

A SOLUTION FOR INDOOR LOCATION USING WIFI FINGERPRINTS AND MACHINE LEARNING

José Kellison De Almeida Silva - UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO - Orcid:
<https://orcid.org/0000-0002-6652-8241>

Alixandre Santana - UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO - Orcid:
<https://orcid.org/0000-0003-1910-3360>

Luis Filipe Pereira - UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO - Orcid:
<https://orcid.org/0000-0002-4624-6714>

• Develop component for managing the positions of items located within an environment; • Build the database of predetermined positions with the network signals at each of these points; • Check the applicability of machine learning techniques; • Implement a mobile app Check applicability of machine learning techniques to Implement a mobile app for indoor location. Design Science Research The algorithm with the highest level of acceptance was Adaboost, with an average error of less than 1 meter and with the vast majority of tests resulting in a tolerable margin of error (up to 2 meters) for the context of the solution Statistical tests showed that the Machine Learning application in indoor location context, wifi fingerprints database construction approach A system for the the retail and logistics management of products indoor.

Keywords: position, Machine Learning, Wifi fingerprints, location, indoor

SISTEMA DE GESTÃO PARA LOCALIZAÇÃO INDOOR UTILIZANDO WIFI FINGERPRINTS E MACHINE LEARNING

• Desenvolver componente para gestão das posições de itens localizados no interior de um ambiente; • Construir a base de dados de posições pré-determinadas com os sinais de rede em cada um desses ponto; • Verificar a aplicabilidade de técnicas de aprendizagem de máquina; • Implementar um aplicativo móvel Verificar a aplicabilidade de técnicas de aprendizagem de máquina para Implementar um aplicativo móvel para localização indoor. Design science Research Dito isto, o algoritmo com maior nível de aceitação foi o Adaboost, apresentando um erro médio de menos de 1 metro e com a grande maioria dos testes resultando em uma margem tolerável de erro (até 2 metros) para o contexto da solução Testes estatísticos mostraram que o resultado Aplicação de Machine Learning no contexto de indoor location, abordagem de construção de base de dados wifi fingerprints Para a gestão de varejos e logística.

Palavras-chave: localização, Aprendizagem de máquina, Wifi fingerprints, indoor, varejo

A SOLUTION FOR INDOOR LOCATION USING WIFI FINGERPRINTS AND MACHINE LEARNING

SISTEMA DE GESTÃO PARA LOCALIZAÇÃO INDOOR UTILIZANDO WIFI FINGERPRINTS E MACHINE LEARNING

Author 1

Author 2

Author 3

ABSTRACT

Software solutions that are dependent on the Global Positioning System (GPS) signal may not display good accuracy in indoor spaces (malls, airports, commercial complexes, etc.), as the signal variations are likely to make it unreliable. This work aims to implement an indoor location management solution using signals from local wireless networks to predict the position of a user. To do so, the solution uses WiFi fingerprints, captured by mobile devices with Android operating system, to build bases of instances representing real positions of a user collected in one of the Federal Rural University of Pernambuco - Academic Unit of Garanhuns (UFRPE -UAG) buildings. The mobile application determines the user's position and a product location management web tool allows the insertion of user points of interest on the indoor map. To complete the process, the mobile application plots a vector from the predicted position of the user to the position of a specific target object, thus composing a complete and functional solution of indoor routes. The algorithms used to predict the position of the user were random forest, multilayer perceptron and Adaboost. The best indoor localization result was obtained with Adaboost, presenting an average error just under 1 meter from the actual position and 98.64% of the results were under an acceptable margin of error (up to 2 meters).

Keywords:

Indoor positioning. Machine Learning. Wifi fingerprints

RESUMO

Soluções de software que são dependentes do sinal do Sistema de Posicionamento Global (GPS) podem não apresentar uma boa precisão em espaços fechados ou indoor (shoppings, aeroportos, complexos comerciais etc.), pois as variações do sinal são capazes de inviabilizar seu uso. Este trabalho tem como objetivo implementar uma solução de gestão de localização indoor utilizando sinais de redes sem fio locais para prever a posição de um usuário. Para tanto, a solução utiliza WiFi fingerprints captados por aparelhos celulares com sistema operacional Android, para construir bases de instâncias representando posições reais de um usuário coletadas em um dos prédios da Universidade Federal Rural de Pernambuco - Unidade Acadêmica de Garanhuns. O aplicativo móvel determina a posição do usuário e uma ferramenta web de gestão de localidades de produtos permite a

inserção de pontos de interesse do usuário no mapa indoor. Para completar o processo, o aplicativo móvel realiza a plotagem de um vetor da posição predita do usuário até a posição de um objeto específico de destino compondo assim uma solução completa e funcional de rotas indoor. Os algoritmos utilizados para predição da posição do usuário foram o random forest, multi layer perceptron e Adaboost. Ao final dos experimentos, o melhor resultado de localização indoor foi obtido com o Adaboost, apresentando um erro médio de pouco menos de 1 metro em relação à posição real e 98,64% dos resultados se apresentando dentro de uma margem de erro aceitável (até 2 metros).

Palavras-Chave: Posicionamento *indoor*. Aprendizagem de máquina. *Wifi fingerprints*

1. INTRODUÇÃO

O GPS (Sistema de Posicionamento Global, em português) é um meio de navegação baseado em satélites (EL-RABBANY, 2006), um dos mais usados para a geolocalização de objetos e pessoas. Apesar do seu alcance ser excelente, ele apresenta problemas para realizar o posicionamento com uma precisão aceitável dentro de espaços fechados, uma vez que a sua utilização requer line of sight (LOS, ou linha de visão) dos satélites (SHALA; RODRIGUEZ, 2011), como é o caso em shoppings, aeroportos, complexos comerciais etc. Outros métodos são usados em conjunto para tentar sanar a deficiência do sinal GPS, utilizando diversos tipos de sinais existentes localmente em um ambiente e em dispositivos portáteis, como por exemplo: Bluetooth, WiFi, dados de giroscópio, sinais emitidos por torres de rádio etc.

Existe uma gama de métodos que visam tentar predizer a posição de um dispositivo com base em sinais de rede sem fio, como por exemplo: detecção de proximidade (FARID et al., 2013), técnica baseada em ângulo (AOA, do inglês Angle Of Arrival) (COSTA, 2014), técnica do menor polígono (COSTA, 2014), fingerprinting (ALTINTAS; SERIF, 2011; ZEGEYE et al., 2016) entre outros. Uma das soluções mais utilizadas na literatura é o uso de WiFi fingerprints, em outras palavras, o uso de sinais de redes WiFi recebidos por um dispositivo em uma determinada posição dentro de um espaço. Tais sinais são coletados em pontos de um espaço e armazenados em uma base de dados, durante uma primeira etapa para posteriormente serem usados com algoritmos de aprendizagem de máquina. Neste trabalho selecionamos alguns algoritmos para testes de acurácia, sendo eles: Random Forest, MLP (Multilayer Perceptron) e Adaboost.

O uso do sinais WiFi apresentam uma variância de predição entre 1 e 5 metros a depender da técnica utilizada, sendo que o uso de *fingerprints* é um dos com maiores níveis de acurácia dentro desse intervalo (FARID et al., 2013). Além da acurácia apresentada pela literatura, outro fator que colabora para sua adoção é que na maioria dos lugares já existe uma rede com dispositivos WiFi instalados e prontos para uso.

Para realizar a estimativa da posição atual de um dispositivo mobile podese adotar o uso de algoritmos de aprendizagem de máquina. Muitos trabalhos implementam algoritmos comumente usados para classificação (SHALA; RODRIGUEZ, 2011; FARID et al., 2013; ALTINTAS; SERIF, 2011), como o KNN e o K-Means. Entretanto, a análise do problema

mostra-se mais próxima de um problema de regressão, uma vez que valores numéricos reais tentam ser preditos, por isso buscou-se validar os resultados de algoritmos de regressão, como os algoritmos Adaboost e MLP que são utilizados para experimentação neste trabalho. Neste sentido, o objetivo geral deste trabalho é o *desenvolvimento de uma solução de geolocalização indoor baseada no uso de algoritmos de machine learning e WiFi fingerprints*. Os objetivos específicos estão listados abaixo:

- Desenvolver componente para gestão das posições de itens localizados no interior de um ambiente;
- Construir a base de dados de posições pré-determinadas com os sinais de rede em cada um desses pontos;
- Verificar a aplicabilidade de técnicas de aprendizagem de máquina;
- Implementar um aplicativo móvel para localização *indoor*.

