RSSI-Based Passenger Movement Classification for Non-Intrusive Public Transport Monitoring

Author One*, Author Two†, Author Three‡ and Author Four§
Department of Whatever, Whichever University
Wherever
Email: *author.one@add.on.net, †author.two@add.on.net, ‡author.three@add.on.net, §author.four@add.on.net

***Abstract*—Accurate monitoring of passenger flows in public transport is essential for service optimization and network planning, yet traditional counting methods face limitations in coverage, cost, and privacy preservation. This paper presents a novel approach for classifying passenger movements using temporal sequences of Wi-Fi Received Signal Strength Indicator (RSSI) measurements. We introduce a dataset collected in a controlled experimental environment that simulates public transport scenarios, capturing the distinctive signal patterns associated with four fundamental movement classes: boarding the vehicle, alighting from the vehicle, remaining inside, and remaining at the bus stop. By analyzing the temporal evolution of RSSI values over ten-second observation windows, our approach enables non-intrusive distinction between static and transitional states without requiring specialized hardware or compromising passenger anonymity. Experimental evaluation using multiple machine learning classifiers demonstrates the feasibility of RSSI-based movement classification, providing a cost-effective complement to existing automatic passenger counting systems for intelligent transportation applications.**

***Index Terms*—Passenger counting, RSSI fingerprinting, Wi-Fi sensing, public transport, machine learning, intelligent transportation systems, urban mobility**

I. Introduction

Accurate passenger flow data is essential for public transport optimization, yet traditional counting approaches—manual surveys, ticketing systems, infrared sensors, and Automatic Passenger Counting (APC) devices—suffer from high costs, incomplete coverage, and privacy concerns [1]. The ubiquity of Wi-Fi-enabled devices offers new avenues for non-intrusive sensing: Received Signal Strength Indicator (RSSI) signatures enable movement pattern recognition while preserving anonymity, using only standard networking equipment [2]–[4].

This paper proposes a passenger boarding and alighting classification framework based on temporal RSSI sequences. By tracking signal evolution over ten-second observation windows, the methodology distinguishes four movement patterns: remaining inside the vehicle, remaining at the stop, boarding, and alighting. Without requiring precise localization or specialized hardware. The principal contributions are:

1) An RSSI-based movement classification framework exploiting temporal signal evolution;
2) An experimental dataset comprising approximately 1,360 labelled samples across four classes, collected from four smartphone models spanning three brands;
3) A comprehensive evaluation of 38 machine learning classifiers with Bayesian hyperparameter optimization;
4) Feature importance analysis revealing the discriminative role of initial and mid-trajectory RSSI measurements;
5) A privacy-preserving sensing approach that operates without device identification.

II. Related Work

A. Automatic Passenger Counting

Conventional APC systems rely on infrared sensors, pressure mats, or video-based detection [5], yet claimed accuracies of 98% frequently drop to 53–74% under real-world conditions [1]. Vision-based approaches can reach 94% accuracy [6] but remain hampered by occlusion, lighting variability, and privacy concerns.

B. Wi-Fi-Based Sensing and Research Gap

Wi-Fi-based passenger counting has been demonstrated with up to 100% detection in static settings [2]. While Channel State Information (CSI)-based systems yield richer information with over 94% accuracy [7], they require specialized hardware. RSSI-based methods remain practical with off-the-shelf equipment; Simončič et al. [3] attained over 96% accuracy for presence monitoring despite Media Access Control (MAC) randomization challenges.

Despite this progress, machine learning classification of movement patterns from RSSI time series has received comparatively little attention. Existing work treats RSSI primarily as static fingerprints for indoor localization [4] rather than as temporal sequences encoding movement dynamics. Our work addresses this gap by casting passenger movement detection as a supervised classification problem over temporal RSSI evolution, enabling real-time trajectory classification with a single access point at the vehicle door.

III. Experimental Setup

This section outlines the physical data collection environment and the machine learning experimental framework devised for passenger movement classification.

A. Physical Data Collection Setup

Controlled experiments were carried out in an indoor environment that emulates public transport interactions, allowing reproducible data collection under both isolated and noisy conditions.

1) Environmental Configuration: The environment was divided into two zones (Figure 1). Zone A (Vehicle Interior) consisted of a closed room simulating a bus interior, with a Wi-Fi access point positioned adjacent to the doorway. Zone B (Bus Stop) was the corridor outside, representing the boarding area. The wall and door between zones introduce signal attenuation, generating distinctive RSSI patterns during transitions.

