Trajectory Prediction and Intelligent RSU Handover for Connected Vehicles Using Deep Sequential and Ensemble Learning

Muhammad Azfar Yaqub
Free University of Bozen-Bolzano
Bolzano, Italy
muhammadazfar.yaqub@unibz.it

Attaullah Buriro
University of Essex
Colchester, United Kingdom
attaullah.buriro@essex.ac.uk

Malik Muhammad Saad
Kyungpook National University
Daegu, South Korea
maliksaad@knu.ac.kr

Amir Aieb
Free University of Bozen-Bolzano
Bolzano, Italy
amir.aieb@student.unibz.it

Antonio Liotta
Free University of Bozen-Bolzano
Bolzano, Italy
antonio.liotta@unibz.it

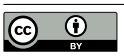
Muhammad Rehan Usman
Kingston University London
London, UK
r.usman@kingston.ac.uk

Abstract

Accurate trajectory prediction and proactive Road-Side Unit (RSU) handover are essential for maintaining seamless connectivity and low-latency Multi-Edge Computing (MEC) based services in Cooperative Adaptive Cruise Control (CACC) systems. Conventionally, mobility forecasting and migration decisions are treated as separate problems, neglecting their inherent dependence, thus leading to premature or delayed handovers. In this work an integrated learning-based pipeline is introduced that first predicts the vehicle mobility using deep sequential models and the leverages these predictions to trigger intelligent RSU migrations. We employ a LSTM network to learn temporal dynamics from the large-scale FHWA dataset and generate multi-step displacement predictions over short horizons. These predictions are then used to infer future coverage boundaries and connectivity risks. Further, we model RSU migration as a binary-class classification task and train Random Forest (RF) and Multilayer Perceptron (MLP) as classifiers on engineered mobility features, such as velocities deltas, cumulative drift and distance to RSU markers. To handle skewed migration labels, we incorporate fallback heuristics and class-balanced strategies. Experiments on the 500,000 real vehicular samples reveal that the RF-based migration model achieves up to *73% accuracy*. Meanwhile LSTM maintains stable short-horizon displacement accuracy with competitive Average Displacement Error (ADE) and Final Displacement Error (FDE) scores. Together, the models enable anticipatory RSU handover decisions that outperform naive threshold-based methods.

CCS Concepts

• **Networks** → *Network performance modeling*; • **Computing methodologies** → **Machine learning algorithms**.



SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779985

Keywords

Multi-access Edge Computing, Cooperative Adaptive Cruise Control, LSTM, Mobility Forecasting, Edge Intelligence

ACM Reference Format:
Muhammad Azfar Yaqub, Attaullah Buriro, Malik Muhammad Saad, Amir Aieb, Antonio Liotta, and Muhammad Rehan Usman. 2026. Trajectory Prediction and Intelligent RSU Handover for Connected Vehicles Using Deep Sequential and Ensemble Learning. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748522.3779985

1 Introduction

Cooperative Adaptive Cruise Control (CACC) and vehicle platooning promise substantial gains in highway throughput, fuel economy, and safety by coordinating longitudinal motion across dynamic vehicular networks [1, 2]. Realizing these benefits in practice increasingly depends on connected and edge-intelligent infrastructure; particularly RSUs that provide low-latency computation and reliable connectivity for inference and control loops. However, highly dynamic traffic, variable radio conditions, and heterogeneous edge loads make it difficult to decide *when* to hand over a vehicle's session to a new RSU and *how* to anticipate near-future motion for predictive resource allocation. Suboptimal handovers can cause stalls or jitter in safety-critical loops, while inaccurate trajectory forecasts may degrade headway control or disrupt the stability of platoons.

To address these challenges, two technical capabilities are central: (i) *short-horizon trajectory prediction* of the ego platoon vehicle in local road coordinates, and (ii) *RSU migration (handover) decisions* that are proactive and load-aware. In the broader literature, edge computing frameworks such as Multi-access Edge Computing (MEC) provide the architectural substrate for offloading real-time inference to roadside servers [3, 4, 5]. Meanwhile, learning-based trajectory prediction has advanced rapidly from recurrent models to interaction-aware graph and vectorized scene encoders [6, 7, 8, 9], supported by large motion datasets and standardized metrics [10]. Yet, most works study trajectory forecasting *in isolation from* RSU control logic, or approach RSU handover using simple mobility or coverage heuristics that fail to exploit learned motion

forecasts. This disconnect limits end-to-end performance in realistic, nonstationary highway scenarios. In this paper, we address this gap by formulating and validating an integrated pipeline that couples (a) *sequential deep learning* for ego-trajectory prediction with (b) *supervised migration decision* at the RSU layer. Concretely, we: (1) learn a short-term motion model using an LSTM that predicts future displacements $(\Delta x, \Delta y, \Delta v)$ over a K-step horizon from a past window of kinematics and context; and (2) derive *leakage-free* handover features from the same past window (optionally augmented with distance/load descriptors) to train an RSU migration classifier. To ensure operational robustness and interpretability, we study two migration classifiers in parallel: a deep MLP with batch normalization and focal loss, and a non-parametric Random Forest (RF) baseline with permutation and Gini importances. This pairing enables a trade-off between raw accuracy and model explainability and training simplicity.

The proposed framework applies simple, reproducible preprocessing steps (imputation, scaling, windowing), trains on normalized deltas to stabilize optimization, and evaluates using RSU-relevant decision metrics (PR-AUC, F1 at an operating threshold) alongside standard forecasting metrics (ADE/FDE). We further incorporate a configurable, policy-aware labeling rule for migration (coverage change and minimum dwell), with a calibrated fallback to preserve class balance when coverage signals are degenerate. To summarize, the main contributions of this work are:

- An end-to-end formulation that couples short-horizon trajectory prediction with RSU migration, bridging two communities that are typically evaluated in isolation.
- A leakage-free RSU feature design derived from past windows, optionally augmented with per-RSU distance/load descriptors, together with a policy-aware migration labeler and balanced fallback mechanism.
- Dual migration models (MLP and RF) that enable both strong PR-AUC/F1 performance and model interpretability via feature importances.

