

Redução de Inconsistências no Monitoramento da Frota de Ônibus Urbanos

Fernando D. M. Silva, Pedro Cruz, Rodrigo S. Couto, Luís Henrique M. K. Costa

Resumo—A Internet das Coisas é ferramenta fundamental para as cidades inteligentes. No contexto de sistemas de ônibus urbanos, ela permite monitorar o desempenho e a qualidade do serviço oferecido. Frequentemente, porém, os dados coletados em uma grande rede de sensores possuem inconsistências. Neste trabalho, é proposto um algoritmo que utiliza os dados de posição dos ônibus para classificá-los nas suas diferentes rotas e com isso lidar com falhas nos dados oferecidos diretamente pelos ônibus. Com base nesse algoritmo, é construído um método de medição do número de ônibus em circulação ao longo do tempo, permitindo avaliar a qualidade do serviço oferecido à população.

Palavras-Chave—Sistemas de transportes inteligentes, IoT, GPS, rastreamento veicular.

Abstract—The Internet of Things is an essential tool for smart cities. In the context of urban buses, it allows for monitoring the performance and quality of this public service. Nevertheless, data collected in such a large sensor network is prone to inconsistencies. In this paper, we propose an algorithm that uses positional data from the buses to classify them into their specific routes and deal with possible failures in the given bus data. Based on this algorithm, a method to measure the number of buses in operation and the quality of service provided to the population can be built.

Keywords—Intelligent transportation systems, IoT, GPS, vehicular tracking.

I. INTRODUÇÃO

Em muitas cidades, o serviço de ônibus é uma concessão municipal e sua auditoria é necessária para avaliar se os critérios estabelecidos pelas licitações são atendidos, além de identificar problemas no sistema. No modelo tradicional de auditoria, fiscais da prefeitura verificam manualmente a atuação das empresas de transporte. Esse procedimento exige uma grande quantidade de fiscais, sendo pouco eficiente, lento e sujeita a falhas humanas.

A adoção da Internet das Coisas (*Internet of Things* – IoT) possibilita a auditoria automatizada nos serviços de ônibus. Sensores nos próprios veículos coletam informações sobre a sua operação e repassam os dados para aplicações na nuvem [1], que podem analisar o comportamento do sistema como um todo. As informações de coordenadas de GPS coletadas por ônibus podem ser utilizadas em diversos cenários [1]–[6]. No entanto, frequentemente surge um desafio

com relação à confiabilidade dos dados. É necessário garantir sua consistência, evitando possíveis problemas gerados pela falta de robustez de dispositivos IoT ou por erros de configuração, seja por parte das empresas de ônibus ou dos motoristas.

Este trabalho busca solucionar problemas de confiabilidade do monitoramento de um sistema de transporte público. No sistema considerado, cada ônibus envia periodicamente dados de seu posicionamento GPS e sobre a linha na qual opera. Porém, como as informações das linhas são configuradas manualmente, os dados podem ser inconsistentes. Assim, este trabalho propõe um método que, com base nas trajetórias de cada ônibus, corrige inconsistências nas informações de suas linhas. O método é aplicado a dados de GPS dos ônibus da cidade do Rio de Janeiro e são gerados resultados do desempenho da frota que podem ser usados para fiscalização. Em uma das linhas, detectamos uma quantidade de ônibus 30% menor do que a estabelecida nos contratos de licitação.

O artigo está organizado da seguinte forma. A Seção II apresenta trabalhos relacionados. A Seção III descreve as características dos dados que são avaliados. A Seção IV detalha o algoritmo que é utilizado. A Seção V descreve os testes que são realizados no algoritmo. A Seção VI apresenta resultados dos testes descritos. A Seção VII mostra os dados que podem ser aproveitados para a auditoria. A Seção VIII conclui o trabalho e identifica direções futuras.

II. TRABALHOS RELACIONADOS

Diversas propostas aplicam IoT em ônibus urbanos. Os trabalhos Opensense [6], Mosaic [5] e Sensing-Bus [1] são exemplos da utilização de ônibus para o sensoriamento da cidade, e não da própria frota de ônibus. Esses estudos mostram que a adoção do paradigma IoT nos ônibus pode ser benéfica mesmo para setores outros que o de transportes.