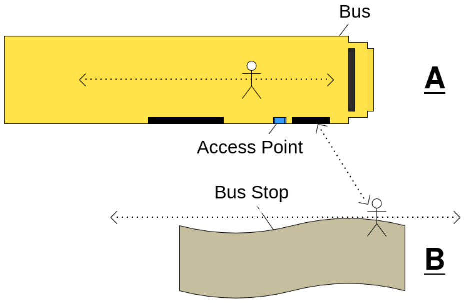

Figure 1. Experimental environment simulating a public transport scenario.

2) Data Acquisition: Data was collected using a Python script interfacing with the Access Point via Ethernet. Each trial comprised a **10-second window** with **10 RSSI samples at 1 Hz**. Devices maintained periodic low-overhead traffic (ICMP) to ensure consistent RSSI reporting. Four smartphones from three brands were employed to introduce hardware and chipset variability: Samsung Galaxy S20 (Android 13), Samsung Galaxy S23 (Android 14), POCO X7 Pro (Android 14), and Xiaomi Redmi 4 (Android 6). This selection spans a wide range of Wi-Fi chipset generations and transmit power characteristics, improving the generalizability of trained models.

Four movement classes were defined: **A→A** (remaining inside), **B→B** (remaining at stop), **A→B** (alighting), and **B→A** (boarding). Collection occurred under two scenarios: *Isolated* (single device, 20 repetitions per class per device) and *Noisy* (four devices simultaneously performing paired movements).

3) Preprocessing and Dataset Structure: Raw data underwent the following preprocessing steps: (1) *Temporal Aggregation*, where 10 RSSI measurements per trial were aggregated into a feature vector $\mathbf{R} = [r_1, \ldots, r_{10}]$; (2) *Device Isolation* using MAC addresses; (3) *Labeling* with movement class and noise indicator; and (4) *Feature Filtering* to retain only RSSI values.

Device isolation relies on MAC addresses reported by the access point for associated clients. Although modern mobile operating systems randomize MAC addresses during probe requests, this randomization does not affect our methodology: once a device establishes a Wi-Fi association, it retains a consistent MAC address for the duration of the session. Since each 10-second observation window operates well within a single association period, device identification remains stable throughout the measurement interval.

Prior to model training, *Feature Scaling* was applied using standardization (z-score normalization), which transforms each feature to zero mean and unit variance. This step is essential for classifiers sensitive to feature magnitudes, such as Support Vector Machines (SVMs), Gaussian Process (GP) classifiers, Multi-Layer Perceptrons (MLPs), and logistic regression, as these distance-based and gradient-based methods may otherwise suffer from degraded convergence speed and classification performance when operating on raw RSSI value ranges.

The resulting dataset contains 1,356 samples, each representing a 10-second trajectory with a corresponding movement class label, enabling analysis of both absolute signal strength and temporal evolution.

4) Experimental Scenarios: To comprehensively evaluate the impact of data collection conditions on classification performance, three distinct experimental scenarios were defined:

Combined Dataset: The complete dataset containing all 1,356 samples from both isolated and noisy collection conditions. This scenario represents the most realistic deployment setting, where the classifier must generalize across varying environmental conditions.

Isolated-Only Dataset: A subset containing exclusively samples collected under isolated conditions (single device, $n = 160$). This scenario provides an upper-bound estimate of classification performance under ideal conditions with minimal signal interference.

Noisy-Only Dataset: A subset containing exclusively samples collected with simultaneous multi-device activity ($n = 1,196$). This scenario evaluates classifier robustness under challenging conditions that closely approximate real-world public transport environments.

Comparing across these scenarios allows for a quantitative assessment of how environmental noise affects classification accuracy and sheds light on the operational boundaries of RSSI-based movement detection.

B. Machine Learning Experimental Framework

The following subsection details the classifier selection rationale, evaluation methodology, and performance metrics adopted in our experimental protocol.

1) Classifier Selection: A total of 38 classification algorithms spanning multiple paradigms were evaluated to ensure thorough benchmarking. The classifier families were chosen based on their established effectiveness in RSSI-based classification tasks [8], [9]:

Support Vector Machines (SVM): SVMs with Radial Basis Function (RBF) and linear kernels were included due to their demonstrated superiority in Wi-Fi fingerprinting tasks. Prior studies on indoor localization using RSSI have shown SVMs achieving accuracies exceeding 90% for location classification [4], [10]. The RBF kernel effectively captures nonlinear relationships in signal strength patterns.