The rest of this paper is organized as follows: Section 2 surveys the recent and most relevant studies centered around the problem. Sections 3 and 4 present the system architecture and methodology of the proposed framework. Section 5 reports the obtained results, which are further analyzed in Section 6. Finally, Section 7 concludes the paper with the summary of our findings and the potential future work.

2 Related Work

In this section, we survey the most recent and most relevant papers, to compare our scheme.

Frequent handovers between roadside units (RSUs) in highly mobile vehicular networks can lead to deterioration in quality of service if not managed intelligently. A large body of research has explored machine learning–based handover algorithms. For instance, Tan *et al.* [11] proposed a Double Deep Q-Learning approach for vehicle-to-network (V2N) handovers, reducing packet loss while keeping signaling overhead low. Similarly, Siriwardhana *et al.* [12] introduced a deep learning and metaheuristic model combining a Deep Maxout Network with a Dung Beetle Optimizer to reduce latency, throughput degradation, and ping-pong handovers during heterogeneous network transitions (5G, LTE, DSRC). Other works have treated handover decisions as a supervised classification problem: Khoder *et al.* [13] compared multiple algorithms in a platoon context and showed that RF outperformed other methods.

Predicting future positions of vehicles is vital for intelligent transportation systems and proactive connectivity management. Recurrent neural networks (RNNs), especially LSTMs, remain dominant in this space. Altché *et al.* used LSTMs to predict highway vehicle motion and outperformed kinematic baselines [14]. More recent approaches leverage attention and Transformers. Cheng *et al.* [15] demonstrated that a vision-transformer-based model substantially reduces long-horizon trajectory error on datasets such as NGSIM and HighD. Ensemble-based or hybrid methods also exist: Choi *et al.* [16] combined an LSTM encoder–decoder with a RF classifier to improve motion prediction by leveraging V2V communication signals and on-board sensors.

Several recent studies advocate combining vehicle mobility prediction with network handover logic. Lv *et al.* [17] used trajectory forecasts for proactive edge offloading and RSU association, demonstrating reduced latency and resource wastage. In the millimeter-wave context, Szeto *et al.* [18] presented B-AWARE, a blockage-aware RSU scheduling framework using predicted paths to avoid signal disruption; it improved throughput by 5% and cut outage time by 40%. Spring *et al.* [19] proposed a multi-agent RSU-centric handover scheme (MACH) integrated with short-term trajectory knowledge to reduce unnecessary handovers in dense urban networks.

Despite these advances, there are still gaps. First, most studies treat trajectory prediction and handover decisions separately, rather than embedding forecasting knowledge directly into the decision pipeline. Deep reinforcement learning approaches are often computationally heavy [11, 19] and require specialized infrastructure to deploy. Even hybrid or predictive models can suffer from generalization problems when tested beyond controlled scenarios [12]. Moreover, many solutions rely on centralized coordination or high communication overhead, making them difficult to scale.

We address these gaps by tightly coupling an LSTM-based trajectory forecaster with an ensemble-learning (RF) classifier to determine if and when a handover should occur. Unlike RL-based solutions, this approach is straightforward to train offline, efficient to deploy at the network edge, and can adapt to different road and traffic conditions. Integrating predicted mobility into the handover decision enables proactive rather than reactive RSU reassignments, improving both reliability and latency.

3 Our Approach

3.1 System Overview

Connected and automated vehicles (CAVs) with multi-access edge computing (MEC) support requires intelligent mechanisms to maintain seamless connectivity and performance. A central challenge is enabling the network edge to make proactive decisions (e.g., when to hand off a vehicle's service between roadside units, or RSUs) by anticipating the vehicle's short-horizon motion [20]. To address this, we propose a two-module system that jointly performs RSU handover prediction and vehicle trajectory forecasting. Figure 1 illustrates the overall architecture.

The workflow consists of collecting the data initially. Each vehicle periodically transmits telemetry (position, speed, acceleration, heading) and network context (current RSU ID, signal readings, neighbor RSU metrics). Next, the RSU prediction model uses RF classifier to determine whether a handover is needed and Trajectory Prediction Module uses an LSTM-based recurrent model to forecast near-term motion. Finally, the system fuses both outputs to enact proactive or delayed RSU handovers based on predicted vehicle position and target coverage zones.

3.2 RSU Prediction Module

We formulate handover triggering as a binary classification problem (migrate / don't migrate). A RF is used due to its robustness, interpretability, and ability to capture non-linear feature interactions [21].

At each time step t, we construct a feature vector containing, vehicle state (speed, acceleration, heading), radio context (received RSSI/RSRP from the current and neighboring RSUs), RSU load/traffic context to capture network-side conditions, distance to the serving RSU coverage boundary, and predicted local displacement hints from the LSTM for short-horizon position change.

All the features are normalized, and categorical IDs are one-hot encoded where needed. This feature representation provides a consistent, temporarily aligned input for the migration classifier.

Since handover events are relatively rare, we handle class imbalance through a combination of class weighting in the RF, oversampling of positive (handover) samples, and decision threshold tuning based on F_1 or ROC-PR trade-offs [22].

For training and evaluation, we use 10-fold cross-validation with temporal ordering for sequence safety. Model hyperparameters which include the number of trees and depth are tuned on a validation set. Feature importance from the RF helps confirm domain relevance with attribtes such as RSSI and coverage edge distance emerge as primary predictors. Performance evaluation metrics includes precision, recall, F_1 score, AUROC, and false-handover rates.

