Enhancing Data Offloading in Urban Networks via Machine Learning-based Mobility Prediction

Ilaria Mangiacapra*, Paulo Martins Carvalho†, Antonio Montieri*,
Antonio Pescapè*, Emanuel Ribeiro Lima†, Solange Rito Lima†
*University of Napoli Federico II (Italy), †University of Minho (Portugal)
*{ilaria.mangiacapra, antonio.montieri, pescape}@unina.it, †{pmc, emanuellima, solange}@di.uminho.pt

***Abstract*—In the context of smart cities, mobile data offloading (i.e. redirecting network traffic from cellular infrastructure to alternatives such as Wi-Fi) is essential to ensure high-quality connectivity under growing urban demand. A key enabler of this strategy is the accurate prediction of *Offloading Regions* (ORs), i.e. geographic zones where users are likely to transfer data to alternative networks. This paper introduces a Machine Learning-based framework for predicting the next OR visited during user mobility. Using urban mobility traces from the city of Beijing, we develop novel spatio-temporal features and adopt a progressive evaluation strategy aimed at identifying the best-performing model under closed-world assumptions and assessing its robustness under open-world conditions. The approach enhances robustness to both dynamic mobility patterns and unseen ORs at inference time, leveraging an Open Set Recognition (OSR) mechanism. Our optimized framework achieves up to $+42\%$ accuracy gain in the closed-world scenario and $+16\%$ relative gain in AUC when evaluated in an open-world setting. These results show that the integration of feature engineering and OSR significantly enhances both predictive performance and generalization capabilities, paving the way for intelligent and adaptive data offloading in next-generation urban networks.**

***Index Terms*—Smart City, Offloading Regions, Mobility Prediction, Machine Learning, Open Set Recognition**

I. Introduction

The digital transformation of urban environments is increasingly driven by the convergence of enabling technologies, such as the Internet of Things, Artificial Intelligence (AI), and Big Data analytics. These technologies underpin the development of Smart Cities by supporting intelligent infrastructures that optimize urban mobility and improve the overall efficiency of transportation systems. One of the most pressing challenges in this context is the sustained growth of mobile data traffic, fueled by the proliferation of always-online applications running on handheld devices that rely on mobile connectivity. According to the latest Ericsson Mobility Report [1], global mobile data traffic has grown by 19% between Q1 2024 and Q1 2025 and is projected to reach 430 EB per month by 2030—approximately a 2.6× increase over current levels.

This steep surge places considerable strain on cellular networks, highlighting the need for scalable strategies that can sustain Quality of Experience (QoE) under high demand. A promising solution is *mobile data offloading*, which aims to alleviate cellular congestion by redirecting traffic to alternative infrastructures such as Wi-Fi access points. Indeed, the widespread deployment of Wi-Fi networks in homes, public spaces, and commercial buildings of Smart Cities has substantially increased Internet availability in densely populated urban areas, making these networks an attractive option for offloading mobile traffic. However, offloading decisions must often be made while users are in motion, where Wi-Fi connectivity is intermittent and cellular networks are the default. Simply deferring transmissions until a known Wi-Fi zone is reached may reduce load, but it often conflicts with application latency requirements. We therefore define an *Offloading Region (OR)* as a geographic area where user mobility conditions make offloading feasible [2]. Proactively *predicting the next ORs* enables timely and efficient offloading, improving both network utilization and user QoE.

To this end, we propose a *Machine Learning* (ML) framework that predicts ORs using *specifically designed spatio-temporal features* and integrates an *Open Set Recognition* (OSR) mechanism [3] to detect ORs unseen at training time, addressing the inherent variability of urban mobility scenarios. Our approach improves adaptability and robustness under realistic mobility dynamics, supporting intelligent offloading in next-generation wireless infrastructures for Smart Cities.

The main contributions of this paper are the following: (i) we formulate the prediction of next ORs as a classification task over urban mobility trajectories, and provide a structured methodology to address it; (ii) we engineer and analyze a set of spatio-temporal features, including both mobility-agnostic and mobility-aware descriptors, specifically tailored to improve OR classification; (iii) we conduct a systematic evaluation of several ML models using the Geolife dataset related to Beijing city, assessing predictive performance in both closed-world and open-world settings; (iv) we integrate a confidence-based OSR mechanism into the OR prediction workflow to detect ORs not seen during training of ML models and improve robustness to evolving mobility patterns.

