remote sensing

MDPI

Article

Multipath-Characteristic-Based Clustering for GNSS Pseudorange Error Compensation Coupled with Reliability-Indicator-Based Adaptive Kalman Filtering in Complex Urban Navigation

Jianhui Quan [1], Yuhao Chen [2], Hengyu Zhang [2], Rui Sun [2,*], Tinglun Song [1] and Washington Yotto Ochieng [3]

[1] College of Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; jianhui.quan@nuaa.edu.cn (J.Q.); songtinglun@nuaa.edu.cn (T.S.)
[2] College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; yuhao_chen@nuaa.edu.cn (Y.C.); hengyu.zhang@nuaa.edu.cn (H.Z.)
[3] Department of Civil and Environmental Engineering, Imperial College London, London SW7 2AZ, UK; w.ochieng@imperial.ac.uk
* Correspondence: rui.sun@nuaa.edu.cn

Highlights

What are the main findings?

- An IDEC-based clustering method groups GNSS signals by signal features rather than spatial grids, and cluster-specific AutoML models are built for better pseudorange error prediction in complex urban areas.
- A prediction reliability indicator dynamically adjusts measurement weights in the adaptive Kalman filter, helping MCC-AKF achieve an overall 3D RMSE of 14.53 m, improving positioning accuracy by 40.96%, 20.70%, and 13.38% over traditional EKF, 2D-EKF, and 3D-EKF, respectively.

What are the implications of the main findings?

- Grouping GNSS measurements by signal features avoids the mixing of different multipath conditions inside fixed spatial grids.
- Measuring prediction reliability using validation errors and feature deviations protects GNSS/IMU sensor fusion from poor corrections, supporting more reliable navigation in complex urban areas.

Abstract

The integration of the global navigation satellite system (GNSS) and inertial measurement unit (IMU) is crucial for reliable vehicular positioning. However, multipath interference (MI) and non-line-of-sight (NLOS) reception introduce severe GNSS pseudorange measurement errors, leading to significant positioning degradation of GNSS/IMU integrated navigation systems of the vehicle in challenging urban environments. Existing two-dimensional (2D) and three-dimensional (3D) spatial grid-based pseudorange error prediction algorithms suffer from environmental heterogeneity and lack robustness in noise covariance tuning. To tackle these issues, this paper proposes a multipath-characteristic-based clustering algorithm integrated with an automated machine learning (AutoML) framework for pseudorange error compensation and adaptive GNSS/IMU integration system (MCC-AKF) in complex urban environments. First, an improved deep embedded clustering (IDEC) algorithm partitions multipath environments using physical signal traits rather than spatial grids. Second, cluster-specific regression models are established via an AutoML framework to predict and compensate for pseudorange errors. Finally, a prediction reliability indicator

Check for updates

Academic Editor: Gino Dardanelli

Received: 2 November 2026
Revised: 25 November 2026
Accepted: 26 November 2026
Published: 27 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

(PRI) is designed to dynamically scale the measurement noise covariance in an adaptive tightly coupled extended Kalman filter (EKF). Experimental results in complex urban areas demonstrate that the proposed algorithm achieves a 3D root mean square error (RMSE) of 14.53 m, improving positioning accuracy by 40.96%, 20.70%, and 13.38% compared with the traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based fusion algorithms, respectively.

Keywords: adaptive Kalman filter; deep embedded clustering; automated machine learning; pseudorange error compensation; urban navigation

1. Introduction

The emergence of mission-critical applications within intelligent transportation systems (ITS) demands high-quality positioning, navigation, and timing (PNT) information to support accurate and reliable navigation performance [1,2]. A global navigation satellite system (GNSS) integrated with an inertial measurement unit (IMU) represents one of the most widely adopted solutions for delivering PNT information to vehicles in urban areas [3,4]. However, in complex urban environments, raw GNSS measurements suffer from signal blockages, multipath interference (MI), and non-line-of-sight (NLOS) reception, leading to the rapid degradation or even divergence of traditional GNSS/IMU integrated navigation systems [5–7]. Consequently, effectively mitigating multipath and NLOS errors prior to or during the information fusion process remains a critical practical challenge for GNSS/IMU integrated navigation systems in urban areas.

To enhance the performance of GNSS/IMU integrated navigation systems in urban environments, current research mainly focuses on two critical aspects: (1) GNSS multipath mitigation and (2) the improvement of filtering algorithms for sensor fusion.

In terms of the first aspect, early GNSS multipath mitigation algorithms primarily relied on physical hardware optimization. At the receiver and antenna level, techniques such as narrow correlator spacing [8], multipath-estimating delay-locked loops [9,10], choke ring groundplanes [11], and various dual-polarization architectures [12] were proposed to suppress multipath effects physically. Recently, extensive studies have adopted machine learning (ML) [13] to explore the correlations between GNSS measurement variables and signal reception types. For instance, Phan et al. [14] adopted a support vector machine (SVM); Li et al. [15] utilized a random forest (RF); Sun et al. [16] implemented a gradient-boosted decision tree (GBDT); and Yozevitch et al. [17] explored a decision tree (DT) to classify signals into line-of-sight (LOS) and NLOS categories. These classification models typically utilize features such as the carrier-to-noise density ratio (C/N_0) [18] and multi-frequency correlations [19], while consistency checks and C/N_0-based weighting schemes were also incorporated to assess signal reception quality [20]. Furthermore, advanced techniques, including fuzzy logic [21], ML classifiers [22], and multiple fault detection algorithms [23], have been developed to assign soft measurement weights or eliminate abnormal GNSS measurements. However, these classification algorithms suffer from a fundamental drawback: excluding measurements classified as multipath or NLOS can reduce the number of available satellites and weaken satellite geometry, potentially leaving insufficient measurements for reliable positioning in urban areas.

As inaccurate pseudorange measurements are the main source of positioning errors arising from multipath effects, regression-based prediction models have been proposed to characterize and correct pseudorange errors. Specific modeling algorithms include GBDT regression [24] and adaptive neuro-fuzzy inference [25]. To build these prediction models,

selecting optimal input variables is crucial. Early regression algorithms were developed for static positioning using ML models with selected variables to predict pseudorange errors and correct measurements directly. For dynamic positioning, the point-based correction (PBC) algorithm was proposed [26]. With prior information collected from the target area, pseudorange error prediction rules are extracted by incorporating reference point coordinates, carrier-to-noise density ratios, and satellite elevations. Meanwhile, grid-based correction (GBC) algorithms were proposed to partition the dataset. By closely arranging hexagonal grids to divide urban areas, GBC establishes corresponding prediction models for different multipath environments. Based on this idea, algorithms employing two-dimensional (2D) grids [26] and subsequently three-dimensional (3D) grids [27] have been proposed to predict and correct pseudorange errors via ML techniques.

However, geometric grids divide the urban area solely based on spatial coordinates, assuming that points close in space share similar multipath conditions, an assumption that often fails in complex urban environments. In reality, surrounding buildings can cause completely different reflections within just a few meters, and even at the same spot, signals from different satellites can vary greatly depending on their directions and elevations. Rigid geometric grids cannot distinguish these physical differences and simply treat all signals in the same cell as identical. To solve these problems, recent studies have started to use clustering algorithms to group complex GNSS data. For example, clustering algorithms have been used to improve ambiguity resolution in complex environments [28]. Point cloud clustering is used to detect dynamic obstacles for NLOS signal exclusion [29]. These studies have proven the effectiveness of clustering-based approaches.

In terms of the second aspect, the improvement of filtering algorithms for sensor fusion focuses on adaptively adjusting state estimates when abnormal GNSS measurements occur. Conventional loosely and tightly coupled extended Kalman filters (EKFs) have served as the standard baseline [30]. To handle measurement errors, researchers developed innovation-based adaptive Kalman filters (AKFs) [31–35], dual-w-test quality control algorithms [36], and robust adaptive filtering [37–39]. Furthermore, predicted pseudorange errors have been used to build adaptive indicators to adjust the measurement noise covariance matrix (**R**). For example, using predicted pseudorange errors to adjust the measurement noise covariance improved 3D positioning accuracy by 55% in tightly coupled systems [40]. Recently, Sun et al. [27] proposed a model reliability indicator (MRI) in 3D grids to adjust **R** based on satellite geometry and predicted pseudorange errors, improving 3D positioning accuracy by 52%. However, these adaptive fusion strategies still have weaknesses. They rely on the accuracy of grid-based predictions. If the predicted error is incorrect, the adaptive indicator can make positioning results even worse.

In conclusion, existing algorithms have two main drawbacks. First, existing algorithms divide urban environments using rigid spatial grids rather than multipath characteristics. This fails to ensure similar multipath conditions within each grid, leading to inaccurate pseudorange error predictions. Second, existing adaptive fusion strategies fail to robustly quantify prediction reliability in urban environments, which may degrade positioning accuracy when inaccurate error predictions are introduced.

In order to address these drawbacks and improve the positioning accuracy of GNSS/IMU integrated navigation systems in complex urban environments, we propose a multipath-characteristic-based clustering algorithm integrated with an automated machine learning (AutoML) framework for pseudorange error compensation and an adaptive GNSS/IMU integration system (MCC-AKF) in complex urban environments. Regarding environmental dependence, both the multipath-characteristic-based clustering model and the cluster-specific pseudorange error prediction models are affected by the building distributions and signal propagation environments in the training area. When the proposed

method is applied to other urban areas not included in the training dataset, differences in building layouts, street structures, and reflection surfaces can change the input feature distributions and the relationship between signal features and pseudorange errors. Because studying model generalization across different cities requires massive datasets covering diverse urban areas, this paper focuses on multipath-characteristic-based clustering and error modeling within a specific complex urban environment to achieve better positioning as a foundation for robust GNSS navigation, while studies on the generalization of the regression models are beyond the scope of this paper and are left for future work. The main contributions of this paper are as follows:

(1) A multipath-characteristic-based clustering algorithm using the improved deep embedded clustering (IDEC) combined with an AutoML framework is proposed, which partitions heterogeneous urban environments based on multipath characteristics and compensates for pseudorange errors via cluster-specific regression models, thus solving the problem of environmental heterogeneity and inaccurate predictions caused by rigid spatial grid partitioning.
(2) A prediction reliability indicator (PRI)-based adaptive Kalman filtering strategy is proposed, which dynamically scales the measurement noise covariance by jointly evaluating model cross-validation accuracy and observational distribution deviations, thus solving the problem of filter degradation caused by unreliable pseudorange error predictions in complex urban environments.

