmetro para o classificador. Utilizar a função "*GridSearchCV()*"passando como parâmetro, o classificador ("*SVC()*"), as listas de parâmetros e o valor do *CrossValidation*, setado no valor cinco pois depende diretamente do tamanho da base. Como as bases colhidas têm no máximo seis classes foi padronizado o valor cinco para todos os casos (PEDREGOSA *et al.*, 2011).

Segue valores para os hiperparâmetros:

- **Kernels**: "rbf", "poly", "sigmoid", "linear";

- **Tolerância a Erros ou Custo (C)**: 1, 10, 100, 1000;
- **Gamma**: 2e-3, 2e-2, 2e-1, 'auto';
- ***CrossValidation* (CV)**: 5.

Após o treinamento, é realizada a predição utilizando a base teste e por fim a classificação. Dessa forma possui-se um modelo criado, treinado e testado, podendo apresentar suas taxas de acurácia. Pode-se também, verificar quais foram os melhores parâmetros no classificador utilizados para obter a melhor acurácia com a menor taxa de erro entre todos os que o *GridSearch* testou.

Por fim, é possível criar visualizações, como a matriz de confusão, gráficos, entre outros. Uma das opções para tais representações visuais é utilizar a biblioteca "*matplotlib*".

4.6 Comparação Modelos Preditivos

A comparação é realizada a partir das acurácias resultantes dos experimentos. É realizado uma análise e apresentado o melhor classificador com os melhores hiperparâmetros para a construção do modelo preditivo. também é realizado uma avaliação das taxas contendo os graus de certezas de pontos específicos dentro de uma mesma célula e então avaliar se existe algum padrão interno à célula.

5 EXPERIMENTOS

5.1 Experimento I

Para o Experimento I, foi considerado apenas um bloco do campus. Foi realizado a coleta contendo apenas 1 ponto de medição por célula, ponto central, exemplificado pela Figura 7. Obteve-se um total de 9 registros no BD. Com base nos registros foi criado um arquivo no formato CSV contendo os identificadores (Identificador do conjunto de serviço (SSID) + Endereço Controle de Acesso de Mídia (MAC)) de cada rede, para então ser utilizado como base de treino para os classificadores K-NN e SVM, utilizados como modelo preditivo.

Figura 7 – Cenário de coleta de pontos centrais para o Experimento I: (a) Andar Térreo, (b) Primeiro Andar

Fonte: Autoria própria (2022).

Como teste, foi utilizado uma base contendo 27 medições em pontos diferentes dos utilizados para o processo de mapeamento das células, de forma distribuídas, englobando o máximo de células contidas na base de treino. Utilizando-se dos identificadores obtidos no pré-processamento da base de treino foi possível ordenar e atribuir valores zerados para os identificadores que não foram identificados na medição. Neste pré-processamento dos dados, foram também descartado RSSIs não identificados na base de treino, garantindo assim que todas as colunas e suas ordenações estejam idênticas, tanto no treino quanto no teste. Este processo funciona da mesma forma nos demais experimentos.

5.1.1 Resultados Experimento I

Segue os resultados obtidos para o cenário e parâmetros descrito no experimento, tanto para o classificador K-NN quanto para o classificador SVM:

5.1.1.1 K-NN

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, a métrica "*chebyshev*"com o valor de *K* vizinhos igual a 3. Obtendo-se uma acurácia de 30%, conforme apresentados na Figura 8.

Figura 8 – Resultados do Experimento I com Classificador K-NN.

	precision	recall	f1-score	support
E001	0.23	1.00	0.38	3
E003	0.29	0.67	0.40	3
E007	0.00	0.00	0.00	4
E100	0.00	0.00	0.00	3
E101	0.50	0.25	0.33	4
E102	1.00	0.67	0.80	3
E103	0.00	0.00	0.00	0
E104	0.00	0.00	0.00	3
E105	0.00	0.00	0.00	4
accuracy			0.30	27
macro avg	0.22	0.29	0.21	27
weighted avg	0.24	0.30	0.22	27

Fonte: Autoria própria (2022).

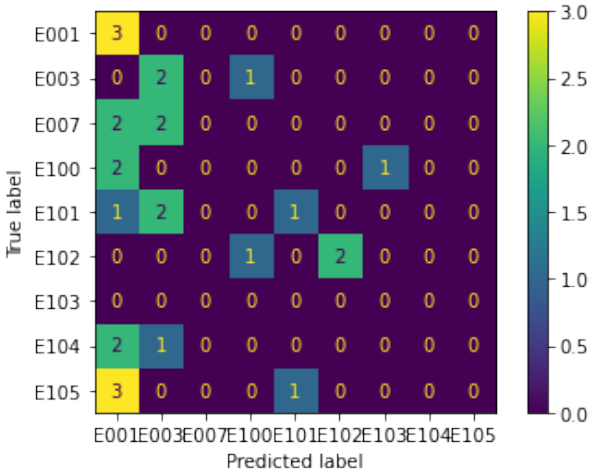
Nota-se que a acurácia é muito baixa, onde grande parte desse resultado se da pelo fato da base de treino possuir dados insuficientes por classe, não sendo possível utilizar a validação Cruzada. A validação Cruzada necessita de pelo menos dois membros por classe para ser possível utiliza-la. No caso do Experimento, a base de treino possui apenas um membro por classe. Neste caso a própria biblioteca lida com o valor "*None*"configurado para o *Cross-Validation* atribuindo valores padrões retornando assim uma acurácia. Tal valor retornado não é confiável.

Analisando a matriz de confusão na Figura 9 nota-se claramente que o resultado é basicamente aleatório, ou melhor dizendo, tende para o erro. Onde a diagonal principal possui poucos valores e os pontos de erros são totalmente sem padrão.

