

Sistema de Localização em Ambientes Internos usando Wi-Fi e algoritmo WKNN

Alípio de Sales Carvalho, José Ferreira de Rezende e Celso Barbosa Carvalho

Resumo— Nos últimos anos, surgiram novas técnicas de localização de objetos ou usuários em ambientes internos. Apesar disso, os sistemas de localização em ambientes internos (*Indoor Location Systems - ILSs*) continuam sendo um desafio uma vez que algumas propostas dependem de sensores com tecnologia específica ou de alto custo para obter acurácia adequada. Além do mais, muitas propostas de ILSs da literatura possuem limitações em seus mecanismos de Interação Homem-Máquina (IHM), dificultando operação e usabilidade. Neste artigo apresentamos proposta de ILS baseado no algoritmo de aprendizagem de máquina WKNN que é executado em plataforma Android e foi construído para permitir fácil operação e usabilidade dos usuários. O ILS proposto apresentou erro de localização RMSE de 2,87m, superando trabalhos recentes da literatura, onde o menor erro médio de localização RMSE obtido foi de 6,7283 m, além de nossa proposta disponibilizar facilidades de configuração IHM.

Palavras-Chave— ILS, WKNN, Machine-Learning, fingerprint Wi-Fi.

Abstract— In recent years, with the evolution of information and communication technologies, new techniques for locating objects or users in indoor environments have emerged. Despite this, indoor location systems (ILSs) continue to be a challenge since some proposals depend on sensors with specific technology and high cost to obtain adequate system accuracy. In addition, many existing ILS proposals have limitations in their Human-Machine Interaction (HMI), making operation and usability difficult. In this article we present an ILS proposal for indoor environments based on WKNN machine learning algorithm that runs under Android platform and was built to allow easy operation and usability for users. The proposed ILS showed a RMSE location error of 2.87 m, surpassing recent work in the literature, where the lowest RMSE location error obtained was 6.7283 m, in addition, our proposal provides HMI configuration facilities.

Keywords— ILS, WKNN, Machine-Learning, fingerprint, Wi-Fi.

I. INTRODUÇÃO

Nos últimos anos, com o surgimento e evolução da Internet das Coisas (*Internet of Things - IoT*) houve uma crescente demanda em pesquisas relacionadas com serviços baseados em localização (*Location Based Services - LBS*). Os LBSs podem ser divididos em externos e internos. Os LBSs externos possuem soluções que utilizam sistema *Global Positioning System* (GPS) para determinar a localização fixa e em tempo real dos dispositivos sem fio [1]. No entanto, os sistemas GPS não funcionam adequadamente em dias nublados, mesmo em ambientes externos e, também, não realizam adequadamente suas operações de localização em ambientes internos, devido à dificuldade dos sinais de satélites em

alcançar o interior de casas e prédios (ex. garagens de edifícios, quartos, banheiros, departamentos de lojas, seção de supermercados, etc...)[2]. Um fato interessante é que grande maioria de pessoas gastam grande parte de seu tempo em ambientes internos, tal como o de escolas, trabalho, shoppings, supermercados, hospitais e outros [3]. Apesar disso, existem muitos desafios a serem vencidos a fim de determinar com precisão a localização de objetos e/ou pessoas em ambientes internos [4].

Em pesquisas da literatura existem propostas que endereçam soluções para sistemas de localização em ambientes internos (*Indoor Location System - ILS*) baseado em tecnologias como RFID (Radio Frequency Identification), UWB (Ultra Wide Band) [5][6], infravermelho, Wi-Fi (Wireless Fidelity) [7], Bluetooth [8], NFC (Near Field Communication) [9][10], sinais magnéticos, processamento de imagens e sinais sonoros [11] e tecnologias combinadas [12]. Nestas pesquisas é possível destacar algumas questões a serem melhoradas em sistemas ILSs tais como a complexidade na instalação, configuração e utilização do sistema além do erro médio de localização que pode afetar a acurácia do sistema. Neste contexto, e a fim de melhorar estes pontos em aberto na literatura, este artigo propõe um sistema ILS que utiliza assinaturas de sinais de redes WiFi resumidas com funções características para compor um método de localização de objetos e/ou pessoas combinada com algoritmo de aprendizagem de máquina WKNN (Weighted K-Nearest Neighbor) que estima a localização de um objeto através da atribuição de pesos entre os atributos de uma instância de teste e dos K vizinhos mais próximos das instâncias de treino.