O trabalho de Boshita *et al.* [3] utiliza a tecnologia LoRaWAN para enviar à nuvem dados de posicionamento dos ônibus. Os autores concluem que a rede LoRaWAN pode ser mais vantajosa em termos de custo para enviar dados de posicionamento, quando comparada com tecnologias como 3G e LTE. Porém, as condições apresentadas pelo autor mostram a necessidade de ser mais eficiente na etapa da transmissão, o que pode acarretar em perda de precisão. O presente

trabalho pode ser usado como correção dos dados gerados pela compressão dos dados de posição apresentada em [3].

Geetha e Cicilia [7] utilizam sensores internos aos ônibus para coletar dados sobre posicionamento, número de ônibus, número de passageiros e tempo de chegada. Os dados são entregues a pontos de acesso WiFi nos pontos de ônibus. Estes trabalhos assumem a validade dos dados, enquanto no presente trabalho, a consistência é verificada para que relatórios mais confiáveis sejam gerados.

Zambada *et al.* [8] e Jisha *et al.* [9] utilizam dados de velocidade e posicionamento de ônibus escolares. A ideia é que seja possível verificar se um determinado ônibus escolar está seguindo o trajeto definido para ele. Enquanto os ônibus escolares são monitorados em um traçado específico, o presente trabalho abre a possibilidade de verificar divergências em diversos traçados distintos e aponta os traçados mais prováveis que um veículo percorreu, mesmo na presença de uma fração de dados inconsistentes.

Os trabalhos [4] e [2] propõem um sistema de ônibus inteligentes com o uso do serviço de GPS (*Global Positioning Service*), no qual sensores integrados em cada ônibus registram posição e quantidade de passageiros para o envio desses dados à nuvem. Esses dados são utilizados para fornecer serviços a usuários do transporte público, tais como a localização em tempo real dos ônibus ou frequência de passagem de um ônibus. Porém, o uso é limitado a informações em tempo real. O presente artigo propõe que, com o armazenamento dessas informações, é possível fazer uma análise completa e mais precisa da execução do trabalho da frota.

Em [10] é proposto um sistema de banco de dados que monitora e salva os dados do percurso do ônibus. Com a ajuda de geocodificação reversa, informações sobre o trecho em que o ônibus está são utilizadas para determinar se o ônibus percorre a rota ou se desviou do curso. Para isso, é necessário criar um banco de dados contendo todas as informações das linhas em um formato específico. O presente trabalho usa apenas o conjunto de coordenadas que definem a linha.

III. DESCRIÇÃO DOS DADOS

Um ônibus segue um trajeto que é definido por uma “linha” de ônibus. Cabe ao poder público averiguar a demanda e definir a quantidade de ônibus que percorre cada linha. No caso da cidade do Rio de Janeiro, esses números são definidos nos contratos de concessão [11].

A IoT permite que informações sobre os ônibus sejam coletadas automaticamente, podendo ser usadas para auditoria dos termos do contrato. Neste trabalho, são explorados os dados de posicionamento dos ônibus ao longo do tempo para obtenção das seguintes medidas sobre o sistema:

- **Contingente pertencente a uma linha:** quantidade de ônibus em circulação para uma linha específica, em um determinado intervalo de tempo;

- **Tamanho total da frota:** Contagem de todos os diferentes ônibus que atuaram durante o dia;
- **Tamanho de pico da frota:** A maior quantidade de ônibus que operou simultaneamente em algum momento do dia (amostras de 30 minutos).

Utilizando como base o sistema do Rio de Janeiro, este trabalho considera que as empresas de ônibus fornecem as posições de seus ônibus na forma de coordenadas GPS, juntamente com um identificador de cada ônibus e a linha que percorre no momento. É considerado que os identificadores e as posições de GPS são consistentes *a priori*. Assim, pode-se reconhecer cada ônibus única e inequivocamente por um identificador e obter suas posições geográficas ao longo do tempo. Por outro lado, considera-se que as informações sobre a linha podem ser inconsistentes, sendo passível de validação. Essas considerações são feitas porque as coordenadas GPS são geradas automaticamente, com procedimentos bem conhecidos. Por outro lado, a identificação das linhas é configurada manualmente. Apesar da configuração manual dos identificadores dos ônibus, este trabalho os considera consistentes pois, para tal, basta que sejam únicos e que não mudem ao longo do período observado. A informação sobre linhas, porém, precisa ser alterada manualmente todas as vezes em que um ônibus altera a linha que ele serve. Por isso, é necessária uma verificação mais refinada dessa informação.