The remainder of this paper is organized as follows: Section 2 details the proposed approach, including the overall architecture, feature engineering, multipath-characteristic-based partitioning, AutoML modeling, and the tightly coupled integration scheme. Section 3 presents the experimental setup, vehicular field tests, and a comparative analysis of the results in challenging urban environments. Section 4 discusses the findings, strengths, limitations, and future directions. Finally, Section 5 concludes the paper and discusses future research directions.

2. Materials and Methods

This section details the proposed multipath-characteristic-based clustering for pseudorange error compensation and adaptive GNSS/IMU sensor fusion. The overall algorithm is organized into two parts. The first part focuses on multipath-characteristic-based clustering for pseudorange error prediction and compensation, which includes extracting key multipath features, labeling pseudorange errors, dividing complex multipath environments using the IDEC algorithm, and training cluster-specific prediction models with AutoML. The second part focuses on tightly coupled GNSS/IMU sensor fusion, which uses the designed PRI to adaptively adjust the Kalman filter noise covariances, providing accurate positioning results.

2.1. Overview of the Framework

The overall framework of the proposed MCC-AKF-based fusion algorithm is illustrated in Figure 1. The architecture consists of an offline training phase and an online positioning phase.

The offline phase focuses on constructing the IDEC clustering model and cluster-specific AutoML regression models. The inputs to this phase are raw GNSS measurements and high-precision reference data. First, feature engineering extracts four key variables (α, θ, v, and C/N_0) from raw GNSS data, outputting the five-dimensional feature vector $\mathbf{x}$, while the pseudorange error labels $\Delta\rho$ are calculated using both the raw GNSS data and reference data. Second, the feature vectors are used as inputs to the IDEC model to perform multipath-characteristic-based clustering, thereby training the network and partitioning

the training data into distinct multipath clusters. Finally, using the feature vectors and pseudorange error labels, AutoML trains customized regression models for each valid cluster, outputting the trained pseudorange error prediction models.

The online phase achieves pseudorange error correction and adaptive sensor fusion. The inputs to this phase comprise raw GNSS measurements, IMU data, the trained IDEC model, and the trained AutoML regression models. First, GNSS measurements are converted into the feature vector **x**, which is classified by the trained IDEC model and then used as input to the corresponding AutoML regression model to output the predicted pseudorange error $\Delta\hat{\rho}$, generating the corrected pseudorange measurements. Second, the PRI is calculated from the cross-validation accuracy of the prediction model and the observational deviation from the training distribution to adaptively adjust the measurement noise covariance matrix (**R**). Finally, the corrected pseudoranges, IMU mechanization outputs, and adjusted covariance matrices are processed by the tightly coupled Kalman filter, outputting the optimal navigation solution.

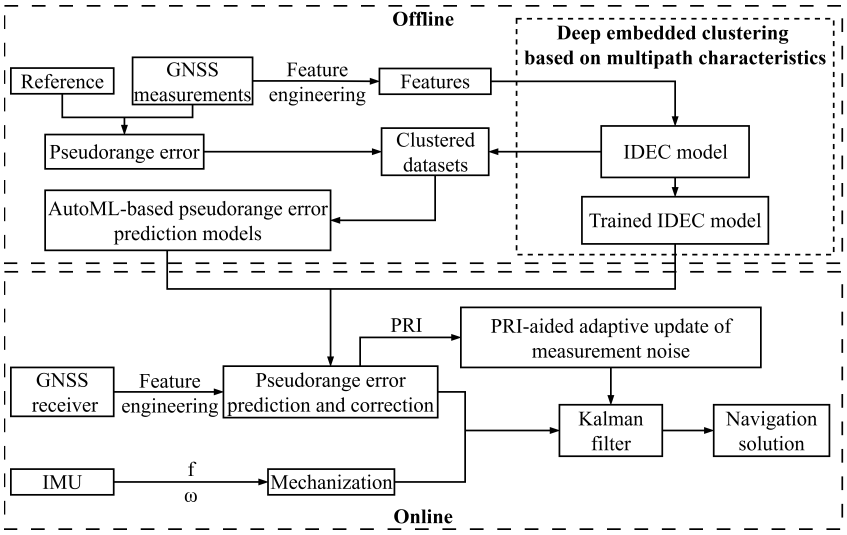

Figure 1. Framework of the MCC-AKF-based fusion algorithm.

2.2. Feature Engineering

Raw GNSS measurements contain various variables that reflect satellite geometry and signal propagation conditions. In previous research [24,26,27,40], it was demonstrated that using too many input variables increases computational cost without guaranteeing higher prediction accuracy. To balance modeling efficiency and prediction performance, four key variables are extracted from the receiver: satellite azimuth angle (α), satellite elevation angle (θ), pseudorange residual (v), and carrier-to-noise density ratio (C/N_0). All these selected variables have clear physical meanings supported by previous studies [24,27,41,42]: (1) the satellite azimuth reflects the direction of surrounding obstacles; (2) the satellite elevation directly indicates signal blockage risks; (3) the pseudorange residual directly shows the measurement error; and (4) C/N_0 indicates signal loss caused by reflection and blockage.

The distributions of these extracted variables are shown in Figure 2. As illustrated in Figure 2a,b, both the azimuth (α) and elevation (θ) angles show several clear peaks. This is because tall buildings along the urban streets only allow satellites to be observed from specific directions and heights. For the pseudorange residual (v) in Figure 2c, most values are concentrated close to zero, but long tails appear on both sides with errors exceeding $\pm$20 m, indicating large multipath errors. Regarding C/N_0 in Figure 2d, although most measurements have strong signal power above 45 dB, many signals drop below 35 dB because of signal blockage, MI, and NLOS reception.

According to their physical properties and statistical distributions, specific processing methods are designed for each variable as follows.

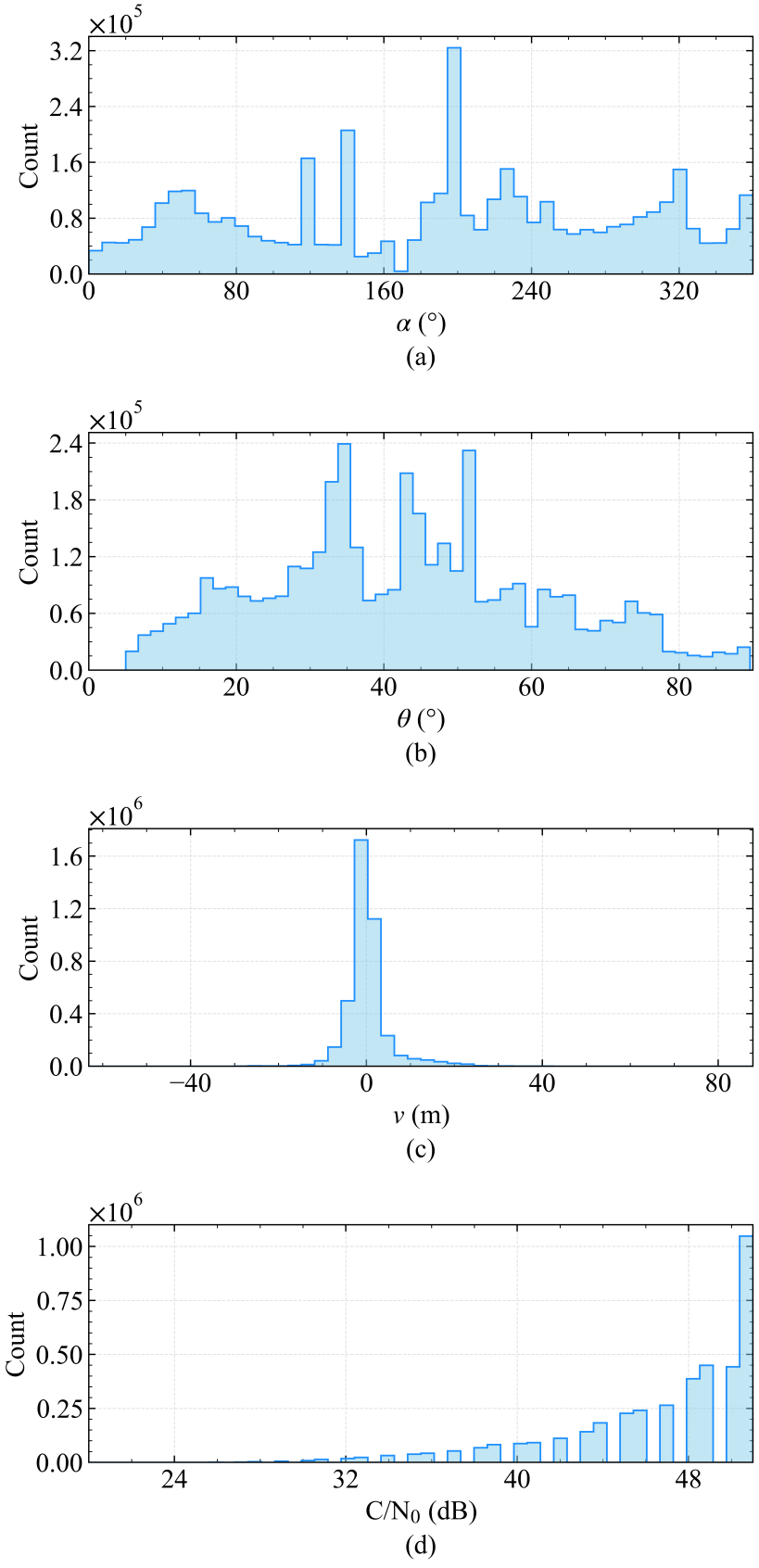

Figure 2. Distributions of extracted variables in the dataset: (**a**) satellite azimuth angle α; (**b**) satellite elevation angle θ; (**c**) pseudorange residual v; and (**d**) carrier-to-noise density ratio C/N_0.