Uma vantagem do ILS proposto é que ele funciona em ambientes internos utilizando tecnologias amplamente utilizadas no dia-a-dia das pessoas, tais como os APs (*Access Points*) do ambiente, utilizados no sistema como pontos de referência, e dispositivos smartphones, para os quais desejamos determinar a localização. Para apresentar os assuntos mencionados, este artigo é organizado como a seguir: A Seção II apresenta a arquitetura e o funcionamento da proposta; A Seção III mostra resultados dos experimentos realizados com diferentes parâmetros e, além disso, discute resultados. A Seção IV apresenta conclusões.

II. PROPOSTA

A arquitetura da proposta pode ser visualizada na Figura 1. A arquitetura é composta de APs existentes no ambiente e utilizados como pontos de referência, um ou mais dispositivos “*smartphone*”, para os quais se deseja determinar a localização e, por fim um computador “*servidor*” do sistema.

Alípio de Sales Carvalho, Faculdade de Tecnologia, Universidade Federal do Amazonas (UFAM), Manaus-AM, Brasil, E-mail: alipiosc@gmail.com, ccarvalho@ufam.edu.br; José Ferreira de Rezende, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro - RJ, Brasil, E-mail: rezende@land.ufrj.br; Celso Barbosa Carvalho Faculdade de Tecnologia, Universidade Federal do Amazonas (UFAM), Centro de P&D em Tecnologia Eletrônica e da Informação (CETELI/UFAM) Manaus-AM, Brasil, E-mail: ccarvalho@ufam.edu.br

Os APs são roteadores sem fio padrão IEEE 802.11 que conectam dispositivos em rede sem fio e que são comumente utilizados em ambientes residenciais e comerciais. Os APs IEEE 802.11 transmitem periodicamente seus SSIDs (*Service Set Identifiers*), utilizando quadros de *beacons*. Em nossa proposta, medimos, em um dispositivo *smartphone*, o *RSS* (*Received Signal Strength*) dos quadros de *beacons* transmitidos por um ou mais APs de referência existentes no ambiente de experimentação.

O *smartphone* coleta os sinais *RSS* de APs próximos através de uma aplicação desenvolvida com o objetivo de montar um mapa de rádio do ambiente, que corresponde ao conjunto de registro dos valores de potências dos sinais *RSS*s em determinado local. Caso os valores de *RSS* de um determinado AP possuíssem valor menor que -90 dBm , em um instante de coleta da amostra, o valor era igualado a -200 , significando fora de alcance de comunicação, e o valor -200 não era considerado nos cálculos das funções características da aplicação. O computador “servidor” foi equipado com uma aplicação desenvolvida para abrigar um algoritmo que estima a localização do *smartphone* com base no algoritmo de aprendizagem de máquina WKNN.

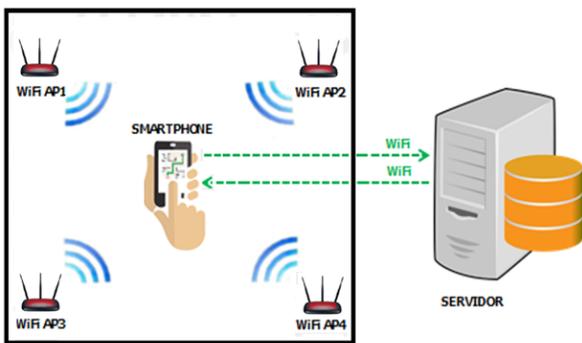


Figura 1 – Arquitetura Geral.