Este trabalho está organizado na forma de cinco seções. A Seção 2 apresenta a fundamentação teórica dos conceitos utilizados durante o desenvolvimento do sistema completo, incluindo tanto a plataforma web quanto a móvel, bem como os trabalhos correlatos. A Seção 3 apresenta a caracterização da pesquisa e a metodologia utilizada, o detalhamento das tecnologias utilizadas para construção do sistema, os requisitos para funcionamento dos componentes web e *mobile*, além de explicar o procedimento utilizado para coleta de dados e traz também um diagrama explicando a conexão entre os componentes. A Seção 4 traz um detalhamento do resultado alcançado para cada um dos objetivos específicos que foram listados acima. Finalmente, a Seção 5 apresenta as conclusões e trabalhos futuros que podem ser realizados a partir deste projeto.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 TÉCNICAS DE GEOLOCALIZAÇÃO

2.1.1 Sistema de Posicionamento Global - GPS

O GPS está atuante desde o início da década de 1990, tendo atingido potencial total no ano de 1995, fornecendo informações precisas e contínuas de posição tridimensionais em escala global para qualquer aparelho compatível (KAPLAN; HEGARTY, 2005). Kaplan e Hegarty (2005) também explica que para ser possível obter a posição de um receptor na superfície do planeta, é necessário que ele esteja ao alcance do sinal de quatro satélites (no mínimo). Devido a este requerimento, o design da constelação de satélites foi construído para que seis satélites sempre estejam visíveis por um usuário. A posição obtida é calculada através do método de trilateração (SHALA; RODRIGUEZ, 2011). Apesar desse meio de geolocalização possuir uma precisão bastante aceitável em ambiente abertos (*outdoor*), ele possui limitações que dificultam o uso confiável dentro de ambientes *indoor*. Sendo assim, o uso para sistemas de geolocalização *indoor* podem não apresentar resultados aceitáveis se utilizassem o GPS única como base de funcionamento.

2.1.2 Baseadas em Redes sem Fio

Sistemas baseados em redes sem fio utilizam os sinais de rádios provindos de aparelhos

transmissores presentes em um ambiente para estimar a posição de um aparelho que recebe esses sinais (COSTA, 2014; ZEGEYE et al., 2016). Como apresentado por Altintas e Serif (2011), os algoritmos disponíveis para geolocalização indoor com uso de redes sem fio podem ser divididos em dois grupos gerais: (a) através de modelos matemáticos, como por exemplo, trilateração, que busca determinar a distância ou ângulo de três transmissores em relação ao receptor de sinais; ou, (b) através do mapeamento de locais, com o armazenamento de sinais em posições pré- definidas para uso posterior por algoritmos, como é o caso do uso do objeto de estudo principal deste trabalho, *WiFi fingerprints*.

Aparelhos que são usados para a transmissão de sinais podem ser dos mais diversos, dado os tipos de rede sem fio que existem, como por exemplo: WiFi utiliza-se de APs (do inglês access points ou pontos de acesso) que são roteadores, switches ou mesmo outros dispositivos móveis; ou Bluetooth, que utiliza beacons. Os sinais emitidos por esses transmissores são medidos através do indicador de intensidade de sinal recebido (Received Signal Strength Indicator ou RSSI (BENKIC et al., 2008; SHALA; RODRIGUEZ, 2011).

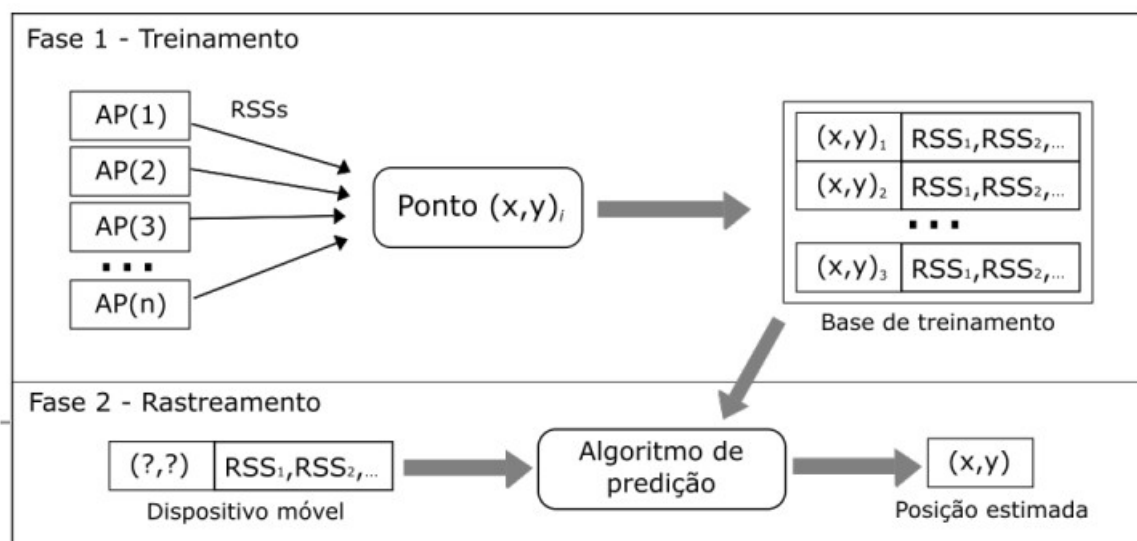
Fingerprinting ou Mapa de Sinais

Esta é uma técnica que utiliza um vetor RSSI de múltiplos transmissores para formar um **mapa de sinais** (ALTINTAS; SERIF, 2011; ZEGEYE et al., 2016). *Fingerprints* funcionam como uma assinatura de um local, onde um conjunto de intensidades de sinais de redes são alcançáveis (COSTA, 2014). Cada indicador é associado ao identificador do aparelho que o transmite, o BSSID (*Basic Service Set Identifier*), sendo este o seu endereço do MAC. Os indicadores emitidos por dispositivos WiFi são medidos em dB (decibéis), variando entre 0 e -100, sendo que quanto maior o valor melhor a qualidade do sinal o que significa uma maior aproximação com do dispositivo mobile (receptor) com o dispositivo transmissor.

A Figura 1 apresenta o funcionamento do método que utiliza WiFi fingerprints, ou seja, os sinais emitidos por dispositivos de redes WiFi. A Fase 1 do processo se dá pela **fase de treinamento** (também chamada de fase offline ou calibração), na qual, dado o mapa de um espaço, os RSSIs oriundos de APs são coletados em pontos pré-definidos (no formato de coordenadas [x,y]) e então armazenados em uma base para posterior treinamento. Situações que geram mudanças no ambiente (troca mudança de posição de access points, por exemplo) requerem que um novo treinamento seja realizado.

A etapa seguinte é a **fase de rastreamento** (*online*), nela os RSSIs detectados pelo dispositivo do usuário são coletados e usados como entrada para um algoritmo de aprendizagem de máquina, de modo que não há o conhecimento prévio da sua posição. O algoritmo usa então os dados coletados na primeira fase do processo para treinamento e os dados coletados na segunda fase, para gerar como saída valores estimados das coordenadas x e y que representam a posição atual do aparelho móvel.

Figura 1 - Fluxograma das fases que compõem o método de localização usando WiFi *fingerprints*
Fonte: Adaptado de Altintas e Serif (2011)



2.2 APRENDIZAGEM DE MÁQUINA

Aprendizagem de Máquina (*Machine Learning* ou ML) é um campo da ciência da computação que visa estudar métodos com os quais máquinas possam ser capazes de aprender baseando-se na observação de dados existentes, extraíndo padrões e realizando predições, de modo a tentar se assemelhar a inteligência humana (RÄTSCH, 2004; NAQA; MURPHY, 2015). Abaixo seguem algumas definições de termos usados neste trabalho (REZENDE, 2003):

Na literatura, existem vários algoritmos que são utilizados nesse campo de pesquisa, os quais podem ser divididos em duas categorias distintas:

- **Supervisionados:** os conjuntos de dados de treino utilizados por estes algoritmos contêm uma classe para as instâncias, de modo que este é o resultado esperado quando um novo exemplo com classe desconhecida é processado pelo algoritmo (NAQA; MURPHY, 2015). Nesta categoria podemos citar os seguintes algoritmos que são descritos nessa pesquisa: *Random Forest*, MLP (*Multilayer Perceptron*) e Adaboost.