(1) Satellite azimuth angle (α): In urban areas, tall buildings are typically arranged along specific street orientations, meaning that multipath reflections strongly depend on satellite direction [27]. Therefore, the satellite azimuth angle is essential for capturing directional obstacle layouts. However, α is a circular variable ranging from 0° to 360°, where 0° and 360° represent the same physical direction. In order to eliminate numerical jumps at the boundary, α is decomposed into two continuous trigonometric components: $\sin(\alpha)$ and $\cos(\alpha)$.

(2) Satellite elevation angle (θ): Higher elevation angles typically correspond to lower obstruction probabilities and fewer reflections, while low-elevation signals are more easily affected by multipath effects [24,26]. In order to standardize the data distribution and eliminate numerical scale differences, z-score normalization is expressed as

$$\theta_{\text{scaled}} = \frac{\theta - \mu_\theta}{\sigma_\theta} \tag{1}$$

where μ_θ and σ_θ denote the mean and standard deviation of θ across the training dataset.

(3) Pseudorange residual (v): The pseudorange residual represents the difference between measured pseudoranges and estimated positions, which captures unmodeled multipath errors [24]. For the i-th satellite at the current epoch, the pseudorange residual v_i is calculated as

$$v_i = \tilde{\rho}_i - \left(\hat{R}_i + c\,\widehat{\delta t}_{\text{r}} + c\,\widehat{\Delta t}_{\text{sys}} - c\,\delta t_{\text{s},i} + \hat{I}_i + \hat{T}_i\right) \tag{2}$$

where $\tilde{\rho}_i$ is the measured pseudorange after satellite group delay and code bias corrections; $\hat{R}_i$ is the geometric distance between the broadcast satellite position and the estimated receiver position $\hat{\mathbf{r}}$; c is the speed of light; $\widehat{\delta t}_{\text{r}}$ is the estimated receiver clock bias; $\widehat{\Delta t}_{\text{sys}}$ is the estimated time offset between other satellite systems and GPS; $\delta t_{\text{s},i}$ is the broadcast satellite clock bias; and $\hat{I}_i$ and $\hat{T}_i$ are the ionospheric and tropospheric delays calculated from standard broadcast models at $\hat{\mathbf{r}}$.

The receiver position $\hat{\mathbf{r}}$ and clock estimates ($\widehat{\delta t}_{\text{r}}$, $\widehat{\Delta t}_{\text{sys}}$) are obtained at each epoch using weighted least squares (WLS) from current GNSS measurements only:

$$\Delta\hat{\mathbf{x}} = \left(\mathbf{H}^\top\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^\top\mathbf{W}\mathbf{v} \tag{3}$$

where $\hat{\mathbf{x}} = [\hat{\mathbf{r}}^\top,\, c\,\widehat{\delta t}_{\text{r}},\, c\,\widehat{\Delta t}_{\text{sys}}]^\top$ is updated step by step until $\|\Delta\hat{\mathbf{x}}\|_2 < 10^{-4}$ m, $\mathbf{H}$ is the design matrix, $\mathbf{W}$ is the weight matrix based on satellite elevation and model variances, and $\mathbf{v}$ is the residual vector. v_i is computed at this final step and can be obtained directly during online processing without any reference trajectory. In order to standardize the data distribution and eliminate numerical scale differences, z-score normalization is expressed as

$$v_{\text{scaled}} = \frac{v - \mu_v}{\sigma_v} \tag{4}$$

where μ_v and σ_v denote the mean and standard deviation of v across the training dataset.

(4) Carrier-to-noise density ratio (C/N_0): Signal power directly reflects the degree of multipath contamination, as reflected signals experience significant power loss compared to direct signals [27,41]. In order to standardize the data distribution and eliminate numerical scale differences, z-score normalization is expressed as

$$(C/N_0)_{\text{scaled}} = \frac{C/N_0 - \mu_{C/N_0}}{\sigma_{C/N_0}} \tag{5}$$

where μ_{C/N_0} and σ_{C/N_0} denote the mean and standard deviation of C/N_0 across the training dataset.

To avoid temporal data leakage, the testing dataset is strictly separated from the training dataset, and the mean and standard deviation used for z-score normalization are computed solely from the training dataset. Moreover, the training data are randomly shuffled using a fixed random seed of 42.

In order to analyze the linear relationship between these processed features and the multipath error $\Delta\rho$, we calculate the Pearson correlation coefficients as shown in Figure 3. The heatmap clearly shows that the scaled pseudorange residual v_{scaled} has the strongest negative correlation of -0.55 with the multipath error. The other four features also show clear positive correlations. Specifically, the correlation values are 0.18 for the sine of the azimuth $\sin\alpha$, 0.36 for the cosine of the azimuth $\cos\alpha$, 0.18 for the scaled elevation θ_{scaled}, and 0.19 for the scaled carrier-to-noise density ratio $(C/N_0)_{\text{scaled}}$. This statistical result proves that all five selected features contain useful information, making it reasonable to combine them for training the prediction models.

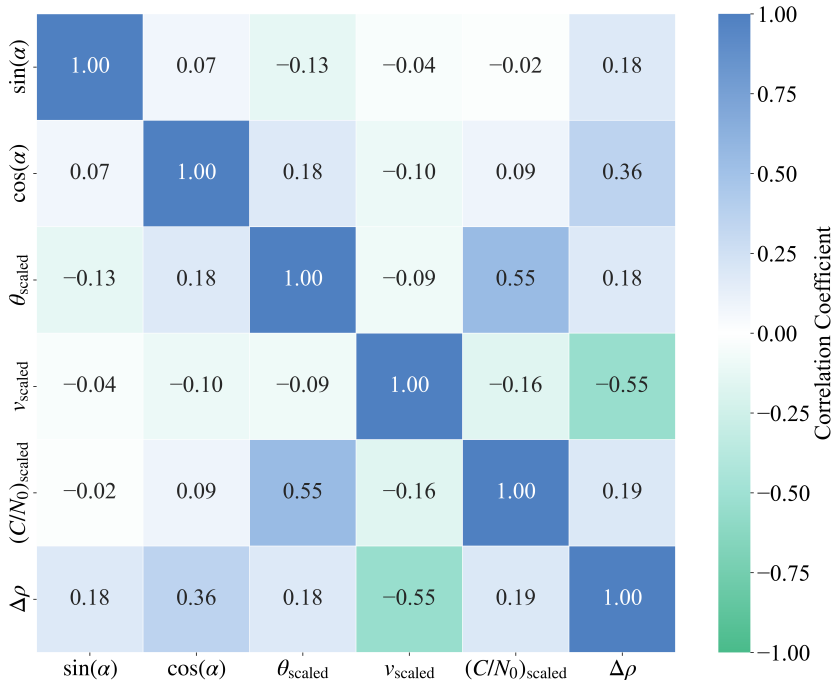

Figure 3. Correlation matrix of the processed features and multipath errors, where the satellite elevation angle, pseudorange residual, and carrier-to-noise density ratio are z-score standardized.

Consequently, the final input feature vector for the i-th measurement is expressed as

$$\mathbf{x}_i = [\sin(\alpha_i), \cos(\alpha_i), \theta_{\text{scaled},i}, v_{\text{scaled},i}, (C/N_0)_{\text{scaled},i}]^\top \tag{6}$$

2.3. Pseudorange Error Labeling

To extract unmodeled multipath errors for supervised learning, the raw GNSS measurements need to be explicitly formulated and corrected. For a specific satellite i, the raw measured pseudorange $\rho^{(i)}$ can be expressed as

$$\rho^{(i)} = R^{(i)} + c(\delta t_{\text{r}} - \delta t_{\text{s}}^{(i)}) + I^{(i)} + T^{(i)} + \varepsilon^{(i)} \tag{7}$$

$$R^{(i)} = \sqrt{\left(x_{\text{s}}^{(i)} - x_{\text{r}}\right)^2 + \left(y_{\text{s}}^{(i)} - y_{\text{r}}\right)^2 + \left(z_{\text{s}}^{(i)} - z_{\text{r}}\right)^2} \tag{8}$$

where $R^{(i)}$ represents the true geometric distance between the receiver and satellite i. The 3D coordinates of the receiver and the i-th satellite are denoted by $(x_{\text{r}}, y_{\text{r}}, z_{\text{r}})$ and $(x_{\text{s}}^{(i)}, y_{\text{s}}^{(i)}, z_{\text{s}}^{(i)})$,

respectively. The parameter c represents the speed of light. The terms δt_r and $\delta t_s^{(i)}$ denote the clock biases of the receiver and the i-th satellite. $I^{(i)}$ and $T^{(i)}$ are the ionospheric and tropospheric delays, respectively. The term $\varepsilon^{(i)}$ accounts for the unmodeled multipath errors, which are primarily caused by MI and NLOS reception in urban environments.

