UAI-FI: UM MÉTODO BASEADO EM APRENDIZADO DE MÁQUINA PARA CONTAGEM AUTOMÁTICA DE PASSAGEIROS UTILIZANDO SINAIS WI-FI

Marcos Paulino Roriz Junior
Ronny Aliagra Medrano
Universidade Federal de Goiás.
Faculdade de Ciências e Tecnologia – Engenharia de Transportes

RESUMO
A contagem de passageiros constitui uma importante parte na gestão e otimização do transporte público. Diversas iniciativas de Sistemas Inteligentes de Transporte têm sido empregadas com intuito de automatizar esta tarefa, através do uso de câmeras estereoscópicas e sensores de infravermelho. Porém, tais sistemas são bastante onerosos. Como alternativa, pode-se explorar os *smartphones* dos passageiros para realizar esta tarefa. Estes emitem periodicamente um quadro (pacote) de rede de sondagem para descobrirem redes Wi-Fi em volta. Neste sentido, este artigo visa investigar a viabilidade de utilizar este sinal para realizar a contagem automática de passageiros. O artigo propõe o método UAI-FI, que utiliza conceitos de aprendizado de máquina, para classificar e contabilizar automaticamente os quadros de rede Wi-Fi recebidos. O método desenvolvido foi aplicado em uma linha de ônibus da cidade de Goiânia. Os resultados iniciais indicam que o método conseguiu contabilizar aproximadamente 83,33% e 88,57% do embarque e desembarque dos passageiros contabilizados manualmente.

ABSTRACT
Counting passengers is an important task in managing and optimizing public transportation. Several Intelligent Transport Systems initiatives have been employed to automate this task through stereoscopic cameras and infrared sensors. However, such systems are quite costly. Alternatively, one can explore the passenger's smartphones to do this task. Such devices periodically send a network frame to discover nearby Wi-Fi networks. Therefore, this article aims to investigate the feasibility of using this signal to automatically count the passengers. The article proposes UAI-FI , a method based on machine learning concepts, that classify and count the received frames automatically. The developed method was applied in a bus line of the city of Goiânia. Initial results indicate that the method was able to detect approximately 83,33% and 88,57% of passengers boarding and disembarking such line when considering the manually counted results.

1. INTRODUÇÃO

Uma parte importante na gestão do transporte público baseia-se na contagem de passageiros que utilizam o sistema (Koffman, 1992; Myrvoll *et al.*, 2017). Através dessas informações, do embarque e desembarque de passageiros, pode-se otimizar a operação da linha, permitindo diminuir os efeitos de lotação e tempo de espera para os usuários do sistema (Oransirikul *et al.*, 2014). Dessa forma, muitas tecnologias têm começado a ser desenvolvidas com o objetivo de facilitar a contagem de passageiros. Um exemplo dessas tecnologias são os sistemas de contagem automático, que são baseados principalmente em câmeras estereoscópicas e detectores de infravermelho acima das portas (Myrvoll *et al.*, 2017).

Atualmente, muitas pesquisas têm começado a trabalhar com uma solução menos dispendiosa baseada na contagem de dispositivos móveis (*smartphones*) que estão a bordo do ônibus (Mikkelsen *et al.*, 2016). Grande parte desses *smartphones* podem conectar-se às redes sem fio IEEE 802.11, também conhecidas como redes Wi-Fi (Crow *et al.*, 1997). O padrão IEEE 802.11 define uma série de quadros (pacotes) de rede e protocolos para realizar a descoberta e conexão do dispositivo com roteadores. Dentre esses pacotes, destaca-se aqueles do tipo *probe requests*, ou requisições de sondagem (Gast, 2005). Esses são emitidos periodicamente pelos dispositivos para varrer e descobrir pontos de acesso próximos, isto é, roteadores que forneçam uma rede Wi-Fi. Todos os dispositivos que possuem um *chip* Wi-Fi ligado emitem esse sinal, mesmo aqueles que não estejam conectados a uma rede Wi-Fi. Isto é feito para descobrir e atualizar a lista de redes Wi-Fi próximas ao dispositivo.

Cada quadro de sondagem de rede possui o identificador do emissor e um indicador de potência do sinal da onda recebida, denominado *Received Signal Strength Indication* (RSSI), que indica quão próximo o emissor está do receptor. Recentemente, têm-se utilizado a quantidade de quadros e a potência dos sinais recebidos como mecanismos de contabilizar os passageiros (Mikkelsen *et al.*, 2016; Oransirikul *et al.*, 2014). Entretanto, a classificação de quais quadros de sonda estão dentro ou fora dos ônibus é realizada utilizando limites arbitrário. Isto torna a aplicação destas abordagens problemática, devido à complexidade de se estimar tais limites. Por exemplo, suponha que um dispositivo seja contabilizado como dentro do ônibus após enviar 100 quadros de sondagem de rede. Neste caso, o sistema poderá contabilizar incorretamente os motoristas que estejam andando ou parados ao lado de um ônibus, *e.g.*, em um engarrafamento.

Neste sentido, este artigo visa investigar a viabilidade de utilizar o sinal Wi-Fi dos passageiros para aprender padrões destes limites e realizar a contagem automática de passageiros. Este trabalho propõe o UAI-FI (*U*tilização de *A*prendizado de máqu*I*na para contagem automática de passageiros utilizando wi-*FI*), um método baseado nos conceitos de aprendizado de máquina para estimar automaticamente a função de classificação para os quadros e, consequentemente, contabilizar os passageiros. O aprendizado de máquina, também conhecido como *machine learning*, visa possibilitar que sistemas aprendam classificar dados sem ser programados explicitamente, de modo que os limites deverão ser aprendidos com base nos dados em si. Para avaliar o método UAI-FI, foi feito a sua programação na linguagem Python e sua implantação no *hardware* Raspberry Pi 3B (Richardson e Wallace, 2014). Utilizou-se este sistema em um teste de campo e paralelamente foi realizada uma pesquisa de embarque e desembarque de passageiros dentro do ônibus para validação dos resultados.

A fim de detalhar o método proposto, divide-se este artigo em seis seções. A seção 2 apresenta a fundamentação teórica. A seção 3 apresenta as etapas do método UAI-FI: captura, treinamento e análise. A seção 4 apresenta o experimento utilizado e a análise sobre os resultados obtidos. A seção 5 discorre sucintamente sobre os trabalhos relacionados. Por fim, a seção 6 apresenta a conclusão e trabalhos futuros para lidar com as limitações do método proposto.

2. FUNDAMENTAÇÃO TEÓRICA

O método proposto neste artigo, UAI-FI, se baseia na utilização dos sinais Wi-Fi emitidos pelos *smartphones* dos passageiros e no conceito de aprendizado de máquina. A fim de embasar o fundamento do método, esta seção discorre sucintamente sobre o funcionamento de ambos.