Os dados coletados dos ônibus são disponibilizados pela prefeitura através de uma API, cujo um *script* foi escrito para fazer a coleta e armazenamento constante desses dados, desde setembro de 2018 até a presente data. Para comparar esses dados, outra API da prefeitura também disponibiliza dados sobre a trajetória de cada linha que é registrada. Esses dados podem ser visualizados nas figuras 1 e 2.

Para identificar e eliminar possíveis inconsistências nas linhas, é proposto um método que compara o trajeto dos ônibus definido por suas coordenadas GPS com o traçado original das linhas de ônibus obtidas. O método então é aplicado aos dados de ônibus da cidade do Rio de Janeiro. A próxima seção descreve um algoritmo para identificar e eliminar possíveis inconsistências nas linhas.

IV. ALGORITMO PROPOSTO

O Algoritmo 1 encontra as linhas que provavelmente foram seguidas por um ônibus. Ele recebe como entradas a trajetória do ônibus e os trajetos de cada linha na forma de coordenadas geográficas. Então, calcula-se um grau de semelhança entre a trajetória do ônibus e o trajeto de cada linha. As linhas retornadas são aquelas que tiverem uma semelhança com a trajetória acima de um determinado patamar. Os parágrafos a seguir detalham o funcionamento do algoritmo.

O algoritmo recebe o dicionário *linhas*, que contém listas de pares de coordenadas indexadas pela sua linha correspondente, a lista *trajetoOnibus* que contém

Algoritmo 1: Método de detecção de linha.

```
Entrada: linhas, trajetoOnibus, tolerância,
          porcentagemValida
Resultado: linhasPercorridas
linhasPercorridas ← ∅;
para listaCoordLinha em linhas faça
  /* Define o tamanho da lista */
  nCoordLinha ← tamanho(listaCoordLinha);
  /* Obter o número mínimo de posições
     que um ônibus deve percorrer */
  minCoordValidas ←
    nCoordLinha × porcentagemValida;
  nCoordValidas ← 0;
  /* Comparar cada coordenada dos
     trajetos da linha e do ônibus */
  para coordLinha em listaCoordLinha faça
    para coordOnibus em trajetoOnibus faça
      distancia =
        haversine(coordLinha, coordOnibus);
      /* Se a distância for menor do
         que uma tolerância, as
         posições são consideradas
         equivalentes */
      se distancia < tolerância então
        nCoordValidas+ = 1;
        encerrarlaço
      fim
    fim
  fim
  /* Se percorre um mínimo do trajeto
     da linha, percorre a linha */
  se nCoordValidas ≥ minCoordValidas então
    linhasPercorridas ←
      linhasPercorridas ∪ linha ;
  fim
fim
retorna linhasPercorridas
```

todos os pares de coordenadas emitidos pelo ônibus em um dia, um valor *tolerância* que estabelece a distância máxima entre duas coordenadas para que sejam consideradas equivalentes e o valor *porcentagemValida*, que estabelece a porcentagem mínima de posições da linha que devem ser percorridas para que se considere que o ônibus percorre a linha.

O Algoritmo 1 opera da seguinte forma: Primeiro executa-se um loop em que cada linha no dicionário retorna um conjunto de pares de coordenadas da linha em *listaCoordLinha*. Então, define-se como *minCoordValidas* a menor quantidade de coordenadas necessárias para validar a linha. A seguir, utiliza-se o conjunto de pares de coordenadas de linhas e cada elemento deste é comparado com todos os outros elementos de *trajetoOnibus*, utilizando a fórmula de *haversine*. A fórmula de *haversine* retorna a distância entre duas coordenadas geográficas na superfície terrestre, ela é descrita em [9]. Se uma coordenada do trajeto do ônibus estiver a menos de *tolerância* de uma coordenada da linha, as coordenadas são consideradas equivalentes. Se ao menos *minCoordValidas* forem equivalentes, considera-se que o ônibus percorreu essa linha e ela é armazenada no conjunto *linhasPercorridas*. O algoritmo retorna o conjunto *linhasPercorridas*, já que um ônibus pode operar mais de uma linha no dia.

Para que o método proposto seja mais eficiente, são utilizadas diferentes estratégias. Uma delas consiste

no monitoramento dos resultados a cada cálculo de distância, para interromper a contagem caso a quantidade de pontos inválidos exceda um valor muito alto. Assim, linhas que não fazem parte da trajetória de um ônibus ocupam apenas uma fração do tempo de processamento para serem declaradas inválidas.