The remainder of the manuscript is structured as follows: Section II discusses the main contributions in the literature related to urban mobility prediction and data offloading strategies. Section III details the adopted methodology, including the problem formulation and the proposed ML approach for OR prediction. Section IV presents the dataset and illustrates the feature extraction, preprocessing operations, and evaluation metrics adopted. Section V reports the experimental evaluation in both closed-world and open-world settings. Section VI concludes the paper and outlines directions for future work.

II. Related Work

Human mobility prediction has been widely explored, with applications ranging from location-based services to smart urban planning. Traditional approaches rely on spatiotemporal data (e.g., GPS traces, smartphone records, or credit card time-series) and include trajectory mining techniques [4, 5] and stochastic models like Markov Chains [6, 7] or Bayesian Networks [8]. More recent advances in ML and Deep Learning (DL) leverage ensemble methods [9] and Recurrent Neural Networks [10] to capture sequential mobility patterns. While effective in modeling static transitions, these approaches often lack adaptability to evolving user behavior and dynamic contexts such as mobile data offloading in Smart Cities.

The growing demand for mobile data traffic in smart urban environments [1] has driven the development of offloading solutions to alleviate network congestion, typically based on protocol-level or architectural approaches such as Wi-Fi offloading [11] or caching in mobile edge computing [12]. However, these solutions often lack integration with predictive mobility models that could enable proactive decision-making. Recent research has introduced AI-driven approaches for urban mobility optimization in Smart Cities [13], but few works explicitly target OR prediction. Clustering-based methods have been employed to define and detect ORs [14], but forecasting transitions between them remains largely underexplored, especially in realistic open-world conditions, modeling actual user mobility dynamics.

The work most closely related to ours is that in [2], which focuses primarily on OR extraction and characterization. While it provides a valuable feature-enriched dataset for downstream tasks, it only marginally explores predictive modeling. In contrast, our work places OR prediction at the core of the methodology, proposing a structured and systematic framework that combines feature engineering, hyperparameter tuning, and OSR to enhance both prediction accuracy and generalization in dynamic urban environments. Building upon expert-driven optimization in controlled settings, and inspired by OSR methods widely applied in computer vision and anomaly detection [3], we incorporate a confidence-based rejection mechanism that allows the model to differentiate between known and previously unseen ORs during inference. To the best of our knowledge, this multi-faceted capability has not yet been explored in the context of mobile data offloading. By moving beyond static, closed-world assumptions and embracing adaptive, uncertainty-aware OR prediction, our proposal enables more resilient and context-aware offloading strategies, better suited for the evolving demands of next-generation Smart City infrastructures.

III. Next OR Prediction Methodology

Herein, we outline the methodology adopted for predicting the next OR visited by a user. An OR is defined as a geographic area where the user remains within a limited spatial range for a sufficient duration, making the location suitable for mobile data offloading. Following the approach in [2], ORs are identified via DBSCAN clustering over GPS trajectories, using spatial-temporal constraints. The spatial threshold $S_{\text{th}} = 44\,\text{m}$ approximates reliable Wi-Fi coverage in urban environments (RSSI $> -85\,\text{dBm}$), while temporal thresholds $T_{\text{th}} \in \{40, 20, 10, 5\}\,\text{s}$ reflect varying offloading scenarios, from prolonged stops to short connectivity windows.

TABLE I

Optimal hyperparameters selected for each ML model.

Model	*Selected Hyperparameters*
LR	C=0.1
DT	min_samples_split=5, max_features='sqrt'

RF	n_estimators=1000, max_depth=12,
	class_weight='balanced', max_samples=0.8

GB	learning_rate=0.01, n_estimators=500,
	subsample=0.9, max_depth=5, min_samples_split=5,
	min_samples_leaf=5, max_features='sqrt'

MLP	hidden_layer_sizes=(50, 20), activation='tanh',
	alpha=1.0, learning_rate_init=0.0005, max_iter=5000

Only hyperparameters differing from the scikit-learn default [15] are reported. *GNB* has been used with all default hyperparameters.