2.1. Protocolo IEEE 802.11 (Wi-Fi)

A tecnologia de rede sem fio Wi-Fi foi concebida e padronizada pelo Instituto dos Engenheiros Eletricistas e Eletrônico (IEEE) para possibilitar a comunicação sem fio entre dispositivos eletrônicos (Crow *et al.*, 1997). Este padrão, conhecido como protocolo IEEE 802.11, define as atividades necessárias, em forma de processo, para realizar a comunicação entre os dispositivos.

O protocolo determina processos para realizar cada uma das atividades envolvidas na conexão dos dispositivos. Dentre essas, pode-se destacar o processo de descoberta de rede (Gast, 2005). Para realizar tal atividade, o protocolo especifica um processo que sonda dispositivos próximos através de um quadro (pacote) de rede denominado *probe request*, ou requisição de sondagem. *Smartphones* emitem periodicamente este quadro com intuito de encontrar pontos de acesso próximos (Freudiger, 2015). Pontos de acesso, *e.g.* roteadores, que recebem este quadro enviam de volta uma resposta para o emissor contendo os dados da rede, como nome e endereço IP.

Os quadros de sondagem (*probe requests*) são difundidos publicamente em uma faixa de onda próxima ao emissor, permitindo que todos dispositivos interessados possam recebê-lo. Cada quadro inclui um número que identifica globalmente aquele aparelho, uma espécie de impressão digital do *smartphone*, chamado de MAC (*Media Access Control*). O alcance dessas ondas depende do subtipo do padrão Wi-Fi sendo utilizado. Como a maioria dos chips são do subtipo IEEE 802.11*.b*, assume-se que a distância alcançada pelas ondas é de até 100 metros em campo aberto e de até 30 metros dentro de residências (Gast, 2005). Para distinguir os quadros, utiliza-se um indicador da potência do sinal da onda recebida, denominado *Received Signal Strength Indication* (RSSI). Ao receber o quadro, utilizando a potência, o receptor computa o indicador RSSI, em decibéis, que representa a proximidade dos aparelhos. Este valor varia de 0 a $-\infty$, sendo que quanto mais próximo de zero, mais próximo os dispositivos se encontram.

Como os quadros de sondagem são enviados periodicamente e em faixas de onda públicas, um computador que esteja monitorando o respectivo espectro de onda também poderá recebê-los. Esses computadores, chamado de *sniffers*, podem utilizar esses quadros para detectar a presença e ausência de *smartphones* em sua vizinhança. Por exemplo, pode-se vincular o primeiro quadro *probe request* recebido de um *smartphone* como a entrada do mesmo em sua proximidade. Já quando o *sniffer* parar de receber quadros, pode-se estimar a saída do dispositivo (passageiro).

Para exemplificar tais conceitos considere o cenário ilustrado na Figura 1, onde um *sniffer* está localizado em um ônibus que se direciona ao ponto de parada. Nela, o *smartphone* do passageiro A envia quadros de sondagem para descobrir e atualizar a lista de redes próximas. Porém, como os quadros estão fora do raio do *sniffer*, o *smartphone* e, consequentemente, o passageiro, não é contabilizado. Por outro lado, o passageiro B está localizado dentro do ônibus. Quando o seu dispositivo enviar um quadro de sonda este será capturado pelo *sniffer*, implicando que B ainda se encontra no veículo. Após B descer, o *sniffer* irá parar de receber quadros oriundos de seu dispositivo. Percebendo a falta destes quadros, pode-se concluir que o passageiro deixou o ônibus e estimar o local de saída. Inclusive, ao combinar os dados do primeiro quadro emitido pelo dispositivo, junto com o último visto, pode-se estimar o par origem-destino do passageiro.

Uma das limitações dessa técnica é de que é necessário que o passageiro possua um aparelho *smartphone* e que este esteja com o Wi-Fi ligado. Note que, independente do *smartphone* estar conectado a uma rede, o dispositivo irá executar a atividade de sondagem periodicamente. Entretanto, isso somente ocorrerá se o Wi-Fi estiver ligado. Apesar desta limitação, estudos indicam que a população cada vez mais possui smartphones e uma grande fração dela deixam o Wi-Fi habilitado por padrão (Bonné *et al.*, 2013; Freudiger, 2015).

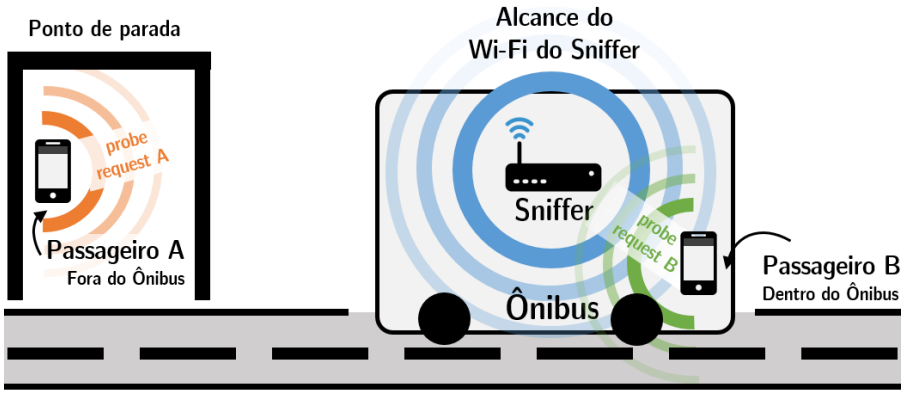


Figura 1: Exemplo de utilização de um *sniffer* para capturar a presença de passageiros.

2.2. Aprendizado de Máquina

Como dito previamente, nos métodos atuais, a classificação de quais quadros de sonda estão dentro ou fora dos ônibus é realizada através da comparação com limites arbitrários (Mikkelsen *et al.*, 2016; Oransirikul *et al.*, 2014). Por exemplo, comparar se a potência do sinal (RSSI) do quadro recebido está abaixo de limiar superior. Entretanto, tal abordagem é problemática devido à complexidade de se estimar *a priori* esses limites. Especificamente, suponha que o limite máximo de RSSI seja definido como -80 dB, aproximadamente 7 metros, para classificar os quadros como dentro do ônibus. Utilizando tal limite, quando um ônibus parar em um ponto de parada, o sistema poderá incorretamente classificar quadros de passageiros parados naquele local como interno ao veículo devido à proximidade dos dispositivos.