- **Não supervisionados:** diferente dos métodos supervisionados, os dados utilizados aqui não possuem rótulos. Os métodos que se encaixam nesta categoria são utilizados para identificar padrões ou anomalias através da identificação de *clusters*, por exemplo (NAQA; MURPHY, 2015)

Como mencionado nos objetivos deste trabalho, a solução proposta funciona com base em geolocalização de objetos, portanto os algoritmos utilizados aqui tem o objetivo de identificar sua coordenada dentro de um espaço. Sendo assim, os esforços dos algoritmos de aprendizagem de máquina resultarão em dados numéricos. Neste caso, os métodos demonstrados são **algoritmos supervisionados de regressão** como MLPs, AdaBoost, Random Forest, Árvores de Decisão.

2.3 TRABALHOS CORRELATOS

Shala e Rodriguez (2011) apresentam uma avaliação de uma solução para geolocalização *indoor* utilizando WiFi *fingerprints* (assim como neste trabalho) em conjunto com sensores embutidos em dispositivos móveis com o sistema operacional Android, como é o caso do acelerômetro, giroscópio e outros. Os autores do artigo citado utilizaram um algoritmo de KNN (do inglês *k-nearest neighbour* ou *k* - vizinhos mais próximos) para estimar a posição do dispositivo móvel do usuário. Como meio de tentar contornar as variações de sinais que resultam em outliers, os autores utilizaram pesos atribuídos a outras técnicas implementados na aplicação: detector de movimento, contador de passos, bússola, entre outros. A técnica de fusão conseguiu chegar a um erro médio de menos de 2m entre a posição real e a estimada. Altintas e Serif (2011) utiliza o método baseado no *K-Means Clustering* utilizando o RSSI de transmissores de redes sem fio, comparando seus resultados com o KNN comum. Pelos experimentos realizados, com diferentes variações de argumentos, o *k-Means* mostrou resultados que variam de 2,68m à 4,11m. Enquanto que Altintas e Serif (2011) utilizam técnicas de classificação e clusterização.

Zegeye et al. (2016) realizaram experimentos utilizando apenas WiFi *fingerprints* em um corredor de um prédio. Para os experimentos, foram utilizados 10 APs e 93 pontos de localização para treinamento no local. Realizaram dois tipos de experimentos: um inicial utilizando todos os dados coletados e posteriormente, realizaram alguns melhoramentos em como os dados eram coletados e analisados. Para o primeiro experimento, obtiveram um distanciamento médio de 7,46m entre os pontos reais e estimados, sendo que em 67% dos casos a distância estava abaixo de 10m. O segundo experimento obteve uma distância de 3,46m e em 80% dos testes a distância estava abaixo dos 5m. Enquanto que o primeiro experimento se assemelhou ao usado aqui, eles não deixam claro se foi feita uma coleta com diferença temporal, e no segundo experimento realizaram mudanças na base que não foram feitas nesta pesquisa. Outra diferença se dá pelo fato de aqui ter sido usado algoritmos de aprendizagem de máquina, enquanto que eles buscaram resolver o problema configurando-o como um problema de otimização.

Gogolak et al. (2013), apresentam uma aproximação utilizando método de WiFi fingerprint juntamente com uma rede neural. Além disso, como meio de comparação, um algoritmo KNN com pesos (WKNN). A coleta de dados para treino da rede neural é feita em um espaço pré-definido com pontos colocados em uma coordenada (x, y) do ambiente. Gogolak et al. (2013), apresentam uma aproximação utilizando método de WiFi fingerprint juntamente com uma rede neural. Além disso, como meio de comparação, um algoritmo KNN com pesos (WKNN). A coleta de dados para treino da rede neural é feita em um espaço pré-definido com pontos colocados em uma coordenada (x, y) do ambiente.

3. ASPECTOS METODOLÓGICOS

3.1 CARACTERIZAÇÃO DA PESQUISA

Usou-se a taxonomia de Wohlin e Aurum (2015) para classificar este trabalho, tomando oito pontos (que podem ser reduzidos a sete) de decisão para sua classificação. Deste modo, esta pesquisa pode ser considerada de cunho aplicado, uma vez que busca fornecer uma solução para um problema específico (geolocalização indoor) com base em conhecimento extraído de práticas conhecidas (WiFi fingerprints e machine learning). A lógica da pesquisa é do tipo dedutiva, onde o desenvolvimento parte do objetivo mais geral para os mais específicos, confirmando a hipótese de que uma das técnicas escolhidas para testes atendem a um resultado satisfatório com base nos dados coletados. A pesquisa é de propósito avaliativo, uma vez que busca avaliar um método que pode solucionar o problema de geolocalização indoor.

Quanto à sua aproximação, esta pesquisa é de cunho quantitativo, uma vez que trabalha com métodos estatísticos e de aprendizagem de máquina. O método de pesquisa é classificado como design science research, tal que tem como objetivo a geração de um artefato que solucione um dado problema. Durante a construção desse estudo, utilizou do método de experimentação para as coleções de dados trabalhados, utilizando dados reais para testes a fim de chegar nos resultados desejados. Por fim, os resultados obtidos foram analisados e tidos como aceitáveis através de um método estatístico.

3.2 DESIGN SCIENCE RESEARCH

Como denotado na seção anterior, a metodologia utilizada aqui foi o Data Science Research (DSR). Na área de Sistemas de Informação, existem dois paradigmas que caracterizam os métodos de pesquisa (ALAN et al., 2004): o chamado de ciência comportamental (do inglês, behavioral science), que busca desenvolver e verificar teorias; e, ciência do design (do inglês, design science), cujo objetivo é o desenvolvimento de artefatos. No contexto da pesquisa feita aqui, o artefato gerado é uma implementação de um software.

O DSR é dividido em seis passos (PEFFERS et al., 2007), os quais são contextualizados nessa pesquisa na Tabela 1:

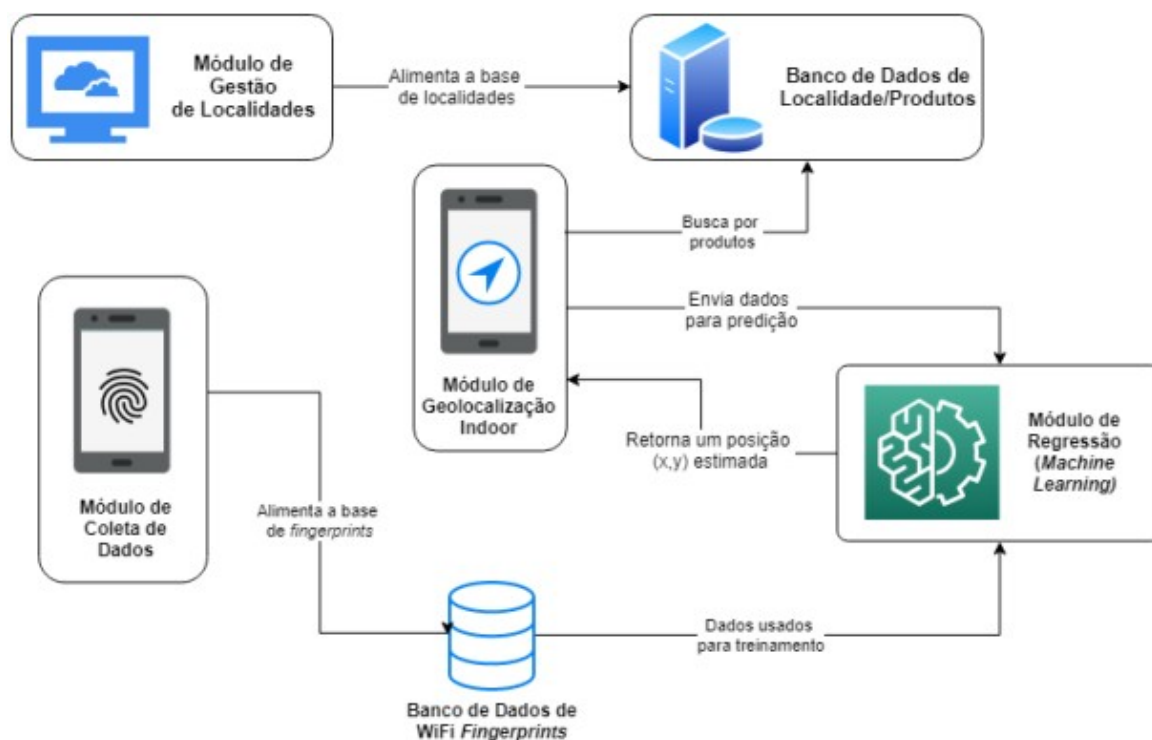
Tabela 1 – Passos que compõem o DSR

Passo	Contextualização
Identificação do problema e motivação	Geolocalização através do GPS apresenta ineficiência quando usado em locais fechados, por isso há a necessidade de uma outra aproximação
Objetivo	Desenvolver uma sistema para geolocalização de usuários em um ambiente e traçar direção entre um usuário e um ponto localizados no mesmo ambiente
Design e desenvolvimento	A solução foi construída com base em conceitos de aprendizagem de máquina e técnicas de localização
Demonstração	Foram realizados experimentos com o método que apresentou resultados no estado da arte
Avaliação	Uso de métodos estatísticos para demonstrar que o resultado obtido se encontra dentro do esperado
Comunicação	Relatório de desenvolvimento no formato de monografia

3.3 DIAGRAMA DE COMPONENTES DA SOLUÇÃO

O sistema em sua completude é composto por quatro componentes e duas bases de dados. A Figura 2 traz uma representação visual das conexões entre estas bases e componentes do sistema. A ferramenta de gestão de localidades é responsável por alimentar a base de dados com as localidades dos produtos. A base de localidades é acessada pelo aplicativo móvel (versão cliente ou usuário final) durante a pesquisa por produtos. O aplicativo móvel, usado pelo cliente, captura os sinais de redes WiFi ao seu alcance em um momento t e envia como entrada para o algoritmo de aprendizagem de máquina responsável por estimar sua posição. O algoritmo é alimentado também com a base de dados de WiFi fingerprints que foram coletados na fase de rastreamento (invisível para o usuário final). A fase de rastreamento é feita com o auxílio de um aplicativo móvel, que obtém e salva os dados das redes WiFi ao seu alcance.

Figura 2– Diagrama de conexão entre os componentes que integram o sistema
Fonte: O Autor.



O funcionamento do sistema como um todo começa pelo Módulo de Gestão de Localidades (LocalFácil), onde os mapas dos ambientes são construídos com base nas plantas baixas de uma filial cadastrada pelo sistema. Dentro de uma filial é possível trabalhar com um ou mais mapas. A construção de cada mapa é feita como uma espécie de decalque da planta, em que elementos gráficos são posicionados sobre a imagem original, indicando locais de interesse. Após a construção de um mapa de locais, o Módulo de Coleta de Dados (implementado na forma de um aplicativo móvel) poderá ser utilizado para a construção da base de dados de treinamento que será utilizada pelo Módulo de Regressão. Este, por sua vez realiza a execução do algoritmo de predição da posição atual

do usuário dentro de um ambiente, com base nos dados de sinais WiFi coletados pelo AcheFácil (nome dado ao Módulo de Geolocalização Indoor). O AcheFácil é um aplicativo móvel que realiza buscas por locais de interesse de acordo com termos informados pelo usuário. Os dados obtidos nas buscas, inclusive os mapas criados, são oriundos do banco de dados de localidades que é gerenciado pelo módulo de gestão. O detalhamento de cada um dos módulos é encontrado nas seções seguintes.

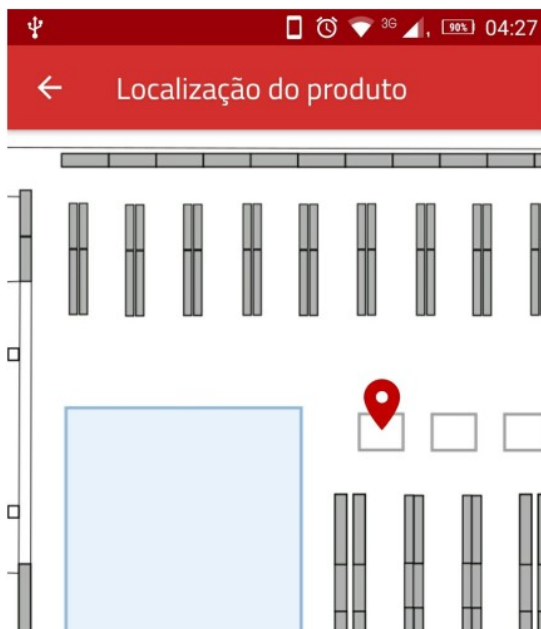
As tecnologias utilizadas na solução foram: JavaScript, a MEAN Stack (MongoDB5, Express, Angular e NodeJS), APIs REST(REpresentational State Transfer), Banco de dados NoSQL e o Sistema Operacional Android.

3.3 MÓDULO DE GESTÃO DE LOCALIDADES

O sistema trata do cadastro de localidades em um ambiente, através de uma ferramenta gráfica de construção de mapas, como será apresentado Na Seção 4. Neste contexto, uma localidade é um local ou ponto de interesse em que um objeto se encontra, a localidade é identificada por sua posição cartesiana. As localidades cadastradas são utilizadas como referencial de destino pelo aplicativo de geolocalização para traçar um vetor de direção entre elas e o usuário. A Figura 3 exemplifica a visualização (*pinpoint*, ou plotagem no mapa) de um local cadastrado, sendo indicado pelo ícone vermelho fixado no mapa.

Figura 3 – Exemplo do pinpoint realizado para localidade cadastrada e visualizada em um mapa do ambiente.

Fonte: o Autor



Com o intuito de garantir a portabilidade, e assim ser possível a sua utilização em qualquer sistema operacional, o LocalFácil (nome dado a este módulo) foi desenvolvido para ser

executado a partir de navegadores web, sendo estruturado em forma de uma aplicação SPA.

3.4 MÓDULO DE LOCALIZAÇÃO INDOOR

O aplicativo móvel desenvolvido no trabalho foi denominado AcheFácil. As informações obtidas através do app são oriundas de uma base de dados (alimentada pelo sistema de gestão, ver seção 3.3) que se encontra hospedada em servidores na nuvem, e obtidas através do uso de APIs REST. Após a obtenção da localização do objeto desejado, a aplicação captura os dados das redes sem fio ao seu redor para plotar um vetor de direção do usuário até o a posição do objeto, através da execução de um algoritmo de aprendizagem de máquina para predição da sua posição dentro do ambiente.

3.5 MÓDULO DE COLETA DE DADOS

A coleta de dados é uma etapa de suma importância para o funcionamento do sistema aqui descrito. Esta etapa equivale à fase de calibração (que trata dos métodos de geolocalização por sinais de redes sem fio), ou seja, essa é etapa em que dados serão coletados para alimentar a base de treinamento do algoritmo de aprendizagem de máquina.