5.1.1.2 SVM

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, o Kernel "*RBF*", com o valor de custo *C* igual a

Figura 9 – Matriz de Confusão referente ao Experimento I com Classificador K-NN.



Fonte: Autoria própria (2022).

1 e o valor de Gamma igual a 0,002. Obtendo-se uma acurácia de 41%, conforme apresentados na Figura 10.

Figura 10 – Resultados do Experimento I com Classificador SVM.

	precision	recall	f1-score	support
E001	0.60	1.00	0.75	3
E003	1.00	1.00	1.00	3
E007	1.00	0.25	0.40	4
E100	0.00	0.00	0.00	3
E101	0.00	0.00	0.00	4
E102	0.00	0.00	0.00	3
E103	0.00	0.00	0.00	0
E104	0.00	0.00	0.00	3
E105	0.44	1.00	0.62	4
accuracy			0.41	27
macro avg	0.34	0.36	0.31	27
weighted avg	0.39	0.41	0.34	27

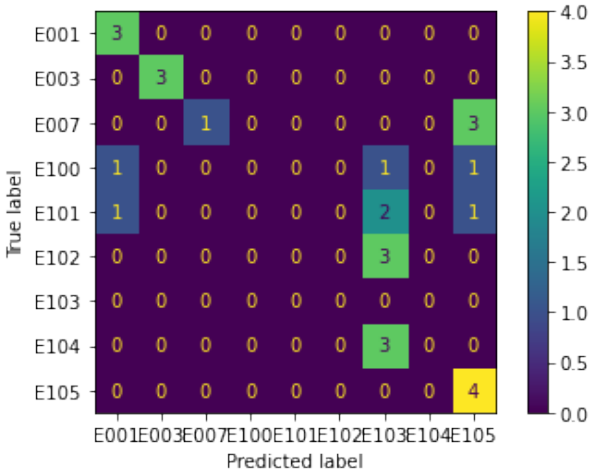
Fonte: Autoria própria (2022).

Assim como no caso do classificador K-NN o classificador SVM se comporta da mesma forma. Analisando a matriz de confusão na Figura 9 é possível verificar claramente que o resultado tende para o erro. Onde a diagonal principal possui poucos valores.

5.2 Experimento II

Para o Experimento II, foi considerado apenas um bloco do campus. Foi realizado a coleta contendo 5 pontos de medição por célula, distribuídas entre as extremidades e ponto central, exemplificado pela Figura 12. Obteve-se um total de 45 registros no BD. Seguindo os mesmos passos do Experimento anterior.

Figura 11 – Matriz de Confusão referente ao Experimento I com Classificador SVM.



Fonte: Autoria própria (2022).

Figura 12 – Cenário de coleta de pontos centrais para o Experimento II: (a) Andar Térreo, (b) Primeiro Andar

(a) Andar Térreo **(b) Primeiro Andar**

Fonte: Autoria própria (2022).

5.2.1 Resultados Experimento II

Segue os resultados obtidos para o cenário e parâmetros descrito no experimento, tanto para o classificador K-NN quanto para o classificador SVM:

5.2.1.1 K-NN

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, a métrica "*braycurtis*"com o valor de *K* vizinhos igual a 7. Obtendo-se uma acurácia de 74%, conforme apresentados na Figura 13.

Figura 13 – Resultados do Experimento II com Classificador K-NN.

	precision	recall	f1-score	support
E001	1.00	1.00	1.00	3
E003	1.00	1.00	1.00	3
E007	1.00	1.00	1.00	4
E100	0.67	0.67	0.67	3
E101	0.50	0.75	0.60	4
E102	0.00	0.00	0.00	3
E103	0.00	0.00	0.00	0
E104	1.00	0.33	0.50	3
E105	1.00	1.00	1.00	4
accuracy			0.74	27
macro avg	0.69	0.64	0.64	27
weighted avg	0.78	0.74	0.74	27

Fonte: Autoria própria (2022).

Nota-se que a acurácia relativamente alta, parte desse resultado se da pelo fato do algorítimo conseguir aplicar o *CrossValidation* de maneira a dividir a base de treino em 5 aplicando o classificador em cada parcela, obtendo uma acurácia para cada parte. Ao fim é realizado uma média para obter a acurácia final. A quantidade de pontos por célula também acaba influenciando este resultado pois abrange mais as variações dos RSSIs dentro das regiões da célula.

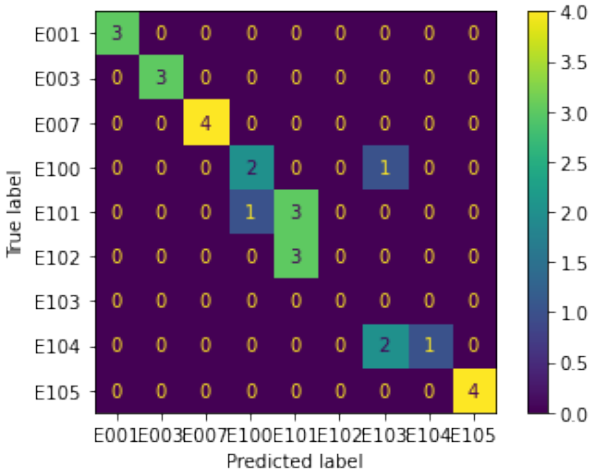
Analisando a matriz de confusão na Figura 14 nota-se claramente uma maioria de valores na diagonal principal, e poucos erros próximos ao valores corretos. Dentre os erros, em sua grande maioria foi realizada a predição para a célula imediatamente ao lado da célula real. Apenas um dos pontos apresentou um erro muito distante.

5.2.1.2 SVM

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, o Kernel "*RBF*", com o valor de custo *C* igual a 1 e o valor de Gamma igual a 0,02. Obtendo-se uma acurácia de 70%, conforme apresentados na Figura 15.