Ensemble Methods: Random Forest and Extra Trees were selected for their robustness to noise and ability to model complex decision boundaries without extensive hyperparameter tuning [8]. Gradient boosting variants (XGBoost, LightGBM, CatBoost) were included based on their state-of-the-art performance in tabular classification tasks, with CatBoost demonstrating particular effectiveness for categorical features [11].

Gaussian Process Classifier: GPs provide probabilistic predictions with uncertainty quantification, particularly valuable for RSSI data where signal variability is inherent. The RBF kernel enables automatic adaptation to the intrinsic dimensionality of temporal RSSI sequences.

Neural Networks: MLPs with varying architectures (small, medium, large) were evaluated to assess whether deeper representations improve classification over traditional methods for this feature space dimensionality.

Stacking and Voting Ensembles: Meta-learning approaches combining heterogeneous base learners were included to leverage complementary classifier strengths, a strategy shown to improve robustness in transportation sensing applications [6].

2) Data Partitioning Strategy: The dataset was partitioned using **stratified sampling** with an 80%/20% train-test split. Stratified sampling ensures that class distributions are preserved in both partitions, which is critical for multi-class classification problems where class imbalance could otherwise bias model evaluation. This approach maintains the original proportion of each movement class (AA, BB, AB, BA) in both training and testing sets.

3) Cross-Validation Protocol: Model training employed **5-fold stratified cross-validation**, a methodology widely recommended for robust classifier evaluation. Stratified K-fold cross-validation maintains class ratios across all folds, ensuring that minority classes receive adequate representation during training and validation. This technique reduces variance in performance estimates compared to simple hold-out validation.

To assess result stability, experiments were repeated with three random seeds (3, 5, and 42), and metrics were aggregated across runs. This multi-seed evaluation quantifies classifier sensitivity to random initialization and data shuffling.

4) Hyperparameter Optimization: To ensure fair comparison and optimal performance across classifier families, systematic hyperparameter optimization was conducted using Optuna [12], a state-of-the-art Bayesian optimization framework. Optuna employs the Tree-structured Parzen Estimator (TPE) algorithm to efficiently explore high-dimensional hyperparameter spaces, focusing search efforts on promising regions.

For each tunable classifier, extensive hyperparameter searches were performed with budgets ranging from 50 to over 1,300 trials depending on model complexity. The optimization objective was accuracy during 5-fold stratified cross-validation on the training set. This approach yields classifier configurations specifically adapted to the RSSI feature space characteristics.

The hyperparameter search spaces were defined based on recommended ranges from the literature and practical considerations. Ensemble methods (Random Forest, Extra Trees, Gradient Boosting variants) were tuned over tree depth, number of estimators, and regularization parameters. SVMs were optimized for the regularization parameter C and kernel coefficients. Neural network configurations explored layer architectures, learning rates, and regularization strengths. The complete hyperparameter search spaces and optimal configurations for each classifier are documented in the Appendix A.

5) Performance Metrics: Four complementary metrics were employed to provide comprehensive performance characterization:

Accuracy: The proportion of correctly classified samples. While intuitive, accuracy can be misleading for imbalanced datasets.

Weighted F1-Score: The harmonic mean of precision and recall, weighted by class support. This metric balances false positives and false negatives while accounting for class distribution.

Balanced Accuracy: The arithmetic mean of per-class recall values, ensuring equal contribution from each class regardless of prevalence.

Matthews Correlation Coefficient (MCC): Selected as the primary evaluation criterion, as this metric offers a reliable and balanced evaluation of classification models, particularly in scenarios involving imbalanced datasets or when assessing performance across multiple classes [13]. Matthews Correlation Coefficient (MCC) ranges from -1 to $+1$, producing high scores only when all confusion matrix categories achieve strong results.

IV. Exploratory Data Analysis

Ahead of classifier training, an exploratory data analysis was carried out to gauge the discriminative potential of temporal RSSI signatures and to characterize signal behaviour across different passenger movement classes.

A. Dataset Composition

The final dataset is roughly balanced, comprising around 340 samples per movement class. For each class, 40 samples were gathered under isolated conditions, while the remaining samples were obtained during simultaneous device activity, thereby introducing controlled signal interference. This balance ensures that the exploratory analysis and subsequent results reflect both ideal and realistic operating conditions.