We model the prediction of the next OR visited by a user as a *supervised multiclass classification* task. Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the space of input feature vectors extracted from user mobility traces (see Tab. II for the full feature set) and $\mathcal{Y} = \{1, \ldots, K\}$ the label space of known ORs. Given a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, the goal is to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ that generalizes to new samples. To this aim, we consider a set of various ML models: (i) classical statistical classifiers, i.e. Logistic Regression (LR) and Gaussian Naïve Bayes (GNB); (ii) decision tree-based methods, i.e. Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB); and (iii) neural networks, i.e. Multi-Layer Perceptron (MLP). Their hyperparameters are optimized via grid search with cross-validation (not shown for the sake of brevity). Optimal hyperparameter values are summarized in Tab. I. Such models are then evaluated on a disjoint test set $\mathcal{D}_{\text{test}} = \{(x_j, y_j)\}_{j=1}^{M}$, where $x_j \in \mathcal{X}$ and $y_j \in \mathcal{Y}_{\text{test}}$.

Firstly, we consider a **closed-world** setting, providing an upper bound for ML models' performance. Formally, we assume $\mathcal{Y}_{\text{test}} = \mathcal{Y}$, namely, all test instances refer to the same ORs encountered during training.

To reflect realistic deployment scenarios where user mobility and urban dynamics may evolve, we extend our evaluation to an **open-world** setting, where ORs unseen during training may occur at inference (i.e. operational) time. In the open-world setting, $\mathcal{Y}_{\text{test}} = \mathcal{Y} \cup \mathcal{Y}_{\text{unknown}}$, where $\mathcal{Y}_{\text{unknown}}$ denotes ORs not present in training. To operate reliably in such a realistic setting, we integrate an OSR mechanism that enables the model to reject uncertain predictions instead of forcing classification into a known OR. Given a posterior probability distribution $P(y \mid x)$, we compute the maximum predicted confidence as $\hat{p}_c = \max P(y \mid x)$. A confidence threshold τ determines whether to accept or reject the prediction:

$$\hat{y} = \begin{cases} \arg\max P(y \mid x), & \text{if } \hat{p}_c \geq \tau \\ \texttt{Unknown}, & \text{if } \hat{p}_c < \tau \end{cases}$$

This mechanism allows the classifier to detect and reject test samples associated with ORs not seen during training, thus improving robustness to dynamic mobility patterns.

Finally, we extend the OSR mechanism by framing the task as a multiclass classification problem over the augmented label space $\mathcal{Y}' = \mathcal{Y} \cup \{\texttt{Unknown}\}$, where Unknown is treated as an auxiliary class. At inference time, if the classifier's confidence $\hat{p}_c = \max P(y \mid x)$ exceeds the threshold τ, the sample is assigned to a known OR as in the closed-world setting. Otherwise, it is classified as Unknown. Known-class predictions are evaluated using standard multiclass metrics (e.g., accuracy, macro-averaged F1-score), while the true positive rate for the Unknown class reflects the ability to correctly detect ORs unseen during training (see Sec. IV-D for details).

In summary, the proposed framework integrates predictive modeling and OSR to jointly address known and unknown ORs, enhancing the robustness of offloading decisions under realistic mobility dynamics in smart urban environments.

IV. Experimental Setup

This section presents the experimental setup adopted to assess the performance of the proposed OR prediction framework. We begin by introducing the dataset (Sec. IV-A) and describing the set of mobility-aware features engineered for ML model training (Sec. IV-B). Next, we outline the data preprocessing steps, including filtering and class balancing (Sec. IV-B), followed by the evaluation metrics adopted under both closed-world and open-world settings (Sec. IV-D).