Como parte do resultado apresentado se da pelo fato do algorítimo também conseguir aplicar o *CrossValidation*, assim como no classificador anterior. A maior quantidade de pontos por célula aumenta a representatividade da classe resultando em uma melhor acurácia, pois abrange mais as variações dos RSSIs dentro das regiões da célula.

Figura 14 – Matriz de Confusão referente ao Experimento II com Classificador K-NN.



Fonte: Autoria própria (2022).

Figura 15 – Resultados do Experimento II com Classificador SVM

	precision	recall	f1-score	support
E001	1.00	0.67	0.80	3
E003	1.00	1.00	1.00	3
E007	0.80	1.00	0.89	4
E100	0.50	0.67	0.57	3
E101	0.60	0.75	0.67	4
E102	0.50	0.33	0.40	3
E103	0.00	0.00	0.00	0
E104	1.00	0.33	0.50	3
E105	1.00	0.75	0.86	4
accuracy			0.70	27
macro avg	0.71	0.61	0.63	27
weighted avg	0.80	0.70	0.72	27

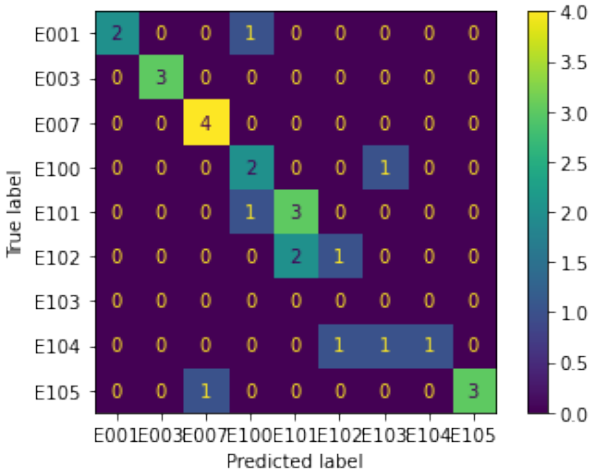
Fonte: Autoria própria (2022).

Analisando a matriz de confusão na Figura 16 nota-se uma matriz mais distribuída. Mesmo com sua taxa de acerto, representada pela diagonal principal, existem muitos erros dispersos. Alguns dos pontos apresentam sim alguma relação física, como salas imediatamente ao lado ou até mesmo acima, entretanto, erros distantes estão mais presentes.

5.3 Experimento III

Para o Experimento III, foi considerado apenas um bloco do campus. Foi realizado a coleta contendo 6 pontos de medição por célula, distribuídas entre as extremidades e pontos centrais, exemplificado pela Figura 17. Obteve-se um total de 54 registros no BD.

Figura 16 – Matriz de Confusão referente ao Experimento II com Classificador SVM



Fonte: Autoria própria (2022).

Figura 17 – Cenário de coleta de pontos centrais para o Experimento III: (a) Andar Térreo, (b) Primeiro Andar

(a) Andar Térreo **(b) Primeiro Andar**

Fonte: Autoria própria (2022).

5.3.1 Resultados Experimento III

Segue os resultados obtidos para o cenário e parâmetros descrito no experimento, tanto para o classificador K-NN quanto para o classificador SVM:

5.3.1.1 K-NN

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, a métrica "*braycurtis*"com o valor de *K* vizinhos igual a 7. Obtendo-se uma acurácia de 78%, conforme apresentados na Figura 18.

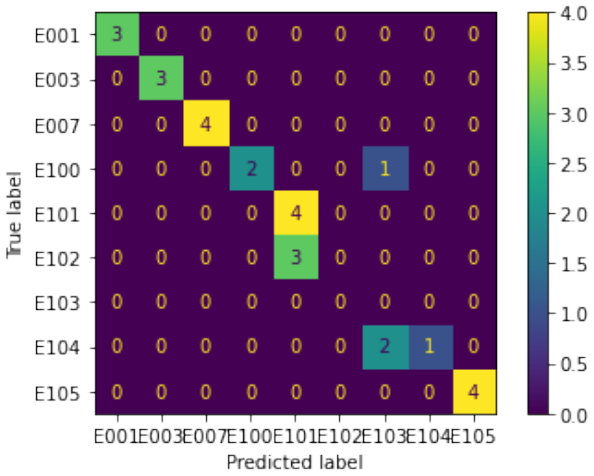
Figura 18 – Resultados do Experimento III com Classificador K-NN

	precision	recall	f1-score	support
E001	1.00	1.00	1.00	3
E003	1.00	1.00	1.00	3
E007	1.00	1.00	1.00	4
E100	1.00	0.67	0.80	3
E101	0.57	1.00	0.73	4
E102	0.00	0.00	0.00	3
E103	0.00	0.00	0.00	0
E104	1.00	0.33	0.50	3
E105	1.00	1.00	1.00	4
accuracy			0.78	27
macro avg	0.73	0.67	0.67	27
weighted avg	0.83	0.78	0.77	27

Fonte: Autoria própria (2022).

Análogo ao Experimento II, a acurácia aumentou principalmente pelo aumento da representatividade das classes na base de treino, além da aplicação do *CrossValidation*. Analisando a matriz de confusão na Figura 19 nota-se claramente que dentre todos os erros apresentados apenas um é discrepante, os demais são de células que fazem fronteira uma com a outra ou que estão imediatamente próximas.

Figura 19 – Matriz de Confusão referente ao Experimento III.



Fonte: Autoria própria (2022).

5.3.1.2 SVM

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, o Kernel "*RBF*", com o valor de custo *C* igual a 10 e o valor de Gamma igual a 0,02. Obtendo-se uma acurácia de 74%, conforme apresentados na Figura 20.