A. Fase de treinamento

Na Figura 2, visualizamos o fluxograma que mostra o funcionamento das aplicações residentes no aparelho *smartphone* (ex. cor azul) e no servidor (cor laranja) do trabalho proposto. Conforme figura, o processo de localização envolve duas fases principais: treinamento e teste. Na fase de treinamento, ou *off-line*, são executados os seguintes passos conforme fluxograma: (1) No bloco 1, “*Cadastro de RP*” o usuário se posiciona em cada local de medição, chamada de ponto de referência (*Reference Point - RP*) e, utiliza o aplicativo Android desenvolvido para cadastrar o $RP\ r = \{1, \dots, R\}$, onde R é o total de RPs. No cadastro do RP, utilizamos mapa, previamente desenhado do ambiente de experimentação, para graficamente posicionar o RP no mapa. No cadastro do RP, o aplicativo identifica os APs que estão no raio de comunicação do *smartphone*. O aplicativo utiliza os *SSIDs* (*Service Set Identifier*) e endereços físicos (*MAC-Medium Access Control*) ou *BSSID* (*Basic Service Set Identification*) para identificar cada AP de maneira única. Ao marcar o RP no mapa do ambiente, o aplicativo associa ao RP, os valores das coordenadas x, y do RP ao índice r do RP sendo cadastrado; (2) No bloco 2, “*Coleta amostras de RSS*”, os APs são utilizados

no ILS como equipamentos transmissores de *beacons* cujos valores de *RSS* são medidos no RP definido no bloco 1.

No processo de coleta das amostras de *RSS*, a aplicação do *smartphone* coletou n amostras de *RSS* representadas pelo conjunto S_n na Equação 1, para cada um dos m APs existentes no ambiente e visíveis a partir do RP r de coordenada x, y . Conforme Equação 2, os m conjuntos de amostras S foram agrupados para criar uma instância i de assinatura $FP_{i,r}$ ou instância de *fingerprint*, de um dado RP r , de coordenadas x, y .

Na Figura 2 bloco 3, chamado de “*Resumo de amostras*”, executamos uma função (*func(.)*) sobre a instância de *fingerprint* $FP_{i,r}$ de *RSS* de cada RP. A ideia foi criar um vetor de valores que resumisse as características da instância de *fingerprint* $FP_{i,r}$, sem a necessidade de transmitir todas as amostras de *RSS* para o servidor. Além disso, em trabalhos da literatura, verificamos que devido ao multipercorso dos sinais de rádio transmitidos e obstáculos móveis existentes nos ambientes internos, existem cenários em que, por exemplo, dois RPs diferentes, mesmo estando a diferentes distâncias de um certo AP, podem coletar amostras de *RSS* semelhantes deste AP específico. Isto pode gerar valores de *RSS* que afetam a acurácia de localização [14]. Para evitar isso, aplicamos funções características de média, máximo, mínimo ou mediana sobre as instâncias de *fingerprint* $FP_{i,r}$ a fim de minimizar esses efeitos.

O resultado do cálculo de cada função aplicado na Equação 2, originou a Equação 3 onde *func*, representa uma das funções características média, máximo, mínimo ou mediana; (4) No bloco 4, “*Envio de dados ao servidor*” o vetor resultante $func(FP_{i,r})$ foi transmitido para o servidor de aplicação.

$$S_n = \{RSS_1, \dots, RSS_n\} \quad (1)$$

$$FP_{i,r} = \{S_1, S_2, \dots, S_m\} \quad (2)$$

$$func(FP_{i,r}) = \{func(S_1), func(S_2), \dots, func(S_m)\} \quad (3)$$

Para transmissão de informações entre o *smartphone* e o servidor Web utilizamos os métodos PUT e GET do HTTP (Hyper Text Transfer Protocol) que foi aplicado para transmitir objetos no formato JSON (Java Script Object Notation). O servidor Web foi configurado para fornecer serviços utilizando API (Application Program Interface) Web REST (Representational State Transfer), permitindo comunicação aplicação-servidor.

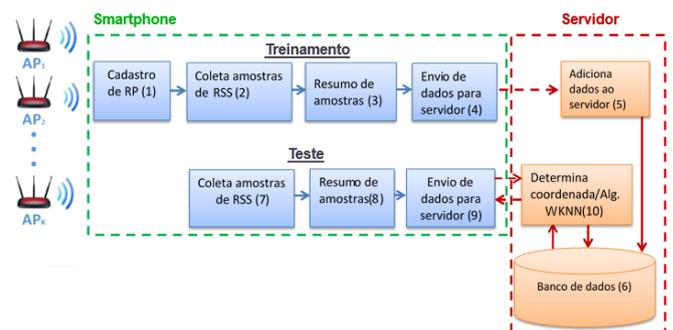


Figura 2 – Fluxograma da aplicação do *smartphone* e do Servidor.