B. Temporal RSSI Characteristics

The temporal evolution of RSSI values serves as the primary discriminative feature between movement classes. Figure 2 depicts the mean RSSI trajectory over the 10-second observation window for each class.

Static states (**AA and BB**) exhibit relatively stable signal levels over time, albeit with distinct average magnitudes due to their spatial separation from the access point. In contrast, transitional movements display clear monotonic trends. The boarding class (**B** $\rightarrow$ **A**) shows a consistent increase in RSSI as the devices move toward the access point, while the alighting class (**A** $\rightarrow$ **B**) presents a pronounced decrease as physical obstructions attenuate the signal.

These contrasting temporal patterns offer a compelling rationale for leveraging RSSI sequences in movement classification.

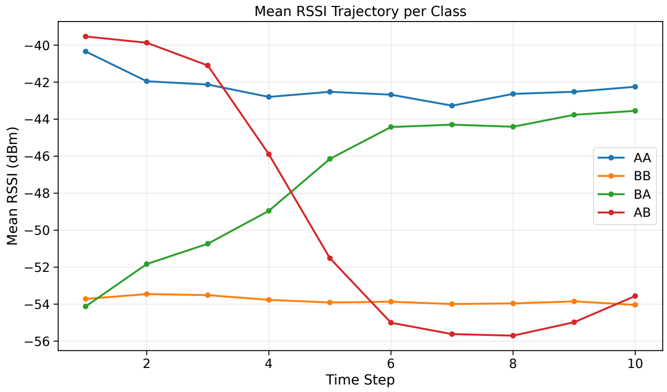

Figure 2. Temporal evolution of RSSI values over the 10-second observation window for each movement class.

V. Results and Discussion

This section presents the experimental results from evaluating 38 machine learning classifiers on the RSSI-based passenger movement dataset and analyses their implications across three experimental scenarios.

A. Comparative Analysis Across Scenarios

Table I presents the performance comparison of top classifiers across the three experimental scenarios, revealing the substantial impact of data collection conditions on classification performance.

Table I
Performance Comparison Across Experimental Scenarios (MCC)

Classifier	Combined	Isolated-Only	Noisy-Only
KNN (k=5)	0.692 ± 0.029	**0.907 ± 0.061**	0.704 ± 0.025
KNN (k=3)	0.690 ± 0.046	0.882 ± 0.068	0.702 ± 0.043
LinearSVC	0.726 ± 0.039	0.867 ± 0.060	0.716 ± 0.031
LogisticRegression (L2)	0.743 ± 0.044	0.866 ± 0.028	0.731 ± 0.009
SVC (Linear)	0.750 ± 0.023	0.850 ± 0.088	0.731 ± 0.037
StackingEnsemble	0.749 ± 0.020	0.851 ± 0.122	0.768 ± 0.023
ExtraTrees	0.737 ± 0.013	0.836 ± 0.041	0.755 ± 0.021
GaussianProcess	**0.756 ± 0.033**	0.414 ± 0.052	0.755 ± 0.028
SVC (RBF)	0.755 ± 0.021	0.825 ± 0.101	0.754 ± 0.016
CatBoost	0.746 ± 0.017	0.782 ± 0.063	**0.770 ± 0.013**

The isolated-only scenario yielded markedly superior performance, with K-Nearest Neighbors (KNN) (k=5) reaching an MCC of 0.907—a 31% relative improvement over the combined dataset. This increase stems from the absence of inter-device signal interference, producing cleaner RSSI patterns with more distinct class separations. Simpler classifiers such as KNN, which rely on local neighbourhood structure, benefit disproportionately from this increased separability.

Conversely, the GP classifier suffered notable degradation in the isolated scenario (MCC: 0.414) despite attaining the highest MCC (0.756) on the combined dataset. This paradox is explained by the limited sample size ($n = 160$), which proves insufficient for reliable kernel hyperparameter estimation. On the combined dataset, the GP's probabilistic framework and RBF kernel flexibility enable effective modelling of non-linear decision boundaries without extensive manual tuning.

The noisy-only scenario, representing the most realistic operational conditions, showed performance on par with the combined dataset, with CatBoost attaining the highest MCC (0.770). This consistency indicates that the combined dataset's performance is largely driven by the noisy samples, which constitute 88% of the total data. Gradient boosting methods demonstrated particular robustness to signal interference.

B. Per-Class and Error Analysis

Table II presents the per-class metrics for the best classifier (GP).

