A Real-Time, Privacy-Preserving Crowd Management System Using Density-Based Clustering of Simulated Network Signals

Ankur Yadav
Reg. No.: 24BCE1836
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Chennai, India
ankur.yadav2024@vitstudent.ac.in

Nithish Kannan M
Reg. No.: 24BCE1842
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Chennai, India
nithishkannan.m2024@vitstudent.ac.in

Namish Gupta
Reg. No.: 24BCE1934
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Chennai, India
namish.gupta2024@vitstudent.ac.in

***Abstract*—This paper presents a novel, end-to-end framework for real-time crowd management that prioritizes user privacy by design. The escalating need for intelligent monitoring in dynamic environments, such as university campuses and public venues, is often hindered by the privacy implications of traditional surveillance and the unreliability of network-based methods due to MAC address randomization. Our proposed system addresses these challenges by leveraging the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to analyze passively collected, anonymized network signals. The methodology integrates a reproducible network emulation environment using Mininet for realistic data generation with a scalable, full-stack MERN (MongoDB, Express.js, React, Node.js) application for data processing and visualization. Real-time data dissemination is achieved via the WebSocket protocol. The results demonstrate the system's capacity to accurately estimate crowd density, identify high-traffic zones, and visualize population dynamics in near real-time. By analyzing the spatio-temporal density of signals rather than tracking unique devices, the framework is inherently resilient to MAC randomization and provides actionable crowd intelligence without compromising individual privacy.**

***Index Terms*—Crowd Management, Real-Time Systems, DBSCAN, MERN Stack, Network Activity Simulation, Socket.IO, Privacy Preservation, Smart Campus, Mininet**

I. Introduction

The proliferation of smart devices and the advancement of Internet of Things (IoT) technologies have paved the way for the development of smart environments, capable of optimizing resource allocation, enhancing public safety, and improving operational efficiency. Within this domain, real-time crowd analytics has emerged as a critical component for managing large-scale public spaces such as university campuses, transportation hubs, and event venues. Understanding crowd patterns is essential for effective space utilization, safety protocol enforcement, and emergency response planning.

However, the deployment of effective crowd monitoring systems is constrained by a fundamental trade-off between analytical accuracy and the preservation of individual privacy. Conventional camera-based surveillance systems, while capable of high accuracy, are inherently privacy-intrusive, computationally expensive, and susceptible to environmental factors like poor lighting and physical occlusions [5]. As an alternative, network-based sensing, which utilizes ubiquitous Wi-Fi signals, has gained traction. Early iterations of these systems relied on counting unique Media Access Control (MAC) addresses from device probe requests. This approach has been rendered largely obsolete by the widespread adoption of MAC address randomization in modern mobile operating systems—a security feature designed specifically to prevent such tracking [2]. This has forced researchers into a complex "arms race" of device fingerprinting, attempting to create stable identifiers from other non-randomized fields in wireless frames [6], [13], a solution that reintroduces implementation complexity and potential privacy vulnerabilities [2].

This paper proposes a paradigm shift in network-based crowd monitoring. Instead of attempting to circumvent privacy-enhancing features like MAC randomization, our system embraces the anonymity they provide. We present a framework that derives crowd intelligence from the collective density of anonymized network signals, rendering device identity irrelevant. The primary contributions of this work are threefold:

- **A Novel Privacy-by-Design Architecture:** We introduce a system that analyzes the spatio-temporal density of

This work was conducted under the guidance of Dr. S A AMUTHA JEEVAKUMARI, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Chennai, India.

anonymous network signals [2]. By focusing on "how many" signals are present in a location rather than "who" they belong to, the system is inherently robust to MAC randomization and upholds user privacy.

- **Effective Application of DBSCAN for Crowd Analytics:** We demonstrate the implementation of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm as the core analytical engine [1]. Its unique ability to identify arbitrarily shaped clusters and automatically filter noise makes it exceptionally well-suited for discovering dynamic crowd formations in complex environments [4].
- **An End-to-End Reproducible Framework:** We present a complete and verifiable system, encompassing a scientifically valid network emulation environment using Mininet for controlled data generation and a scalable MERN and WebSocket-based application for real-time data persistence, analysis, and visualization[1][2]. This provides a full lifecycle model from research validation to practical deployment.

II. Literature Review

The field of network-based crowd sensing has evolved significantly, moving from simple device counting to sophisticated signal processing and machine learning techniques. This evolution has been largely driven by the need to overcome the challenge of MAC address randomization while improving accuracy and respecting privacy.