O servidor foi equipado com o algoritmo de aprendizagem de máquina WKNN que utilizamos para propor o ILS deste artigo. Adotamos a arquitetura cliente-servidor a fim de transferir os cálculos computacionais dos algoritmos para o servidor, uma vez que o servidor possuía maior capacidade de processamento quando comparado ao dispositivo *smartphone*. Na Figura 2, bloco 5 “**Adiciona dados ao servidor**” e, ainda na fase de treinamento, foi implementado o serviço que permite adicionar vetores $func(FP_{i,r})$ ao “**banco de dados**” localizado no servidor, bloco 6.

Na Figura 3, observa-se o ambiente de experimentação com um esquema de RPs organizado tal como [13]. Este ambiente é formado pelo andar térreo do bloco (Centro de P&D em Tecnologia Eletrônica e da Informação - CETELI), localizado na UFAM (Universidade Federal do Amazonas). No ambiente, definimos um total de $R=85$ RPs (quadrados representados na cor verde e azul na Figura 3). Cada RP possuía área de $1m \times 1m$. Em cada RP foram coletadas $n=10$ amostras de RSS de cada AP visível no RP, sendo este conjunto de amostras utilizado para criar uma instância $i=\{1, \dots, I\}$ de *fingerprint* $FP_{i,r}$ e, nos experimentos utilizou-se $I=10$.

Como exemplo, cita-se que se em um RP_1 hipotético, fossem percebidos 7 APs pela aplicação Android, teriam sido coletadas um total de 700 amostras de RSS (ex. 7 APs \times 10 instâncias i de *fingerprint* \times 10 amostras de RSS/AP/instância) no RP_1 . É importante comentar que neste exemplo hipotético, as 700 amostras seriam inicialmente armazenadas no aplicativo Android e, em seguida, o aplicativo resumiria as 700 amostras em 70 instâncias de *fingerprint* para cada função característica de média, máximo, mínimo e mediana das amostras de RSS. Estas 70 instâncias seriam então transmitidas pelo aplicativo Android para o servidor de banco de dados, conforme realização dos experimentos individuais de utilização das funções características de média, máximo, mínimo e mediana.

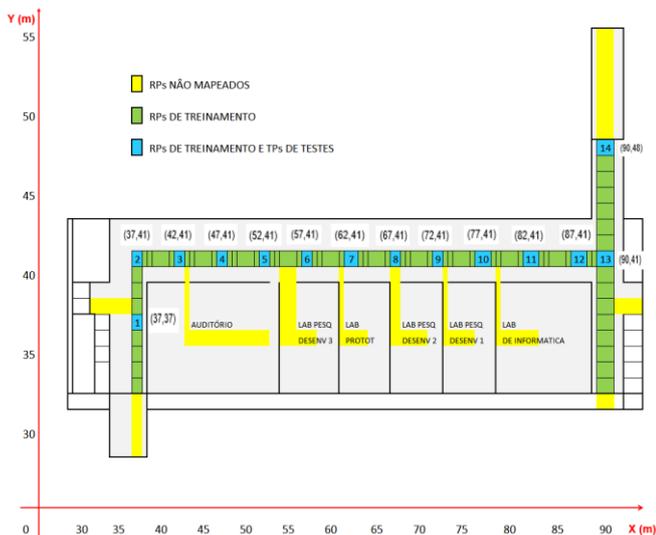


Figura 3 – Layout de experimentação no Bloco CETELI da UFAM.

B. Fase de validação e testes

Na fase de validação, realizada antes da fase de testes, dividimos a base de dados de treinamento, utilizando 80% da base para treinamento do algoritmo e 20% da base para validação do modelo. No experimento de validação, o

parâmetro \tilde{K} foi variado de 1, 2, 3, 4, 5, 6, 7, 8 e 9 com o objetivo de avaliar o modelo em relação ao erro médio de localização. Ainda nesta etapa o modelo alcançou acurácia de 98,05% na coordenada X e de 99,01% para a coordenada Y para o cenário de experimentação.