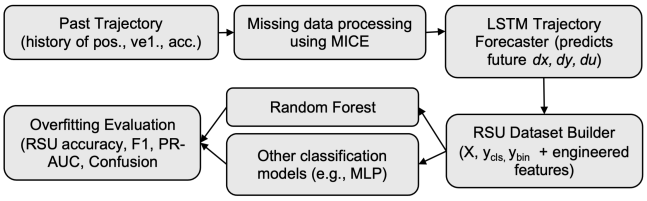


Figure 1: Pipeline for RSU migration prediction

To produce short-term forecasts of vehicle motion in the trajectory prediction module, we design a sequence-to-sequence LSTM architecture [23]. We take the last L time steps:

$$X_t = [\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{L\times F},$$

where each $\mathbf{x}_i$ includes position, velocity, and relevant context. Features are standardized.

The LSTM predicts a short-horizon sequence of length K in the form of relative state changes:

$$\hat{Y}_t = [\Delta\hat{s}_{t+1}, \ldots, \Delta\hat{s}_{t+K}] \in \mathbb{R}^{K\times C},$$

where $\Delta\hat{s}_{t+1}$ represents incremental changes in position and velocity (Δx, Δy, and Δv). This modeling at meter scales increments helps stabilize training and improves convergence robustness [24].

We use two stacked LSTM architecture, where each layer comprising 256 hidden units with dropout (0.2) for regularization, followed by a MLP head that processes the relative changes (Δx, Δy, Δv). The model is tuned using the Smooth L1 (Huber) loss and optimized with Adam. To enhance stability, gradient clipping is used to prevent exploding gradients and the Learning rate follows cosine annealing schedule.

The training samples are generated using sliding-window approach, reserving 10% windows for validation. Typical hyperparameters choices are input length $L = 30$, and prediction horizon $K = 10$. We monitor Average Displacement Error (ADE) and Final Displacement Error (FDE) as evaluation metrics [25].

Predicted displacements are then de-normalized and cumulatively summed to get absolute forecasts (trajectories). By mapping predicted trajectories to RSU coverage zones, we can infer which RSU a vehicle is likely to enter and the handover timing.

3.3 Integration Strategy

The framework integration strategy combines the output of RF and LSTM to yield proactive handover decisions. Specifically:

- When the RF indicates an imminent handover and the LSTM forecasts that the vehicle will soon exit from current RSU coverage area, the system proactively triggers handover early.
- If RF is uncertain but the LSTM trajectory forecasts movement towards a neighbour RSU's region, the system adapts by lowering thresholds or plan the handover accordingly.
- Conversly, if the RF suggests a handover while the LSTM shows minimal displacement, the decision maye be sppressed to avoid false trigger.

By aligning predicted trajectories with data-driven RF-based classification, the framework effectively reduces both unnecessary handovers (ping-pong) and missed handovers [26]. This integrated approach thus combines the strengths of ensemble classification and deep sequential forecasting to realize a robust, forward-looking edge handover mechanism suitable for dynamic vehicular environments.

4 Methodology

Figure 1 in the previous section summarizes the dataflow; here we provide complete algorithmic and implementation specifics. In this section, we describe our proposed framework. We start by describing the end-to-end pipeline for (i) preparing a supervised learning corpus from raw connected-vehicle platoon traces, (ii) training a sequence model to forecast short-horizon vehicle motion, and (iii) developing a RSU migration decision policy from historical and context aware features.

4.1 Data Sources and Preprocessing

We ingest the data from the filed tests conducted by the Federal Highway Administration (FHWA) in collaboration with the Vope center for a platooning proof f concept based on CACC and ACC [27] that contains hundreds of millions of time-stamped records; 478 raw columns.

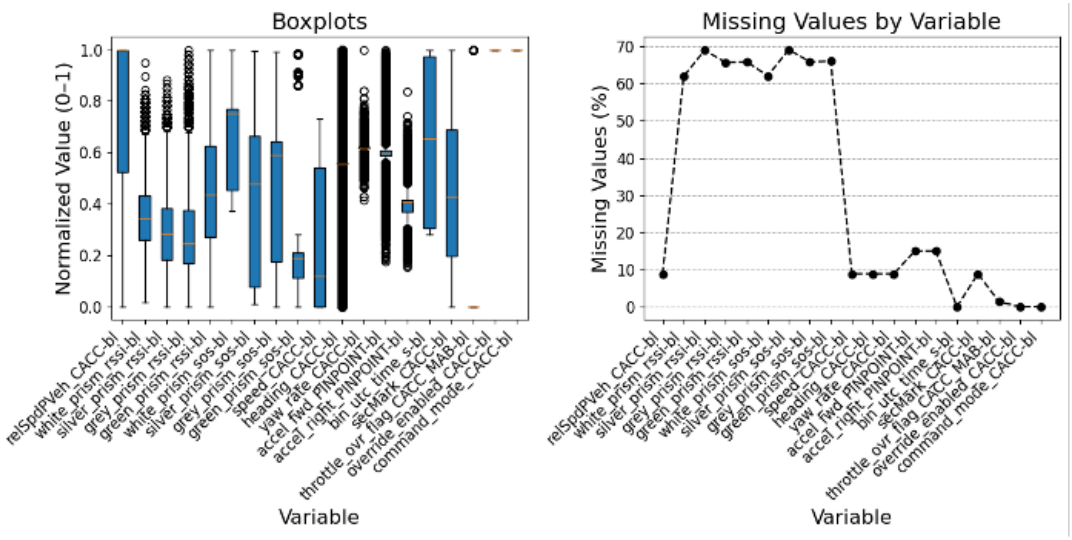


Figure 2: Boxplots display the distribution of the vehicle dataset variables, followed by a plot showing the percentage of missing data for each variable.