A. Wi-Fi Probe Request-Based Systems

Systems based on passively capturing 802.11 probe requests represent a significant body of work in this area. While early methods are now unreliable, recent research has focused on creating resilient device fingerprints. The AFOROS system [6], for instance, analyzes non-randomized fields within 802.11 frame headers to construct stable identifiers for more accurate tracking. Similarly, the CrowdWatch framework [13] leverages multi-modal fingerprints from Wi-Fi frames for privacy-preserving monitoring. While effective, these fingerprinting techniques can be complex to implement and maintain [6]; moreover, they operate in a gray area regarding user privacy by attempting to re-identify devices, even if pseudonymously [2]. Our proposed system diverges from this approach by forgoing identification entirely.

B. Device-Free Sensing with Channel State Information (CSI)

An alternative and powerful approach is device-free sensing, which does not require individuals to carry any devices. These systems measure how human bodies disturb wireless signals propagating between a transmitter and a receiver. Channel State Information (CSI), which provides fine-grained physical layer data about the communication channel, has proven particularly effective. Research has shown that multi-link CSI from commercial off-the-shelf hardware can achieve high accuracy in counting small groups [9] and can be used to predict outdoor human flow [10]. CSI-based methods offer excellent privacy preservation, as they sense human presence directly. However, their performance can degrade in high-density scenarios, and they often entail higher computational overhead compared to probe-based methods [6].

C. Clustering Algorithms in Network Data Analysis

The choice of analytical algorithm is crucial for interpreting raw network data. While many systems employ standard classifiers like Support Vector Machines (SVMs) [10], these models are often ill-suited for discovering the dynamic, arbitrarily shaped formations characteristic of human crowds. Density-based clustering algorithms, particularly DBSCAN [1], offer a more robust alternative [4]. DBSCAN's ability to identify clusters of any shape and its inherent mechanism for noise detection make it ideal for this application. Prior work has extended DBSCAN for semantic trajectory analysis (E-DBSCAN) [7] and for improving Wi-Fi positioning accuracy [11]. However, its application as the core engine for a real-time, privacy-preserving crowd counting system that leverages anonymity as a feature represents a novel contribution. Table I provides a comparative summary of these approaches and highlights the gap addressed by our proposed system.

III. Proposed System Architecture and Methodology

The proposed system is architected as a multi-stage, end-to-end pipeline that transforms raw, simulated network signals into actionable crowd intelligence. The architecture is logically divided into a **Validation Layer**, where the core sensing and clustering hypothesis is tested in a controlled environment, and a **Deployment Layer**, which represents a scalable, real-world application for visualizing the results. The complete data flow is illustrated in Fig. 1.

[1]The source code for the implementation is available at: https://github.com/Namish-Gupta/Real-Time-Crowd-Management-System

[2]A live deployment of the system can be accessed at: https://real-time-crowd-management-system-cnproject-1h1zuoha9.vercel.app/

TABLE I
COMPARISON OF CROWD MONITORING TECHNIQUES

Reference	Methodology	Primary Limitation	Contribution of Proposed System
Vega-Barbas et al. [6]	Wi-Fi Probes + Fingerprinting	Raises privacy concerns; complex implementation.	Avoids fingerprinting by focusing on signal density, ensuring privacy.
Brena et al. [9]	Multi-link Wi-Fi CSI + ML	Accuracy degrades in high-density scenarios.	Designed for scalability in dense environments using a lightweight clustering approach.
Ogawa & Munetomo [10]	Wi-Fi CSI + SVM	SVM is less effective for discovering arbitrarily shaped crowds.	Employs DBSCAN, which excels at identifying dynamic, non-linear crowd formations.
Delzanno et al. [8]	Lightweight DBSCAN on Wi-Fi Probes	Achieves moderate precision with a limited feature set.	Proposes a full-stack architecture for robust, real-time visualization and alerting.
Proposed System	Simulated Network Signals + DBSCAN + MERN/WebSocket	Relies on simulation; requires real-world calibration.	Provides a complete, privacy-by-design, and reproducible framework resilient to MAC randomization.