A. Dataset Description

For our experiments, we start with the widely used *Microsoft Geolife GPS Trajectories* dataset [16], a large-scale human mobility dataset collected in Beijing between 2007 and 2012, comprising high-resolution, time-stamped GPS traces from 182 users. Geolife offers fine-grained temporal granularity, with over 91% of samples recorded at intervals of 1 to 5 seconds, making it particularly suitable for mobility analysis and OR prediction tasks.

To ensure consistency with prior work, we adopt the OR definition introduced in [2], where each OR is identified by a combination of user identifier (device_id), discretized time of day (day_period), temporal threshold (contDur), and a location-specific index (loc_idx_cat). The day_period variable discretizes the day into semantically meaningful intervals (e.g., sleeping, commuting, working, lunch, social activities), thus providing coarse-grained temporal context. ORs are filtered considering only one fixed temporal threshold (contDur = 5 s) and ordered chronologically to define OR transitions correctly. We refer the reader to [2] for full details on the OR extraction process.

B. Feature Extraction

To support the prediction of the next OR, we extend the original Geolife dataset with contextual and mobility-related features. Building upon the feature set introduced in [2], which captures user location and temporal context, we further enhance the dataset with engineered features that aim to improve the model's discriminative power across transitions between ORs.

TABLE II
Feature set for OR prediction with ML models.

Feature	*Description*
loc_idx_cat	Unique identifier of the current location
last_cell_lon	Longitude of the last point in current OR
last_cell_lat	Latitude of the last point in current OR
leaveOR_speed	Speed of the user while leaving the OR
hour	Hour of the day the user was in the OR
weekday_cat	Day of the week the user was in the OR
relevanceOR_cat	Relevance category of the OR
previousOR_cat	Identifier of the previously visited OR
time_inOR_cat	Categorized duration the user stayed in the OR
movement_angle	**Movement angle between two consecutive ORs**
min_dist	**Minimum distance between two consecutive ORs**
time_in_previousOR	**Time the user spent in the previous OR**

Novel engineered features are typeset in **boldface**.
Features marked with cat are categorical; all others are numerical.

In particular, we design *three additional novel features* to better capture inter-OR mobility dynamics:

(i) **Movement Direction (movement_angle)** indicates the heading of the user between two consecutive ORs, capturing directional mobility patterns. It is computed via the two-argument arc-tangent function:

$$\theta = \arctan 2(\Delta\phi, \Delta\lambda)$$

with

$$\Delta\phi = \phi_{\text{current}} - \phi_{\text{previous}}, \quad \Delta\lambda = \lambda_{\text{current}} - \lambda_{\text{previous}}$$

where ϕ and λ are the latitude and longitude, respectively, used to define the coordinates of the first position in the current OR and the last position in the previous OR.

(ii) **Minimum Distance between ORs (min_dist)** measures the shortest spatial distance between any two ORs visited consecutively by a user. For ORs OR_x and OR_y, the distance is calculated as the minimum Euclidean distance between all GPS points in the two ORs:

$$d_{\text{min}}(OR_x, OR_y) = \min_{i \in OR_x, j \in OR_y} \left(\sqrt{(\phi_i - \phi_j)^2 + (\lambda_i - \lambda_j)^2} \right)$$

where (ϕ_i, λ_i) and (ϕ_j, λ_j) are the GPS coordinates of all points $i \in OR_x$ and $j \in OR_y$.

(iii) **Time in Previous OR (time_in_previousOR)** records the total time the user spent in the previous OR, offering insights into user behavior and dwell tendencies before transitioning to a new region.

These features are computed for each OR transition, where the current OR is identified by the tuple of attributes described in Sec. IV-A, and are used as input to the supervised learning models. Table II summarizes the complete set of features employed in our experiments, with the newly engineered ones highlighted in bold. The resulting dataset provides a clean and temporally ordered collection of user mobility segments, suitable for supervised OR prediction.

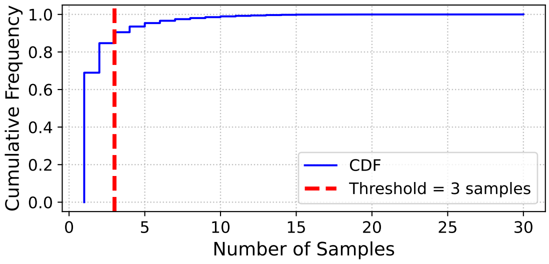


Fig. 1. Cumulative Distribution Function of the target variable *nextOR*.