To obtain structured features, we begin with column pruning, where the columns 100% missing data are dropped and cast numeric-typed fields (coercing invalid tokens to `NaN`) using a schema-aware loader. Numerical features are imputed on a feature basis, median by default, while categorical columns are one-hot encoded with an `UNK` bucket for missing values. Local displacements are computed $(\Delta x, \Delta y)$ in meters from (lat, lon) using a linear approximation:

$$\Delta y \approx 111{,}320 \cdot \Delta\text{lat}, \qquad \Delta x \approx 111{,}320 \cdot \cos(\text{lat}_0) \cdot \Delta\text{lon},$$

where lat_0 is the first non-missing latitude in a continuous segment. Velocity changes Δv are derived either from measured speed (forward-fill then first-difference) or directly from $\Delta x, \Delta y$ if speed is unavailable. Finally, a feature scaler S_x is fit using training-only statistics and applied to all features, while optionally a target scaler S_y is used to standardize $(\Delta x, \Delta y, \Delta v)$ for numerically stable training.

To support reproducibility, we persist (i) a processed CSV for auditing, (ii) an NPZ tensor container with sliding-window sequences (Section 4.2), (iii) joblib scalers S_x, S_y, and (iv) a metadata JSON describing shapes and configuration.

4.2 Temporal Windowing and Feature Engineering

We convert the flat time series into supervised sequence pairs using a sliding window with stride s. Given past length L and prediction horizon K, for each valid index t we form:

$$\mathbf{X}_t \in \mathbb{R}^{L\times F}, \quad \mathbf{Y}_t \in \mathbb{R}^{K\times 3}, \quad \mathbf{Y}_t = [(\Delta x, \Delta y, \Delta v)_{t+1}, \ldots, (\Delta x, \Delta y, \Delta v)_{t+K}],$$

where F is the feature count. To build *leakage-free* features for migration, we flatten the past window $\text{vec}(\mathbf{X}_t)$ and augment it with additional context: (i) Per-RSU distances at the last past step $t = 0$computed from past coordinates, (ii) summary kinematics, such as mean, standard deviation, and last vlaue of of the last Δv over the window. The resulting migration-design matrix $\mathbf{Z}_t \in \mathbb{R}^D$ is used for MLP and RF.

4.3 Label Generation and Migration Policy

For trajectory predictions, the sequence regression target is $\mathbf{Y}_t$ defined in the sliding window formulation. To construct the RSU Class label, let $\mathcal{R} = \{r_i = (x_i, y_i, \rho_i)\}_{i=1}^R$ denote RSUs with centers and coverage radii. Using the cumulative future displacements $\hat{\mathbf{p}}_{t+\tau} = \sum_{j=1}^{\tau}(\Delta x, \Delta y)_{t+j}$ anchored at the current position, the RSU class at the horizon end is set as the index of the nearest RSU:

$$y_t^{\text{cls}} = \arg\min_{i\in\{1,\ldots,R\}} \|\hat{\mathbf{p}}_{t+K} - (x_i, y_i)\|_2 .$$

A binary migration table y_t^{mig} is defined conservatively to indicate an imminent handover. An event is marked if either the predicted class differs from the current RSU cor if the vehicle remains within the current RSU coverage for fewe that κ steps:

$$y_t^{\text{mig}} = \mathbb{1}\Big(y_t^{\text{cls}} \neq y_t^{\text{now}} \;\vee\; \sum_{\tau=1}^{K} \mathbb{1}\big[\|\hat{\mathbf{p}}_{t+\tau} - (x_{y_t^{\text{now}}}, y_{y_t^{\text{now}}})\|_2 \le \rho_{y_t^{\text{now}}}\big] < \kappa\Big),$$

To prevent degenerate class balance (e.g., all zeros), a distance-based fallback is applied with an adaptive threshold percentile, and this adjustment is recorded to ensure reproducibility.

4.4 LSTM-Based Trajectory Forecasting

We adopt a stacked LSTM [23], optionally bidirectional and with dropout, to predict the target sequence $\mathbf{Y}_t$ from the input $\mathbf{X}_t$. Given encoded past $\mathbf{X}_t$, the model outputs $\widehat{\mathbf{Y}}_t \in \mathbb{R}^{K\times 3}$, training is performed by minimizing a per-step Smooth L1 (Huber) loss::

$$\mathcal{L}_{\text{traj}} = \frac{1}{K}\sum_{\tau=1}^{K} \text{Huber}(\widehat{\mathbf{y}}_{t+\tau}, \mathbf{y}_{t+\tau}),$$

Optimzed for with Adam [28] and gradient clipping to ensure stable training. Targets are optionally standardized with inverse-transform for metrics. to improve numerical stabilize.

For evaluation we report Average Displacement Error (ADE) and Final Displacement Error (FDE), computed in meters adter inverse

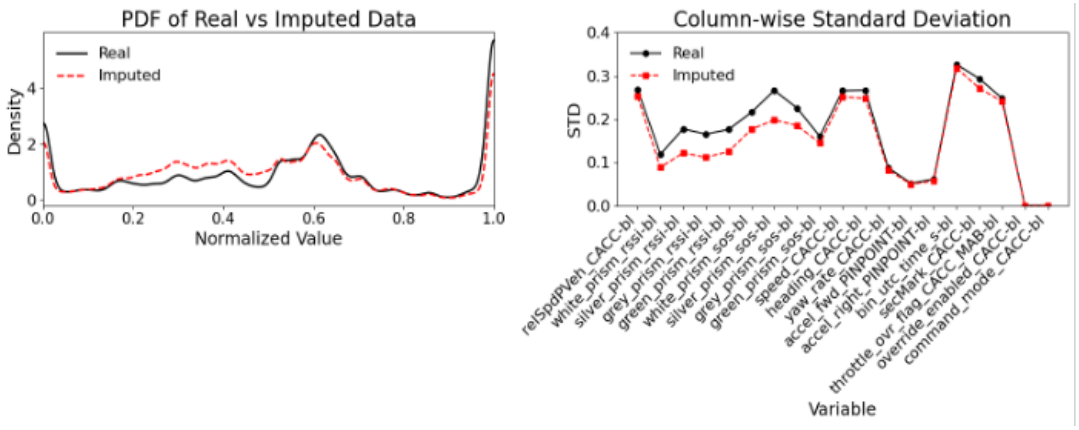


Figure 3: Evaluation of the MICE Model Under 70% Missingness, Including Probability Density Functions (PDF) and Column-wise Standard Deviation Plots Comparing Real and Imputed Data.