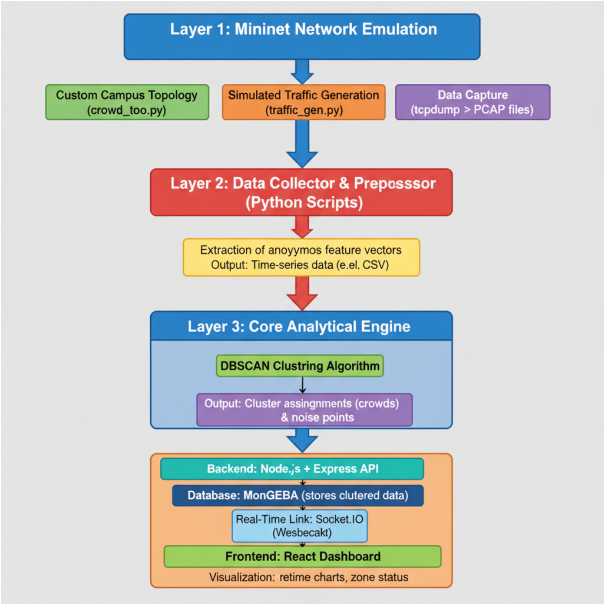

Fig. 1. System Workflow Flowchart

A. Layer 1: Network Emulation and Data Generation

The foundation of the system's validation is a virtualized laboratory built with the Mininet network emulator. This environment allows for the creation of a custom network topology that simulates a university campus with nine distinct zones, such as 'Library', 'Admin Block', and 'North Square'. Python scripts are used to control virtual hosts within this topology, programming them to generate realistic network traffic. Specifically, these scripts simulate the movement of students by emitting Wi-Fi probe requests with randomized MAC addresses from different points in the network. Virtual sensors, implemented as listeners on switches corresponding to each zone, capture this traffic using `tcpdump`. The resulting PCAP files constitute a ground-truth dataset where the number and location of simulated users at any given time are known, providing a reproducible framework for testing the system's accuracy.

B. Layer 2: Data Acquisition and Feature Extraction Pipeline

The raw data captured by Mininet is processed by a data pipeline consisting of Python-based collector and preprocessor modules. These scripts parse the PCAP files or associated logs to extract a non-personally identifiable feature vector for each detected network event. This vector contains only essential, anonymized information: a timestamp, the `zone_id` (determined by which virtual sensor captured the signal), and a proxy for signal strength (RSSI). This minimalist feature set is intentionally designed to be privacy-preserving, containing no information that could be used to track a specific device or individual over time.

C. Layer 3: Core Analytical Engine: DBSCAN Clustering

The analytical core of the system is the DBSCAN algorithm [1], a seminal work in density-based clustering. DBSCAN groups points that are closely packed together, marking as outliers points that lie alone in low-density regions. It is defined by two key parameters:

- **epsilon (ϵ):** The maximum distance between two points for them to be considered neighbors. In our system, this is a conceptual distance in a 2D space representing the campus map, set to 30 units.
- **MinPoints:** The minimum number of points required to form a dense region (a cluster core). This is set to 2, meaning at least two simulated devices in close proximity are needed to begin forming a crowd cluster.

The extracted feature vectors are fed into the DBSCAN engine, which processes the spatio-temporal data points. The algorithm's output is then directly mapped to the crowd management domain:

- **Clusters:** These are dense regions of network signals and directly correspond to detected crowds. The number of points in a cluster provides the population count, and the geometric centroid of the points indicates the crowd's location.

- **Noise/Outliers:** These are isolated data points that do not belong to any cluster. They represent sparse, transient network signals or background chatter and are automatically filtered out by the algorithm, significantly enhancing the signal-to-noise ratio of the final crowd estimate.

D. Layer 4: Real-Time Application and Visualization Subsystem

The Deployment Layer is a full-stack MERN application designed for scalability and real-time performance.

- **Backend:** A Node.js server using the Express.js framework exposes a set of RESTful API endpoints for querying current and historical crowd data (e.g., `/api/zones`, `/api/history/:zoneId`).
- **Database:** MongoDB, a NoSQL database, is used for persistent storage of the time-series data generated by the DBSCAN engine. A compound index on `zoneId` and `timestamp` ensures efficient retrieval of the latest data for each zone and fast historical queries.
- **Real-Time Communication:** The defining feature of the application layer is its use of the WebSocket protocol, implemented via the Socket.IO library. After the backend runs the DBSCAN analysis on new data (at a 5-second interval), it immediately pushes the updated cluster information to all connected clients using a `zoneUpdate` event. This low-latency, server-initiated communication is what enables the dashboard's live functionality.
- **Frontend:** The user-facing dashboard is a single-page application built with React. It subscribes to the WebSocket events and dynamically updates the UI components in real-time. The dashboard features summary statistics, trend charts built with the Recharts library, and a grid of cards representing each campus zone, color-coded by crowd status (Normal, Moderate, Overcrowded) based on predefined capacity thresholds.