Table II
Per-Class Performance Metrics (GP)

Class	Accuracy	Recall	F1-Score	MCC
AA (Inside)	0.838	0.750	0.791	0.785
BB (Stop)	0.838	0.897	0.878	0.785
BA (Boarding)	0.838	0.824	0.855	0.785
AB (Alighting)	0.838	0.882	0.828	0.785
Weighted Avg	**0.838**	**0.838**	**0.838**	**0.785**

The bus stop state (BB) achieved the highest recall (89.7%) and F1-score (0.878), attributable to the consistent low RSSI values produced by the physical barrier separating Zone B from the access point. The boarding movement (BA) achieved strong results (recall: 82.4%, F1-score: 0.855), benefiting from its distinctive increasing RSSI pattern. The alighting class (AB) demonstrated high recall (88.2%) but lower F1-score (0.828), indicating some false positives from the AA class.

The static state inside the vehicle (AA) presented the lowest recall (75.0%), primarily due to misclassification as boarding (BA). This confusion arises from the spatial proximity of both classes to the access point, producing similar high-RSSI signatures. Although their temporal dynamics differ, AA maintains relatively stable values while BA exhibits an increasing trend, this distinction may be subtle within the 10-second observation window. The overall accuracy of 83.8% and MCC of 0.785 confirm robust multi-class discrimination.

C. Confusion Matrix Analysis

Figure 3 presents the normalized confusion matrix for the GP classifier.

The confusion matrix confirms that classification errors concentrate between spatially adjacent classes. The AA–BA confusion reflects RSSI magnitude similarity when devices are near the access point, while the AB–BB confusion arises from both classes sharing lower RSSI values characteristic of the exterior zone. These patterns suggest that additional features or longer observation windows could improve discrimination between static and transitional states.

D. Model Stability

Figure 4 illustrates the MCC variability for top classifiers across the three random seeds.

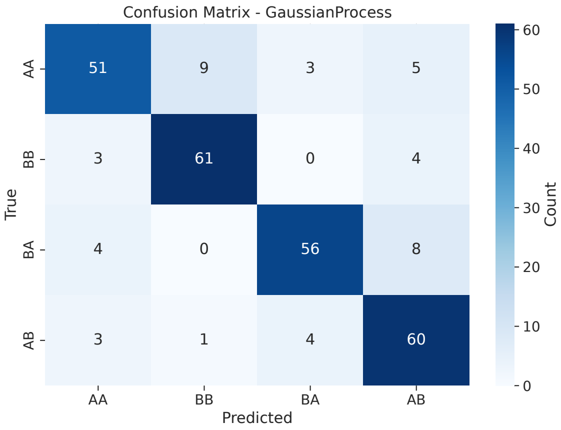

Figure 3. Confusion matrix for the GP classifier, demonstrating strong diagonal dominance with minimal inter-class confusion.

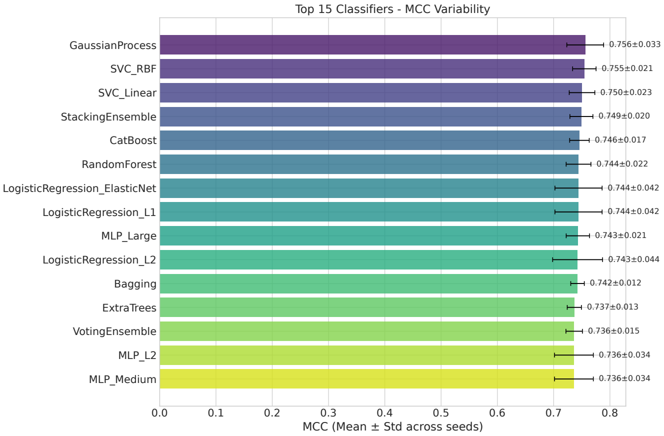

Figure 4. MCC variability for top classifiers, in combined dataset, demonstrating consistent ranking stability.

Top-performing classifiers maintain consistent relative rankings across seeds. Kernel-based methods (GP, SVC) and ensemble approaches (Stacking, CatBoost) displayed the lowest variability, a desirable property for deployment scenarios where model retraining may occur with different data partitions.

E. Feature Importance

Figure 5 illustrates the mean feature importance across interpretable classifiers.

Initial RSSI measurements (features 1–3) contribute most significantly to classification decisions, capturing the starting position and enabling immediate distinction between static and transitional states. Feature 1 exhibits the highest mean importance (normalized score: 0.99) with universal agreement among classifiers (standard deviation: 0.009). The elevated importance of feature 6 (mid-trajectory) indicates that classifiers also rely on signal evolution to confirm movement direction, validating the use of sequential RSSI measurements over aggregate statistics.