TABLE III
TRAIN AND TEST ACCURACY IN THE BASELINE SCENARIO.

Model	*Train Accuracy [%]*	*Test Accuracy [%]*
DT	89.62	20.65
GNB	95.26	22.15
GB	70.57	20.37
LR	96.07	23.51
MLP	95.20	20.54
RF	**98.66**	**24.94**

Best performance figures are typeset in **boldface**.

C. Preprocessing Operations

To enable a consistent comparison with [2], we adopt a series of preprocessing operations aligned with their filtering criteria. For each user, a sliding window of four consecutive months is applied, and the interval with the highest number of days with GPS data is retained. We select only users with at least five weekdays containing a minimum of eight hours of data, restricting the analysis to mobility traces collected within the metropolitan area of Beijing. Observations occurring during the sleeping period (i.e. 01:00–06:59) are excluded, as this `day_period` exhibits limited mobility and scarce transitions between ORs. Also, we retain only current ORs (grouped as in Sec. IV-A) with at least 12 valid observations to ensure training consistency. A temporally consistent $75\%/25\%$ split is applied to obtain the training and test sets for ML models.

Nevertheless, a preliminary analysis of the target variable `nextOR` still revealed a strong class imbalance. As shown in Fig. 1, which reports the cumulative distribution function of class frequencies, approximately 70% of the classes appear fewer than 3 times (red dashed vertical line in the figure). To reduce the impact of underrepresented classes and ensure model generalization, we exclude from the training set all classes with fewer than 3 samples. The resulting dataset is well-suited for supervised learning in both closed-world and open-world OR prediction tasks.

D. Evaluation Metrics

To evaluate the performance of our OR prediction framework, we employ standard classification metrics suited for both *closed-world* and *open-world* settings. In the closed-world setting (and for the known classes in the open-world), we report *accuracy* and *macro-averaged F1-score*. In the open-world setting, these metrics are computed only on test samples confidently assigned to one of the known ORs, excluding those rejected as unknown.

To assess the effectiveness of the OSR mechanism, we compute the *True Positive Rate* (TPR), defined as the proportion of truly unknown ORs correctly rejected by the model, and the *False Positive Rate* (FPR), defined as the fraction of known ORs mistakenly rejected as unknown. Additionally, we report the *Receiver Operating Characteristic* (ROC) curve, which plots TPR against FPR across varying confidence thresholds, and the corresponding *Area Under the Curve* (AUC), which quantifies the model's ability to distinguish between known and unknown ORs.

Following the definition of current OR adopted in this work, all experiments are conducted on temporally ordered transitions grouped by `(device_id, day_period, contDur)`. Evaluation metrics are computed for each group and then averaged to ensure consistent and fair comparison across varying user behaviors and mobility contexts.

V. EXPERIMENTAL RESULTS

This section presents the results of our experimental evaluation on `nextOR` prediction using ML models. We consider a structured pipeline comprising *four scenarios*, designed to reflect increasing levels of realism and complexity. Section V-A evaluates a *Baseline* scenario, where ML models are evaluated on the entire test set, including unseen ORs, replicating the setup of [2]. Section V-B introduces the *Closed-World* scenario, which restricts evaluation to known ORs (i.e. seen during the training of the model), enabling controlled assessment without optimization. Section V-C presents the *Optimized Closed-World* scenario, where feature engineering and hyperparameter tuning are applied to improve performance under ideal assumptions. Finally, Sec. V-D addresses the *Open-World* scenario, where an OSR mechanism enables the model to reject unseen ORs based on confidence, while continuing to classify known ones. Performance is measured using the metrics defined in Sec. IV-D, highlighting the impact of the proposed methodological components on both predictive accuracy and robustness to mobility variability.