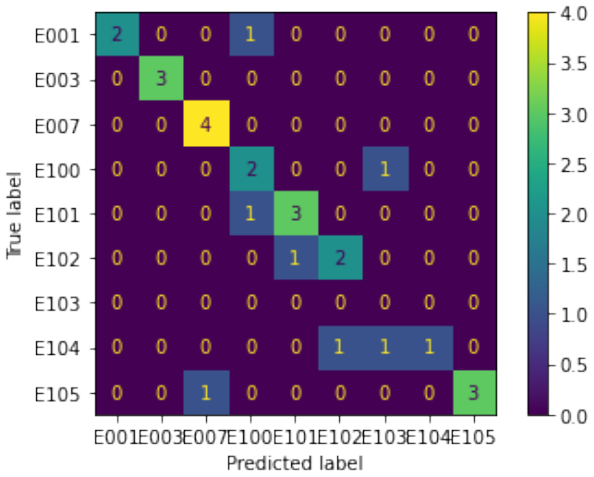
Figura 20 – Resultados do Experimento III com Classificador SVM

	precision	recall	f1-score	support
E001	1.00	0.67	0.80	3
E003	1.00	1.00	1.00	3
E007	0.80	1.00	0.89	4
E100	0.50	0.67	0.57	3
E101	0.75	0.75	0.75	4
E102	0.67	0.67	0.67	3
E103	0.00	0.00	0.00	0
E104	1.00	0.33	0.50	3
E105	1.00	0.75	0.86	4
accuracy			0.74	27
macro avg	0.75	0.65	0.67	27
weighted avg	0.84	0.74	0.76	27

Fonte: Autoria própria (2022).

Asiim como no classificador anterior os ganhos também para o SVM. Analisando a matriz de confusão na Figura 21 apresenta uma quantidade dispersa de erros, nos quais não se repetem. Tais erros são apresentados para células vizinhas, tanto lado a lado quanto abaixo, além de células distantes, mais que o classificador anterior.

Figura 21 – Matriz de Confusão referente ao Experimento III com Classificador SVM.



Fonte: Autoria própria (2022).

5.4 Experimento IV

Para o Experimento IV, foi considerado apenas um bloco do campus. Foi realizado a coletas a cada um metro e meio, variando de 14 a 36 pontos de medição por célula. Obteve-se um total de 205 registros no BD.

Como teste, foi utilizado uma base contendo 49 medições distribuídas entre as células, garantindo que tanto pontos das extremidades quanto centro das mesmas estivessem contidas no teste. Utilizando-se dos identificadores obtidos no pré-processamento da base de treino foi possível ordenar e atribuir valores zerados para os identificadores que não foram identificados na medição. Neste pré-processamento dos dados, foram também descartado RSSIs não identificados na base de treino, garantindo assim que todas as colunas e suas ordenações estejam idênticas, tanto no treino quanto no teste.

5.4.1 Resultados Experimento IV

Segue os resultados obtidos para o cenário e parâmetros descrito no experimento, tanto para o classificador K-NN quanto para o classificador SVM:

5.4.1.1 K-NN

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, a métrica "*manhattan*"com o valor de *K* vizinhos igual a 3. Obtendo-se uma acurácia de 71%, conforme apresentados na Figura 22.

Figura 22 – Resultados do Experimento IV com Classificador K-NN.

	precision	recall	f1-score	support
E001	1.00	0.33	0.50	6
E003	0.67	1.00	0.80	6
E007	0.86	1.00	0.92	6
E100	1.00	0.67	0.80	6
E101	0.00	0.00	0.00	6
E102	0.42	0.83	0.56	6
E103	0.00	0.00	0.00	1
E104	0.75	1.00	0.86	6
E105	1.00	1.00	1.00	6
accuracy			0.71	49
macro avg	0.63	0.65	0.60	49
weighted avg	0.70	0.71	0.67	49

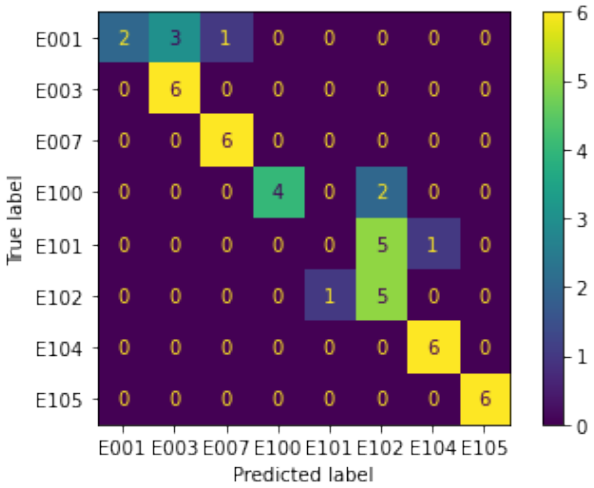
Fonte: Autoria própria (2022).

Nota-se que a acurácia relativamente alta, parte desse resultado se da pelo fato do algorítimo conseguir aplicar o *CrossValidation* de maneira a dividir a base de treino em 5 aplicando o classificador em cada parcela, obtendo uma acurácia para cada parte. Ao fim é realizado uma

média para obter a acurácia final. A quantidade de pontos por célula também acaba influenciando este resultado pois abrange maior representatividade das classes, uma vez que coleta maiores variações dos RSSIs dentro das regiões da célula.

Analisando a matriz de confusão na Figura 23 nota-se claramente uma maioria de valores na diagonal principal, e poucos erros próximos ao valores corretos.

Figura 23 – Matriz de Confusão referente ao Experimento IV com Classificador K-NN



Fonte: Autoria própria (2022).

5.4.1.2 SVM

Por meio dos hiperparâmetros, a função *GridSearch*, através da busca exaustiva, retornou como melhores parâmetros, para este caso, o Kernel "*linear*", com o valor de custo *C* igual a 1 e o valor de Gamma igual a 0,002. Obtendo-se uma acurácia de 82%, conforme apresentados na Figura 24.