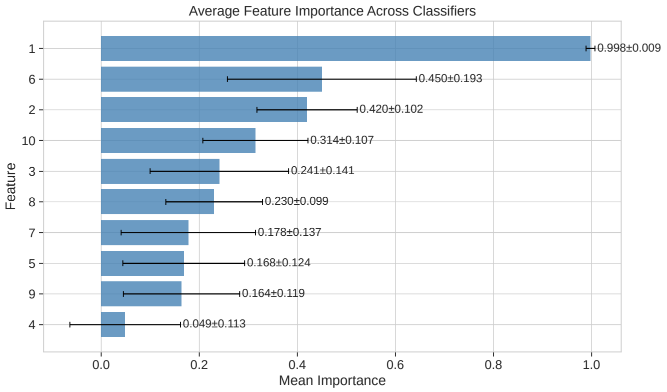

Figure 5. Mean feature importance across classifiers, with standard deviation, to the $[0, 1]$ range using min-max normalization

F. Hyperparameter Configuration

While extensive hyperparameter optimization was performed for nine classifier families using Optuna (detailed in section A), the GP classifier employed default configurations with a RBF kernel. The GP configuration is presented in Table III.

Table III
GP HYPERPARAMETERS

Parameter	Value
Kernel	1.0 × RBF(1.0)
Kernel Length Scale	Optimized during fitting
Optimizer	L-BFGS-B
Max Iterations	100
Multi-class Strategy	One-vs-Rest

The RBF kernel automatically learns optimal length scale parameters during training, adapting to the intrinsic dimensionality of RSSI sequences. Notably, the GP achieved top performance without requiring the extensive hyperparameter search applied to other classifiers, suggesting that its probabilistic formulation is well-suited to the RSSI feature space.

G. Deployment Considerations

The experimental results inform practical classifier selection. For high-interference environments with multiple simultaneous devices, gradient boosting methods (CatBoost, XGBoost) and ensemble approaches offer the best balance of accuracy and robustness. In controlled settings with minimal device density, simpler classifiers such as KNN achieve superior performance with reduced computational overhead. For general deployment across varying conditions, SVC with RBF kernel provides consistent performance with acceptable variance.

The notable performance improvement in isolated conditions (MCC up to 0.907) indicates that signal interference is the chief limiting factor. Deployment strategies that mitigate interference, like dedicated frequency channels or directional antennas, could markedly enhance performance. Nevertheless, the results under noisy conditions (MCC > 0.77) confirm that RSSI-based movement classification remains viable as a complementary passenger counting technology even in challenging environments.

H. Limitations

Several limitations warrant discussion. The controlled experimental environment, while designed to simulate public transport conditions, may not capture all sources of variability present in operational settings, such as passenger density fluctuations, vehicle movement, and diverse access point placements. The 10-second observation window, while suitable for typical boarding and alighting actions, may be insufficient for slower movements; adaptive window lengths could improve accuracy. The limited isolated dataset size ($n = 160$) further constrains reliability estimates for complex classifiers in that scenario.

Regarding energy consumption, the proposed approach imposes minimal overhead on passenger devices, as it leverages existing Wi-Fi associations without requiring additional active scanning. The access point performs passive RSSI sampling, introducing no extra power drain on user devices. However, maintaining continuous Wi-Fi connectivity may affect battery life on devices with aggressive power-saving policies. On the infrastructure side, the access point's processing and energy overhead, while generally modest, may increase with higher device densities and more computationally demanding classification models, necessitating efficient implementation for real-time inference. Furthermore, certain router operating systems may enter power-saving modes that could disrupt continuous RSSI monitoring; careful firmware configuration is therefore required to ensure uninterrupted data collection. A quantitative characterization of energy impact, on both client devices and access point hardware, across different deployment scales remains an avenue for future investigation.

Modern mobile operating systems employ MAC address randomization during Wi-Fi probe requests, which can complicate device tracking in passive sensing scenarios. Our approach mitigates this challenge by relying on established Wi-Fi associations rather than passive scanning: once a device associates with the access point, it maintains a consistent MAC address throughout the session, and the 10-second observation window operates well within a single association period, ensuring stable device identification. Nevertheless, devices that do not actively connect to the access point (or that cycle their MAC address between association events) remain untrackable, imposing a practical ceiling on detection coverage. Variations in device hardware and operating system behaviour may further affect RSSI reporting consistency in practice.