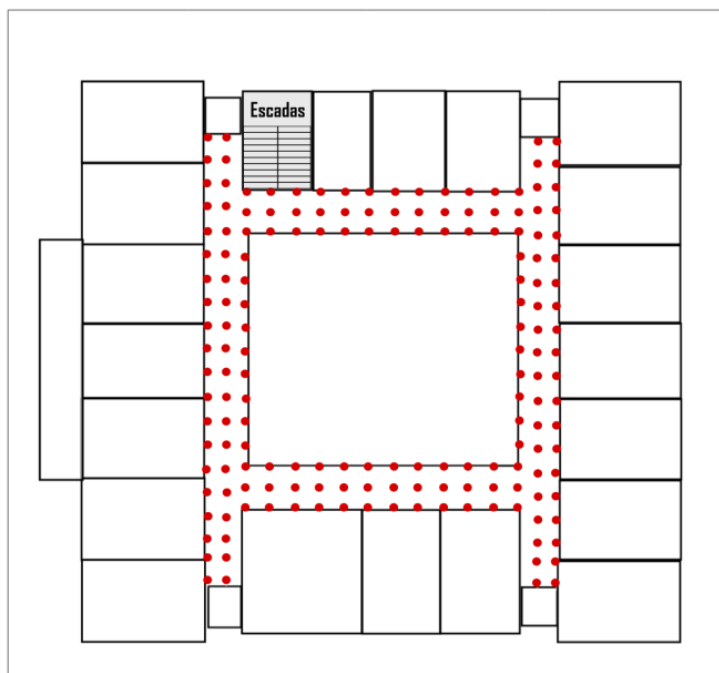
As informações necessárias foram obtidas através do mapeamento do primeiro andar do Prédio II dos professores da Universidade X. Com o objetivo de obter o maior número possível de padrões de modo que fosse possível capturar os sinais de WiFi com mais precisão, foi construído um mapa de pontos fixos a cada 1m (aproximadamente) nos corredores do andar. A Tabela 2 apresenta a equivalência das medidas em pixels para metros no mundo real, utilizando o mapa do andar utilizado como referência. A Figura 4 apresenta o layout de coleta de fingerprints por pontos pré-definidos, onde os pontos em vermelho indicam os locais em que é feita a obtenção de sinais em um dado momento.

Tabela 2 – Valores das medidas de distância e tamanhos utilizados nos experimentos (Fonte: o Autor)

Valor em pixels (px)	Valor em metros (m)
19,75	0,5
39,5	1
79	2

Foram construídas quatro bases de dados, sendo três delas compostas por dados obtidos em três dias consecutivos e uma foi construída através do cálculo da média dos valores dos RSSIs coletados anteriormente. As redes existentes em um ambiente podem sofrer alterações através do tempo ou por perda de sinal por quaisquer que sejam os motivos, sendo assim decidiu-se realizar a construção do mapa de sinais em múltiplos dias e após isso realizar a média desses dias para que fosse possível analisar qual tipo de base se mostraria mais precisa na fase de aprendizagem.

Figura 4 – Mapa de pontos pré-fixados do primeiro andar do prédio para a fase de calibração (coleta de fingerprints, ou fase offline):



As informações salvas nessas bases são instâncias compostas por valores numéricos, sendo eles: dois valores que representam a posição, x e y , de um ponto do plano cartesiano que representa o andar; e, uma quantidade n de valores de RSSI de todos os sinais de WiFi captados pelo aplicativo no momento da coleta, sendo associados ao BSSID (endereço MAC) do aparelho transmissor. Deste modo, na quarta base, a instância identificada pelo par (x_i, y_i) para o atributo MAC_p possui o valor obtido pelo cálculo $([MAC_{pDia1} + MAC_{pDia2} + MAC_{pDia3}]/3)$.

4. RESULTADOS

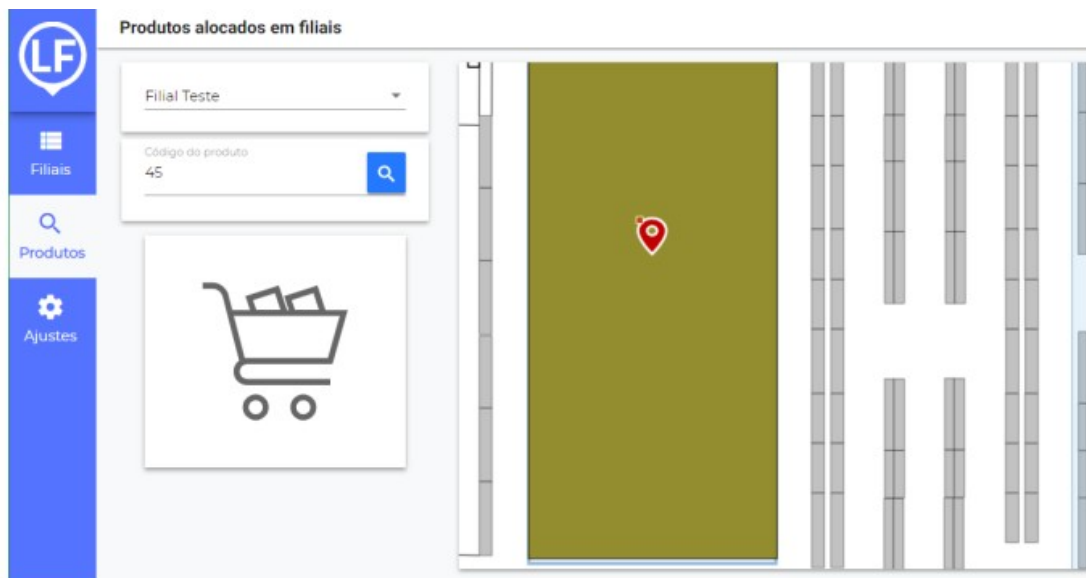
4.1 MÓDULOS DA SOLUÇÃO

- Módulo de gestão de Localizações

O sistema de gerenciamento de localizações recebeu o nome de LocalFácil, sendo ele um sistema web que tem como objetivo auxiliar o gerenciamento das localidades de produtos dentro de ambientes extensos, de modo que também serve como um fornecedor de dados para a aplicação móvel de geolocalização de usuários.

O usuário possui a opção de procurar por produtos alocados em uma filial, através do código do produto é possível verificar qual o andar em que o mesmo se encontra dentro da unidade, sendo exibido com um *pinpoint* no mapa do andar em que ele foi alocado. A tela de pesquisa e resultado é vista na Figura 5.

Figura 5 – Tela de pesquisa de produtos alocados em uma filial
Fonte: o Autor



- Módulo de Coleta de Posições Indoor

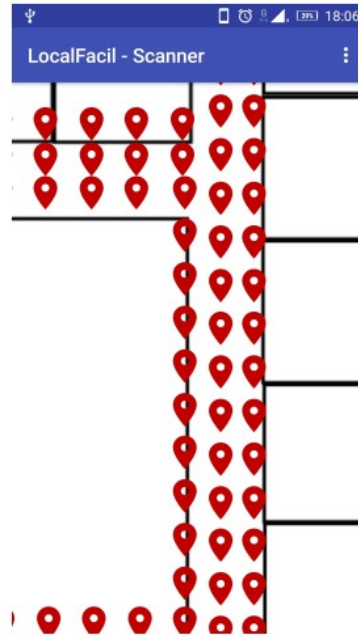
De acordo com o mapa de pontos construídos, para cada base foram coletados os fingerprints de 169 locais espalhados no ambiente. O conjunto de características da base de dados “Médias” é formado pela interseção das características das bases diárias (Dia 1, Dia 2 e Dia 3). O resumo das informações coletadas são descrito na Tabela 3.

Tabela 3 – Quantidade de características (BSSIDs) coletados em cada base
(Fonte: o Autor)

Base	Nº de Características
Dia 1	18
Dia 2	18
Dia 3	23
“Médias”	27

A Figura 6 mostra um *screenshot* da tela do aplicativo coletor de *fingerprints* (na fase de rastreamento).

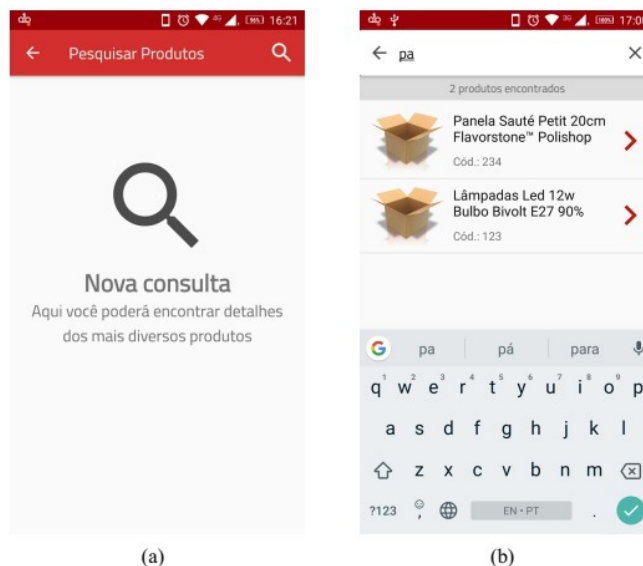
Figura 6 - Screenshot da tela do aplicativo coletor de *fingerprints*
Fonte: O Autor