Na fase de testes (*online*), o ILS desenvolvido foi utilizado para permitir a um usuário de *smartphone* utilizar a aplicação Android para obter sua localização. A localização foi realizada a partir da coleta de amostras de RSS dos APs em TPs (*Test Points*) cujas coordenadas coincidiam com as coordenadas dos RPs. A Figura 2 mostra o fluxograma do processo de localização utilizado na fase de testes. O processo de testes possuiu os blocos 7, 8 e 9 com funções similares aos blocos 2, 3 e 4 utilizados na fase de treinamento, e um novo bloco (bloco 10), cuja função será explicada mais adiante.

O bloco 7 da fase de testes possuiu função semelhante ao bloco 2 da fase de treino. Contudo, na fase de testes, utilizamos as Equações 1 e 2, para coletar amostras de RSS formando conjuntos S_n com $n=\{2, 4, 6, 8, 10\}$ amostras para cada um dos m APs existentes no ambiente e visíveis a partir do TP de índice $t=\{1, \dots, T\}$ de coordenada x, y a serem determinadas. O bloco 8 da fase de testes (Bloco “**Resumo de amostras**”) possuiu função idêntica ao do bloco 3 da fase de treino, sendo executado uma função ($func(.)$) sobre a instância de *fingerprint* $FP_{i,t}$ de RSS de cada TP de índice t , onde $func(.)$ pode ser a função média, mediana, mínimo ou máximo, igual a função de sumarização de dados utilizada na fase de treino. O bloco 9 da fase de testes (“**Envio de dados para servidor**”) possuiu mesma função do bloco 4 da fase de treinamento.

No bloco 10 (“**Determina coordenada**”) foi implementado o algoritmo de localização do ILS proposto e que utiliza o algoritmo WKNN. O bloco utiliza os valores de RSS coletados da instância de teste num instante de tempo e atribui pesos conforme a distância entre os atributos da instância de teste e os atributos do(s) K elemento(s) mais próximo(s), obtendo assim coordenadas de posição do TP a ser localizado.

Neste bloco, o algoritmo de localização do ILS, inicialmente calcula a distância euclidiana $D(r,t)$, em espaço de sinais, entre os atributos dos resumos das instâncias de *fingerprint* de um dado TP t , determinado na fase de testes, e os atributos de RP r da base de treinamento, conforme representado na Equação 4, onde $func(FP_{i,r})$ e $func(FP_{i,t})$ são os resumos das instâncias de *fingerprint* i das fases de treinamento e teste. Em seguida é feita uma seleção dos $D(r,t)$ de menor distância no espaço de sinais com base no parâmetro $k=\{1, \dots, \tilde{K}\}$, para um dado TP t da fase de testes. Estes vetores são armazenados em um vetor $Dist$ em ordem crescente de distância de sinais conforme representado na Equação 5.

Os \tilde{K} RPs vizinhos mais próximos de um TP t foram utilizados para determinar as coordenadas (x', y') do TP t , conforme Equações 7 e 8. Nestas equações, os pesos $dist_k$, aplicados às coordenadas dos \tilde{K} RPs mais próximos, são proporcionais à distância, em espaço de sinais, de cada um dos K RPs para o TP t da fase de testes [14]. Desta forma, pôde-se estimar a localização dos TPs conforme Equações 7 e 8, onde x_k e y_k são as coordenadas dos \tilde{K} vizinhos mais próximos e x' e y' são as posições estimadas para o TP a ser localizado.

$$D(r, t) = \sqrt{\sum_{u=1}^i \sum_{v=1}^m [(func(FP_{i,r}))_{uv} - (func(FP_{i,t}))_v]^2} \quad (4)$$

$$Dist = \{dist_1, dist_2, \dots, dist_{\tilde{K}}\} \quad (5)$$

$$x' = \frac{\sum_{k=1}^{\tilde{K}} \left(\frac{x_k}{dist_k} \right)}{\sum_{k=1}^{\tilde{K}} \left(\frac{1}{dist_k} \right)} \quad (6)$$

$$y' = \frac{\sum_{k=1}^{\tilde{K}} \left(\frac{y_k}{dist_k} \right)}{\sum_{k=1}^{\tilde{K}} \left(\frac{1}{dist_k} \right)} \quad (7)$$

Utilizamos a biblioteca científica da linguagem de programação Python *scikit-learn*, empregada para desenvolver aplicações de aprendizagem de máquina [15], a fim de desenvolver o algoritmo de localização do ILS proposto.