scaling:

$$\text{ADE} = \frac{1}{K}\sum_{\tau=1}^{K} \|\widehat{\mathbf{p}}_{t+\tau} - \mathbf{p}_{t+\tau}\|_2\,, \quad \text{FDE} = \|\widehat{\mathbf{p}}_{t+K} - \mathbf{p}_{t+K}\|_2\,,$$

4.5 ML Classifier for RSU Migration

In this study, we use a RF classifier as a baseline for RSU migration prediction and compare its performance with a Multi-Layer Perceptron (MLP) classifier to better evaluate potential overfitting. Both models operate on past-only features $\mathbf{Z}_t$ and jointly predict (i) a binary migration decision and (ii) the RSU class. The RF model implemented using scikit-learn [29, 30], applies

```
class_weight=balanced_subsample
```

to address class imbalance and consists of several hundred trees, offering strong tabular performance and interpretability. The MLP architecture comprises two fully connected layers with Batch Normalization and dropout, optimized for joint prediction using a composite loss: Binary Cross-Entropy (BCE) or Focal Loss [31] for migration, and cross-entropy for RSU classification. The binary loss is defined as,

$$\mathcal{L}_{\text{BCE}} = \text{BCEWithLogits}\left(\hat{z}_t^{\text{mig}},\, y_t^{\text{mig}};\, \text{pos\_weight}\right),$$

and the joint loss combines both tasks as $\mathcal{L}_{\text{Focal}} = \alpha(1-p_t)^{\gamma}\cdot\text{BCE}$, where $p_t = \sigma(\hat{z}_t^{\text{mig}})$, $\alpha \in (0,1]$, $\gamma \geq 0$. The RSU class head uses cross-entropy and the joint loss is defined as $\mathcal{L}_{\text{MLP}} = \mathcal{L}_{\text{mig}} + \lambda\,\mathcal{L}_{\text{cls}}$, where λ a small weighting scalar.

For decision thresholding, we sweep $\theta \in [0.1, 0.9]$ on the validation set to maximize F1 and select the optimal $\theta^{\star}$.We additionally report PR-AUC and ROC-AUC, which provide operating points reflecting the trade-offs between handover aggressiveness and stability.

4.6 Training Configuration and Losses

For optimization, we used Adam for the LSTM and AdamW for the MLP, with optional cosine annealing for the learning rate. Batch sizes range from 256–1024, and gradient clipping (ℓ_2 norm ≤ 1) for the LSTM.

For validation, sequence forecasting reserves 10% of the data as temporally contiguous hold-out segments. Migration models use stratified splits on the binary label when feasible. We report per-epoch metrics, including loss, PR-AUC, ROC-AUC, F1, and accuracy, with best checkpoints selected by PR-AUC.

To ensure proper evaluation , we enforce strict leakage controls Only past window features are used as model inputs, while future-only quantities (e.g., end-of-horizon displacement) are reserved for label construction and never used as inputs during training.

4.7 Evaluation Metrics and Experimental Setup

We report ADE/FDE (meters), per-step ADE, and RMSE on each output channel after inverse scaling for trajectory. For migratinn, we report PR-AUC, ROC-AUC, accuracy, F1, precision/recall, threshold $\theta^{\star}$, and confusion matrices. Additionally save learning curves, PR/ROC curves, and feature importance plots to artifacts. To test the effectiveness, we vary: (i) target standardization on/off, (ii) bidirectional vs. unidirectional LSTM, (iii) MLP depth/width and dropout, (iv) BCE vs. Focal Loss, and (v) inclusion of per-RSU distance features and kinematic statistics.

5 Results

5.1 Dataset Exploration

Figure 2 summarizes the characteristics of the vehicle dataset employed in this study, detailing the variables analyzed, including the speed, heading, acceleration, yaw rate that define the ego-motion states of the vehicle and relative dynamics (`relSpdPVeh`, `distToPVeh`), localization data and control flags. Additionally, inter-vehicle and V2I infrastructure communication features captured via RSSI and SOS measurements. All variables were normalized to enable consistent comparison and visualization. Dynamic variables such as `heading` and `accel_fwd` exhibit broad distributions with numerous outliers, whereas RSSI/SOS-related features are more tightly clustered and predominantly skewed toward lower values. The figure also illustrates variable-specific missingness patterns: prism measurements show substantial data gaps (60–70%),