Para mitigar este problema, o método proposto (UAI-FI) utiliza os conceitos de aprendizado de máquina para estimar limites automaticamente a partir dos próprios dados (quadros) exemplos. O aprendizado de máquina visa possibilitar que sistemas computacionais aprendam classificar dados sem ser programados explicitamente, isto é, sem definir explicitamente esses limites. Isto ocorre através da utilização de quadros de sonda de rede dos quais já conhecemos a resposta.

Dentre os métodos de aprendizado de máquina destaca-se a *Support Vector Machine* (SVM – Máquinas de Vetores de Suporte), que permite classificar dados lineares e não lineares (Gama *et al.*, 2011). A idéia central da SVM é encontrar o melhor hiperplano que separe os dados, no nosso caso, os quadros recebidos. Como melhor hiperplano, entende-se aquele que possui a maior margem (vetores suporte) entre os dados de classe diferentes. Para ilustrar isto, considere a Figura 2 (a). Todos os hiperplanos traçados separam os dados, entretanto, H_2 possui a maior margem entre eles. É desejável uma margem maior para que se tenha uma maior precisão.

Para exemplificar o funcionamento da SVM, observe o item (b) da Figura 2. Considere um conjunto de treinamento com m dados vetoriais n-dimensionais, sendo $\mathrm{x}^{(i)}$ o i-ésimo exemplo, neste caso o quadro recebido. Cada dado possui uma classe $\mathrm{y}^{(i)} = \{-1, +1\}$, onde -1 representa fora do ônibus e $+1$ dentro do ônibus. Note que qualquer $\mathrm{x}^{(i)}$ localizado no hiperplano satisfaz a equação $\mathrm{w} \cdot \mathrm{x} + b = 0$, onde w é um vetor de pesos (w_1, w_2, ..., w_n), que também é a normal do hiperplano e b é uma constante que indica o quão distante o plano se encontra da origem.

Especificamente, deseja-se encontrar w tal que:

$$\mathrm{w} \cdot \mathrm{x} + b \geq +1 \quad \forall \mathrm{x}^{(i)} \text{ que possui } \mathrm{y}^{(i)} = +1, \text{ e} \tag{1}$$

$$\mathrm{w} \cdot \mathrm{x} + b \leq -1 \quad \forall \mathrm{x}^{(i)} \text{ que possui } \mathrm{y}^{(i)} = -1 \tag{2}$$

Estas duas inequações podem ser combinadas em uma só:

$$\mathrm{y}^{(i)}(\mathrm{w} \cdot \mathrm{x} + b) \geq 1 \quad \forall \mathrm{x}^{(i)} \tag{3}$$

Todos os vetores localizados nas margens, vetores de suporte, satisfazem estritamente (3). Logo, sejam x^+ e x^- dois vetores de suporte com classes $+1$ e -1 respectivamente. Pode-se calcular o tamanho da margem através do produto vetorial do vetor unitário $\hat{\mathrm{w}}$ com a diferença desses dois vetores, precisamente:

$$\hat{\mathrm{w}} \cdot (\mathrm{x}^+ - \mathrm{x}^-) = \frac{\mathrm{w}}{\|\mathrm{w}\|} \cdot (\mathrm{x}^+ - \mathrm{x}^-) = \frac{(\mathrm{w} \cdot \mathrm{x}^+ - \mathrm{w} \cdot \mathrm{x}^-)}{\|\mathrm{w}\|} = \frac{(1-(-1))}{\|\mathrm{w}\|} = \frac{2}{\|\mathrm{w}\|} \tag{4}$$

A SVM busca maximizar esta margem. Isto pode ser feito com a redução da norma $\|\mathrm{w}\|$, que é equivalente a reduzir $\frac{1}{2}\|\mathrm{w}\|^2$. Logo, o problema clássico da SVM pode ser enunciado como:

$$\min_{\mathrm{w},b} \frac{1}{2}\|\mathrm{w}\|^2 \quad \text{Sujeito à restrição: } \mathrm{y}^{(i)}(\mathrm{w} \cdot \mathrm{x} + b) \geq 1 \quad \forall \mathrm{x}^{(i)} \tag{5}$$

As restrições desta formulação requerem que todos os dados (quadros) sejam linearmente separáveis, o que nem sempre acontece. Para lidar com isso, é introduzido uma penalidade linear que permite que alguns dados violem esta restrição. A variável $\xi^{(i)}$ representa o erro da classificação em um dado $\mathrm{x}^{(i)}$. Assim, o problema é reformulado para:

$$\begin{gathered} \min_{\mathrm{w},b,\xi} \frac{1}{2}\|\mathrm{w}\|^2 + C\left(\sum_{i=1}^{m} \xi^{(i)}\right) \\ \text{Sujeito à restrição: } \mathrm{y}^{(i)}(\mathrm{w}\cdot\mathrm{x}+b) \geq 1-\xi^{(i)} \quad \forall \mathrm{x}^{(i)}, \text{ onde} \\ \xi^{(i)} = \max\left(0, 1-\mathrm{y}^{(i)}\left(\mathrm{w}\cdot\mathrm{x}^{(i)}+b\right)\right) \end{gathered} \tag{6}$$

O parâmetro C calibra a minimização, de modo que se C for pequeno é preferível maximizar a margem e ignorar algumas restrições, enquanto que se C for grande o erro terá maior impacto na função objetivo e consequentemente teremos uma margem um pouco menor.

O problema (6) é tipicamente resolvido na versão dual com multiplicadores de Lagrange:

$$\begin{gathered} \max_{\alpha} \sum_{i=1}^{m} \alpha^{(i)} - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\alpha^{(i)}\alpha^{(j)}\mathrm{y}^{(i)}\mathrm{y}^{(j)}\left(\mathrm{x}^{(i)}\cdot\mathrm{x}^{(j)}\right)\right) \\ \text{Sujeito às restrições: } 0 \leq \alpha^{(i)} \leq C, \\ \sum_{i=1}^{m} \alpha^{(i)}\mathrm{y}^{(i)} = 0 \end{gathered} \tag{7}$$