Após a fase de treinamento no cenário do experimento, prosseguimos com a fase de testes definindo um caminho de 14 TPs marcados em azul na Figura 3 e coincidentes com os RPs já mapeados no ambiente na etapa anterior. Depois realizamos a predição dos dados com o mesmo conjunto dos TPs variando-se o número n de amostras de RSS coletadas do conjunto S_n com $n=\{2, 4, 6, 8, 10\}$ para cada um dos m APs visíveis em cada TP do ambiente e para cada valor de \tilde{K} . Com isso podemos avaliar o desempenho do sistema ILS através de cálculos de desvios dos valores de coordenadas reais (x_t, y_t) em comparação as coordenadas preditas (x'_t, y'_t) pelo sistema ILS.

Para conseguir isso, utilizamos indicadores estatísticos para quantificar os desvios através do cálculo do erro médio absoluto (*Mean Absolut Error - MAE*) que dá um peso igual a todos os desvios, e do cálculo da raiz do erro médio quadrático (*Root Mean Squared Error - RMSE*) [16] que resulta em um peso maior aos desvios grandes. As equações 8, 9 e 10 possibilitam calcular o erro MAE de cada coordenada e sua resultante enquanto as equações 11, 12 e 13 permitem o cálculo do erro RMSE de cada coordenada e sua resultante. Os resultados da avaliação de desempenho do sistema ILS são apresentados na seção III.

$$MAE_x = \frac{\sum_{t=1}^T |x_t - x'_t|}{n} \quad (8)$$

$$MAE_y = \frac{\sum_{t=1}^T |y_t - y'_t|}{n} \quad (9)$$

$$MAE = \sqrt{MAE_x^2 + MAE_y^2} \quad (10)$$

$$RMSE_x = \sqrt{\frac{1}{n} \sum_{t=1}^T (x_t - x'_t)^2} \quad (11)$$

$$RMSE_y = \sqrt{\frac{1}{n} \sum_{t=1}^T (y_t - y'_t)^2} \quad (12)$$

$$RMSE = \sqrt{RMSE_x^2 + RMSE_y^2} \quad (13)$$

III. RESULTADOS

A aplicação desenvolvida permite ao usuário coletar amostras de RSS dos APs do ambiente, enviando-os ao servidor de aplicação e retornando como resultado a posição nas coordenadas X e Y. Utilizando diferentes funções características nos resumos das instâncias de *fingerprint*, foram

obtidos os resultados apresentados nas Figuras 4, 5, e 6 nas quais \tilde{K} assume os valores 1, 2, 3, 4, 5, 6, 7, 8 e 9.

A Figura 4 apresenta os resultados dos cálculos do erro médio de localização em função da variação do parâmetro \tilde{K} para cada função característica média, máximo, mínimo e mediana. Neste gráfico podemos verificar que a **função média** apresentou menor erro médio de localização MAE com valor de desvio calculado igual a **2,12 metros** com o parâmetro $\tilde{K}=5$, quando comparado às funções de mediana, mínimo e máximo.

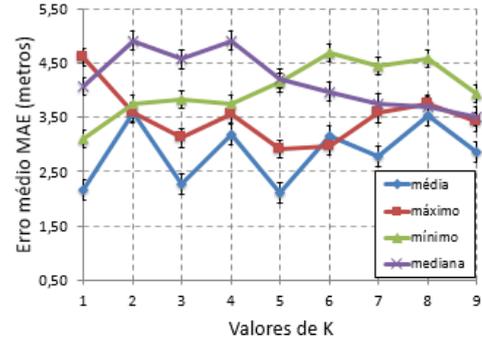


Figura 4 – Erro médio de localização MAE versus Valor K

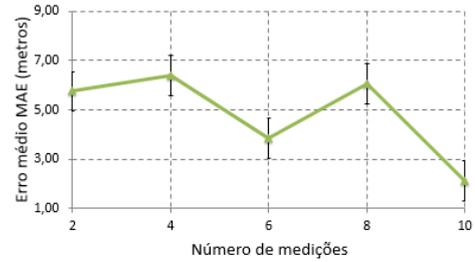


Figura 5 – Erro médio de localização MAE versus número de medições por instância (K=5)