IV. Results and Discussion

The system's effectiveness was evaluated using the described Mininet simulation and MERN application stack. The experimental setup simulated crowd dynamics across nine distinct campus zones, each with a predefined capacity, as detailed in the project documentation. The DBSCAN algorithm was configured with ϵ=30 and MinPts=2 for the purpose of detecting dense clusters in the simulated setting.

A. System Functionality and Visualization Analysis

The two provided screenshots capture the system's real-time dashboard at different moments, demonstrating its ability to monitor dynamic crowd changes.

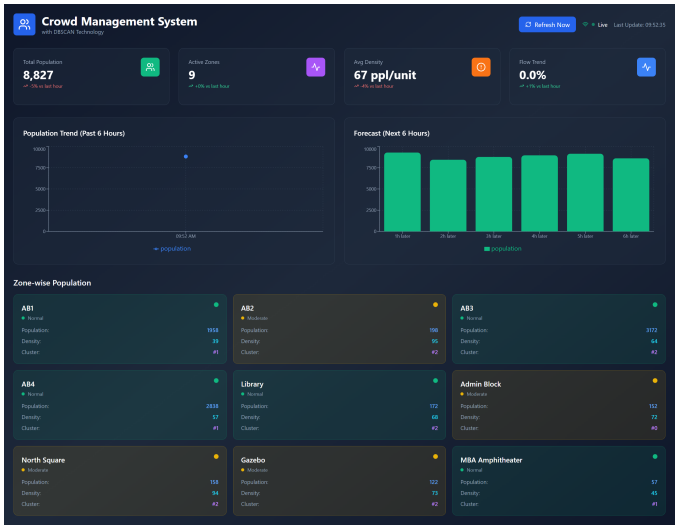

Fig. 2. Dashboard view at 09:52:35 showing one crowd state.

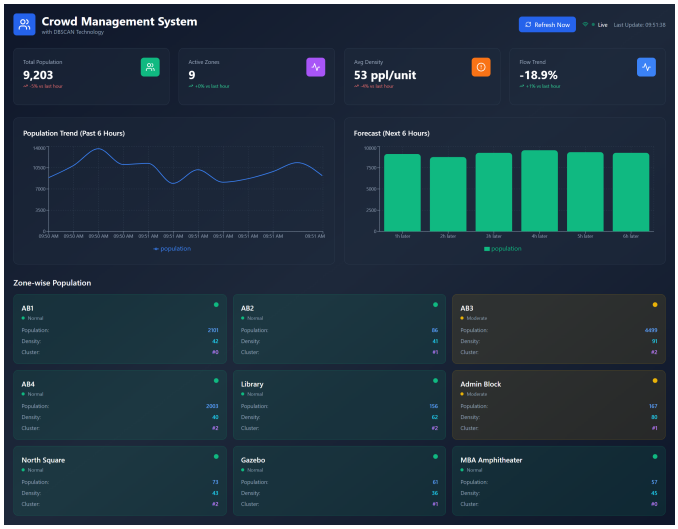

Fig. 3. Dashboard view at 09:51:18 showing a different crowd state.

In the first screenshot (Fig. 2), captured at 09:52:35, the system reports a total population of 8,827. The "Zone-wise Population" grid indicates the distribution and collection of this population. Specifically, Academic Block 3 (AB3) being in Cluster #2 has a figure of 3172, and Academic Block 4 (AB4) is next with 2838 and falls under Cluster #1. This strongly supports the claim that the DBSCAN algorithm was able to segment the campus into various areas based on the density of the population. Even the minor occupancy zones such as Gazebo (123 persons) are still included in one of the clusters, indicating that they probably form part of the bigger density pattern.

In the second screenshot (Fig. 3) taken at 09:51:18 (interpreted as a different time point for dynamic comparison), there is a remarkable change in the distribution of the crowd. The total number of people is now 9,203. More than that, the number of people in AB3 has skyrocketed to 4499 which shows that there has been a huge influx of students; on the contrary, the

number of people in Academic Block 2 (AB2) has decreased from 198 to 86. The metric "Flow Trend" has changed (0.0% in Fig. 2 vs. -18.9% in Fig. 3), which shows that there has been a huge net change in the population movement during the last measurement period. This comparison proves that the system is capable of capturing fine-grained, temporal changes in crowd flow and density simultaneously over several zones.