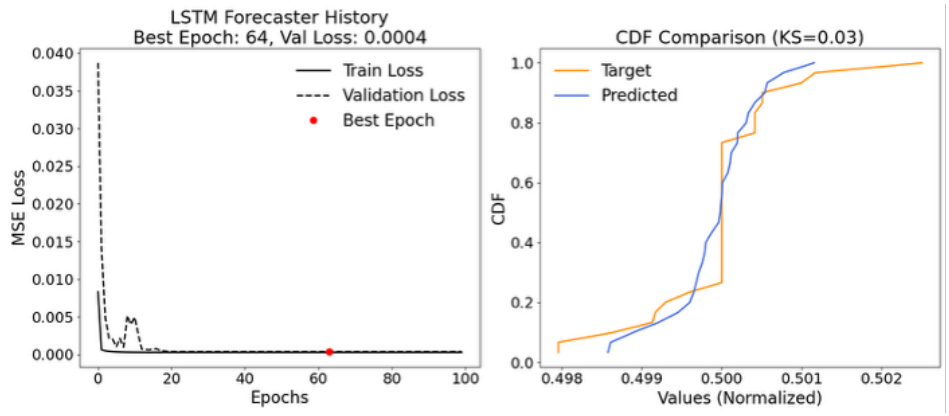


Figure 4: Evaluation of LSTM model performance through training and validation loss, followed by a comparison of the cumulative distribution curves of predicted and target values.

in contrast to minimal missingness (<10%) for variables such as `relSpdVeh`, `speed`, and command/override signals. These disparities underscore the necessity for targeted imputation strategies to address high-missingness features while preserving the integrity of low-missingness variables.

To address `NaN` values in the vehicle dataset, a Multiple Imputation by Chained Equations (MICE) [32] was applied and evaluated by introducing a scenario with 70% missingness to mimic the observed pattern. Figure 3 presents probability density functions comparing original and imputed data, showing strong distributional similarity with minor deviations in specific regions. Column-wise standard deviation plots further indicate that most variables maintain comparable variability post-imputation, with slight reductions for certain prism-related and sensor features. These findings confirm that MICE provides robust imputations while preserving the dataset's overall structure and variability

5.2 Overall Performance on the Test Set

We evaluated the models using a data split of 60% for training, 15% for validation, and 25% for testing. The baseline deep learning model (multi-layer perceptron (MLP)) was compared against an ensemble-based RF classifier. Both models were trained and tuned on the same dataset and evaluated on the held-out test set. The final thresholds applied were 0.40 for MLP and 0.50 for RF, selected to balance precision and recall.

Figure 5 provides a comprehensive side-by-side comparison of the two classifiers, including ROC curves, Precision–Recall (PR) curves, and confusion matrices. Both classifiers achieve strong ROC performance (figure 5 top left), with the MLP slightly outperforming RF (AUC = 0.846 vs. 0.837). Across most false positive rates, the MLP maintains a marginally higher true positive rate. This suggests that the neural baseline is slightly better at overall class separation when measured in ROC space. In the PR analysis (Figure 5, top right), the MLP again yields a modest advantage with PR-AUC = 0.653 compared to 0.606 for RF. This is particularly relevant given the class imbalance in the dataset (roughly 30% migration events). The MLP maintains higher precision at moderate recall levels, reducing unnecessary migration triggers. The RF curve, however, remains competitive and closely tracks the MLP for much of the recall range.

5.3 LSTM Accuracy

Figure 4 presents the performance of an LSTM model in forecasting tasks, with two key components. The MSE loss curve shows the training and validation loss over epochs, with a sharp drop early on, stabilizing near zero. The best model performance is marked at epoch 64, where the validation loss reaches its lowest value of 0.0004, indicating the optimal training point. The CDF comparison between the target and predicted distributions reveals a close alignment, with a Kolmogorov-Smirnov (KS) statistic of 0.03, suggesting minimal deviation between the predicted and actual data. This indicates that the LSTM model provides highly accurate predictions in relation to the target distribution.

Overall, the results demonstrate that both RF and MLP are effective for proactive RSU migration prediction, with complementary strengths. The MLP achieves higher ROC and PR AUC, and reduces false negatives, making it preferable when missed migration events are costly. The RF, on the other hand, offers competitive accuracy and fewer false positives, which may be desirable in deployments where unnecessary handovers are disruptive. These findings underscore that both deep and ensemble learning paradigms are viable, and that deployment priorities (avoiding missed migrations vs. reducing false triggers) should guide model selection.

6 Discussion

Our evaluation of RSU migration prediction for connected vehicles, using a deep sequential MLP and an ensemble RF classifier, reveals a more nuanced outcome than initially expected. Contrary to our earlier hypothesis that the ensemble RF would dominate, the final results show that both models perform at a comparable level, with each offering complementary strengths. Specifically, the MLP achieves slightly higher scores on ROC AUC and PR AUC, while the RF demonstrates competitive accuracy and fewer false positives in confusion matrix analysis. These findings indicate that