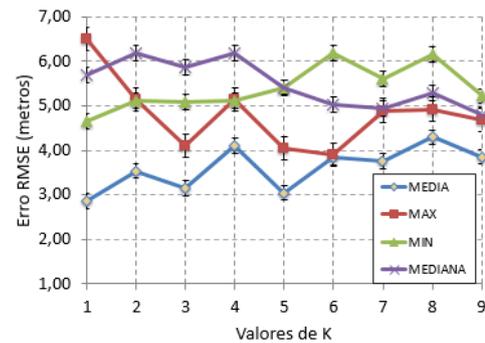


Figura 6 – Erro de localização RMSE versus Valor K

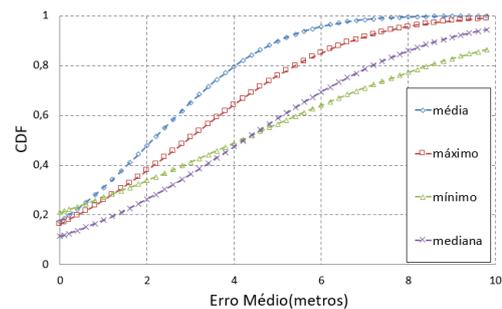


Figura 7 – Função de distribuição acumulada do erro médio MAE (K=5)

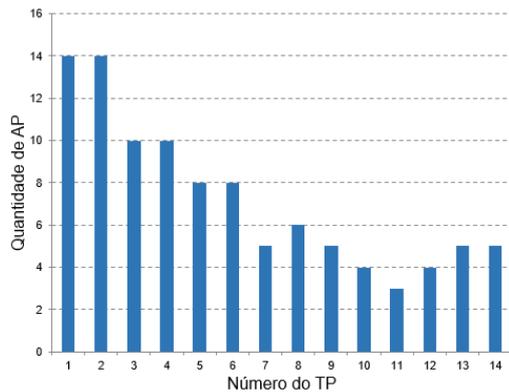


Figura 8 – Quantidade de APs visíveis versus TP.

Ajustando o modelo de localização proposto para operar na **função média** com o parâmetro $\tilde{K}=5$ fixo, podemos verificar na Figura 5 que ao variarmos o número de medições por instância durante o processo de coleta de amostras RSS na fase de testes, o sistema ILS alcança valores de erros médios menores quando aumentamos a quantidade de medições.

Na Figura 6 o sistema ILS proposto alcançou o menor valor para o erro de localização RMSE com valor de desvio calculado igual a **2,87 metros** na **função média** com o parâmetro $\tilde{K}=1$ superando alguns trabalhos na literatura (ex. erro médio de localização RMSE de 6,7283m [17]).

Na figura 7 podemos visualizar o gráfico da Função de Distribuição Acumulada (*Cumulative Distribution Function* – CDF) para cada função característica em relação ao erro médio de localização MAE com parâmetro $\tilde{K}=5$. No gráfico verificamos que em **79,69%** das estimativas de localização, a função característica da média apresenta erro médio de localização inferior a **4 metros** superando as demais funções que alcançaram o mesmo valor de erro médio com probabilidades menores: máximo=64,27%, mínimo=48,90% e mediana=47,50%.

Como foi comentado, nem todos os APs selecionados para o experimento foram detectados em todos os pontos de testes TP. Isto acontece devido que os níveis RSS de alguns APs apresentam perdas conforme o aumento da distância em relação a um dado TP, alcançando valores tão baixos que são desconsiderados para o cálculo das coordenadas de localização. Na Figura 8 é possível visualizar a quantidade APs detectados ou visíveis para cada ponto de teste TP no cenário do experimento. No gráfico podemos verificar que enquanto os TPs 1 e 2 detectaram 14 APs, o TP 11 detectou apenas 3 APs.

IV CONCLUSÕES

Neste artigo, foi apresentada proposta de ILS utilizando dispositivo *smartphone* equipado com aplicação Android que desenvolvemos para medir aplicando funções características e armazenar em banco de dados o resumo de RSS dos sinais emitidos por APs existentes no ambiente. Além da aplicação Android, desenvolvemos aplicação, localizada em computador servidor, baseada no algoritmo de aprendizagem de máquina WKNN e que estima a localização de um dispositivo *smartphone* equipado com a aplicação Android desenvolvida.