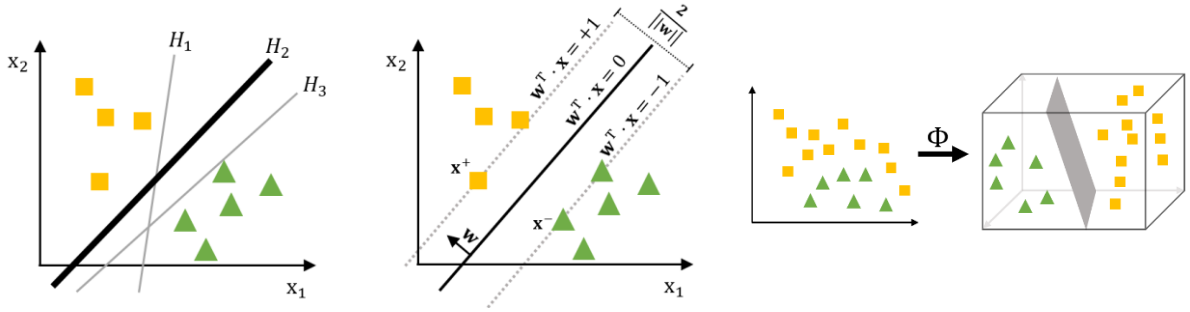
onde $\alpha^{(i)}$ representa o lagrangiano associado com a restrição da variável $\mathrm{x}^{(i)}$ em (6). Nesta formulação o vetor w pode ser reconstruído através da equação $\mathrm{w} = \sum_{i=1}^{m} \alpha^{(i)}\mathrm{y}^{(i)}\mathrm{x}^{(i)}$. Esta versão do problema apresenta uma das principais vantagens da SVM sobre outros métodos de classificação, como Regressão Logística e Redes Neurais, a facilidade de lidar com dados não lineares (Gama *et al.*, 2011). Note que a única parte que depende dos dados de entrada em (7) é o cálculo de $\mathrm{x}^{(i)} \cdot \mathrm{x}^{(j)}$. Neste sentido, para lidarmos com a não linearidade, pode-se substituir tal parte por $\Phi(\mathrm{x}^{(i)}) \cdot \Phi(\mathrm{x}^{(i)})$, onde Φ é um função que realiza o mapeamento de $\mathrm{x} \in \mathbb{R}^n$ para $\mathrm{x}' \in \mathbb{R}^k$. Por exemplo, a Figura 2 (c) ilustra a aplicação da função $\Phi\colon \mathbb{R}^2 \to \mathbb{R}^3$ em um conjunto de dados que não é linearmente separável em $\mathbb{R}^2$ para torná-lo separáveis em $\mathbb{R}^3$.

Usualmente utiliza-se a função de mapeamento de base radial gaussiana (*radial basis function*):

$$K(\mathrm{x}^{(i)}, \mathrm{x}^{(j)}) = \Phi(\mathrm{x}^{(i)}) \cdot \Phi(\mathrm{x}^{(j)}) = \exp\left(-\gamma\left\|\mathrm{x}^{(i)} - \mathrm{x}^{(j)}\right\|^2\right), \tag{8}$$

onde γ é o grau de influência de um exemplo, de modo que se este valor for alto indica que o raio de alcance do exemplo é pequeno, caso contrário será grande. A vantagem de se aplicar tal função é que ela pode realizar o mapeamento para infinitas dimensões. Além disso, não é necessário expandir os vetores de entrada para realizar o cálculo, ou seja, não é necessário gerar cada elemento do mapeamento $\Phi(\mathrm{x}^{(i)})$, basta calcular o valor direto de (8).

Neste sentido, o método UAI-FI baseia-se na SVM como meio de estimar uma função de classificação automaticamente usando o conjunto de quadros recebidos e classificados *a priori*.



(a) Possíveis hiperplanos (b) Hiperplano com maior margem (c) Mapeamento de dados para uma dimensão maior.

Figura 2: Exemplo de funcionamento da SVM. Adaptado de (Gama *et al.*, 2011) .

3. METODOLOGIA

Com base nos conceitos descritos previamente, esta seção apresenta o método UAI-FI. Este é dividido em três etapas: construção da plataforma de captura, treinamento e validação, como descritas ao longo desta seção.

3.1. Plataforma de Captura

Para capturar os quadros de sondagem emitidos pelos *smartphones* dos passageiros é necessário utilizar um computador intermediário (*sniffer*). Como este deverá situar dentro do veículo, recomenda-se utilizar computadores embarcados, tais como Raspberry Pi e Arduino, devido ao tamanho reduzido e facilidade de estendê-los com sensores (Richardson e Wallace, 2014).

Neste trabalho, utilizou-se a plataforma de *hardware* Raspberry Pi 3B como *sniffer* devido o mesmo vir por padrão com um *chip* Wi-Fi (IEEE 802.11.b/g/n) que pode ser utilizado em modo monitor para captura de quadros de rede. Também foi acoplado um sensor de GPS (BU-353) para receber os dados de posicionamento do *sniffer* e, consequentemente, dos passageiros. O dispositivo, ilustrado na Figura 3, possui uma dimensão de 85,6 x 53,98 x 17 mm.

Para controlar o *sniffer* recomenda-se utilizar um sistema operacional Linux voltado à análise forense de dados de comunicação, tais como Kali Linux e BackBox Linux, devido estes serem otimizados para captura de dados (Najera-Gutierrez, 2016). Especificamente, nesse trabalho, optou-se por utilizar o sistema operacional Kali Linux, devido o mesmo incluir uma série de modificações no Linux para facilitar a ativação do modo monitor da placa de rede Wi-Fi, necessárias para captura de pacotes, e também para leitura de dados de GPS por programas.

Utilizando os componentes de Wi-Fi e GPS do sistema embarcado, é possível obter os seguintes dados de entrada ao receber o quadro de sondagem do *smartphone*:

- Endereço MAC (*Media Access Control*), também conhecido como impressão digital do *smartphone*, sendo único para cada dispositivo.
- Indicador de força do sinal recebido (RSSI), um valor numérico, em decibéis, que indica a potência do sinal Wi-Fi entre o *smartphone* e o *sniffer*. Quanto menor o valor, mais próximo os dispositivos se encontram.
- Latitude e longitude do *sniffer* ao capturar um quadro de sonda. Este valor indica o local aproximado em que se recebeu o quadro de um determinado passageiro.
- Velocidade do veículo, em m/s, indicando o quão rápido o ônibus se encontrava quando o *sniffer* recebeu o quadro. Este dado é obtido através do sensor GPS.
- Horário do *sniffer*. Este valor é utilizado para indicar o horário em que o dado é coletado.

Para captura desses dados, recomenda-se a utilização de bibliotecas livres, como *Scapy*, *tshark*, *pygps* e *gpsd* (Bonné *et al.*, 2013). Essas bibliotecas facilitam o acesso e interação com os componentes da placa de rede e módulo GPS, permitindo a interceptação e a manipulação dos dados capturados por programas de terceiros. Por exemplo, a biblioteca *Scapy* possibilita indicar um tipo de quadro que se deseja capturar e uma rotina específica para tratá-lo. Neste trabalho foi utilizado a linguagem Python e as bibliotecas *Scapy* e *gpsd*.

Figura 3: *Sniffer* construído: Raspberry PI 3B, com sensor GPS e placa Wi-Fi.