The corrected pseudorange $\rho_c^{(i)}$ is expressed as

$$\rho_c^{(i)} = R^{(i)} + c\left(\Delta\delta t_r - \Delta\delta t_s^{(i)}\right) + \Delta I^{(i)} + \Delta T^{(i)} + \varepsilon^{(i)} \tag{9}$$

where the geometric range $R^{(i)}$ is calculated from the known reference position of the receiver and the satellite position from the broadcast ephemeris; $\Delta\delta t_r$ is the receiver clock residual after removing the receiver clock bias estimated from the pseudorange equations with the known reference position; $\Delta\delta t_s^{(i)}$ is the remaining satellite clock residual after applying the broadcast ephemeris satellite clock offset and satellite group delay and code bias corrections; $\Delta I^{(i)}$ is the remaining ionospheric delay residual after applying the Klobuchar broadcast model; $\Delta T^{(i)}$ is the remaining tropospheric delay residual after applying the Saastamoinen model; and $\varepsilon^{(i)}$ is the unmodeled error mainly caused by multipath and NLOS reception.

The pseudorange error $\Delta\rho^{(i)}$ is then expressed as

$$\Delta\rho^{(i)} = \rho_c^{(i)} - R^{(i)} = c\left(\Delta\delta t_r - \Delta\delta t_s^{(i)}\right) + \Delta I^{(i)} + \Delta T^{(i)} + \varepsilon^{(i)} \tag{10}$$

Because the large common receiver clock bias has been estimated and removed and the broadcast models reduce most of the systematic satellite clock and atmospheric errors, the remaining residuals $\Delta\delta t_r$, $\Delta\delta t_s^{(i)}$, $\Delta I^{(i)}$, and $\Delta T^{(i)}$ are relatively small compared with severe urban multipath and NLOS errors. This leaves the unmodeled term $\varepsilon^{(i)}$ as the dominant part of $\Delta\rho^{(i)}$. Accordingly, $\Delta\rho^{(i)}$ is extracted as the target label.

To understand how different signal conditions affect the multipath errors, the relationship between the processed features and the corresponding multipath errors was evaluated. Figure 4 illustrates the data distribution of this relationship to show why these features are chosen for the prediction model.

These scatter plots show the relationship between the processed features and the pseudorange errors. The color bar on the right shows the number of data points, where yellow means more points and blue means fewer. In all plots, most data points cluster around the zero line, meaning the pseudorange errors are usually small. However, large pseudorange errors clearly occur under certain conditions.

First, in Figure 4a,b for $\sin(\alpha)$ and $\cos(\alpha)$, the pseudorange errors are evenly spread across the entire horizontal range without showing obvious changes from left to right. Second, in Figure 4c for the scaled satellite elevation angle θ_{scaled}, the spread of pseudorange errors changes with the angle. When θ_{scaled} takes negative values on the left side, the pseudorange errors become much larger and spread widely. As the angle increases to positive values on the right side, the pseudorange errors remain closer to zero. Third, in Figure 4d for the scaled pseudorange residual v_{scaled}, the data points form a clear downward shape, showing a strong negative relationship between v_{scaled} and the pseudorange error $\Delta\rho$. Finally, in Figure 4e for the scaled carrier-to-noise density ratio $(C/N_0)_{\text{scaled}}$, strong signals on the right side produce small pseudorange errors that are tightly concentrated near zero. However, as $(C/N_0)_{\text{scaled}}$ drops to negative values on the left side, the pseudorange errors become much larger. In conclusion, this figure clearly shows that large pseudorange errors usually occur under specific feature conditions. This proves that these extracted features contain useful information and are suitable for training the prediction models.

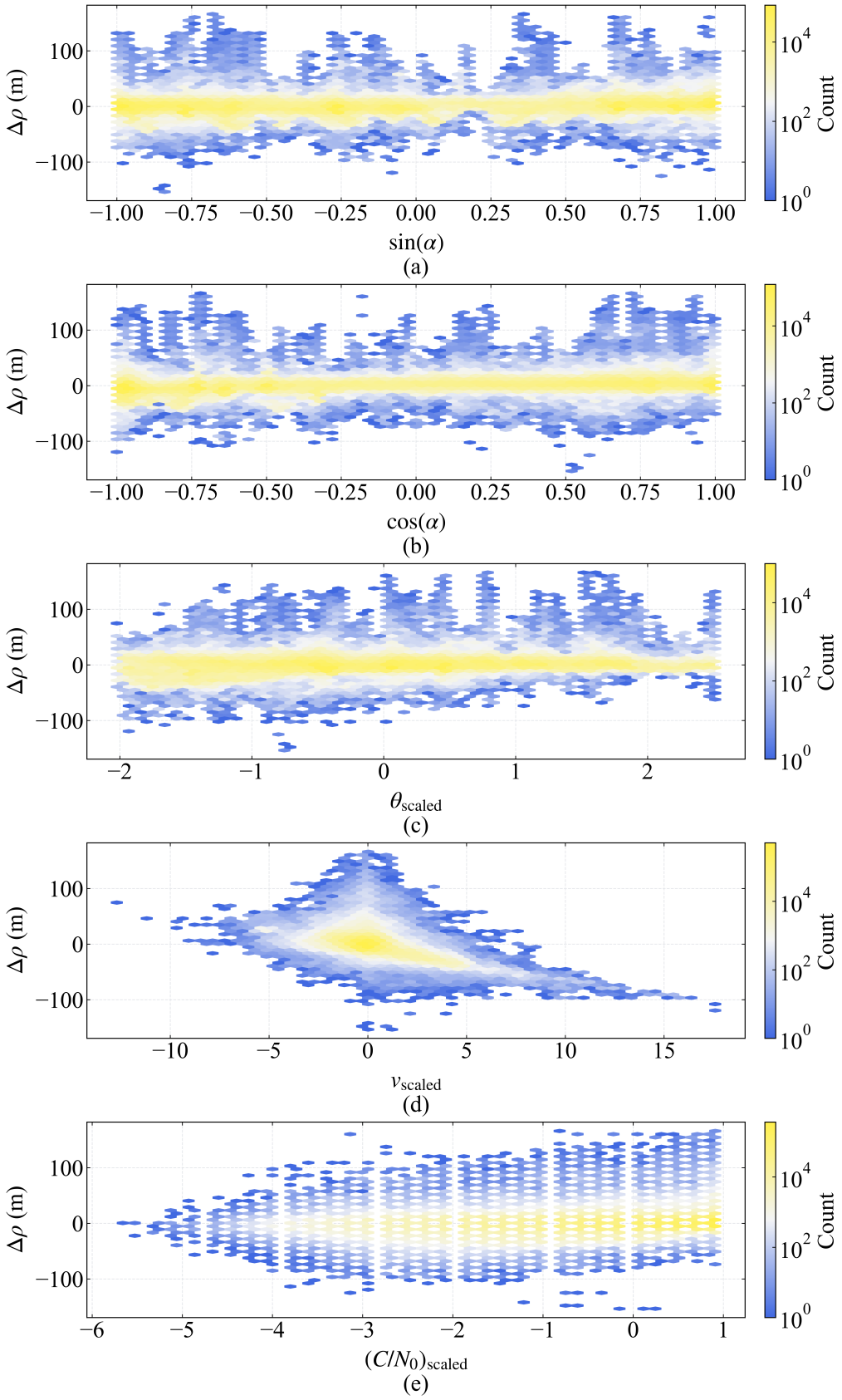

Figure 4. Data distribution of the processed features and multipath errors, where the satellite elevation angle, pseudorange residual, and carrier-to-noise density ratio are z-score standardized: (**a**) sine of satellite azimuth angle $\sin(\alpha)$; (**b**) cosine of satellite azimuth angle $\cos(\alpha)$; (**c**) scaled satellite elevation angle θ_{scaled}; (**d**) scaled pseudorange residual v_{scaled}; and (**e**) scaled carrier-to-noise density ratio $(C/N_0)_{\text{scaled}}$.

2.4. IDEC for Multipath Environment Partitioning

To overcome the limitations of geometric grids, we adopt the IDEC algorithm [43,44] to perform multipath-characteristic-based clustering, grouping GNSS measurements directly by physical signal traits instead of spatial coordinates. Specifically, IDEC incorporates an autoencoder to preserve the local structure of the data-generating distribution while jointly optimizing cluster assignments.

The framework of the IDEC algorithm is shown in Figure 5. It takes the five-dimensional feature vector $\mathbf{x}$ as input and outputs a specific cluster label c for each measurement, grouping signals with similar multipath conditions together while separating different multipath environments. To achieve this, the model jointly optimizes two tasks: a reconstruction task to preserve local structural features and a clustering task to map the data into target clusters.