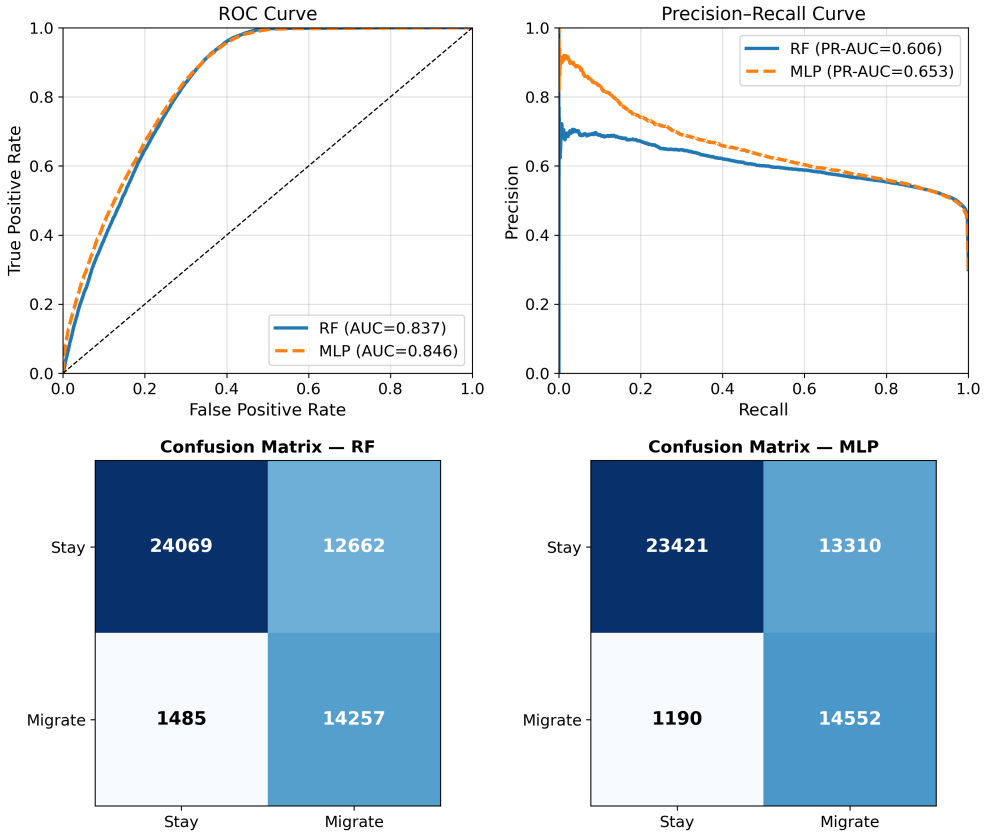


Figure 5: Comparison of RF and MLP classifiers on the test set. Top: ROC and PR curves with AUC scores. Bottom: confusion matrices at selected thresholds (0.50 for RF, 0.40 for MLP).

both deep and ensemble learning paradigms are effective, but the preferred choice may depend on deployment priorities.

While both models demonstrated solid performance, the achieved metrics also highlight limitations. PR AUC values (0.606–0.653) suggest that precision decreases at high recall, which could result in unnecessary migrations in highly sensitive deployments. Moreover, overall error rates remain non-trivial: each model misclassified over 12,000 "stay" events in the test set. This underscores the challenge of ensuring near-perfect reliability in safety-critical vehicular networks. Another limitation is that our experiments were restricted to only two model families (RF and MLP). More advanced sequential methods such as LSTMs or Transformers, or gradient boosting ensembles, may capture temporal dependencies or non-linear interactions more effectively.

From this comparative analysis, several lessons emerge: (1) Deep neural baselines like MLPs can indeed compete with ensemble methods on structured vehicular data, but ensembles may still offer valuable robustness. (2) Calibration of thresholds is critical—using 0.40 for MLP and 0.50 for RF significantly improved the balance between precision and recall. (3) Both models show trade-offs: RF reduces false positives but at the cost of slightly higher false negatives, while MLP improves recall but increases false positives. This suggests that model selection should be guided not only by global metrics (AUC) but also by the operational priorities of the target deployment (e.g., minimizing missed migrations vs. reducing needless handovers).

In summary, our study shows that both RF and MLP are promising for RSU migration prediction. The MLP slightly outperforms RF in discriminative metrics (ROC AUC and PR AUC), while the RF offers better control of false positives. These complementary strengths highlight that no single approach is universally superior; instead, careful consideration of deployment priorities and potential hybridization of methods will be key to achieving highly reliable handover prediction in vehicular edge computing.

7 Conclusions and Future Work

In this paper, we address the challenging problem of improving RSU handover efficiency and short-term, trajectory-informed decision-making in connected vehicular networks. We propose a data-driven solution that integrates deep learning and ensemble methods, using a MLP as a deep sequential baseline and a RF as a classical ensemble approach, to predict handover decisions from vehicle trajectories.

By anticipating mobility patterns, the framework enables proactive RSU migration, thereby reducing service interruptions and enhancing connectivity in vehicular edge networks.

Our experimental results show that both models achieve strong performance on the RSU migration prediction task, albeit with complementary strengths. The MLP demonstrates slightly higher discriminative capability, reflected in superior ROC AUC (0.846) and PR AUC (0.653) values. Conversely, the RF classifier delivers competitive accuracy and F1 scores while producing fewer false positives in confusion matrix analysis. Together, these findings suggest that both deep neural and ensemble learning methods are viable for this task, with their relative advantages depending on operational priorities—whether the goal is to maximize sensitivity to handover events (favoring the MLP) or to minimize unnecessary handovers (favoring the RF). Importantly, the comparative evaluation highlights that effective RSU migration prediction is not the exclusive domain of deep models; classical ensembles can perform equally well under certain conditions.

Despite these promising results, several limitations remain. The absolute accuracy and precision–recall balance are not yet sufficient for fully reliable deployment in safety-critical vehicular environments, as both models still produce a non-trivial number of misclassifications. Moreover, the scope of our study was limited to two modeling families; additional baselines such as gradient boosting machines, LSTMs, or Transformer-based predictors could provide further insight into optimal architectures for this problem. Finally, our experiments relied on a single dataset split; additional testing under varied network and traffic conditions is required to validate generalizability.

Looking ahead, several directions for future work emerge. One promising avenue is the use of LightGBM as a faster and more efficient alternative to Random Forest for large-scale classification, particularly with high-dimensional categorical datasets, while maintaining a strong balance between accuracy and performance. Another exciting direction is the adoption of Liquid Neural Networks (LNNs) in place of traditional LSTMs for real-time forecasting. LNNs offer dynamic adaptability to evolving data streams, enabling more accurate and responsive predictions.

Acknowledgement

This work was carried out in the context of project SELF4COOP (Self-optimizing Networked Edge Control for Cooperative Vehicle Autonomy), funded by the European Union under Next Generation EU, Mission 4 Component 2 - CUP E53D23000910001. Views and opinions expressed are those of the authors only and do not necessarily reflect those of the EU or the EU REA. Neither the EU nor the granting authority can be held responsible for them.