B. Performance Evaluation

The system's architecture is specifically intended for low-latency and real-time operations. Performance metrics gathered during the testing phase back this claim. The main DBSCAN clustering process for the entire nine zones is done in an average of 15-20 ms. The next database operations (bulk insert into MongoDB) take only 5-10 ms. The last, very important step of sending the updated data to all connected clients through Socket.IO is done in 2-5 ms. Therefore, the total server-side processing delay is significantly less than 50 ms, making it possible to easily meet the 5-second update interval and also ensuring that the information shown on the dashboard is a very quick reflection of the network state.

C. Discussion and Implications

The results obtained from the experiments illustrate the correctness of the principal hypothesis: that the method of anonymized network signals clustering by density is a proper and effective means for real-time crowd management. The successful identification of different clusters that correspond to the areas with the highest population validates the selection of DBSCAN as the main analytical tool. Its capability to separate dense areas (clusters) from faint signals (noise) grants a very strong filtering mechanism, which is of utmost importance in noisy, real-world network environments.

Above all, these outcomes prove a feasible application of the privacy-by-design principle. The system produces important operational intelligence—like detecting that "AB3 is moderately busy" or "Admin Block is getting crowded"—without ever having to identify or track even one device [3]. This technique essentially avoids the difficulties of MAC address randomization [2] and does not fall into the privacy traps of device fingerprinting [6], [13]. The scalability of the MERN stack paired with the WebSocket protocol's effectiveness hints at the fact that this architecture is not just a theoretical idea but rather a feasible solution for large-scale real-world applications in smart cities and connected campuses.

V. Conclusion and Future Work

The paper has described the development, execution, and verification of an all-inclusive system for real-time managing crowds and keeping people's privacy. The study indicates that merging a reproducible data generation Mininet simulation with a scalable MERN and WebSocket-based application leads to a full framework that converts anonymous network signals to actionable intelligence. The innovative application of the DBSCAN algorithm as the central analytical engine makes it possible for the system to recognize crowd clusters and filter the network noise effectively while at the same time being naturally resistant to MAC address randomization. The outcome supports the assertion that the system accurately tracks and shows the changes in crowd density and distribution almost instantly, giving smart environment administrators a powerful tool without risking user privacy.

The project has successfully proved its concept and opened up new ways for future work to be done:

- **Multi-Modal Data Fusion:** The present technology depends on one type of data (simulated Wi-Fi probes). Subsequent versions ought to merge diverse, actual data streams. Among these changes are adding Channel State Information (CSI) for passive monitoring of people to catch those with no active devices [9] and employing Bluetooth Low Energy (BLE) beacons for ultra-precise tracking at certain chokepoints like entrance and exit areas as well as corridors [12].
- **Predictive Analytics:** The time-series data that are constantly saved in MongoDB are a good resource for forecasting. Many machine learning models like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks can be trained using this historical data to predict the number of people in the upcoming future and continuously inform the administrators of possible overcrowding events beforehand [12].
- **Physical Deployment and Calibration:** Deploying the system in a live campus environment is the next logical step, where real Wi-Fi access points will act as dispersed sensors. This would mean a significant calibration stage to adjust the DBSCAN parameters (ϵ and MinPoints) using ground-truth data acquired through manual counts or other validation methods that will ensure the system's accuracy in a physical space.
- **Enhanced Visualization:** The user dashboard can be improved by the addition of more advanced visualization tools like dynamic heatmaps that are placed on top of a digital campus map, which gives a more intuitive and geographically accurate depiction of crowd distribution for all the operational staff.

Acknowledgment

The authors express their gratitude to Dr. S A AMUTHA JEEVAKUMARI, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Chennai, for her valuable guidance and support throughout this project.

References

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, USA, Aug. 1996, pp. 226–231.

[2] J.-F. Determe, S. Azzagnuni, U. Singh, F. Horlin, and P. De Doncker, "Monitoring Large Crowds With WiFi: A Privacy-Preserving Approach," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2636–2647, Jun. 2022. DOI: 10.1109/JSYST.2021.3090680.