Com a execução do Algoritmo 1, um novo conjunto de dados é criado, relacionando cada ônibus testado com todas as linhas que foram detectadas para o mesmo. A partir disso, é possível chegar na quantidade total de ônibus em cada linha.

O método proposto é exemplificado na Figura 1. Destacam-se a trajetória do ônibus B25605 (em azul), ao longo do dia 16/01/2020, e a trajetória da linha 238 (em amarelo), uma linha candidata no processo. As partes na qual a linha segue uma rota diferente do trajeto do ônibus estão destacadas com um círculo vermelho. O total de pontos que são avaliados correspondem aos pontos marcados no mapa (destacados em vermelho). Os pontos marcados pelo círculo vermelho não são válidos, portanto, a razão a ser calculada será mais baixa e a linha é identificada como não percorrida.

V. DESCRIÇÃO DOS EXPERIMENTOS

O algoritmo de detecção de linha é utilizado para analisar dados reais, disponibilizados publicamente pela prefeitura do Rio através de uma API (*Application Programming Interface*)¹ e atualizados a cada minuto. A cada requisição à API web, são fornecidos uma *timestamp* do momento de registro dos dados, a identificação do ônibus, a linha informada pelo ônibus e as coordenadas GPS observadas naquele instante. Os resultados das análises deste trabalho são apresentados na Seção VI.

Uma primeira análise consiste no teste do algoritmo proposto. Primeiro, realiza-se a coleta manual do identificador de 17 ônibus e suas linhas correspondentes, no dia 6 de abril de 2020 das 15h às 17h. Esses ônibus passaram na Avenida Maracanã, próximo à coordenada (-22.915265, -43.229110). O resultado da coleta manual é comparado com os dados corrigidos com o método proposto. O método utiliza uma listagem de linhas menor, contendo além das linhas corretas outras linhas erradas para comparação, escolhidas aleatoriamente. Espera-se com isso obter uma visão preliminar da taxa de sucesso em um conjunto de ônibus quando parâmetros da comparação são determinados. Neste caso, define-se o valor de tolerância do algoritmo para 30 metros.

Uma segunda análise avalia o método em condições adversas, escolhendo-se os ônibus das linhas 455, 456, 457 e 474; escolhidos por meio dos dados da API e confirmados visualmente por exibição da trajetória em um mapa. Essas linhas foram selecionadas devido à sua grande similaridade. A Figura 2 exemplifica essa semelhança. Pode-se notar que apenas os trechos em

¹<https://www.data.rio/datasets/c47f8bb36e2e4760a761111ac58f5ec0>



Fig. 1. Comparação das trajetórias do ônibus B25605 (azul) e da linha 238 (amarelo).

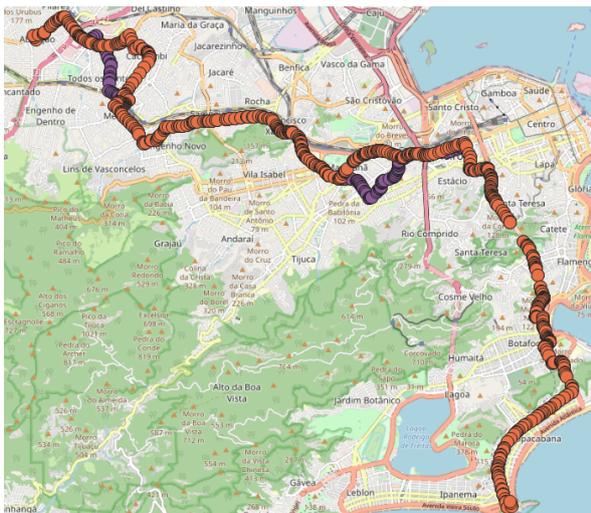


Fig. 2. Comparação entre as linhas 456 e 457.

roxo (pertencentes à linha 456) diferem da linha 457 (em laranja). Nessa segunda análise, seis ônibus são considerados com o mesmo valor de tolerância e taxa de sucesso mínima da primeira análise.

VI. RESULTADOS

O resultado da primeira análise, com os ônibus da Avenida Maracanã, é mostrado na Tabela I. Na tabela, a coluna válidas mostra o valor da taxa de sucesso encontrada para a linha no sentido que obteve o pior resultado. A análise mostra que 15 dos 16 ônibus possuem taxa de sucesso maior do que 95%. Dessa forma, considera-se, na próxima análise, 95% como limiar para definir se o algoritmo detectou a linha.