VI. Conclusions

This paper has presented a new approach for classifying passenger movements in public transport using temporal sequences of Wi-Fi RSSI measurements. The proposed methodology enables non-intrusive distinction between four fundamental movement classes using only standard Wi-Fi access point infrastructure.

The GP classifier attained the highest performance on the combined dataset, with an MCC of 0.756, validated across multiple random seeds. SVMs and regularized logistic regression variants yielded comparable results, confirming that both kernel-based and linear methods can effectively exploit the temporal structure of RSSI sequences.

Per-class analysis revealed that static states at the bus stop and transitional movements are more readily distinguished owing to their characteristic signal patterns, whereas the static state inside the vehicle posed the greatest classification challenge because of its proximity-based similarity with the boarding class.

The results demonstrate that RSSI-based passenger movement classification offers a viable, cost-effective, and privacy-preserving complement to existing APC technologies, requiring no specialized hardware beyond standard Wi-Fi infrastructure.

Future work includes validation in operational public transport environments, integration of complementary sensor modalities, development of adaptive observation windows, and extension of the methodology to estimate complete Origin-Destination (OD) matrices through temporal aggregation of boarding and alighting events.

References

[1] C. Pronello and X. R. Garzón Ruiz, "Evaluating the performance of video-based automated passenger counting systems in real-world conditions: A comparative study," *Sensors*, vol. 23, no. 18, p. 7719, 2023.

[2] M. Nitti, F. Pinna, L. Pintor, V. Pilloni, and B. Barabino, "iabacus: A wi-fi-based automatic bus passenger counting system," *Energies*, vol. 13, no. 6, p. 1446, 2020.

[3] A. Simoni, M. Mohori, and A. Hrovat, "Non-intrusive privacy-preserving approach for presence monitoring based on wifi probe requests," *Sensors*, vol. 23, no. 5, p. 2588, 2023.

[4] C. Agualimpia-Arriaga, S. Govindasamy, B. Soni, C. Páez-Rueda, and A. Fajardo, "Rssi-based indoor localization using machine learning for wireless sensor networks: A recent review," in *IEEE ANDESCON 2024*, 2024.

[5] B. Barabino, M. Di Francesco, and S. Mozzoni, "An offline framework for handling automatic passenger counting raw data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2443–2456, 2014.

[6] C. Wiboonsiriruk, E. Phaisangittisagul, C. Srisurangkul, and I. Kumazawa, "Efficient passenger counting in public transport based on machine learning," in *IEEE TENCON 2023*, 2023.

[7] J. Guo, W. Zhuang, Y. Mao, and I. Ho, "Rssi-assisted csi-based passenger counting with multiple wi-fi receivers," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2025.

[8] N. Singh, S. Choe, and R. Punmiya, "Machine learning based indoor localization using wi-fi rssi fingerprints: An overview," *IEEE Access*, vol. 9, pp. 127 150–127 174, 2021.

[9] B. Zholamanov, A. Saymbetov, M. Nurgaliyev, A. Bolatbek, G. Dosymbetova, N. Kuttybay, S. Orynbassar, A. Kapparova, N. Koshkarbay, and Ö. F. Beyca, "Rssi fingerprint-based indoor localization solutions using machine learning algorithms: A comprehensive review," *Smart Cities*, vol. 8, no. 5, p. 153, 2025.

[10] C. Jain, G. V. S. Sashank, V. N, and S. Markkandan, "Low-cost ble based indoor localization using rssi fingerprinting and machine learning," in *IEEE WiSPNET 2021*, 2021.

[11] L. Fabre, C. Bayart, Y. Kone, O. Manout, and P. Bonnel, "A machine learning approach to estimate public transport ridership using wi-fi data," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 2623–2631.

[13] D. Chicco and G. Jurman, “The matthews correlation coefficient (mcc) should replace the roc auc as the standard metric for assessing binary classification,” *BioData Mining*, vol. 16, no. 1, p. 4, 2023.

Appendix

This appendix documents the hyperparameter optimization process conducted using Optuna [12], including the search spaces explored and the optimal configurations identified for each classifier family.

A. Optimization Methodology

The Tree-structured Parzen Estimator (TPE) sampler was employed with consistent random seeds to ensure reproducibility. Each classifier underwent extensive optimization with the number of trials shown in Table IV. The optimization objective was 5-fold stratified cross-validation accuracy on the training set.