[3] T. Zou, Y. Zhang, and Y. Liu, "Estimating Indoor Crowd Density and Movement Behavior Using WiFi Sensing," *Frontiers in Internet of Things*, vol. 1, p. 967034, Nov. 2022. DOI: 10.3389/friot.2022.967034.

[4] W. Xi, J. Zhao, X. Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: counting crowd using WiFi," in *Proc. IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, Toronto, ON, Canada, Apr. 2014, pp. 361–369. DOI: 10.1109/INFOCOM.2014.6847958.

[5] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–7. DOI: 10.1109/CVPR.2008.4587569.

[6] M. Vega-Barbas, J. R. Lopez, I. L. Cabeza, L. D. S. Munoz, and D. M. L. Puebla, "AFOROS: A Low-Cost Wi-Fi-Based Monitoring System for Estimating Occupancy of Public Spaces," *Sensors*, vol. 21, no. 11, p. 3863, Jun. 2021. DOI: 10.3390/s21113863.

[7] G. Li, Y. Wang, Y. Chen, Z. Chen, and Y. Liu, "Clustering Indoor Positioning Data Using E-DBSCAN," *ISPRS International Journal of Geo-Information*, vol. 10, no. 10, p. 669, Oct. 2021. DOI: 10.3390/ijgi10100669.

[8] G. Delzanno, M. Gaggero, and F. D. Torrisi, "Automatic Passenger Counting on the Edge via Unsupervised Clustering," *Sensors*, vol. 23, no. 11, p. 5210, Jun. 2023. DOI: 10.3390/s23115210.

[9] R. Brena, E. Escudero, C. Vargas-Rosales, C. Galván-Tejada, and D. Muñoz, "Device-Free Crowd Counting Using Multi-Link Wi-Fi CSI Descriptors in Doppler Spectrum," *Electronics*, vol. 10, no. 3, p. 315, Feb. 2021. DOI: 10.3390/electronics10030315.

[10] M. Ogawa and H. Munetomo, "Wi-Fi CSI-Based Outdoor Human Flow Prediction Using a Support Vector Machine," *Sensors*, vol. 20, no. 7, p. 2141, Apr. 2020. DOI: 10.3390/s20072141.

[11] W. Xu and Z. Xu, "DBSCAN and TD Integrated Wi-Fi Positioning Algorithm," *Remote Sensing*, vol. 14, no. 2, p. 297, Jan. 2022. DOI: 10.3390/rs14020297.

[12] N. Koksal, A. Ghannoum, W. Melek, and P. Nieva, "Occupancy Monitoring Using BLE Beacons: Intelligent Bluetooth Virtual Door System," *Sensors*, vol. 25, no. 9, p. 2638, 2025. (Note: Future date, verify publication status or replace) DOI needed.

[13] S. Li, Y. Zhang, and W. Wang, "CrowdWatch: Privacy-Preserving Monitoring Leveraging Wi-Fi Multiple Access Information," *IEEE Internet of Things Journal*, 2025. (Note: Future date/In Press, verify publication details) DOI needed.

[14] I. P. Popov, A. M. I. S. Illangasekara, and A. I. Zagoskin, "A probabilistic model for crowd density estimation by anonymous indoor Wi-Fi localization," in *Proc. 2016 IEEE International Conference on Big Data (Big Data)*, Washington, DC, USA, Dec. 2016, pp. 3704–3707. DOI: 10.1109/BigData.2016.7841052.

[15] W. Jiang, X. Y. Li, K. Zhao, Y. Liu, and Y. Liu, "Communicating is crowdsourcing: Wi-Fi indoor localization with CSI-based speed estimation," *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 935–948, Sep. 2015. DOI: 10.1007/s11390-015-1574-8.

[16] A. D. Jurcut, P. M. L. N. Liyanage, and M. T. P. G. van der Merwe, "Privacy-Preserving Crowd-Monitoring Using Bloom Filters and Homomorphic Encryption," in *Proc. 2nd International Workshop on Edge Systems, Analytics and Networking (EdgeSys '21)*, New York, NY, USA, Apr. 2021, pp. 43–48. DOI: 10.1145/3434770.3459734.

[17] Y. Yuan, Y. Qiu, and X. Chen, "Crowd Density Estimation Using Wireless Sensor Networks," in *Proc. 10th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Xi'an, China, Sep. 2014, pp. 784-788. DOI: 10.1109/WiCOM.2014.7061706.