O programa de captura opera da seguinte forma. Primeiramente, o programa ativa o modo monitor da placa de rede Wi-Fi para receber todos os quadros de rede. Após entrar nesse modo, aplica-se um filtro no espectro para receber apenas os quadros de sondagem. Quando um quadro desse tipo é identificado, é marcado o horário de coleta e inicializado a ação de obtenção dos outros dados. Primeiramente, obtêm-se a potência RSSI e o endereço MAC do quadro. Após isso, é realizado uma consulta ao módulo GPS para obter a velocidade e posição atual do *sniffer*. Estes dados, aliado ao horário de recebimento do quadro, possibilita derivar os seguintes dados:

- Distância ao ponto de parada mais próximo em metros. Para computar este dado utiliza-se a localização do *sniffer* ao receber o quadro. Este dado é importante pois permite verificar se o recebimento do quadro Wi-Fi ocorreu perto ou não de um ponto de parada.
- Distância percorrida pelo dispositivo em metros. Ao combinar a localização do primeiro quadro de rede Wi-Fi de um respectivo passageiro, usando o identificador MAC, com o último é possível estimar o deslocamento do passageiro até o momento.
- Tempo de viagem do dispositivo em segundos. Semelhantemente, pode-se combinar o primeiro e último quadro para obter o tempo de viagem até o momento.
- Total de quadros de redes recebidos de um mesmo endereço MAC em um determinado período de tempo Δ, *e.g.*, o número de quadros recebidos nos últimos 2 minutos.

Utilizando estes dados pode-se construir as componentes do vetor de entrada $x \in \mathbb{R}^6$ do modelo de classificação. Este conterá as seguintes componentes: RSSI, velocidade e todos os dados derivados. Note que o endereço MAC, horário e latitude e longitude não são inclusos no vetor x pois são específicos do emissor, nesse sentido não servem para classificar o quadro.

Para exemplificar o vetor de entrada, considere um vetor x oriundo de um quadro de rede de dentro do ônibus, descrito na Tabela 1. Este vetor representa um quadro recebido com potência RSSI de -44, sendo que o sinal foi obtida a uma distância de 751.9 metros do ponto de parada mais próximo. O veículo estava a uma velocidade de 4.3 m/s quando o recebeu. Além disso, utilizando o endereço MAC e a primeira posição e horário de quadro de rede do emissor, o *sniffer* observou que o passageiro percorreu 5524 metros, num período de 902 segundos, no qual emitiu 63 quadros de sondagem. A classe do quadro é $y = +1$, pois este se encontra dentro do ônibus. O vetor x e o vetor y correspondente são escritos em um arquivo de saída para serem posteriormente analisados. Esse processo se repete para cada quadro de sondagem recebido.

Tabela 1: Exemplo do vetor $x \in \mathbb{R}^6$ calculado a partir de um quadro de rede

RSSI	Velocidade (m / s)	Distância ao ponto parada mais próximo	Distância percorrida	Tempo em viagem	Total de quadros de Sonda enviados
-44	4.3	751	5524	902	63

3.2 Treinamento dos Dados

A segunda etapa consiste em realizar o treinamento do modelo utilizando a SVM. Neste sentido, primeiramente, é necessário capturar e discriminar manualmente os quadros de rede em classes fora do ônibus e dentro do ônibus. Estes serão associados a uma classe $y = \{-1, +1\}$, onde -1 representa fora do ônibus e $+1$ dentro do ônibus. Para isso, deve-se realizar coletas controladas.

Em particular, no primeiro momento deve-se realizar uma coleta de amostras negativas, isto é, de quadros de rede em que os passageiros não se encontram dentro do veículo. Deve-se realizar a captura com um ônibus vazio que percorrerá a linha em análise. Os pesquisadores e motoristas deverão desativar seus respectivos *smartphones* para não interferir na captura. Durante o trajeto, todos os quadros recebidos serão marcados como amostras negativas, $y = -1$.

Semelhantemente, deve-se realizar uma captura positiva de quadros. Para isso, deve-se realizar um trajeto com um ônibus que contenha apenas pesquisadores que possuem *smartphones* com Wi-Fi ligado. Utilizando o endereço MAC dos pesquisadores pode-se filtrar os quadros de rede recebidos. Estes devem ser marcados como amostras positivas, $y = +1$. Além disso, durante este trajeto pesquisadores devem adentrar e sair do veículo em pontos de paradas específicos para capturar os quadros de Wi-Fi em relação ao sobe e desce dos passageiros. Durante este período, os quadros desses pesquisadores devem ser marcados como amostras positivas.

Com estes dois bancos de dados pode-se iniciar a etapa de treinamento da SVM. Para facilitar o treinamento pode se utilizar bibliotecas de programação de aprendizado de máquina, como *Scikit-Learn, libSVM* e *SVMTorch*. Na implementação deste artigo foi usada a biblioteca *Scikit-Learn* (Pedregosa *et al.*, 2011) devido a simplicidade para treinar a função e a flexibilidade de calibrá-lo com diferentes parâmetros, além da mesma ser escrita em Python.

Para realizar o treinamento da função de classificação da SVM divide-se o conjunto de dados, obtidas nas coletas controladas, em dois conjuntos: *treinamento* e *teste*. Utiliza-se o conjunto de dados de treinamento para escolher os vetores que separam as classes. Neste momento, pode-se testar vários parâmetros de calibração γ e C para descobri-los. Por exemplo, pode-se variar γ e C no intervalo de $[10^5, 10^4, 10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$ resultando em uma combinação de 81 conjuntos de parâmetros diferentes, ou seja, será descoberto 81 funções de classificação. Entretanto, após realizar o treinamento com todas as combinações de parâmetros e obter as respectivas funções de classificações, deve-se escolher aquela que apresente a maior precisão e *recall* em relação ao conjunto de dados de *teste*, que foi separado previamente. Esta escolha é feita pois tal função de classificação possui a maior capacidade de generalização, visto que não se conhecia previamente os valores dos dados no conjunto *teste*.

3.3 Validação dos Dados

Após encontrar a função de classificação deve ser feita a validação da mesma. Para isso, devem ser realizadas diversas coletas de quadros de rede de sondagem com passageiros normais na linha em análise. Em cada captura, pesquisadores internos no ônibus devem contabilizar manualmente o embarque e desembarque dos passageiros. Como resultado disto, se tem um conjunto de dados que pode ser testado pela função de classificação descoberta.