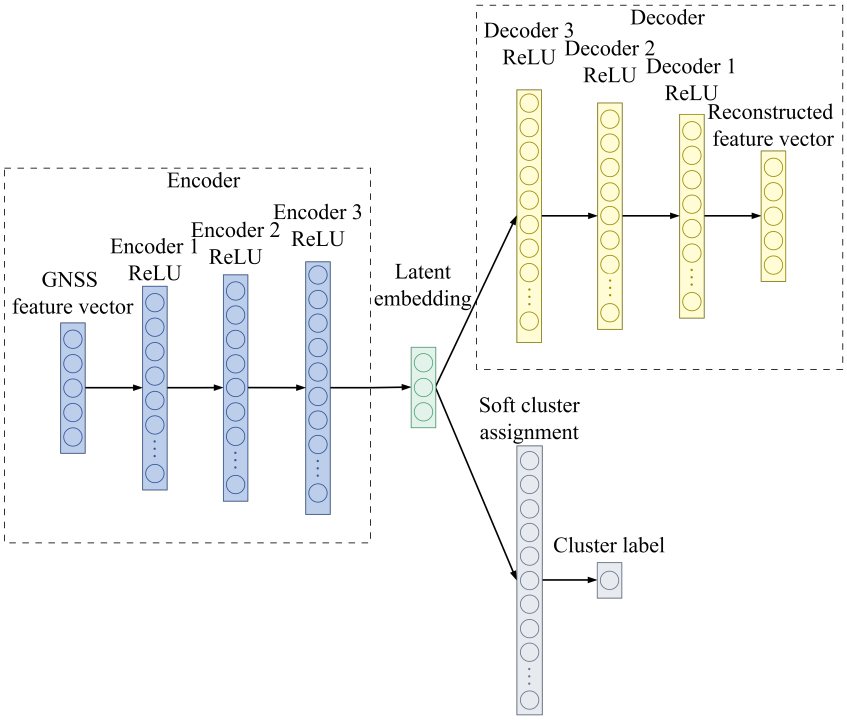

Figure 5. Neural network architecture.

The IDEC algorithm proceeds through the following phases:

(1) Pre-training the autoencoder. First, an autoencoder is trained to reconstruct the original input. This is done by minimizing the mean squared error (L_{MSE}) between the input vector $\mathbf{x}_i$ and the output vector $\mathbf{x}'_i$, which is expressed as

$$L_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{x}'_i\|_2^2 \tag{11}$$

where N is the batch size. This step helps the network learn the physical properties of the multipath features and transform them into hidden features.

(2) Initializing the cluster centers. Next, the trained encoder maps the training dataset into hidden features $\{\mathbf{z}_i\}_{i=1}^{M}$, where M is the total number of measurements in the training dataset. Then, the standard K-means algorithm is used to find K initial cluster centers $\{\boldsymbol{\mu}_j\}_{j=1}^{K}$ among these hidden features. We choose $K = 500$ because it achieves a good balance between environmental details and sample numbers. Specifically, if K is too small, signals with different multipath conditions are mixed into the same

cluster, which lowers the accuracy of error modeling. On the other hand, if K is too large, many clusters have too few samples for reliable model training, so $K = 500$ is selected to provide fine partitioning while keeping enough samples in most clusters.

(3) Calculating the assignment probabilities. After finding the centers, the algorithm calculates how similar each hidden feature $\mathbf{z}_i$ is to each cluster center $\boldsymbol{\mu}_j$. It uses the Student's t-distribution to generate a soft probability q_{ij}, which is expressed as

$$q_{ij} = \frac{\left(1 + \|\mathbf{z}_i - \boldsymbol{\mu}_j\|_2^2\right)^{-1}}{\sum_{j'=1}^{K}\left(1 + \|\mathbf{z}_i - \boldsymbol{\mu}_{j'}\|_2^2\right)^{-1}} \tag{12}$$

To improve cluster purity and put more emphasis on data points assigned with high confidence, a target distribution p_{ij} is calculated based on q_{ij}, which is expressed as

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'=1}^{K}\left(q_{ij'}^2 / f_{j'}\right)} \tag{13}$$

where $f_j = \sum_{i=1}^{M} q_{ij}$ represents the total soft frequency of cluster j in the training dataset.

(4) Joint training and final clustering. Finally, the network is trained by minimizing a combined loss function. This function balances the reconstruction error L_{MSE} and the clustering error L_{KL}, which is expressed as

$$L = L_{\mathrm{MSE}} + \gamma L_{\mathrm{KL}}(P \parallel Q) = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i - \mathbf{x}_i'\|_2^2 + \frac{\gamma}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} p_{ij}\log\frac{p_{ij}}{q_{ij}} \tag{14}$$

where the balance parameter γ is set to 0.1 [44]. The target distribution P is updated in each training round. The training stops when fewer than 0.1% of the measurements change their cluster groups [44]. After training, each GNSS measurement receives its final cluster label by choosing the center with the highest probability.

During IDEC training, the shuffled training data are fed into the network, and a fixed random seed of 42 is used for network initialization, K-means center initialization, and batch shuffling to ensure reproducible clustering results.

As illustrated in Figure 5, the network combines a symmetric autoencoder with a clustering layer. First, the encoder takes the five-dimensional feature vector as input and passes it through three hidden layers with 16, 32, and 64 nodes. Each hidden layer uses the rectified linear unit (ReLU) activation function. Next, a linear layer maps the features into a three-dimensional latent vector. From this three-dimensional vector, the network completes two tasks at the same time. On one path, the decoder takes this three-dimensional vector, passes it through three hidden layers with 64, 32, and 16 nodes using the same ReLU function, and outputs a five-dimensional vector to reconstruct the original input. On the other path, the clustering layer takes the same three-dimensional vector to calculate its distance to 500 cluster centers, outputting the soft probability of assigning each measurement to each cluster.

For model optimization, the adaptive moment estimation (Adam) optimizer is utilized with an initial learning rate of 10^{-3} for both the autoencoder pre-training and the IDEC joint optimization stages [44]. During pre-training, a cosine annealing schedule decays the learning rate to a minimum of 10^{-6} over 200 epochs with an early-stopping patience of 20 epochs. The batch size is set to 4096, and gradient clipping with a maximum norm threshold of 5.0 is applied to ensure numerical stability during backpropagation.

2.5. AutoML-Based Prediction Model

After the training dataset is partitioned via multipath-characteristic-based clustering using IDEC, each cluster represents a distinct multipath environment. The relations between multipath features and pseudorange errors may vary across clusters because of differences in satellite geometry and signal propagation conditions. Therefore, a single fixed regression model may not provide accurate predictions for all clusters. To address this problem, we employ fast and lightweight AutoML (FLAML) [45] to train a specific model for each cluster. FLAML automatically and jointly optimizes both the choice of algorithm and its hyperparameters. This algorithm avoids manual configuration and ensures that each cluster uses a customized model that best fits its own feature distribution and pseudorange error characteristics.

For each GNSS measurement i, the input is the feature vector $\mathbf{x}_i$, and the target output is the pseudorange error $\Delta\rho^{(i)}$. The regression model is trained to learn the mapping from the feature vector $\mathbf{x}_i$ to the pseudorange error $\Delta\rho^{(i)}$, which is expressed as

$$\Delta\rho^{(i)} = f_c(\mathbf{x}_i) + \xi_c^{(i)} \tag{15}$$

where c is the cluster label assigned to measurement i by IDEC, $f_c(\cdot)$ is the regression function for cluster c, and $\xi_c^{(i)}$ is the prediction residual. The predicted pseudorange error is expressed as

$$\Delta\hat{\rho}^{(i)} = \hat{f}_c(\mathbf{x}_i) \tag{16}$$

where $\hat{f}_c(\cdot)$ denotes the trained regression model for cluster c.

For each cluster $c \in \{1, \ldots, K\}$, the training set is constructed from the measurements assigned to it, which is expressed as

$$\mathcal{I}_c = \{i \mid c_i = c\}, \tag{17}$$

$$\mathcal{D}_c = \left\{ \left(\mathbf{x}_i, \Delta\rho^{(i)} \right) \mid i \in \mathcal{I}_c \right\} \tag{18}$$

where c_i is the cluster label of measurement i, $\mathcal{I}_c$ is the set of measurement indices belonging to cluster c, and $N_c = |\mathcal{I}_c|$ is the number of training samples in that cluster. If a cluster contains fewer than five samples, it is excluded because 5-fold cross-validation cannot be performed.

For each cluster containing at least five samples, FLAML searches among four candidate regression algorithms: light gradient boosting machine (LightGBM), extreme gradient boosting (XGBoost), RF, and extremely randomized trees (Extra Trees). These tree-based algorithms can capture nonlinear relationships between the input feature vector $\mathbf{x}_i$ and the pseudorange error $\Delta\rho^{(i)}$.

The model selection objective is to minimize the 5-fold cross-validation root mean square error (CV-RMSE). For each candidate configuration, random sample splitting is adopted to divide the training set $\mathcal{D}_c$ of each cluster into five folds using a fixed random seed of 42. In each round, four folds are used to train the regression model, while the remaining fold is used to evaluate its prediction error. This process is repeated until every fold has been used for validation. While random sample splitting is used within cross-validation for model selection, the test dataset is strictly separated from the training dataset to prevent temporal data leakage during final evaluation. The CV-RMSE for cluster c is expressed as

$$e_{\mathrm{cv},c}(m, \boldsymbol{\lambda}) = \frac{1}{5} \sum_{j=1}^{5} \sqrt{\frac{1}{|\mathcal{V}_c^{(j)}|} \sum_{i \in \mathcal{V}_c^{(j)}} \left[\Delta\rho^{(i)} - \hat{f}_{m,\boldsymbol{\lambda},c}^{(-j)}(\mathbf{x}_i) \right]^2} \tag{19}$$

where $\mathcal{V}_c^{(j)}$ is the set of measurement indices in the j-th validation fold, and $\hat{f}_{m,\boldsymbol{\lambda},c}^{(-j)}$ is the regression model trained on the other four folds using algorithm m and hyperparameters $\boldsymbol{\lambda}$. A lower cross-validation error indicates a smaller prediction error on the validation data. The model selection problem is expressed as

$$(m_c^*, \boldsymbol{\lambda}_c^*) = \underset{\substack{m \in \mathcal{M} \\ \boldsymbol{\lambda} \in \boldsymbol{\Lambda}_m}}{\arg\min} e_{\mathrm{cv},c}(m, \boldsymbol{\lambda}) \tag{20}$$

where m_c^* is the selected regression algorithm, $\boldsymbol{\lambda}_c^*$ is its selected hyperparameter vector for cluster c, $\mathcal{M}$ is the set of candidate algorithms, and $\boldsymbol{\Lambda}_m$ is the hyperparameter space for algorithm m. The corresponding minimal validation error achieved by the optimal model is denoted as $e_{\mathrm{cv},c}^* = e_{\mathrm{cv},c}(m_c^*, \boldsymbol{\lambda}_c^*)$.