- Módulo de Localização Indoor

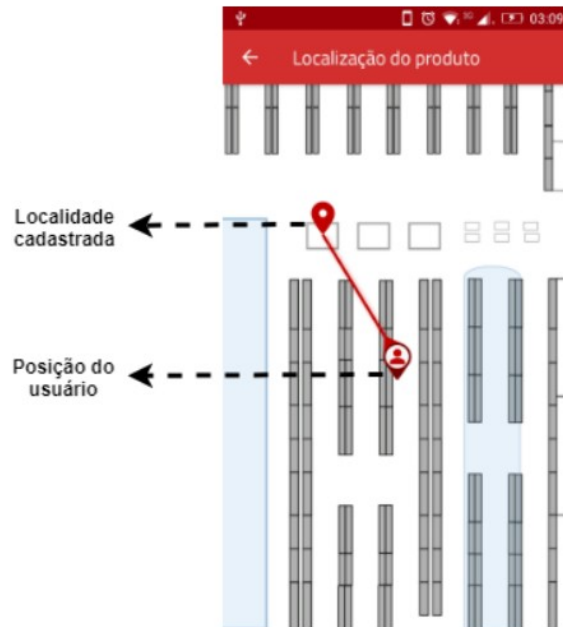
Além do LocalFácil, o usuário pode utilizar uma aplicação móvel desenvolvida para a plataforma Android, chamada de AcheFácil. Nessa aplicação é possível pesquisar por produtos existentes em uma filial. Pesquisando por sua descrição ou código, o usuário pode identificar o local exato do produto na filial. Na pesquisa, é exibida uma lista dos produtos que correspondem a sua pesquisa, a partir dessa lista é possível acessar o mapa do andar em que o produto está alocado, sendo indicado por um ícone de localização no local em que se encontra. O processo de pesquisa e exibição dos resultados pelo aplicativo móvel (módulo de geolocalização) podem ser vistos de pesquisa e resultado podem ser vistos na Figura 7.

Figura 7 – Aplicativo móvel: tela do processo de pesquisa no aplicativo móvel: (a) Tela de pesquisa de produtos no aplicativo móvel; (b) Tela de resultados de pesquisas de produtos



A Figura 8 mostra o vetor direcional entre a posição do usuário dentro do ambiente em relação ao ponto da localização do produto (outrora cadastro pelo módulo de gestão) selecionado na tela de resultados (Figura 7b).

Figura 8 – Aplicativo móvel: tela com o vetor direcional da posição do usuário até uma localidade cadastrada

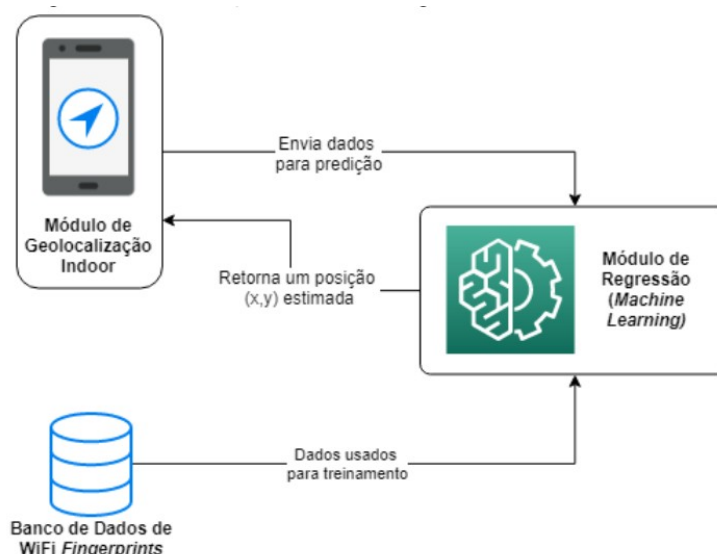


- Módulo de Regressão

As bases construídas durante a fase de calibração, possuem duas classes que identificam cada padrão usado para treinamento dos modelos primitivos, sendo elas: x e y . Utilizando os algoritmos *AdaBoost*, *Random Forest* e *MLP*, os modelos preditivos foram treinados e testados utilizando a técnica de validação cruzada *Leave One Out*. Esta técnica foi escolhida devido ao fato de que as bases usadas possuem poucos dados, e deste modo, compensou a falta de padrões em quantidade satisfatória para treinamento dos algoritmos. Os experimentos executados foram feitos com os valores das classes em pixels (px).

Para aceitação dos resultados obtidos, o valor da distância da entre o a posição real e posição estimada de um ponto (x, y) deve ser menor ou igual a 79 pixels, ou em uma escala do mundo real, aproximadamente 2 metros levando em consideração a imagem da planta baixa do primeiro andar do local de testes. O módulo tem acesso a base de dados da fase de calibração e executa duas vias de comunicação com o aplicativo de geolocalização, recebendo dados para predição de posições e enviando os resultados obtidos. A Figura 9 ilustra como este módulo se comunica com outros elementos que compõem o sistema.

Figura 9 – Comunicação do módulo de regressão com outros módulos



4.2 RESULTADOS DOS MÉTODOS DE MACHINE LEARNING

As Tabelas 4, 5, 6, e 7 mostram os resultados de predição para as classes obtidos com cada base de dados. Os valores apresentados nas tabelas são: MSE (média do erro quadrático, do inglês *mean square error*), MAE (média do erro absoluto, do inglês *mean absolute error*):

Tabela 4 – Resultado dos testes realizados com base de dados “Dia 1

Método	Classe X		Classe Y	
	MSE	MAE	MSE	MAE
AdaBoost	2350,922	29,007	2358,084	28,131
Random forest	6321,431	54,836	4436,222	42,302
Neural Network	9832,279	63,995	7154,644	55,822

Tabela 5 – Resultado dos testes realizados com base de dados “Dia 2”

Método	Classe X		Classe Y	
	MSE	MAE	MSE	MAE
AdaBoost	2218,797	26,131	771,139	18,590
Random forest	3894,990	41,448	3115,381	38,729
Neural Network	5019,841	46,306	4015,228	41,111

Tabela 6 – Resultado dos testes realizados com base de dados “Dia 3”

Método	Classe X		Classe Y	
	MSE	MAE	MSE	MAE
AdaBoost	1556,850	22,311	897,890	18,446
Random forest	3052,945	37,000	1802,758	27,335
Neural Network	2193,997	32,787	2888,768	34,863

Tabela 7 – Resultado dos testes realizados com base de dados “Médias”

Método	Classe X		Classe Y	
	MSE	MAE	MSE	MAE
AdaBoost	936,100	20,272	553,492	16,314
Random forest	2086,023	31,268	1356,877	23,970
Neural Network	2239,775	34,301	3519,694	43,539

Após a execução dos algoritmos de regressão, os resultados das classes foram analisados formando pontos em um plano cartesiano para comparação com a posição real do padrão testado. As Tabelas 8, 9, 10 e 11 apresentam os erros médios e os desvios padrões das distâncias entre os pontos real e predito de cada uma das bases. As tabelas citadas também mostram a quantidade de pontos com uma distância maior do que 2 metros (ou 79px).