A função de classificação treinada pode ser validada sobre um conjunto de quadros conforme descrito a seguir. Primeiramente, o programa deve ler um quadro da rede. A partir do endereço MAC do quadro calcula-se os dados adicionais usando os quadros anteriores do dispositivo. Caso esses não existam, inicialize os dados adicionais com o valor 0, com exceção da distância ao menor ponto de parada. Após isso, aplica-se a função de classificação. Caso o resultado desta seja positivo, $y = +1$, o quadro se encontra dentro do veículo e deverá ser contabilizado. Novamente, utiliza-se o identificador MAC, para verificar se o dispositivo já foi contabilizado. Caso positivo, atualize a última posição e quadro deste passageiro. Caso contrário, contabilize o passageiro e associe a subida do passageiro ao ponto mais próximo.

Paralelamente, para detectar a saída do passageiro é adotado o seguinte procedimento. Caso um endereço MAC deixe de enviar quadros de rede válidos por mais de Δ segundos, diz-se que o passageiro deixou o veículo. Tecnicamente, a IEEE 802.11 não especifica a periodicidade com que os dispositivos devem emitir este quadro. Entretanto, estudos de Freudiger (2015) e Mikkelsen *et al.* (2016) indicam que o envio do quadro de sonda ocorre no intervalo de 66 a 72 segundos. Como alternativa a este valor, pode ser utilizado o tempo médio da frequência

observada nos quadros da coleta positiva. Por fim, associa-se o local de desembarque do passageiro como o ponto de parada mais próximo em que houve a perda do sinal. Utilizando estes dois procedimentos, é possível iniciar a contagem dos passageiros. Assim, compara-se os resultados da contagem com os obtidos na contagem manual.

4. EXPERIMENTOS E RESULTADOS

Para avaliar o método UAI-FI foi realizado um estudo em uma linha de ônibus do Arco Oeste da Rede Metropolitana de Transportes Coletivos de Goiânia (RMTC). A linha escolhida (305), ilustrada na Figura 4, possui uma extensão de 12.41 km (RMTC, 2018). Esta foi escolhida devido interligar dois terminais de ônibus e pela dificuldade em capturar os dados de embarque e desembarque por outros métodos (como bilhetes eletrônicos) devido uma grande parte dos passageiros iniciarem ou terminarem as viagens nos terminais (Oransirikul *et al.*, 2014).

4.1 Captura dos dados

No experimento foram realizadas as três coletas: positiva, negativa e com passageiros normais. As coletas foram realizas em dias úteis no período das 09:00 às 12:00. A coleta negativa foi realizada no dia 6 de Junho (quarta-feira), enquanto que a coleta positiva foi realizada no dia 12 de Junho (quarta-feira). Já a coleta com passageiros normais foi realizada no dia 20 de Junho (sexta-feira). Nas coletas foi observado que os quadros positivos possuem um tempo médio de frequência de envio de quadros de 96 segundos. Neste sentido, adotou-se um valor semelhante a este, com $\Delta = 120$ segundos, como tempo máximo esperado para quadros subsequentes.

4.2 Treinamento

Utilizando o conjunto de quadros capturados, obteve-se uma função de classificação da SVM baseada no mapeamento de base radial gaussiana Φ com os parâmetros $C = 100000$ e $\gamma = 0.01$. A obtenção destes parâmetros indicam que para classificar corretamente os quadros foi necessário penalizar bastante os erros cometidos C e, consequentemente, a função reduziu a margem entre os exemplos de treinamento. Isto também é refletido no pequeno valor de γ, que aumenta a influência dos vetores de suporte, aqueles localizados na margem, na classificação.

A função treinada apresentou uma precisão na classificação dos quadros de dentro do ônibus de 95.53%, enquanto que obteve 98,07% para quadros localizados fora do veículo. O *recall* obtido, isto é, o número total de quadros associados as classes corretas, foi de 97,30% e 96,79% para os quadros de dentro e de fora do ônibus respectivamente. Com base nestes dados, nota-se que a função foi capaz de aprender a classificar corretamente quase todos os quadros obtidos.

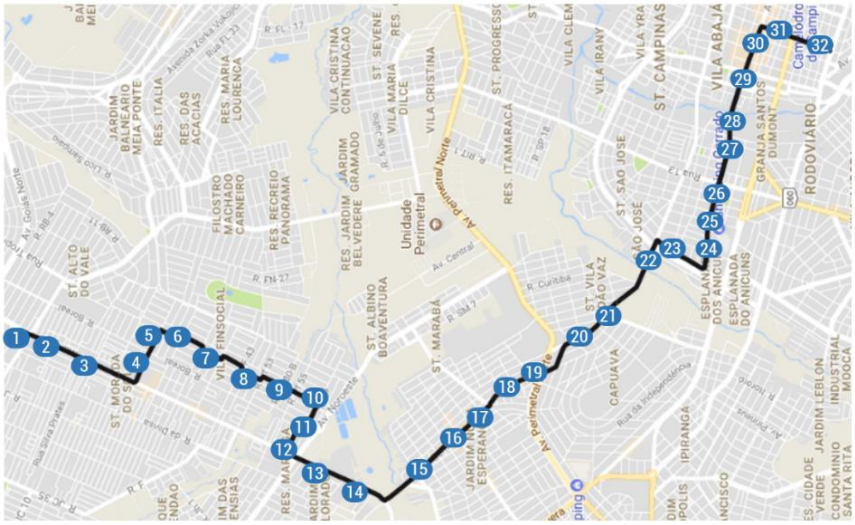


Figura 4: Linha 305 da RMTC de Goiânia. Extraído de (RMTC, 2018).