FLAML is configured for regression, with root mean square error (RMSE) as the optimization metric and 5-fold cross-validation as the evaluation method. The search time budget is empirically limited to 2 s per cluster. Rather than checking every possible configuration, FLAML adaptively searches the candidate algorithms and their hyperparameter spaces within the available time. Thus, the selected configuration is the best one found during the search. After the search, the best model found by FLAML is saved for each cluster.

To ensure the reliability of online pseudorange correction, a cluster-specific model is accepted for online prediction only if it satisfies three conditions: the number of training samples N_c must be at least 500, the sample density κ_c must be at least 500, and the validation error $e_{\mathrm{cv},c}^*$ must be less than 10 m. Although five samples are enough to perform 5-fold cross-validation, the sample threshold is empirically set to 500 to ensure that the model has enough data to learn error patterns. The density threshold is empirically set to 500 to guarantee that the data points in the cluster stay close to each other, which means they truly share similar multipath conditions. Moreover, the error limit is empirically set to 10 m because an inaccurate model can give wrong corrections and degrade the positioning accuracy.

When an online measurement is assigned to a cluster that fails any of these acceptance criteria or has no trained model, a simple fallback procedure is applied. Specifically, no pseudorange error correction is performed, so $\Delta\hat{\rho}^{(i)} = 0$, and the raw pseudorange measurement is used directly.

The sample density κ_c evaluates how closely training samples are distributed around their center within cluster c. It is expressed as

$$\kappa_c = \frac{N_c}{\frac{1}{N_c}\sum_{i \in \mathcal{I}_c} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|_2 + \epsilon_\kappa} \tag{21}$$

where $\epsilon_\kappa = 10^{-6}$ is a small constant added to prevent division by zero, and $\boldsymbol{\mu}_c$ is the mean feature vector of cluster c, expressed as

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i \in \mathcal{I}_c} \mathbf{x}_i \tag{22}$$

2.6. PRI-Based Adaptive Fusion

The proposed algorithm builds on the EKF structure. The system error state vector $\mathbf{X}$ is expressed as

$$\mathbf{X} = [\delta\mathbf{r}_{\mathrm{IMU}}^\top, \delta\mathbf{v}_{\mathrm{IMU}}^\top, \delta\boldsymbol{\phi}_{\mathrm{IMU}}^\top, \mathbf{b}_{\mathrm{g}}^\top, \mathbf{b}_{\mathrm{a}}^\top, \mathbf{s}_{\mathrm{g}}^\top, \mathbf{s}_{\mathrm{a}}^\top, \delta t_{\mathrm{r}}, \delta \dot{t}_{\mathrm{r}}]^\top \tag{23}$$

where $\delta\mathbf{r}_{\mathrm{IMU}}$, $\delta\mathbf{v}_{\mathrm{IMU}}$, and $\delta\boldsymbol{\phi}_{\mathrm{IMU}}$ represent the position, velocity, and attitude error vectors of the IMU, respectively. $\mathbf{b}_{\mathrm{g}}$ and $\mathbf{b}_{\mathrm{a}}$ denote the bias vectors of the gyroscope and accelerometer;

$\mathbf{s}_g$ and $\mathbf{s}_a$ are the scale factor error vectors of the gyroscope and accelerometer; and δt_r and $\delta \dot{t}_r$ represent the GNSS receiver clock bias and clock drift.

In the tightly coupled GNSS/IMU integration framework, the discrete-time system state equation and observation equation are expressed as

$$\mathbf{X}_k = \mathbf{\Phi}_{k,k-1}\mathbf{X}_{k-1} + \mathbf{W}_{k-1} \tag{24}$$

$$\mathbf{Z}_k = \mathbf{H}_k\mathbf{X}_k + \mathbf{V}_k \tag{25}$$

where $\mathbf{X}_k$ and $\mathbf{X}_{k-1}$ are the system error state vectors at epochs k and $k-1$, respectively; $\mathbf{\Phi}_{k,k-1}$ is the state transition matrix; and $\mathbf{W}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ represents the equivalent discrete-time process noise vector with covariance matrix $\mathbf{Q}_{k-1}$, in which the noise mapping matrix has already been absorbed. In the observation equation, $\mathbf{Z}_k$ is the measurement vector, $\mathbf{H}_k$ is the measurement design matrix, and $\mathbf{V}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the measurement noise vector with nominal covariance matrix $\mathbf{R}_k$.

The standard EKF algorithm proceeds recursively through two main stages: state prediction and measurement update.

In the prediction stage, the estimated state vector and the error covariance matrix are propagated from epoch $k-1$ to k, as follows:

$$\hat{\mathbf{X}}_{k,k-1} = \mathbf{\Phi}_{k,k-1}\hat{\mathbf{X}}_{k-1} \tag{26}$$

$$\mathbf{P}_{k,k-1} = \mathbf{\Phi}_{k,k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k,k-1}^{\top} + \mathbf{Q}_{k-1} \tag{27}$$

where $\hat{\mathbf{X}}_{k,k-1}$ and $\mathbf{P}_{k,k-1}$ denote the predicted state vector and predicted error covariance matrix, respectively; $\hat{\mathbf{X}}_{k-1}$ and $\mathbf{P}_{k-1}$ represent the estimated error state vector and error covariance matrix at epoch $k-1$, respectively.

In the update stage, the filter calculates the Kalman gain, updates the state vector, and refines the error covariance matrix, as follows:

$$\mathbf{K}_k = \mathbf{P}_{k,k-1}\mathbf{H}_k^{\top}(\mathbf{H}_k\mathbf{P}_{k,k-1}\mathbf{H}_k^{\top} + \mathbf{R}_k)^{-1} \tag{28}$$

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k,k-1} + \mathbf{K}_k(\mathbf{Z}_k - \mathbf{H}_k\hat{\mathbf{X}}_{k,k-1}) \tag{29}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k,k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^{\top} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\top} \tag{30}$$

where $\mathbf{K}_k$ denotes the Kalman gain matrix, and $\mathbf{I}$ represents the identity matrix.

In the standard Kalman filter, the measurement noise covariance $\mathbf{R}_k$ is usually fixed. However, in complex urban environments, the predicted pseudorange errors may become inaccurate due to model performance limitations and data distribution shifts. If these inaccurate predictions are directly applied, they may cause significant positioning errors. To solve this problem, we propose a PRI-based adaptive adjustment strategy to evaluate the reliability of each prediction and dynamically adjust the covariance matrix.

The framework of the PRI-based adaptive adjustment strategy is illustrated in Figure 6. The whole process relies on four inputs: raw GNSS measurements, raw IMU measurements, the trained IDEC model, and the cluster-specific AutoML models. To make the procedure clear, the operation is detailed in the following steps:

(1) State prediction. The inputs are the raw IMU measurements. Through IMU mechanization, the system processes the raw IMU measurements to propagate the current navigation states. The outputs of this step are the predicted system state ($\hat{\mathbf{X}}_{k,k-1}$) and the state covariance matrix ($\mathbf{P}_{k,k-1}$).

(2) GNSS error correction and measurement weighting. The inputs include raw GNSS measurements, the trained IDEC model, and the cluster-specific AutoML models. In this process, the IDEC model first determines the cluster assignment for each GNSS

measurement. Subsequently, the corresponding AutoML regression model predicts the pseudorange error to correct the raw pseudorange measurement. Then, a PRI is calculated for each satellite to evaluate the reliability of the prediction. If the prediction is unreliable, the system assigns a lower weight to the corresponding measurement. The outputs are the corrected pseudoranges and the adjusted measurement noise covariance matrix ($\bar{\mathbf{R}}_k$).

(3) Measurement update. The inputs include the corrected GNSS measurements, the predicted system state, the predicted state covariance matrix, and the adjusted measurement noise covariance matrix ($\bar{\mathbf{R}}_k$). The output is the final optimal navigation solution, which provides accurate positioning results.

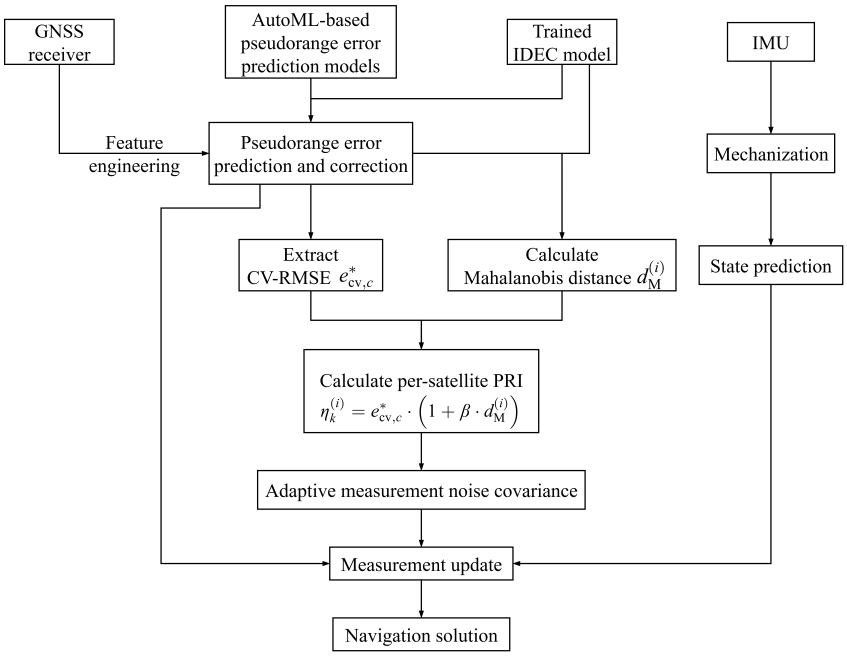

Figure 6. PRI-based adaptive adjustment strategy.