A. Baseline Scenario

The *Baseline* scenario is the starting point of our evaluation. In this scenario, the training set is filtered to exclude ORs with fewer than 3 samples, as described in Sec. IV-C, while the test set is left unchanged, resulting in the presence of ORs not observed during training, as in [2]. We evaluate the ML models introduced in Sec. III using default scikit-learn parameters [15] and without incorporating any novel engineered features. This configuration mirrors realistic deployment conditions but poses a significant challenge, as standard supervised models are not equipped to handle unseen classes at inference time without specific mechanisms to manage them.

As reported in Tab. III, most models achieve high training accuracy—always higher than $\approx 90\%$, except for GB—demonstrating proper learning ability. However, generalization

TABLE IV
TEST ACCURACY IN BASELINE VS. CLOSED-WORLD VS. OPTIMIZED CLOSED-WORLD SCENARIOS.

Model	*Baseline [%]*	*Closed-World [%]*	*Optimized Closed-World [%]*	*Gain [%]*
DT	20.65	58.03	90.80	**+32.77**
GNB	22.15	59.85	95.06	**+35.21**
GB	20.37	55.69	95.60	**+39.91**
LR	23.51	63.48	63.50	**+ 0.02**
MLP	20.54	56.57	**98.43**	**+41.86**
RF	**24.94**	**67.31**	97.61	**+30.30**

Best performance figures are typeset in **boldface**.
Gains indicate the absolute improvement in accuracy between the Optimized Closed-World and Closed-World scenarios.

remains poor. For instance, the best-performing model (i.e. RF) reaches an almost ideal training accuracy of $\approx 99\%$ but only $\approx 25\%$ test accuracy, indicating that unseen classes severely limit predictive capability.

These results highlight the limitations of conventional supervised learning under open-world assumptions and motivate the need for tailored solutions that take such real conditions into account. To this aim, the next section introduces a closed-world scenario, where the set of ORs is fixed across training and testing to enable proper optimization of the considered ML models under constrained conditions.

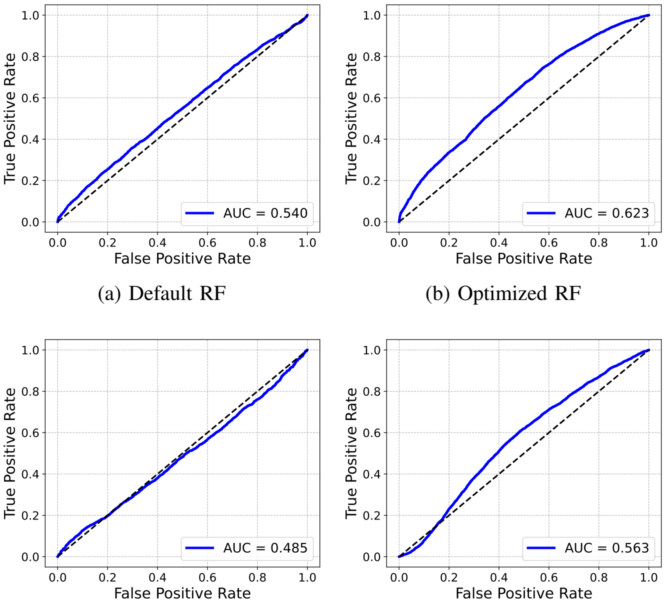
B. Closed-World Scenario

The *Closed-World* scenario provides a controlled setting in which only ORs present in the training set are retained in the test set. This setup enables the assessment of each model's ability to generalize within a known class space, without the additional complexity introduced by unseen ORs at inference time. As reported in Tab. IV, all classifiers benefit from this restricted setting, showing markedly improved performance, as theoretically expected. The RF model achieves the highest test accuracy, with a gain of approximately $+42\%$ over the baseline. Similarly, the MLP model improves by $+36\%$, reaching an accuracy of 57%. These results confirm that `nextOR` prediction performance is strongly influenced by the presence of previously unseen ORs. Also, this scenario serves as a controlled reference environment to evaluate the effects of expert-driven optimizations under ideal conditions, as discussed in the next section. Indeed, despite the observed performance gains, there is still room for improvement, motivating the next step of our evaluation.