Figura 24 – Resultados do Experimento IV com Classificador SVM

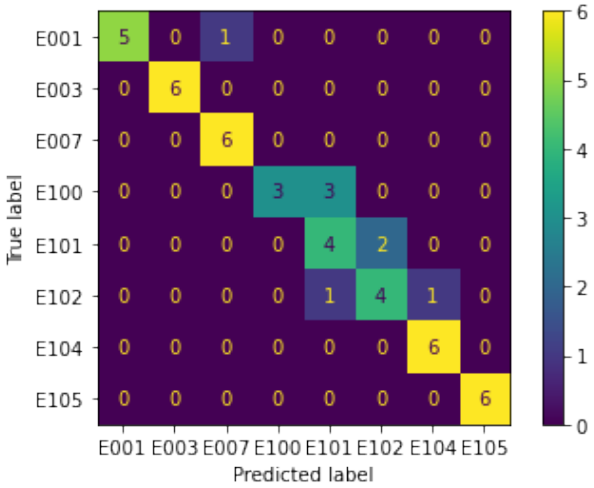
	precision	recall	f1-score	support
E001	1.00	0.83	0.91	6
E003	1.00	1.00	1.00	6
E007	0.86	1.00	0.92	6
E100	1.00	0.50	0.67	6
E101	0.50	0.67	0.57	6
E102	0.57	0.67	0.62	6
E103	0.00	0.00	0.00	1
E104	0.86	1.00	0.92	6
E105	1.00	1.00	1.00	6
accuracy			0.82	49
macro avg	0.75	0.74	0.73	49
weighted avg	0.83	0.82	0.81	49

Fonte: Autoria própria (2022).

Neste caso a acurácia foi alta, pois a grande quantidade de pontos por célula teve um peso grande no momento de utilizar o classificador. Como o SVM trabalha bem com bases maiores apresentou uma boa acurácia em relação aos demais. Parte desse resultado se da, também, pela aplicação do *CrossValidation*.

Analisando a matriz de confusão na Figura 25 nota-se claramente uma maioria de valores na diagonal principal. Os poucos erros que o modelo apresenta estão muito próximos aos corretos, em sua maioria células fisicamente ao lado da correta.

Figura 25 – Matriz de Confusão referente ao Experimento IV com Classificador SVM



Fonte: Autoria própria (2022).

5.5 Resultados Gerais

A Figura 26, representa de forma simultânea os resultados dos quatro experimentos realizados.

Pode-se notar que o Experimento I, que contem apenas um ponto de mapeamento por célula, obteve os piores resultados, tanto para os classificador K-NN quanto para o classificador SVM. Este ultimo com uma melhora de 10% em sua acurácia, mas, ainda assim um péssimo resultado. Tais acurácias são resultados da não aplicação da validação cruzada, pois a base de treino deve conter pelo menos dois membros em cada classe (uma classe equivale a uma célula), mas, o fato de conter apenas um elemento por classe torna o modelo preditivo falho.

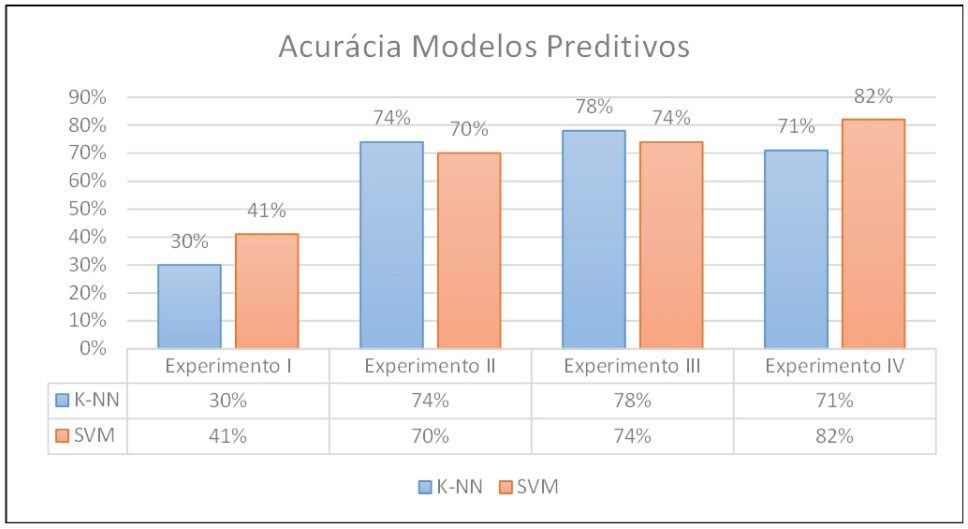
O Experimento II apresentou uma melhora significativa se comparado ao Experimento I. A principal diferença é a maior representatividade de pontos por célula, resultando em um maior numero de membros por classe, possibilitando também, uma melhor aplicação da Validação Cruzada. A coleta de, além do ponto central, pontos das extremidades de cada célula apresentaram uma melhora significativa na acurácia. Esta rasão justifica a melhora em ambos os classificadores de forma considerável.

O Experimento III, apresenta dois pontos centrais e nas extremidades de cada célula, o que acabada melhorando o peso dos valores para o interior da célula no momento da predição. Por conseguinte, apresentando uma melhora ainda mais significativa em relação ao Experimento II.

Nestes três primeiros experimentos um ponto de erro aparece sempre inalterado, E100-E103. Este ponto para não ser por nenhum caso dos experimentos anteriores, provavelmente se trata de uma anomalia no momento da coleta do ponto, no processo de mapeamento.

Por fim, o Experimento IV com o classificador K-NN apresentou uma piora em relação ao experimento anterior. Parte do resultado se da pelo aumento da quantidade de pontos. Já SVM consegue lidar melhor com bases mais complexas e maiores de dados, conseguindo encontrar um resultado mais eficiente de acurácia que os experimentos anteriores.