Before updating the filter, we adjust the measurement noise covariance matrix ($\mathbf{R}_k$) for each satellite. The value of a predicted pseudorange error itself does not fully prove its reliability. Therefore, the PRI is designed using two parts: the cross-validation accuracy of the prediction model and the deviation of the current measurement from the training dataset. For a satellite i assigned to cluster c, we use the Mahalanobis distance to measure the distance between its current feature vector $\mathbf{x}_i$ and the cluster center, which is expressed as

$$d_M^{(i)} = \sqrt{(\mathbf{x}_i - \boldsymbol{\mu}_c)^\top \boldsymbol{\Sigma}_c^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_c)} \tag{31}$$

where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ denote the statistical mean vector and covariance matrix of the input feature vectors belonging to cluster c in the training dataset. A large $d_M^{(i)}$ means that the current multipath feature vector deviates significantly from the training distribution of its assigned cluster, indicating that the corresponding AutoML prediction model might output large errors.

By combining this distance with the cross-validation error of the optimal prediction model for cluster c ($e^*_{cv,c}$), the per-satellite PRI ($\eta_k^{(i)}$) is expressed as

$$\eta_k^{(i)} = e^*_{cv,c} \cdot \left(1 + \beta \cdot d_M^{(i)}\right) \tag{32}$$

where $\beta = 0.5$ is a penalty coefficient. Based on the PRI, we calculate a variance scaling factor $s_k^{(i)}$ for each satellite, which is expressed as

$$s_k^{(i)} = \max\left(s_{\min}, \min\left(s_{\max}, \left(\frac{\eta_k^{(i)}}{E_{\text{base}}}\right)^2\right)\right) \tag{33}$$

where $E_{\text{base}} = 20.0$ m is empirically set to define the baseline reference level, while $s_{\min} = 1.0$ and $s_{\max} = 2.0$ are empirically chosen to determine the lower and upper bounds of the scaling factor.

Finally, the adjusted measurement noise covariance matrix $\bar{\mathbf{R}}_k$ is formed by multiplying the nominal variance $R_{k,ii}$ by this scaling factor, which is expressed as

$$\bar{\mathbf{R}}_k = \text{diag}\left(\bar{R}_{k,11}, \bar{R}_{k,22}, \ldots, \bar{R}_{k,m_k m_k}\right) \tag{34}$$

$$\bar{R}_{k,ii} = R_{k,ii} \cdot s_k^{(i)} \tag{35}$$

where m_k represents the total number of valid satellite measurements at epoch k.

For a measurement assigned to an ineligible cluster, the corresponding prediction reliability indicator defaults to the baseline value $\eta_k^{(i)} = E_{\text{base}} = 20.0$ m, which yields a scaling factor of $s_k^{(i)} = 1.0$. As a result, the measurement noise variance falls back to its nominal value $\bar{R}_{k,ii} = R_{k,ii}$.

In the proposed algorithm, this adjusted matrix $\bar{\mathbf{R}}_k$ dynamically replaces the nominal $\mathbf{R}_k$ during the calculation of the Kalman gain and the updated error covariance matrix. This ensures that unreliable GNSS predictions are assigned lower weights, preventing them from degrading the final navigation solution.

3. Results

3.1. Overview of the Experiment

To validate the effectiveness of the proposed algorithm, a vehicular field test was conducted in the complex urban area near Nanjing South Railway Station, Nanjing, China. The selected route encompasses typical challenging urban environments, including deep urban canyons and complex viaduct scenarios. The trajectory of the experimental vehicle is illustrated in Figure 7.



Figure 7. Trajectory of the experimental vehicle, with the blue line indicating the driving route.

The data collection was conducted between approximately 06:00 and 09:30 Coordinated Universal Time (UTC) for seven consecutive days at the same experimental site. The experimental multi-sensor platform was equipped with a BDStar Navigation C520-AT

GNSS receiver (Beijing BDStar Navigation Co., Ltd., Beijing, China) and a STIM-300 micro-electro-mechanical system (MEMS) IMU (Safran Sensing Technologies Norway AS, Horten, Norway), operating at sampling rates of 10 Hz and 125 Hz, respectively. Additionally, a high-grade GNSS/IMU integrated system, the Honeywell HGuide n580 (Honeywell International Inc., Plymouth, MN, USA), was utilized to provide a high-accuracy reference navigation solution through forward and backward post-processing. This reference solution is used only to generate the pseudorange error labels for training and to evaluate the positioning accuracy of the tested algorithms. The data collected during the first six days (Datasets 1–6) of the experiment were utilized as the training dataset, while the data collected on the final day (Dataset 7) served as the test dataset. Detailed specifications of the datasets are summarized in Table 1.

Table 1. Description of the experimental datasets.

Dataset	Time Range (UTC)	Duration (min)	Distance (km)
Dataset 1	27 March 2025 06:33:34–07:37:01	63.45	16.11
Dataset 2	28 March 2025 07:54:51–08:53:31	58.67	16.55
Dataset 3	29 March 2025 07:23:58–08:10:06	46.13	16.07
Dataset 4	30 March 2025 06:55:37–07:57:22	61.75	17.62
Dataset 5	31 March 2025 08:32:33–09:34:08	61.58	22.51
Dataset 6	1 April 2025 05:53:03–06:46:42	53.65	22.61
Dataset 7	2 April 2025 06:11:06–07:05:54	54.80	22.97

The multipath environment partitioning strategies of the proposed multipath-characteristic-based clustering algorithm are compared in Figure 8. For the proposed multipath-characteristic-based clustering algorithm, the number of clusters is set to 500. As indicated by the color scale in Figure 8, the cluster labels change dynamically along the trajectory to reflect the actual multipath conditions.



Figure 8. Comparison of multipath environment partitioning strategies, where the yellow boxes highlight an enlarged view of the selected area.

Among the $K = 500$ clusters generated by IDEC on our training dataset, 9 clusters have fewer than five samples and are excluded before 5-fold cross-validation, 1 cluster fails the sample density threshold with $\kappa_c < 500$, and 5 clusters fail the validation error threshold with $e^*_{\mathrm{cv},c} \geq 10$ m, while all remaining trained clusters satisfy $N_c \geq 500$. Therefore, 485 out of the 500 clusters remain eligible for online prediction.

To assess the advantages of the proposed MCC-AKF-based fusion algorithm, it was compared with three algorithms: (1) traditional EKF-based fusion, which serves as the

standard baseline relying solely on nominal measurement noise covariances without pseudorange error compensation or an adaptive fusion strategy; (2) 2D-EKF-based fusion, which uses a 2D geographic grid to model and compensate for pseudorange errors without the adaptive fusion strategy; and (3) 3D-EKF-based fusion, which uses a 3D geographic grid to model and compensate for pseudorange errors without the adaptive fusion strategy. The key features of these four compared algorithms are summarized in Table 2. Specifically, two representative urban scenarios, including a long ramp and a road section under viaducts, were selected to further demonstrate the advantages of the proposed algorithm under severe MI and NLOS reception.

Table 2. Summary of the compared algorithms and their key features.

Algorithm	Environment Partitioning	Measurement Weighting
EKF-based fusion	None	Fixed nominal covariance
2D-EKF-based fusion	2D spatial grid	Fixed nominal covariance
3D-EKF-based fusion	3D spatial grid	Fixed nominal covariance
MCC-AKF-based fusion	Multipath clustering by IDEC	PRI-based adaptive covariance

Figure 9 shows the positioning error comparisons of the four algorithms in the north, east, and down directions, respectively. Table 3 summarizes the positioning accuracy of each algorithm, as well as the improvements achieved by the proposed algorithm over the other algorithms.

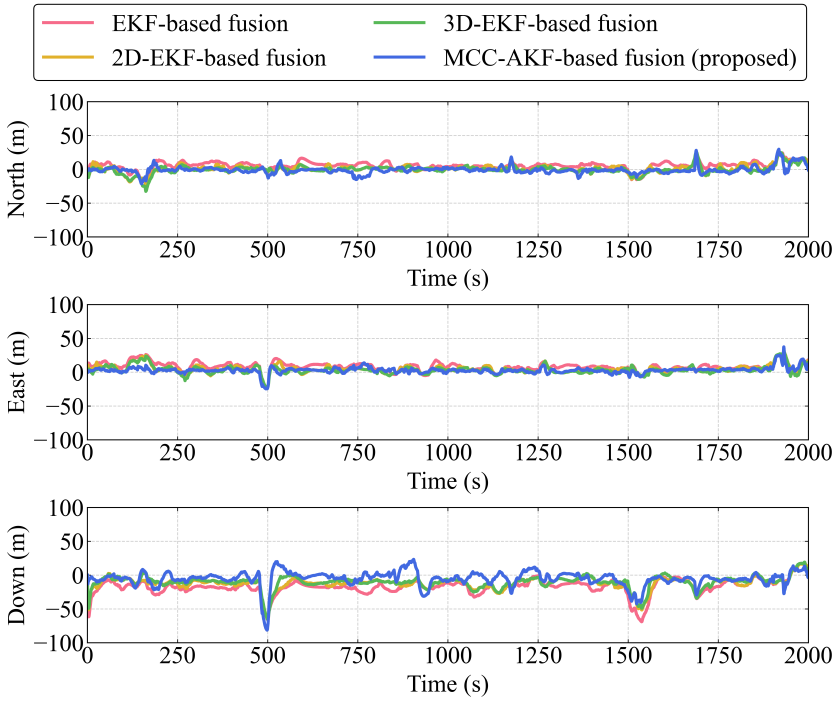

Figure 9. Positioning errors of the algorithms.