O resultado da segunda análise está na Tabela II. A coluna mapa representa as linhas identificadas de forma visual no mapa. A coluna API representa as linhas recebidas da API e a coluna detectado representa o resultado obtido com base no algoritmo proposto. O resultado considera que a linha é detectada se pelo menos 95% das suas amostras são válidas, com base na metodologia da análise anterior. A análise mostra que, mesmo com a alta acurácia do resultado anterior, o algoritmo ainda é suscetível a falha na identificação de linhas que possuem muita similaridade. Um exemplo é o B71056, para o qual não se

TABELA I
RESULTADOS DAS LINHAS DOS ÔNIBUS OBSERVADOS.

ID	Observada	Detectada	Válidas
B42657	457	457	96,2%
B25506	232	232+	100%
C30022	206	232	76,2%
B71047	247	247	96,6%
B71156	247	247	100%
A27616	433	433	99,5%
B71024	238	238	100%
B25605	249	249+	100%
C82534	363	363	95,1%
B42511	457	457	98,8%
B25590	232	232	100%
B71082	247	247	96,6%
A27652	433	433	99,5%
B25543	232	232+	100%
B25538	249	249+	100%
B42660	457	457	98,9%

pôde decidir entre as linhas 456 e 457. Outro exemplo é o ônibus B71074, que obteve taxa de sucesso inferior a 95% na linha 455 e, dessa forma, não há detecção de linha. Isso mostra que problemas podem ocorrer no processo de coleta e armazenamento dos dados, resultando em falhas de detecção. Esses problemas podem ser a perda de pacotes na rede coordenadas GPS imprecisas, o que pode fazer o algoritmo identificar mais de uma linha (p.ex., o ônibus B71056), ou não identificar nenhuma (p.ex., o ônibus B71074).

TABELA II
EXEMPLOS DE DETECÇÃO PARA LINHAS SIMILARES.

ID	Mapa	API	Detectado
B71063	455	455	455
B71074	455	455	-
B71108	455	455	455
B42527	456, 457	456, 457	456, 457
A29153	474	474	474
B71056	455, 457	455	455, 456, 457

VII. EXIBIÇÃO DOS DADOS PARA AUDITORIA

Os dados obtidos pela API, e corrigidos pelo algoritmo proposto, podem ser aglomerados e exibidos em gráficos para a auditoria do sistema de transporte. Assim, esta seção descreve uma aplicação do método proposto em um cenário de auditoria.

A Figura 3 mostra o contingente atuante em alguns dias do ano de 2019. Neste gráfico é possível comparar os tamanhos de pico (barra azul) e totais (barra laranja) que foram atingidas no dia. As informações pertencentes a uma linha podem ser vistas na Figura 4 e na Tabela III. A Figura 4 mostra a quantidade de ônibus atuando na linha 457 ao longo do dia 16/01/2020. É possível comparar os valores obtidos pela análise com o estabelecido pelo órgão público competente. O número de ônibus que deveriam atuar na linha é indicado pelo pontilhado.

A Tabela III mostra valores de interesse para o dia 16/01/2020 que podem ser tirados do gráfico, como médias e máximos. Duas médias são feitas: a total, que usa toda a extensão dos dados; e a de atividades, que faz a média dos valores entre 7 e 21 horas. Finalmente, é possível ver a atuação do algoritmo na auditoria dos dados da linha, ao mostrar o valor total de ônibus com a correção. Constata-se o impacto do algoritmo na correção da quantidade de ônibus total.

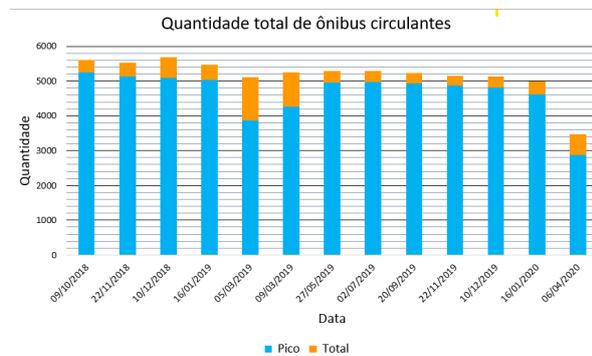


Fig. 3. Tamanho da frota em diferentes dias ao longo de 2019.