C. Optimized Closed-World Scenario

The *Optimized Closed-World* scenario builds upon the results of the previous one by evaluating the effect of *feature engineering* and *hyperparameter tuning* on the performance of ML models. We extend the feature set by incorporating the *newly engineered features* introduced in Tab. II and apply exhaustive hyperparameter tuning (see Tab. I) via *grid search* for each classifier (tuning details are omitted for brevity) [15].

As reported in Tab. IV, compared to the Closed-World scenario, almost all models show absolute accuracy gains above $+30\%$, highlighting the general effectiveness of the adopted optimizations. Only LR reports almost no gain, underpinning its difficulty in tackling this classification task despite optimizations. On the other hand, the MLP exhibits the most substantial improvement, outperforming its non-optimized counterpart by approximately $+42\%$ and establishing itself as the overall best-performing model, achieving a test accuracy of over 98%. It is closely followed by RF, with a difference of less than 1%. These consistent improvements and high absolute performance justify the selection of RF and MLP for the subsequent open-world evaluation.



(a) Default RF

(b) Optimized RF

(c) Default MLP

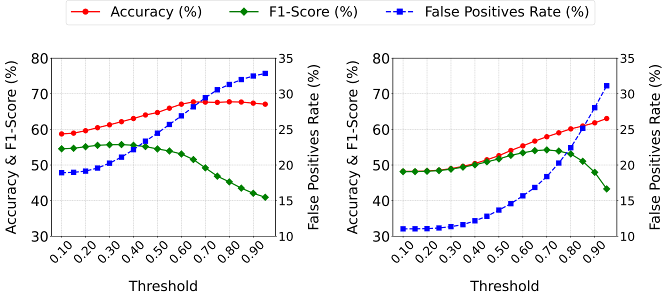
(d) Optimized MLP

Fig. 2. ROC curves and related AUC for default and optimized RF and MLP.

D. Open-World Scenario

Open-World conditions naturally arise in smart urban environments, where evolving mobility patterns and newly visited locations introduce significant variability over time. In such dynamic contexts, supervised models face the critical challenge of encountering ORs that were not observed during training. To address this, we adopt an OSR mechanism based on a confidence threshold, enabling the model to reject uncertain `nextOR` predictions rather than assigning them to incorrect known classes, as detailed in Sec. III. To evaluate the effectiveness of this mechanism, we compare the unoptimized (viz. default) and optimized configurations of the best-performing classifiers, RF and MLP, as identified in Sec. V-C.

Figure 2 shows the ROC curves and corresponding AUC scores for both configurations. RF improves from an AUC of 0.540 to 0.621 (i.e. $+15\%$), while MLP from 0.485 to 0.563 (i.e. $+16\%$) after optimization. These gains highlight how model refinement enhances not only the accuracy in the closed-world but also the ability to reject unseen ORs in open-world conditions. On the other hand, while the task remains inherently challenging, these results confirm that integrating



(a) Optimized RF (b) Optimized MLP

Fig. 3. Accuracy, F1-score, and FPR vs. threshold τ for RF and MLP when integrating multiclass supervised classification with OSR.

OSR with feature and model optimizations strengthens the robustness of classifiers in dynamic mobility scenarios.

To further improve model robustness under open-world conditions, we extend the OSR framework by *integrating supervised multiclass classification with confidence-based rejection*. During inference, samples are either assigned to a known `nextOR` or rejected as `Unknown` if the prediction confidence falls below a threshold τ. Figure 3 reports accuracy, F1-score, and FPR for RF and MLP across varying values of τ. Both models exhibit stable trends for $\tau \leq 0.3$, indicating good confidence calibration in this range. RF achieves a favorable trade-off at $\tau \in [0.4, 0.5]$, with accuracy around 65%, F1-score above 55%, and FPR below 25%. In the same range, MLP attains a slightly lower F1-score ($\approx 51\%$) but also benefits from a lower FPR ($< 14\%$). While RF yields higher predictive performance, it is more prone to falsely rejecting known ORs. Conversely, MLP adopts a more conservative approach, favoring lower FPR at the expense of reduced accuracy and a sharper degradation in performance beyond $\tau > 0.4$.