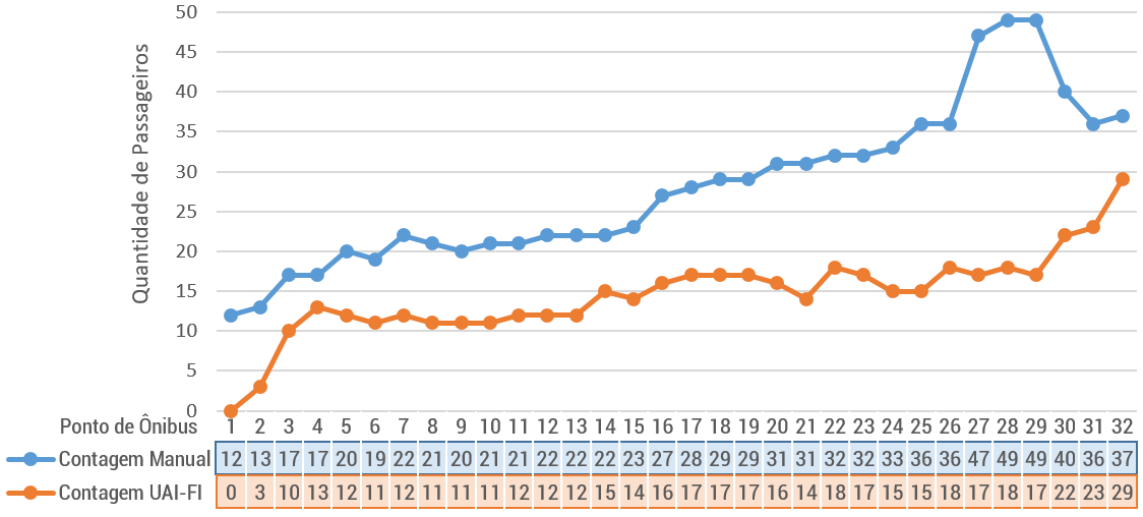
4.3 Teste de Campo

No teste de campo foi contabilizado manualmente o embarque de 72 e desembarque de 35 passageiros ao longo dos 32 pontos de parada. Utilizando o tempo de frequência $\Delta = 120$ e a função aprendida. Já o método UAI-FI, contabilizou o embarque de 60 e desembarque de 31 passageiros, ou seja, o modelo foi capaz de contabilizar 83,33% e 88,57% respectivamente do sobe e desce dos passageiros através dos quadros de rede dos mesmos. O gráfico ilustrado na Figura 5 mostra a evolução da quantidade de passageiros a bordo nas duas abordagens.

Como esperado, a contagem obtida pelo UAI-FI é inferior a contagem manual, visto que nem todos passageiros possuem *smartphones* ou que deixam o aparelho com Wi-Fi ativado. Apesar disso, a quantidade de passageiros contabilizados pelo UAI-FI e sua evolução é semelhante à realizada manualmente. Especificamente, observe que a curva de passageiros a bordo obtida no método UAI-FI apresenta um atraso em relação a curva obtida manualmente de cerca de 2 a 4 pontos de paradas. Uma das explicações para este fenômeno é que alguns pontos de paradas estão razoavelmente próximos. Neste sentido, o tempo de chegada ao próximo ponto pode ser maior do que a frequência de envio de quadros do dispositivo do usuário. Por exemplo, no teste de campo, o tempo gasto para ir do ponto de parada 2 ao 3 o ônibus foi de 31 segundos. Caso o *smartphone* possua uma frequência de envio de quadros maior que este intervalo, a subida do passageiro será atrelada a um dos próximos pontos de parada, por exemplo 3 ou 4.

Tal fenômeno também se repete nos gráficos de embarque e desembarque de passageiros, ilustrados nas Figuras 6 e 7. Observe que a curva de contagem obtida pelo método UAI-FI para embarque dos passageiros aparece defasada da curva real. Por exemplo, no primeiro ponto da contagem manual, no terminal, subiram 12 passageiros. Entretanto, devido à proximidade dos pontos de paradas 1, 2 e 3, esta contagem só foi refletida pelo UAI-FI no ponto de parada 3.

Semelhantemente, a curva de desembarque encontrada pela função de classificação do método UAI-FI também apresenta uma similaridade com a obtida manualmente. Note que em alguns casos, o método UAI-FI associa o ponto de saída do passageiro a um ponto de parada anterior ao visto manualmente. Uma das razões para esse acontecimento é de que este foi o último local em que o dispositivo emitiu um quadro. Para ilustrar isso, considere os pontos 4 e 5. Considere que o passageiro desceu no ponto 5, mas seu dispositivo enviou o último quadro no meio do caminho entre os pontos, sendo mais próximo do ponto 4 do que dê 5. Devido à proximidade ao ponto 4, o sistema irá marcar o ponto 4 como saída daquele passageiro.



Ponto de Ônibus	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Contagem Manual	12	13	17	17	20	19	22	21	20	21	21	22	22	22	23	27	28	29	29	31	31	32	32	33	36	36	47	49	49	40	36	37
Contagem UAI-FI	0	3	10	13	12	11	12	11	11	11	12	12	12	15	14	16	17	17	17	16	14	18	17	15	15	18	17	18	17	22	23	29

Figura 5: Quantidade de passageiros a bordo ao longo dos pontos de parada.

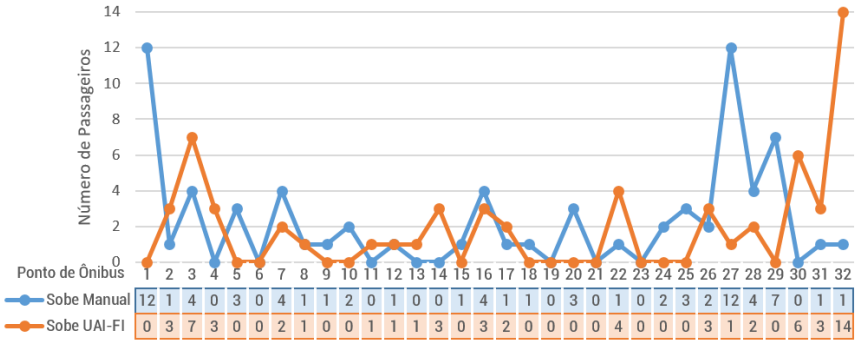


Figura 6: Quantidade de passageiros que embarca ao longo dos pontos de parada.

Por fim, vale salientar a semelhança entre os resultados obtidos, principalmente ao considerar a evolução temporal dos métodos. Os dados sugerem que não somente foi possível detectar uma parte dos passageiros, mas foi possível *acompanhar* a movimentação da contagem manual.

5. TRABALHOS RELACIONADOS

Dentro dos trabalhos relacionados ao tema existe uma maior parte desenvolvida na literatura internacional. Já no contexto brasileiro ainda o tema foi pouco desenvolvido. Mikkelsen *et al.* (2016) também investigaram mecanismos para estimar a lotação dos ônibus usando quadros de sondagem. Entretanto, no decorrer da pesquisa observaram um problema com a superestimação devido a inclusão de dispositivos próximos ao ônibus. Para resolver isto, os autores propuseram filtros SE ENTÃO que utilizam a frequência e o RSSI dos dispositivos com limites definidos arbitrariamente. Os resultados apresentaram alta variação e dificuldade em contabilizar os passageiros devido à complexidade de se estimar os parâmetros. Outro limitante é a simplicidade do filtro, sendo que os autores reconhecem que mais informações são necessárias para construção deste, inclusive, sugerindo o uso de aprendizado de máquina para isso.

Oransirikul *et al.* (2014) também investigaram este problema, mas sob um outro ponto de vista. Especificamente, utilizaram um *sniffer* em pontos de parada para coletar os quadros emitidos. Entretanto, o estudo apresenta algumas complicações. Primeiramente, o estudo também utiliza filtros de regras simples com somente as variáveis de frequência e RSSI dos dispositivos. Além deste problema, uma complicação emerge da dificuldade de capturar os dados em si. Como o *sniffer* está localizado no ponto, pode ocorrer que ao passageiro descer neste, o mesmo se afaste do raio do *sniffer* antes de emitir um quadro de sondagem.