Figura 26 – Resultados Gerais dos Experimentos



Fonte: Autoria própria (2022).

Para confirmar a corretude do modelo criado, foi pego os dois maiores resultados e aplicado novos pontos de uma única célula a fim de identificar se alguma região dentro da célula possui tende para outra célula.

5.5.1 Experimento III - K-NN

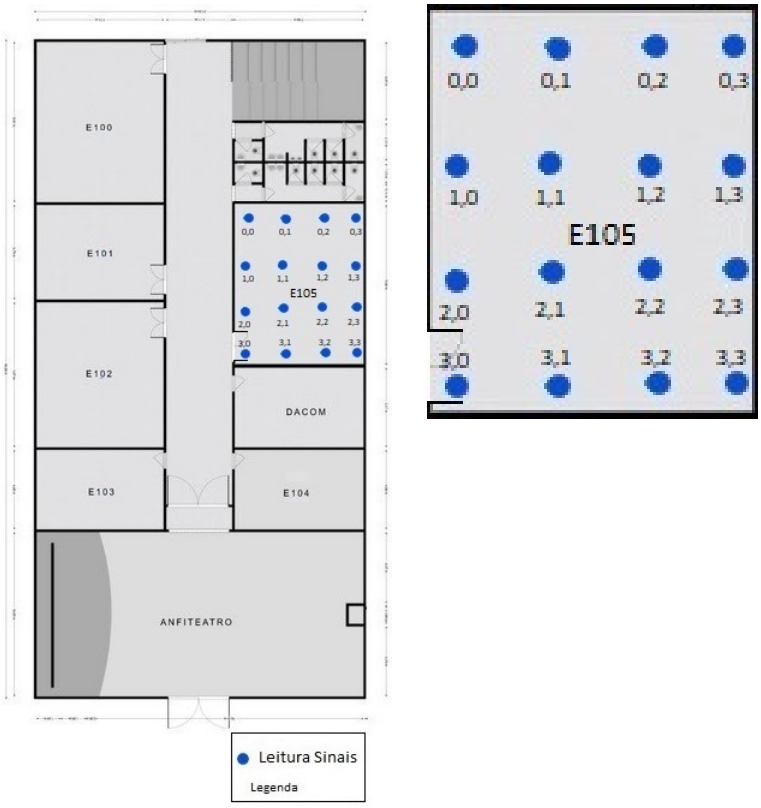
Com uma acurácia de 78% foi utilizado o modelo preditivo para identificar 16 pontos dentro de uma única célula, E105. Os pontos dentro da célula foram distribuídas formando uma matriz 4x4, conforme a Figura 27.

Após aplicar o modelo preditivo produzido pelo experimento, foi possível gerar a matriz de predições correspondente para cada ponto respectivo da célula, juntamente com o valor do nível de certeza para o ponto, conforme a Figura 28.

Pode-se notar que na prática o modelo preditivo é eficiente e que os erros apresentados são de alguma célula próxima à célula estudada, conforme a taxa de acurácia e tabela verdade

Figura 27 – Coletas de Pontos Célula E105 para Verificação de acuratividade.

(a) Primeiro Andar **(b) Sala E105**



Fonte: Autoria própria (2022).

Figura 28 – Matriz de predição e grau de certeza nos pontos da célula E105, respectivamente.

(E105-0.85714286) (E105-0.85714286) (E105-0.71428571) (E105-0.57142857)
(E105-0.71428571) (E105-0.85714286) (E105-0.71428571) *(E104-0.42857143)*
(E105-0.71428571) *(E104-0.57142857)* **(E105-0.85714286) (E105-0.85714286)**
(E105-0.85714286) (E105-0.85714286) (E105-0.71428571) (E105-0.71428571)

Fonte: Autoria própria (2022).

apresentada na criação do modelo preditivo. A célula E105 esta praticamente no centro do bloco, inclusive entre os APs, obtendo uma distância muito parecida entre um e outro. Neste caso, ocorreram dois pontos de erro, um ponto na região central da célula e um pontos na lateral externa da célula. Ambos os casos apontando para a célula imediatamente mais próxima (E104). O ponto de erro central (2,1 na matriz) está muito próximo dos pontos de mapeamentos centrais da célula, o que não justifica tal valor de grau de certeza a não ser que o mapeamento do ponto foi mal realizado.

A tabela Tabela 1, apresenta os graus de certeza para cada ponto no momento da predição. É possível notar que os níveis de certeza para as previsões corretas estão em uma faixa muito próximas, respeitando um padrão de certeza. Nota-se também que um dos pontos que foi retornado como E104 (ponto 1,3 da matriz) se trata de um empate com a classe correta, entretanto o classificador acabou atribuindo o ponto errado como o oficial. Neste ponto de empate é

nítido a variação dos valores de certeza se comparado com os demais, mas fisicamente, neste ponto, não tem nada que justifique tais valores, o mais provável é que também se trata de uma má coleta.

Tabela 1 – Grau de certeza para cada ponto da célula E105 para o classificador K-NN