These findings highlight how the choice of model and threshold τ can be tuned to balance misclassification and rejection, depending on application-specific requirements. Overall, integrating confidence-based rejection into multiclass classification enhances prediction robustness under open-world conditions, particularly when compared to the baseline scenario. Nonetheless, further improvements may be achievable through better calibration or more advanced OSR techniques.

VI. Conclusion

This work proposed a structured ML-based framework for *predicting ORs in smart urban environments to enhance mobile data offloading*, addressing the challenges introduced by dynamic mobility patterns and the emergence of unseen ORs at inference time. We conducted a systematic evaluation of various ML models on the Geolife dataset for the city of Beijing, assessing their predictive capabilities across both *closed-world* and *open-world* scenarios. The integration of *novel spatio-temporal features* and careful hyperparameter tuning led to substantial improvements in closed-world settings, with accuracy gains of up to $+42\%$. To account for real-world deployment conditions, we extended our evaluation to an open-world scenario via a confidence-based OSR mechanism. Experimental results confirmed the effectiveness of our approach, showing relative AUC gains of up to $+16\%$ after optimization and enabling a tunable trade-off between rejecting unknown ORs and accurately classifying known ones.

Despite these promising results, the latter joint task remains a non-trivial challenge. This opens avenues for **future work**, including: (i) exploring advanced DL architectures for automatic feature distillation from mobility patterns; (ii) exploiting newly released and diverse datasets, possibly including location data from cellular network operators, ride-sharing services, or public transit apps; (iii) designing hierarchical frameworks that combine unsupervised and supervised components for OR detection and classification; (iv) integrating more principled OSR methods such as OpenMax or Extreme Value Machines [3]; (v) embedding prediction models into edge computing infrastructures to enable low-latency inference and adaptive offloading decisions.

References

[1] J. Arkko *et al.*, “Ericsson Mobility Report,” *Ericsson, Stockholm, Sweden, Tech. Rep. EAB-25*, vol. 004565 Uen Rev B, 2025.

[2] E. Lima, A. Aguiar *et al.*, “Human Mobility Support for Personalized Data Offloading,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1505–1520, 2022.

[3] C. Geng, S.-J. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.

[4] Y. Zheng, L. Zhang *et al.*, “Mining interesting locations and travel sequences from GPS trajectories,” in *Proceedings of the 18th International Conference on World Wide Web*, 2009, pp. 791–800.

[5] X. Cao, G. Cong, and C. S. Jensen, “Mining Significant Semantic Locations from GPS Data,” *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1009–1020, 2010.

[6] X. Lu, E. Wetter *et al.*, “Approaching the limit of predictability in human mobility,” *Scientific Reports*, vol. 3, p. 2923, 2013.

[7] C. Krumme, A. Llorente *et al.*, “The predictability of consumer visitation patterns,” *Scientific Reports*, vol. 3, no. 1645, 2013.

[8] T. M. T. Do, O. Dousse *et al.*, “A probabilistic kernel method for human mobility prediction with smartphones,” *Pervasive and Mobile Computing*, vol. 20, pp. 13–28, 2015.

[9] Z. Yang, J. Hu *et al.*, “Mobility Modeling and Prediction in Bike-Sharing Systems,” in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 165–178.

[10] Q. Liu, S. Wu *et al.*, “Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 194–200.

[11] F. Rebecchi, M. Dias de Amorim *et al.*, “Data Offloading Techniques in Cellular Networks: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 17, 2015.

[12] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE communications surveys & tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[13] G. Chen and J. wan Zhang, “Intelligent transportation systems: Machine learning approaches for urban mobility in smart cities,” *Sustainable Cities and Society*, vol. 107, p. 105369, 2024.

[14] E. Lima, A. Aguiar *et al.*, “Impacts of Human Mobility in Mobile Data Offloading,” in *Proceedings of the 13th Workshop on Challenged Networks*, Oct. 2018, pp. 39–46.

[15] F. Pedregosa, G. Varoquaux *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] Y. Zheng, X. Xie *et al.*, “GeoLife: A collaborative social networking service among user, location and trajectory.” *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.