No contexto brasileiro, Melo *et al.* (2015) propuseram um método de contagem de passageiros no ônibus através de Bluetooth. Além das tecnologias, a principal diferença entre as abordagens proposta é que o método proposto pelos autores requer que os aparelhos estejam executando uma aplicação específica para o funcionamento do mesmo. Este requisito dificulta a viabilidade do método, visto que todos os usuários deverão ter a aplicação instalada e em execução.

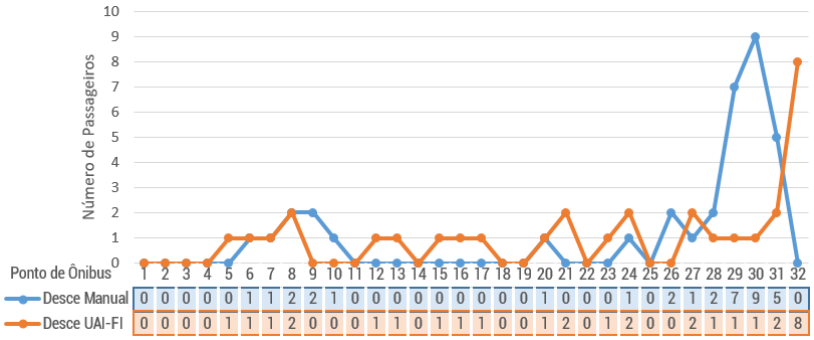


Figura 7: Quantidade de passageiros que desembarca ao longo dos pontos de parada.

8. CONSIDERAÇÕES FINAIS

Este trabalho investigou a viabilidade de se utilizar o sinal Wi-Fi emitido pelos dispositivos de passageiros para contabilizar os mesmos. Para isso, foi proposto um método, UAI-FI, que utiliza aprendizado de máquina para aprender uma função que classifique os quadros de rede emitidos. Para validar o método foi realizado um teste de campo onde os resultados sugerem que o mesmo foi capaz de contabilizar 83,33% e 88,57% respectivamente do sobe e desce dos passageiros através dos quadros de rede emitidos pelos seus dispositivos. Os resultados preliminares sugerem que é possível contabilizar os passageiros a partir dos sinais emitidos pelos dispositivos. Inclusive, esta abordagem tem um potencial de ser aplicada em outros domínios de aplicações, como na geração da matriz de origem destino ou descoberta da taxa de transferência entre diferentes linhas nos terminais.

Apesar da similaridade entre a contagem manual e a descoberta pela função de classificação do método UAI-FI, os resultados indicam uma defasagem da contagem obtida pelo método de alguns pontos de paradas de ônibus ao comparar com o método manual. Acredita-se que o principal fator disto é a frequência de envio dos quadros. Neste sentido, um possível trabalho futuro seria de investigar mecanismos que mitigasse esta falha. Outro problema interessante está relacionado a investigar se é possível descobrir uma função que relaciona a curva obtida no UAI-FI com a obtida manualmente. Especificamente, pode-se usar os resultados entre as curvas como subsídio para investigar a existência de uma função que relacione ambas utilizando métodos de aprendizado de máquina e estatísticos.

REFERÊNCIAS BIBLIOGRÁFICAS

Bonné, B.; Barzan, A.; Quax, P. e Lamotte, W. (2013) WiFiPi: Involuntary tracking of visitors at mass events. *2013 IEEE 14th Int. Symp. on A World of Wireless, Mobile and Multimedia Networks*, WoWMoM, p. 1–6.

Crow, B. P.; Widjaja, I.; Kim, J. G. e Sakai, P. T. (1997) IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, *35*(9), p. 116–126.

Freudiger, J. (2015) How Talkative is your Mobile Device? An Experimental Study of Wi-Fi Probe Requests. *WiSec '15 Proc. of the 8th ACM Conf. on Security & Privacy in Wireless and Mobile Networks*, p. 1–6.

Gama, J.; Carvalho, A.; Faceli, K. e Lorena, A. C. (2011) *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. (1ª Edição). Ed. LTC. Rio de Janeiro.

Gast, M. S. (2005) *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc.

Koffman, J. (1992) Automatic Passenger Counting Data: Better Schedules Improve on-Time Performance. M. Desrochers & J.-M. Rousseau (Eds), *Computer-Aided Transit Scheduling*, p. 259–282,. Springer.

Melo, Â. dos S.; Kraus Junior, W.; Farines, J.-M. e Pieri, G. (2015) ABORDAGEM DE BAIXO CUSTO PARA COLETA DE DADOS DE TRANSPORTE PÚBLICO USANDO SMARTPHONES. *Anais do XXIX Congresso Nacional de Pesquisa em Transporte da Anpet*, ANPET, Ouro Preto, v. 1, p. 1182–1193.

Mikkelsen, L.; Buchakchiev, R.; Madsen, T. e Schwefel, H. P. (2016) Public transport occupancy estimation using WLAN probing. *Proc. of 8th Int. Wksh. on Resilient Networks Design and Modeling*, RNDM, p. 302–308.

Myrvoll, T. A.; Håkegård, J. E.; Matsui, T. e Septier, F. (2017) Counting public transport passenger using WiFi signatures of mobile devices. *IEEE 20th Int. Conf. on Intelligent Transportation Systems*. ITSC. p. 1–6.

Najera-Gutierrez, G. (2016) *Kali Linux Web Penetration Testing Cookbook*. Packt Publishing.

Oransirikul, T.; Nishide, R.; Piumarta, I. e Takada, H. (2014) Measuring Bus Passenger Load by Monitoring Wi-Fi Transmissions from Mobile Devices. *Procedia Technology*, *18*(September), p. 120–125.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M. e Duchesnay, E. (2011) Scikit-learn: Machine Learning in Python. *Jrnl of Machine Learning Research*, *12*, 2825–2830.

Richardson, M. e Wallace, S. (2014) *Getting Started with Raspberry Pi: Electronic Projects with Python, Scratch, and Linux*. (2nd ed). Maker Media, Inc, USA.

RMTC. Informações Instituicionais (2018). Disponível em <http://rmtcgoiania.com.br>. Acesso em 08/07/2018.

Marcos Paulino Roriz Junior (marcosroriz@ufg.br)
Ronny Marcelo Aliaga Medrano (ronnymarcelo@ufg.br)
Engenharia de Transportes, Faculdade de Ciências e Tecnologia, Universidade Federal de Goiás
Rua Mucuri S/N, área 03, Bairro Conde dos Arcos, Aparecida de Goiânia - GO