$Ponto$	$E001$	$E003$	$E007$	$E100$	$E101$	$E102$	$E103$	$E104$	$E105$
0,0	0.	0.	0.	0.	0.	0.	0.14285714	0.	**0.85714286**
0,1	0.	0.	0.	0.	0.	0.	0.	0.14285714	**0.85714286**
0,2	0.	0.	0.	0.	0.	0.	0.	0.28571429	**0.71428571**
0,3	0.	0.	0.	0.	0.	0.	0.	0.42857143	**0.57142857**
1,0	0.	0.	0.	0.	0.14285714	0.	0.	0.14285714	**0.71428571**
1,1	0.	0.	0.	0.	0.	0.	0.	0.14285714	**0.85714286**
1,2	0.	0.	0.	0.	0.	0.	0.	0.28571429	**0.71428571**
1,3	0.	0.	0.14285714	0.	0.	0.	0.	**0.42857143**	**0.42857143**
2,0	0.	0.	0.	0.	0.	0.	0.	0.28571429	**0.71428571**
2,1	0.	0.	0.	0.	0.14285714	0.	0.	**0.57142857**	0.28571429
2,2	0.	0.	0.	0.	0.	0.	0.	0.14285714	**0.85714286**
2,3	0.	0.	0.	0.	0.	0.	0.	0.14285714	**0.85714286**
3,0	0.	0.	0.	0.	0.	0.	0.	0.14285714	**0.85714286**
3,1	0.	0.	0.	0.	0.	0.	0.	0.14285714	**0.85714286**
3,2	0.	0.	0.	0.	0.	0.	0.	0.28571429	**0.71428571**
3,3	0.	0.	0.	0.	0.	0.	0.	0.28571429	**0.71428571**

Fonte: Autoria própria (2022).

5.5.2 Experimento IV - SVM

Com uma acurácia de 82%, também foi utilizada os mesmos 16 pontos apresentados na Figura 27, entretanto dessa vez utilizando o modelo preditivo resultante do Experimento IV.

Após aplicar o modelo preditivo produzido pelo experimento, foi possível gerar a matriz de predições correspondente para cada ponto respectivo da célula, juntamente com o valor do nível de certeza para o ponto, conforme a Figura 29.

Figura 29 – Matriz de predição e grau de certeza nos pontos da célula E105, respectivamente.

(E105-0.65092879) (E105-0.72611912) (E105-0.74160971) (E105-0.6501304)
(E105-0.40093543) (E105-0.6104176) (E105-0.88883634) (E105-0.58048687)
(E105-0.51015578) *(E104-0.43319435)* **(E105-0.66472346) (E105-0.79842323)**
(E105-0.65328124) (E105-0.81279049) (E105-0.45447458) (E105-0.66936638)

Fonte: Autoria própria (2022).

Com base nos valores de apresentados na aplicação do modelo preditivo pode-se notar que os valores de certezas são mais bem distribuídos, isso se dá pelo funcionamento do algoritmo SVM, onde todas as classes acabam por receber um valor de certeza e predomina a classe com maior valor. O erro apresentado aponta para um ponto central, tal ponto pode se tra-

tar de um ruído ou até mesmo uma medição mal realizada pois em ambos os testes apresentou erro.

Com base na Tabela 2 é possível verificar que existe uma distribuição mais uniforme do grau de certeza entre todas as classes, tal comportamento se da pelo funcionamento do algoritmo do classificador. Como no momento do mapeamento das células, neste caso, possui muitos pontos distribuídos por distancias iguais ponto a ponto por célula, os graus de certezas acabam ficando mais distribuídos também, não respeitando uma faixa padrão como na Tabela 2 com o outro classificador. Tal fato não diminui a acurácia do classificador e sua eficiência na prática.

Tabela 2 – Grau de certeza para cada ponto da célula E105 para o classificador SVM

$Ponto$	$E001$	$E003$	$E007$	$E100$	$E101$	$E102$	$E104$	$E105$
0,0	0.00728836	0.00811321	0.01966385	0.03421783	0.10850072	0.12317614	0.0481111	**0.65092879**
0,1	0.00930981	0.007192	0.02463183	0.0262107	0.04694901	0.10580078	0.05378674	**0.72611912**
0,2	0.00667867	0.0063634	0.02598589	0.0338579	0.02177134	0.04844254	0.11529056	**0.74160971**
0,3	0.00716018	0.00529881	0.02956558	0.03139076	0.02824167	0.05142028	0.19679232	**0.6501304**
1,0	0.01374959	0.00672204	0.0269942	0.05509298	0.05657901	0.05487063	0.38505611	**0.40093543**
1,1	0.01035172	0.0071297	0.02640318	0.01941525	0.04974861	0.09851333	0.17802062	**0.6104176**
1,2	0.00310737	0.00214147	0.00754521	0.01729139	0.01401263	0.0228124	0.04425319	**0.88883634**
1,3	0.00611719	0.00610128	0.05157933	0.02977529	0.01573758	0.02510466	0.2850978	**0.58048687**
2,0	0.00619022	0.00588453	0.03632048	0.03312164	0.03211977	0.05203218	0.32417539	**0.51015578**
2,1	0.00835028	0.00972085	0.02173407	0.26322139	0.03678494	0.10572556	**0.43319435**	0.12126855
2,2	0.00791133	0.00809676	0.03813691	0.01746802	0.0409184	0.08550455	0.13724058	**0.66472346**
2,3	0.00596072	0.00539644	0.02879157	0.02227231	0.01911829	0.03690149	0.08313594	**0.79842323**
3,0	0.00715387	0.00551375	0.0194457	0.01918604	0.03458977	0.07944952	0.18138012	**0.65328124**
3,1	0.00411052	0.00320001	0.01138786	0.01136703	0.01844107	0.04345975	0.09524326	**0.81279049**
3,2	0.00758095	0.00585253	0.03329578	0.02526894	0.05348036	0.05738464	0.36266221	**0.45447458**
3,3	0.00456749	0.00424853	0.03373826	0.01909935	0.02890322	0.01676836	0.22330841	**0.66936638**

Fonte: Autoria própria (2022).

5.6 Riscos e Limitações

Esta pesquisa, se tratando de uma pesquisa base, apresenta alguns pontos que devem ser levados em consideração. Primeiramente os mapeamentos foram realizados em um único bloco do Campus, o que pode não representar um comportamento fiel em todos os demais blocos. Na pesquisa foi considerado que o receptor móvel sempre estivesse em uma das células mapeadas, obrigando assim a sempre retornar uma célula mais provável não abrindo brecha para o caso do receptor móvel não estar em nenhuma célula mapeada. No campus existem mais espaços abertos do que possíveis células para serem mapeadas.