Tabela 8–Resultado da comparação entre os pontos real e predito para os padrões da base “Dia 1”

Método	Erro médio (px)	Desvio padrão (px)	Erro real > 2m
AdaBoost	48,074	48.961	20/169
Random Forest	78,232	69,324	57/169
MLP	96,576	87,574	70/169

Tabela 9 –Resultado da comparação entre os pontos real e predito para os padrões da base “Dia 2”

Método	Erro médio (px)	Desvio padrão (px)	Erro real > 2m
AdaBoost	39,345	37,983	8/169
Random Forest	63,648	60,228	35/169
MLP	69,383	64,995	43/169

Tabela 10 –Resultado da comparação entre os pontos real e predito para os padrões da base “Dia 3”

Método	Erro médio (px)	Desvio padrão (px)	Erro real > 2m
AdaBoost	36,306	33,728	5/169
Random Forest	48,377	40,747	22/169
MLP	59,022	49,638	40/169

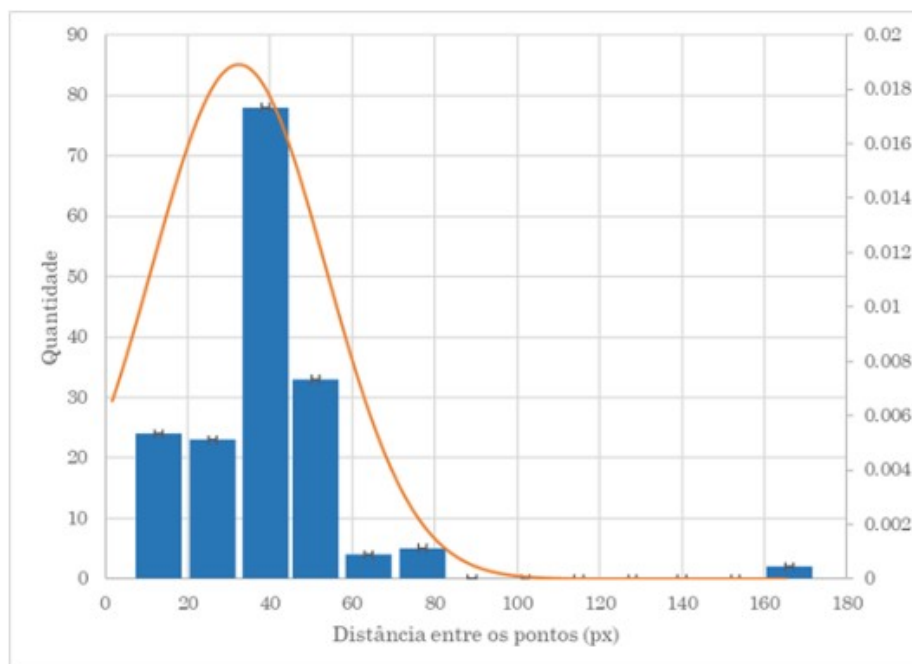
Tabela 11 – Resultado da comparação entre os pontos real e predito para os padrões da base “Médias”

Método	Erro médio (px)	Desvio padrão (px)	Erro real > 2m
AdaBoost	32,379	21,089	2/169
Random Forest	44,460	38,477	12/169
MLP	62,719	42,871	43/169

Com base nas informações contidas nas tabelas 4 - 11, pode-se constatar que o algoritmo de aprendizagem de máquina AdaBoost foi o que obteve melhores resultados em todos os casos de testes. Dentre as bases utilizadas, o algoritmo quando executado com a base “Médias” apresentou resultados extremamente significativos, com um erro médio de menos de 1 metro (menor que 39,5px) e um desvio padrão de pouco mais de meio metro (maior que 19.75px). Além disso, o resultado obtido com a base “Médias” apresentou apenas 2 pontos estimados acima do erro aceitável por este trabalho.

A Figura 10 mostra a distribuição dos erros de distância dos valores estimados da base “Médias”, deixando nítido que a maior parte dos pontos estimados se encontra abaixo do erro tolerado (79px).

Figura 10 - Distribuição dos erros de distância dos valores estimados da base “Médias”



Dada a distância tolerável de dois metros obteve-se um resultado ótimo com o Adaboost e a base de médias. com objetivo de verificar qual seria a quantidade de erros gerados quando esse limiar de tolerância é diminuído, verificou-se os resultados para um metro (a) e um metro e meio (b), sendo que:

(a) Foram obtidos 36 erros de predição, o que é considerado um resultado ruim, chegando a mais de 20% do número de exemplos de treinamento; e,

(b) Foram obtidos 9 erros de predição, marginalizando um resultado bom, visto que apenas representa apenas 5% da base de treinamento.

A Figura 11 mostra o mapa de pontos pré-definidos reais (quadrados) sobrepostos pelos pontos estimados (círculos) pelo AdaBoost na base de dados “Médias”, mostrando o quão próximo o modelo conseguiu chegar do cenário real, contando apenas com poucos outliers, diferente do resultado obtido pelo MLP, como visto na Figura 12. Os outliers visto na na figura (os três pontos posicionados no centro da área do mapa) estão destacados para indicar quais os pontos reais para os quais eles deveriam ser estimados, sendo que cada par (real e estimado) estão na mesma cor.

Figura 11 – Plotagem dos pontos estimados (com AdaBoost) em relação aos pontos reais utilizando a base “Médias”

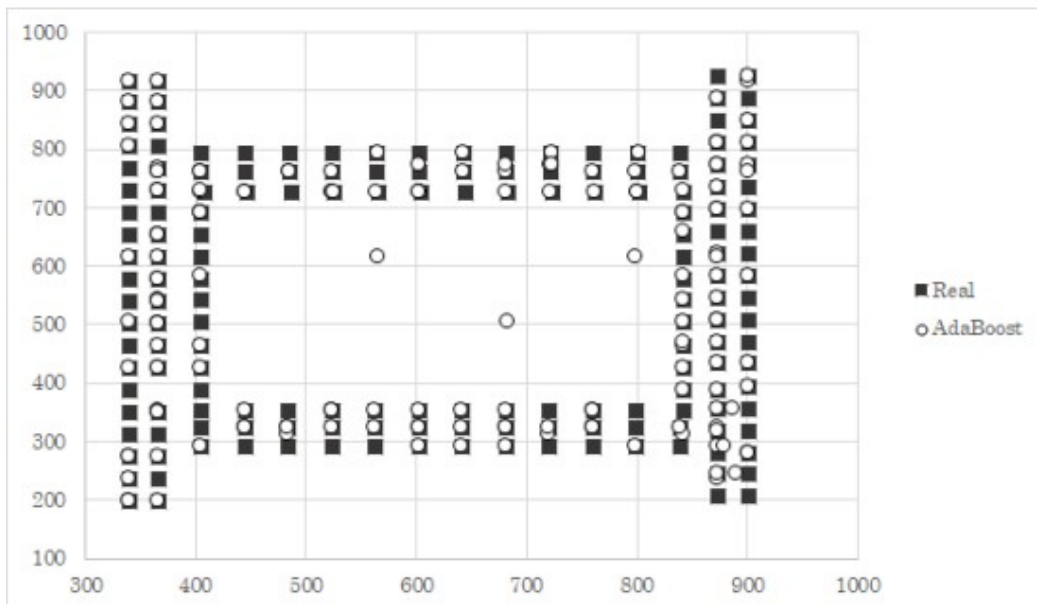
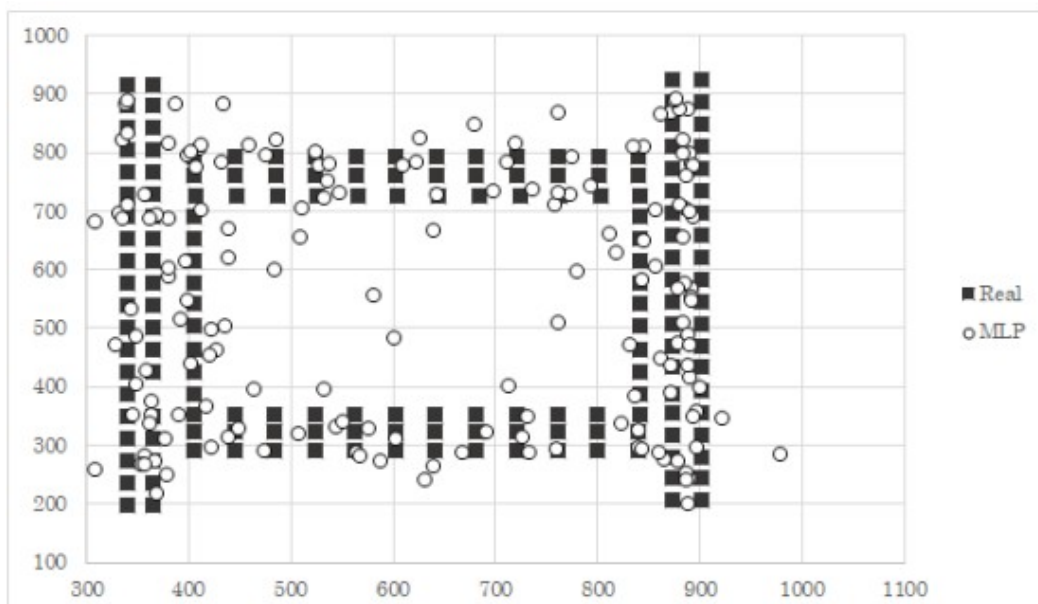


Figura 12 – Plotagem dos pontos estimados (com MLP) em relação aos pontos reais utilizando a base “Médias”



4.2.1 Testes de Hipótese

Como meio de validar os resultados obtidos nos experimentos, foi utilizado o Teste Z sobre os valores das distâncias entre os pontos reais e estimados para cada uma das bases usadas. Para obtenção de um resultado precisamos fixar duas hipóteses: a nula e alternativa. Consideramos como hipótese nula, a afirmação de que a média amostral se iguala ou está acima de 2 metros, enquanto que a hipótese alternativa (a que queremos provar) afirma que a média amostral é inferior a 2 metros. O teste foi executado de modo a considerar um intervalo de confiança de 95%. O cálculo do Teste Z é feito com a Equação 1:

$$Z = \frac{(\mu - \mu_0)}{\sigma} \quad \text{Eq 1.0}$$

Sendo, μ o valor da distância média máxima considerada aceitável (79px ou aproximadamente 2m) por este trabalho, μ_0 é a média da população amostral testada e σ indica o desvio padrão amostral. A hipótese nula será rejeitada quando $Z \geq z_\alpha$.