Concluimos com os resultados dos experimentos que o sistema ILS proposto alcançou resultados promissores com erro

de localização RMSE de 2,87 m apresentando superioridade em relação ao método *fingerprint* tradicional apresentado em [17] onde o erro de localização RMSE alcançou o menor valor igual a 6,7283 m. Propomos como trabalhos futuros avaliar o tempo de resposta do ILS proposto, avaliar o sistema em período de tempos diferentes e implementar o sistema em três dimensões.

AGRADECIMENTOS

Esta pesquisa, conforme previsto no Art. 48 do decreto no 6.008/2006, foi financiada pela Samsung Eletrônica da Amazônia Ltda, nos termos da Lei Federal no 8.387/1991, através de convênio no 004, firmado com o CETELI/ UFAM; pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES); pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM/Programa PPP), CT-INFRA, MCT/CNPQ e SECTI-AM.

REFERÊNCIAS

- [1] KHALAJMEHRABADI, Ali; GATSIS, Nikolaos; AKOPIAN, David. Modern WLAN fingerprinting indoor positioning methods and deployment challenges. *IEEE Communications Surveys & Tutorials*, v. 19, n. 3, p. 1974-2002, 2017.
- [2] LI, Xiangjie et al. Design and implementation of indoor positioning system based on ibeacon. In: 2016 International Conference on Audio, Language and Image Processing (ICALIP). IEEE, 2016. p. 126-130.
- [3] KALLIOLA, Kimmo. Bringing navigation indoors. *The Way We Live Next*, 2008.
- [4] GU, Yanying; LO, Anthony; NIEMEGEREERS, Ignas. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications surveys & tutorials*, v. 11, n. 1, p. 13-32, 2009.
- [5] YANG, Zhuorui; GANZ, Aura. A sensing framework for indoor spatial awareness for blind and visually impaired users. *IEEE Access*, v. 7, p. 10343-10352, 2019.
- [6] ALNAFESSAH, Ahmad et al. Developing an Ultra Wideband Navigation System for Visually Impaired People. *International Journal of Distributed Sensor Networks*, v. 12, n. 7, p. 6152342, 2016.
- [7] FERREIRA, D; CARVALHO, C. B.; MOTA, E. S. Localização em Ambientes Internos Utilizando Redes Wi-Fi. *Simpósio Brasileiro de Processamento de Sinais e Telecomunicações*, 2019. Petrópolis, RJ.
- [8] DOUSH, A. I. et al. ISAB: Integrated indoor navigation system for the blind. *Interacting with Computers*, v. 29, n. 2, p. 181-202, 2016.
- [9] GOMES, J. P. An Indoor navigation architecture using variable data sources for blind and visually impaired persons. In: 2018 13th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2018. p. 1-5.
- [10] MELIONES, Apostolos; SAMPSON, Demetrios. Blind museumtourer: A system for self-guided tours in museums and blind indoor navigation. *Technologies*, v. 6, n. 1, p. 4, 2018.
- [11] SHAIKH, S.; FARHAN, J.; SOSA, K.; NAIR, P. Smart Helmet for Visually Impaired. *International Research Journal of Engineering and Technology (IRJET)*, 2019, 06 (04), pp. 1846-1849.
- [12] ELMANNAI, W.; ELLEITHY, K. Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions. *Sensors*, v. 17, n. 3, p. 565, 2017.
- [13] YIU, Simon et al. Wireless RSSI fingerprinting localization. *Signal Processing*, v. 131, p. 235-244, 2017.
- [14] LIN, Tsung-Nan; LIN, Po-Chiang. Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. In: 2005 international conference on wireless networks, communications and mobile computing. IEEE, 2005. p. 1569-1574.
- [15] PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, v. 12, p. 2825-2830, 2011.
- [16] JAMES, Gareth, et al. *An introduction to statistical learning*. New York: springer, 2013.
- [17] YANG, Zhuorui; GANZ, Aura. A Sensing Framework for Indoor Spatial Awareness for Blind and Visually Impaired Users. *IEEE Access*, v. 7, p. 10343-10352, 2019.