Table IV
Optuna Hyperparameter Optimization Summary

Classifier	Trials	Best CV Accuracy
Random Forest	1,303	89.08%
Extra Trees	1,250	91.45%
Gradient Boosting	1,170	87.51%
XGBoost	1,150	87.51%
LightGBM	51	79.89%
SVC (RBF)	950	92.22%
KNN	950	85.88%
MLP	950	90.68%
Logistic Regression	900	85.97%

B. Optimal Hyperparameter Configurations

The following tables present the best hyperparameter configurations identified through the optimization process.

1) **Random Forest** (Table V)

Table V
Random Forest Best Hyperparameters

Parameter	Value
n_estimators	97
max_depth	13
min_samples_split	2
min_samples_leaf	1
max_features	sqrt
bootstrap	False

2) **Extra Trees** (Table VI)

Table VI
Extra Trees Best Hyperparameters

Parameter	Value
n_estimators	182
max_depth	9
min_samples_split	9
min_samples_leaf	1
max_features	sqrt

3) **Gradient Boosting** (Table VII)

Table VII
Gradient Boosting Best Hyperparameters

Parameter	Value
n_estimators	156
learning_rate	0.0616
max_depth	12
min_samples_split	4
min_samples_leaf	2
subsample	0.950

4) **XGBoost** (Table VIII)

Table VIII
XGBoost Best Hyperparameters

Parameter	Value
n_estimators	104
learning_rate	0.253
max_depth	13
min_child_weight	2
subsample	0.800
colsample_bytree	0.714
reg_alpha (L1)	2.66×10^{-6}
reg_lambda (L2)	1.368

5) **LightGBM** (Table IX)

Table IX
LightGBM Best Hyperparameters

Parameter	Value
n_estimators	83
learning_rate	0.0333
max_depth	8
num_leaves	25
min_child_samples	45
subsample	0.970
colsample_bytree	0.923
reg_alpha (L1)	4.15×10^{-5}
reg_lambda (L2)	1.02×10^{-7}

6) **SVC (RBF Kernel)** (Table X)

Table X
SVC (RBF Kernel) Best Hyperparameters

Parameter	Value
C (regularization)	0.634
gamma	scale
kernel	RBF

7) **K-Nearest Neighbors** (Table XI)

Table XI
KNN Best Hyperparameters

Parameter	Value
n_neighbors	4
weights	distance
metric	minkowski
p	5

8) **MLP Neural Network** (Table XII)

Table XII
MLP BEST HYPERPARAMETERS

Parameter	**Value**
hidden_layer_sizes	(249, 140, 95)
activation	tanh
alpha (L2 penalty)	0.000991
learning_rate	constant
learning_rate_init	0.00696
max_iter	500
early_stopping	True

9) **Logistic Regression** (Table XIII)

Table XIII
LOGISTIC REGRESSION BEST HYPERPARAMETERS

Parameter	**Value**
C (regularization)	36.81
l1_ratio	0.214
solver	saga
max_iter	1000

C. Search Space Definitions

Table XIV summarizes the hyperparameter search ranges explored during optimization.

Table XIV
HYPERPARAMETER SEARCH SPACE RANGES

Classifier	**Parameter**	**Range**	**Scale**
Tree Ensembles	n_estimators	[50, 300]	Linear
	max_depth	[3, 30]	Linear
	min_samples_split	[2, 20]	Linear
	min_samples_leaf	[1, 10]	Linear
	max_features	{sqrt, log2, None}	Categorical
Boosting Methods	learning_rate	[0.01, 0.3]	Log
	subsample	[0.6, 1.0]	Linear
	reg_alpha	[10^{-8}, 10]	Log
	reg_lambda	[10^{-8}, 10]	Log
SVC (RBF)	C	[0.01, 100]	Log
	gamma	{scale, auto}	Categorical
KNN	n_neighbors	[1, 20]	Linear
	weights	{uniform, distance}	Categorical
	metric	{euclidean, manhattan, minkowski}	Categorical
	p	[1, 5]	Linear
MLP	n_layers	[1, 3]	Linear
	n_units_per_layer	[32, 256]	Linear
	alpha	[10^{-5}, 0.1]	Log
	learning_rate_init	[10^{-4}, 0.1]	Log
	activation	{relu, tanh}	Categorical
Logistic Regression	C	[0.001, 100]	Log
	l1_ratio	[0, 1]	Linear