- Base “Médias”: apresentou como resultado o valor $Z = 2; 21$. Logo, rejeitamos a hipótese nula ao nível de 95% de confiança e consideramos como verdadeira a afirmação sobre a distância média ser aceitável para a solução do problema aplicado nesta base.
- Base “Dia 1”: apresentou como resultado o valor $Z = 0; 01$. Logo, a hipótese nula não é rejeitada e consideramos como falsa a afirmação sobre a distância média ser aceitável para a solução do problema aplicado nesta base.
- Base “Dia 2”: apresentou como resultado o valor $Z = 0; 25$. Logo, a hipótese nula não é rejeitada e consideramos como falsa a afirmação sobre a distância média ser aceitável para a solução do problema aplicado nesta base.
- Base “Dia 3”: apresentou como resultado o valor $Z = 0; 63$. Logo, a hipótese nula não é rejeitada e consideramos como falsa a afirmação sobre a distância média ser aceitável para a solução do problema aplicado nesta base.

Com base nos resultados estatísticos obtidos e apresentados anteriormente, podemos concluir que a base formada pela médias dos valores RSSI coletados em momentos distintos podem ser uma melhor opção para uso no modelo de regressão, em relação ao dados oriundos de uma base construída apenas com os valores de um único dia.

5. CONSIDERAÇÕES FINAIS

O presente trabalho dedicou-se a estudar métodos de regressão para solucionar o problema. A aplicação destes algoritmos possibilita estimar uma posição de acordo com dados já conhecidos do ambiente e com informações em tempo real do usuário. Dito isto, o algoritmo com maior nível de aceitação foi o Adaboost, apresentando um erro médio de menos de 1 metro e com a grande maioria dos testes resultando em uma margem tolerável de erro (até 2 metros) para o contexto da solução. Testes estatísticos mostraram que o resultado alcançado pelo Adaboost é aceitável, alcançando uma taxa de aceitação de 98,61%, com nível de confiança de 95% pelo teste de hipótese.

Além da ferramenta móvel desenvolvida, este trabalho gerou um componente para gestão da localização de objetos e ou locais de interesse para usuários em um ambiente, o LocalFácil. Esta ferramenta de gestão pode ser utilizada para diversos usos, como por exemplo, para: gestão de produtos alocados em uma empresa de varejo, identificação de obras em uma exposição de arte, além de aplicações para localização de estabelecimentos em shoppings ou aeroportos.

O uso do RSSI transmitido por redes WiFi, pode gerar distorções na predição dos resultados de acordo com os obstáculos existentes no ambiente. Além disso, em estruturas que possuem mais de um andar, é necessário realizar a identificação do mesmo de forma automática em uma aplicação móvel. Sensores nativos de dispositivos mobile podem ser usados para obter a altitude aproximada de um dispositivo, adicionando um meta dado para identificação do andar na base de treinamento e no momento de predição. Idealmente, as bases de treinamento deveriam possuir o máximo possível de instâncias para uma fase de calibração ter bons resultados. Contudo, neste trabalho o ambiente de testes não possuía uma extensão grande o suficiente, e devido a isto foram coletadas poucas instâncias, e mais tarde utilizou-se do Leave One Out para tentar contornar esse fato.

Possíveis trabalhos futuros incluem o aperfeiçoamento das fases de calibração e rastreamento. Apesar deste trabalho ter apresentado a prova de conceito com sucesso, a solução ainda precisa ser verificada em um ambiente com proporções consideráveis. A fase de calibração pode ser aperfeiçoada com o objetivo de torná-la mais ágil, uma vez que para ambientes muito extensos essa tarefa se torna cansativa e demanda muito tempo. Outro trabalho que pode ser realizado é o uso da fusão de tecnologias (como os sensores embutidos de dispositivos móveis) para melhorar os resultados alcançados com o Adaboost ou mesmo com outros algoritmos. Além destes, um possível caso de estudo seria a análise sobre o número de características da base de dados, buscando encontrar qual seria um tamanho ótimo para diminuição do erro obtido nos algoritmos utilizados. A forma de tratamento utilizada para os missing values nas bases também pode ser utilizada para estudo, buscando encontrar a melhor forma de tratá-los.

REFERÊNCIAS

ALAN, R. H. V. et al. Design science in information systems research. *MIS quarterly*, Springer, v. 28, n. 1, p. 75–105, 2004.

ALTINTAS, B.; SERIF, T. Improving RSS-based indoor positioning algorithm via k-means clustering. In: VDE. 17th European Wireless 2011-Sustainable Wireless Technologies. [S.l.], 2011. p. 1–5.

BENKIC, K. et al. Using RSSI value for distance estimation in wireless sensor networks based on ZigBee. In: IEEE. 2008 15th International Conference on Systems, Signals and Image Processing. [S.l.], 2008. p. 303–306.

COSTA, L. J. S. Técnica de localização em ambientes fechados utilizando padrões de redes sem fio. Dissertação (Mestrado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2014. doi:10.11606/D.3.2016.tde-13072016-161143.

EL-RABBANY, A. Introduction to gps: The global positioning system. Artech house, Londres, 01 2006.

FARID, Z.; NORDIN, R.; ISMAIL, M. Recent advances in wireless indoor localization techniques and system. Journal of Computer Networks and Communications, Hindawi, v. 2013, 2013.

GOGOLAK, L.; PLETL, S.; KUKOLJ, D. Neural network-based indoor localization in wsnenvironments. Acta Polytechnica Hungarica, v. 10, n. 6, p. 221–235, 2013.

GOOGLE NoSQL: as principais vantagens de um banco de dados não relacional. 2017. Disponível em: <<https://www.qinetwork.com.br/google-nosql-e-suas-vantagens/>>. Acesso em:31.01.2019.

KAPLAN, E.; HEGARTY, C. Understanding GPS: principles and applications. [S.l.]: Artech house, 2005.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: Machine Learning in Radiation Oncology. [S.l.]: Springer, 2015. p. 3–11.

PEFFERS, K. et al. A design science research methodology for information systems research. Journal of management information systems, Taylor & Francis, v. 24, n. 3, p. 45–77, 2007.

RÄTSCH, G. A brief introduction into machine learning. Friedrich Miescher Laboratory of the Max Planck Society, 2004.

REZENDE, S. O. Sistemas inteligentes: fundamentos e aplicações. [S.l.]: Editora Manole Ltda, 2003. 89-114 .

SHALA, U.; RODRIGUEZ, A. Indoor Positioning using Sensor-fusion in Android Devices. Dissertação (Mestrado) — School of Health and Society, Department Computer Science, Kristianstad University, Kristianstad, 01 2011.

WHAT IS REST – Learn to create timeless RESTful APIs. Disponível em: <<https://restfulapi.net/>>. Acesso em: 25.01.2019.

WOHLIN, C.; AURUM, A. Towards a decision-making structure for selecting a research design in empirical software engineering. Empirical Software Engineering, Springer, v. 20, n. 6, p. 1427–1455, 2015.

ZEGEYE, W. et al. Wifi RSS fingerprinting indoor localization for mobile devices. 2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), p. 1–6, 10 2016. doi:10.1109/UEMCON.2016.7777834.