References

[1] Veronika Lesch, Martin Breitbach, Michele Segata, Christian Becker, Samuel Kounev, and Christian Krupitzer. 2022. An overview on approaches for coordination of platoons. *IEEE Transactions on Intelligent Transportation Systems*, 23, 8, 10049–10065. doi:10.1109/TITS.2021.3115908.

[2] Rajesh Rajamani. 2012. *Vehicle Dynamics and Control*. (2nd ed.). Springer.

[3] ETSI GS MEC. 2022. Multi-access Edge Computing (MEC); Framework and Reference Architecture. Tech. rep. ETSI.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. 2017. A survey on mobile edge computing: the communication perspective. *IEEE Communications Surveys & Tutorials*, 19, 4, 2322–2358.

[5] P. Mach and Z. Becvár. 2017. Mobile edge computing: a survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19, 3, 1628–1656.

[6] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: human trajectory prediction in crowded spaces. In *Proc. CVPR*, 961–971.

[7] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. 2017. Desire: distant future prediction in dynamic scenes with interacting agents. In *Proc. CVPR*, 336–345.

[8] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. 2020. Trajectron++: multi-agent generative trajectory forecasting with heterogeneous data. In *Proc. ECCV*.

[9] Jingwei Gao et al. 2020. Vectornet: encoding hd maps and agent dynamics from vectorized representation. In *Proc. CVPR*, 11525–11533.

[10] Scott Ettinger and et al. 2021. Large scale interactive motion forecasting for autonomous driving: the waymo open motion dataset. In *Proc. ICCV*.

[11] Kai Tan, Lei Zhang, et al. 2022. Intelligent handover algorithm for v2n communications with double-deep q-learning. *IEEE Transactions on Vehicular Technology*, 71, 7, 6858–6862.

[12] A. C. P. K. Siriwardhana et al. 2025. Optimizing handover mechanism in vehicular networks using deep learning and optimization techniques. *Computer Networks*, 219.

[13] R. Khoder et al. 2024. On using machine learning for vertical handover decision making in a vehicular platoon. In *Springer Preprint*.

[14] Florent Altché and Arnaud de La Fortelle. 2018. An lstm network for highway trajectory prediction. In *IEEE ITSC*.

[15] Rui Cheng et al. 2025. Vit-traj: a spatial–temporal coupling vehicle trajectory prediction model based on vision transformer. *Systems*, 13, 3, 147.

[16] Dabin Choi et al. 2021. Machine learning-based vehicle trajectory prediction using v2v communications and on-board sensors. *Electronics*, 10, 4, 420.

[17] Bin Lv et al. 2021. Task offloading and serving handover of vehicular edge computing networks based on trajectory prediction. *IEEE Access*, 9, 130793–130804.

[18] Michael Szeto et al. 2023. B-aware: blockage aware rsu scheduling for 5g enabled autonomous vehicles. *ACM Transactions on Embedded Computing Systems*, 22, 5s, 111.

[19] Niels Spring et al. 2025. Mach: multi-agent coordination for rsu-centric handovers. *arXiv preprint arXiv:2505.07827*.

[20] Zhengxin Yu, Jia Hu, Geyong Min, Zhiwei Zhao, Wang Miao, and M. Shamim Hossain. 2021. Mobility-aware proactive edge caching for connected vehicles using federated learning. *IEEE Transactions on Intelligent Transportation Systems*, 22, 8, 5341–5351. doi:10.1109/TITS.2020.3017474.

[21] Muhammad Asif Manzoor and Yasser Morgan. 2018. Vehicle make and model recognition using random forest classification for intelligent transportation systems. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 148–154. doi:10.1109/CCWC.2018.8301714.

[22] Wuxing Chen, Kaixiang Yang, Zhiwen Yu, Yifan Shi, and C. L. Philip Chen. 2024. A survey on imbalanced learning: latest research, applications and future directions. *Artificial Intelligence Review*, 57, 137. doi:10.1007/s10462-024-10759-6.

[23] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9, 8, 1735–1780.

[24] Jianxin Shi, Jinhao Chen, Yuandong Wang, Li Sun, Chunyang Liu, Wei Xiong, and Tianyu Wo. 2025. Motion forecasting for autonomous vehicles: a survey. *arXiv:2502.08664 [cs.RO]*. doi:10.48550/arXiv.2502.08664.

[25] Iskandar, Hajiar Yuliana, Hendrawan, Adriel Timoteo, Fabian Rafinanda Benyamin, and Naufal Bhanu Anargyarahman. 2025. User trajectory prediction in cellular networks using multi-step lstm approaches: case study and performance evaluation. *IEEE Access*, 13, 42143–42166. doi:10.1109/ACCESS.2025.3547811.

[26] Md Mehedi Hasan, Sungoh Kwon, and Sangchul Oh. 2019. Frequent-handover mitigation in ultra-dense heterogeneous networks. *IEEE Transactions on Vehicular Technology*, 68, 1, 1035–1040. doi:10.1109/TVT.2018.2874692.

[27] Tim Tiernan, Nicholas Richardson, Philip Azeredo, Wassim G Najm, Taylor Lochrane, et al. 2017. Test and evaluation of vehicle platooning proof-of-concept based on cooperative adaptive cruise control. Tech. rep. John A. Volpe National Transportation Systems Center (US).

[28] Diederik P. Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *CoRR*, abs/1412.6980. https://api.semanticscholar.org/CorpusID:6628106.

[29] Leo Breiman. 2001. Random forests. *Machine Learning*, 45, 5–32.

[30] F. Pedregosa et al. 2011. Scikit-learn: machine learning in Python. In *Journal of Machine Learning Research*. Vol. 12. JMLR.org, 2825–2830.

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

[32] Stef Van Buuren and Karin Groothuis-Oudshoorn. 2011. Mice: multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45, 3, 1–67. doi:10.18637/jss.v045.i03.