Todos os experimentos foram realizados por meio de um notebook simples, com a mesma placa de rede. Placas diferentes podem trazer valores diferentes de leituras. No momento do mapeamento é necessário aguardar o tempo de atualização da placa de rede, para

que a leitura seja a mais fiel possível. Entretanto, mesmo se realizar duas ou mais medições exatamente no mesmo ponto físico, com o mesmo receptor móvel, os sinais não serão iguais.

As leituras foram realizadas em dias com a temperatura amena, por volta dos 22ºC com clima seco e com o Campus vazio. Em dias com variação de temperatura e umidade, ou contendo muitas pessoas, poderá existir também muita interferência de rádio e obstáculos para os sinais, o que prejudica sua eficiência da propagação dos sinais.

6 CONCLUSÃO

Neste trabalho foram desenvolvidos técnicas de coleta de dados e avaliado sistemas de classificação automática de regiões/células com base na força do sinal Wi-Fi dentro da UTFPR-CM. Utilizando o método de coleta de dados com base na impressão digital das forças dos sinais Wi-Fi, com o benefício de não necessitar de *hardware* adicional, foi verificado várias formas de coletar os dados em cada célula. Os experimentos II, III e IV apresentaram resultados com taxas de acurácia aproximadas por meio dos classificadores clássicos K-NN e SVM.

O método de coleta de pontos para o mapeamento de cada célula, com distância de aproximadamente 1,5m entre um ponto e outro, juntamente com a utilização do classificador SVM apresentou a melhor acurácia entre os métodos experimentados, obtendo uma acurácia de 82%.

Embora os resultados obtidos não sejam os melhores, a aplicação deste tema de pesquisa no cenário da UTFPR-CM é inédita, mostrando que o estudo tem potencial e que é possível obter resultados satisfatórios mesmo com classificadores clássicos. Além disto, servem como motivação para exploração de outras técnicas de processamento de sinais e aprendizagem de máquina que permitam uma maior acurácia e maior automatização no momento do mapeamento das células.

Como sugestão para trabalhos futuros, seria interessante melhorar a etapa de mapeamento dos dados, de forma que torne um processo mais automatizado. Uma coleta automática de pontos juntamente com um classificador. A utilização de combinações de classificadores voltados a aprendizado de máquina, não somente os clássicos, poderia trazer acurácias melhores para a pesquisa.Uma forma de manter a base de mapeamento atualizada de forma dinâmica, alterando também o modelo preditivo seria uma solução para utilização a médio e longo prazo. São alguns pontos interessantes para futuros estudos e continuidade da pesquisa.

REFERÊNCIAS

BAHL, P.; PADMANABHAN, V. Radar: an in-building rf-based user location and tracking system. **Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)**, p. 775–784, 2000. Disponível em: http://ieeexplore.ieee.org/document/832252/.

CHENG, Y.-C. *et al.* Accuracy characterization for metropolitan-scale wi-fi localization. **Proceedings of the 3rd international conference on Mobile systems, applications, and services - MobiSys '05**, p. 233–245, 2005. Disponível em: https://www.usenix.org/conference/mobisys2005/accuracy-characterization-metropolitan-scale-wi-fi-localization.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 12, p. 1573–0565, 09 1995. ISSN 1573-0565. Disponível em: https://doi.org/10.1007/BF00994018.

CRISTIANINI, N.; RICCI, E. Support vector machines. *In*: ____. **Encyclopedia of Algorithms**. Boston, MA: Springer US, 2008. p. 928–932. ISBN 978-0-387-30162-4. Disponível em: https://doi.org/10.1007/978-0-387-30162-4_415.

FUKUNAGA, K.; NARENDRA, P. A branch and bound algorithm for computing k-nearest neighbors. **IEEE Transactions on Computers**, C-24, n. 7, p. 750–753, 1975.

GUIDARA, A. *et al.* Impacts of temperature and humidity variations on rssi in indoor wireless sensor networks. **Procedia Computer Science**, v. 126, p. 1072–1081, 2018. ISSN 1877-0509. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia. Disponível em: https://www.sciencedirect.com/science/article/pii/S187705091831322X.

HAN, D. *et al.* Access point localization using local signal strength gradient. **Lecture Notes in Computer Science**, p. 99–108, 2009. Disponível em: https://link.springer.com/chapter/10.1007/978-3-642-00975-4_10.

KJæRGAARD, M. B. **Indoor Positioning with Radio Location Fingerprinting**. 2018. Tese (Doutorado) — Faculty of Science of the University of Aarhus, 2018.

NICULESCU, D.; NATH, B. Vor base stations for indoor 802.11 positioning. *In*: . [*S.l.*: *s.n.*], 2004. p. 58–69.

PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

SADOWSKI, S.; SPACHOS, P. Rssi-based indoor localization with the internet of things. **IEEE Access**, v. 6, p. 30149–30161, 2018.

VAPNIK, V. N. **The nature of statistical learning theory**. [*S.l.*]: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

YOUSSEF, M. Indoor localization. *In*: ____. **Encyclopedia of GIS**. Boston, MA: Springer US, 2008. p. 547–552. ISBN 978-0-387-35973-1. Disponível em: https://doi.org/10.1007/978-0-387-35973-1_622.

ZEGEYE, W. *et al.* Wifi rss fingerprinting indoor localization for mobile devices. *In*: . [*S.l.*: *s.n.*], 2016. p. 1–6.

ZHAO, F. *et al.* An rssi gradient-based ap localization algorithm. **China Communications**, v. 11, n. 2, p. 100–108, 2014. Disponível em: http://ieeexplore.ieee.org/document/6821742/.

ZHU, X.; FENG, Y. Rssi-based algorithm for indoor localization. **Communications and Network**, v. 05, n. 02, p. 37–42, 2013.