TABELA III

VALORES DE INTERESSE DA LINHA 457 NO DIA 16/01/2020.

Valor total com correção	35
Valor total sem correção	45
Valor máximo	41
Média total	33,3
Média em atividade	38,5

VIII. CONCLUSÕES

Este trabalho propôs um algoritmo capaz de reduzir inconsistências em informações de ônibus urbanos, para gerar relatórios mais precisos sobre o desempenho de diferentes linhas. Com o método proposto, é possível corrigir erros nos dados e chegar a resultados que definem mais precisamente o desempenho do sistema de transporte público.

O algoritmo proposto pode ser implementado em diferentes cenários, como procedimentos de auditoria, aplicações de previsão de chegada de ônibus e demais análises de padrões de deslocamento de frota. Ademais, o trabalho fornece inspiração para desenvolver uma ferramenta que possa fazer a análise dos dados de

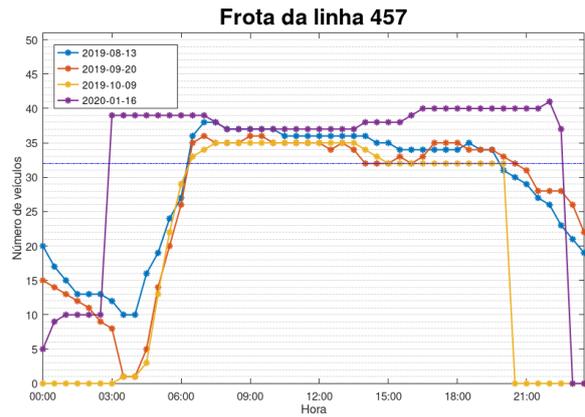


Fig. 4. Quantidade de ônibus na linha 457 ao longo do dia 16/01/2020.

ônibus públicos e dar mais transparência ao serviço de transporte público.

IX. AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001. O trabalho também foi financiado pelo CNPq, FAPERJ, e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), proc. 2015/24494-8.

REFERÊNCIAS

- [1] P. Cruz, F. F. Silva, R. G. Pacheco, R. S. Couto, P. B. Velloso, M. E. M. Campista, and L. H. M. K. Costa, "Sensingbus: Using bus lines and fog computing for smart sensing the city," *IEEE Cloud Computing*, pp. 1–11, 2018.
- [2] U. Farooq, T. u. Haq, M. Amar, M. U. Asad, and A. Iqbal, "Gps-gsm integration for enhancing public transportation management services," in *2nd International Conf. on Computer Engineering and Applications*, vol. 2, 2010, pp. 142–147.
- [3] T. Boshita, H. Suzuki, and Y. Matsumoto, "Iot-based bus location system using lorawan," in *21st Int. Conf. on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 933–938.
- [4] A. J. Kadam, V. Patil, K. Kaith, D. Patil, and Sham, "Developing a smart bus for smart city using iot technology," in *2nd Int. Conf. on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1138–1143.
- [5] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *35th Int. Conf. on Computer Communications (INFOCOM)*. IEEE, 2016, pp. 1–9.
- [6] A. Marjovi, A. Arfire, and A. Martinoli, "High resolution air pollution maps in urban environments using mobile sensor networks," in *Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2015, pp. 11–20.
- [7] S. Geetha and D. Cicilia, "Iot enabled intelligent bus transportation system," in *2nd Int. Conf. on Communication and Electronics Systems (ICES)*. IEEE, 2017, pp. 7–11.
- [8] J. Zambada, R. Quintero, R. Isijara, R. Galeana, and L. Santillan, "An iot based scholar bus monitoring system," in *IEEE Ist Int. Smart Cities Conference (ISC2)*. IEEE, 2015, pp. 1–6.
- [9] R. Jisha, A. Jyothindranath, and L. S. Kumary, "Iot based school bus tracking and arrival time prediction," in *Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017, pp. 509–514.
- [10] M. Al-Jabi, "Toward an iot-enabled adaptive interactive bus transportation system," in *2nd Int. Conf. on the Applications of Information Technology in Developing Renewable Energy Processes Systems (IT-DREPS)*, 2017, pp. 1–4.
- [11] Cadastro da linha de ônibus. [Online]. Available: <http://www.rio.rj.gov.br/web/smr/exibenoticias?id=9437691>