In Table 3, 3D positioning RMSEs of 18.32 m and 16.77 m can be achieved using the 2D-EKF-based fusion and 3D-EKF-based fusion algorithms, compared to 24.60 m for the traditional EKF-based fusion, providing improvements of 25.53% and 31.83%, respectively. This demonstrates the effectiveness of modeling pseudorange errors in complex urban environments. The proposed MCC-AKF-based fusion algorithm can further reduce the 3D positioning RMSE to 14.53 m, providing improvements of 40.96%, 20.70%, and 13.38% compared with the traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based

fusion algorithms, respectively. This is because the proposed multipath-characteristic-based clustering algorithm groups homogeneous multipath environments based on physical multipath features rather than rigid spatial grids, and the PRI-based adaptive fusion strategy effectively reduces the weights of unreliable predictions. Meanwhile, the improvements in the horizontal and down directions are also significant. In the horizontal direction, the proposed algorithm reduces the RMSE from 13.18 m to 7.66 m, achieving an improvement of 41.85% over the traditional EKF-based fusion. In the down direction, the proposed algorithm reduces the RMSE from 20.77 m to 12.34 m, achieving an improvement of 40.61% over the traditional EKF-based fusion.

Table 3. Comparison of positioning accuracy for different algorithms.

Algorithm	RMSE (m)				
	North	East	Down	Horizontal	3D
EKF-based fusion	8.09	10.40	20.77	13.18	24.60
2D-EKF-based fusion	6.12	7.51	15.55	9.69	18.32
3D-EKF-based fusion	5.90	6.66	14.21	8.90	16.77
MCC-AKF-based fusion (proposed)	**5.38**	**5.46**	**12.34**	**7.66**	**14.53**
Improvement (compared with EKF-based fusion)	33.57%	47.49%	40.61%	41.85%	40.96%
Improvement (compared with 2D-EKF-based fusion)	12.18%	27.25%	20.63%	20.89%	20.70%
Improvement (compared with 3D-EKF-based fusion)	8.95%	18.02%	13.18%	13.91%	13.38%

To further evaluate the advantages of the proposed algorithm in specific challenging urban scenarios, two typical environments with severe multipath effects were selected from the entire trajectory for detailed analysis: a long ramp and a road section under the viaducts. In order to show the experimental conditions clearly, Figure 10 displays the locations of these two road sections along with their surrounding urban environments. As illustrated in Figure 10, the first scenario is a long ramp (Scene 1) and the second is a road section under viaducts (Scene 2).

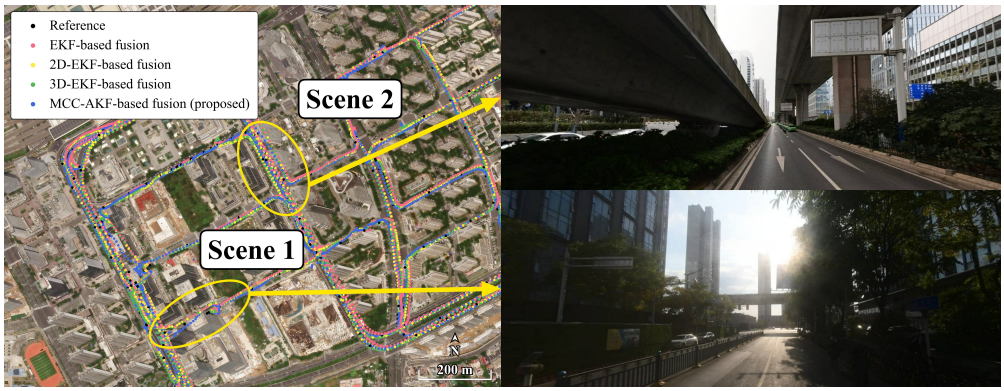

Figure 10. Partial experimental results and experimental scenes.

3.2. Results on a Long Ramp

In this complex urban environment, the surrounding structures cause strong multipath effects on GNSS signals. This selected road section covers the time window from 06:46:07 to 06:50:17 UTC on 2 April 2025. The positioning errors of the four algorithms in this scenario are compared in Figure 11. The specific accuracy statistics are listed in Table 4.

In Table 4, the traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based fusion algorithms yield 3D positioning RMSEs of 26.86 m, 26.09 m, and 25.90 m, respectively. The proposed MCC-AKF-based fusion algorithm reduces the 3D RMSE to 21.88 m, providing improvements of 18.54%, 16.15%, and 15.52% compared with the

traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based fusion algorithms, respectively. This is mainly due to the multipath-characteristic-based clustering mechanism. Instead of using rigid spatial grids, the proposed algorithm groups GNSS signals based on their physical features to model and correct multipath errors. Moreover, the PRI-based adaptive strategy evaluates the reliability of these predictions. If a prediction is unreliable, the adaptive filter increases the measurement noise covariance to reduce the weight of the corresponding signal. In this way, inaccurate corrections are prevented from degrading the final navigation solution.

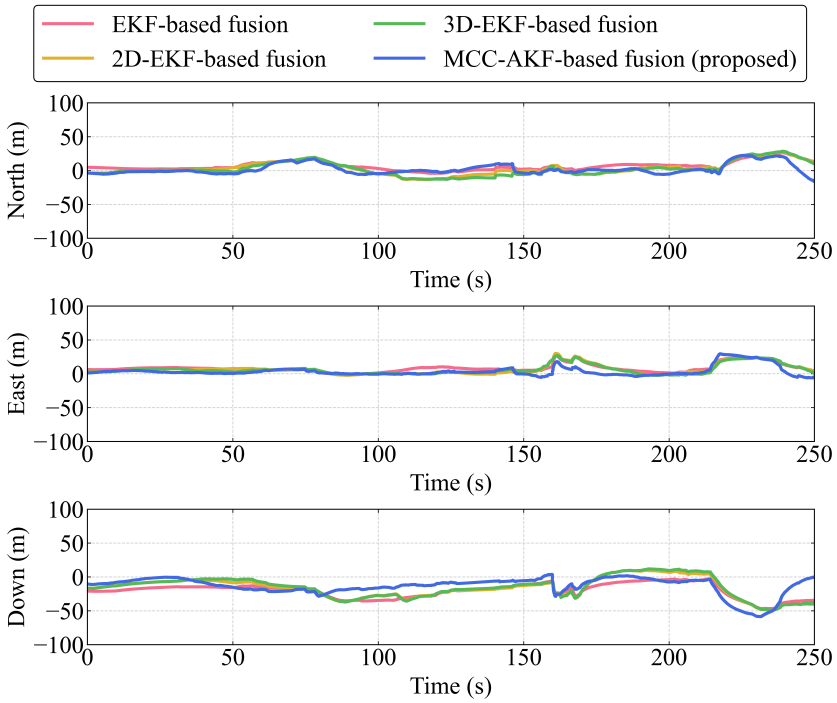

Figure 11. Positioning errors on a long ramp.

Table 4. Comparison of positioning accuracy for different algorithms on a long ramp.

Algorithm	RMSE (m)				
	North	**East**	**Down**	**Horizontal**	**3D**
EKF-based fusion	9.51	10.28	22.92	14.01	26.86
2D-EKF-based fusion	10.11	9.85	21.95	14.12	26.09
3D-EKF-based fusion	10.45	9.50	21.72	14.12	25.90
MCC-AKF-based fusion (proposed)	**7.88**	**8.17**	**18.71**	**11.35**	**21.88**
Improvement (compared with EKF-based fusion)	17.13%	20.53%	18.39%	18.95%	18.54%
Improvement (compared with 2D-EKF-based fusion)	22.01%	17.11%	14.76%	19.58%	16.15%
Improvement (compared with 3D-EKF-based fusion)	24.52%	14.01%	13.86%	19.59%	15.52%

3.3. Results Under the Viaducts

This is a typical harsh urban environment where satellite signals are heavily blocked by the bridge above, causing severe MI and NLOS reception. This selected road section covers the time window from 06:44:07 to 06:48:17 UTC on 2 April 2025. The positioning errors of the four algorithms in this scenario are compared in Figure 12. The specific accuracy statistics are listed in Table 5.

In Table 5, the 3D RMSE values of the traditional EKF-based fusion algorithm, the 2D-EKF-based fusion algorithm, and the 3D-EKF-based fusion algorithm are 23.50 m, 20.71 m, and 20.04 m, respectively. The proposed MCC-AKF algorithm reduces the 3D RMSE

to 14.05 m, providing improvements of 40.22%, 32.17%, and 29.89% compared with the traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based fusion algorithms, respectively. However, in the north direction, the 2D-EKF-based fusion algorithm yields lower positioning accuracy than the traditional EKF-based fusion, with the RMSE increasing from 7.43 m to 7.80 m. This is because the 2D grid layout has limitations in scenes with changing heights, as it applies the same prediction model to environments both over and under the viaduct. In contrast, the proposed algorithm solves this problem through the multipath-characteristic-based clustering algorithm and the adaptive fusion strategy. First, it clusters GNSS measurements based on physical multipath features rather than spatial coordinates, which avoids applying incorrect models when the height changes. Second, under the viaducts, satellite signals suffer from severe multipath effects. In such cases, the PRI-based adaptive strategy calculates the deviation of current signals from the training data to evaluate their reliability. For unreliable signals, the system dynamically increases their measurement noise covariance to reduce their weights. Consequently, the adaptive filter effectively improves the positioning accuracy in all directions.

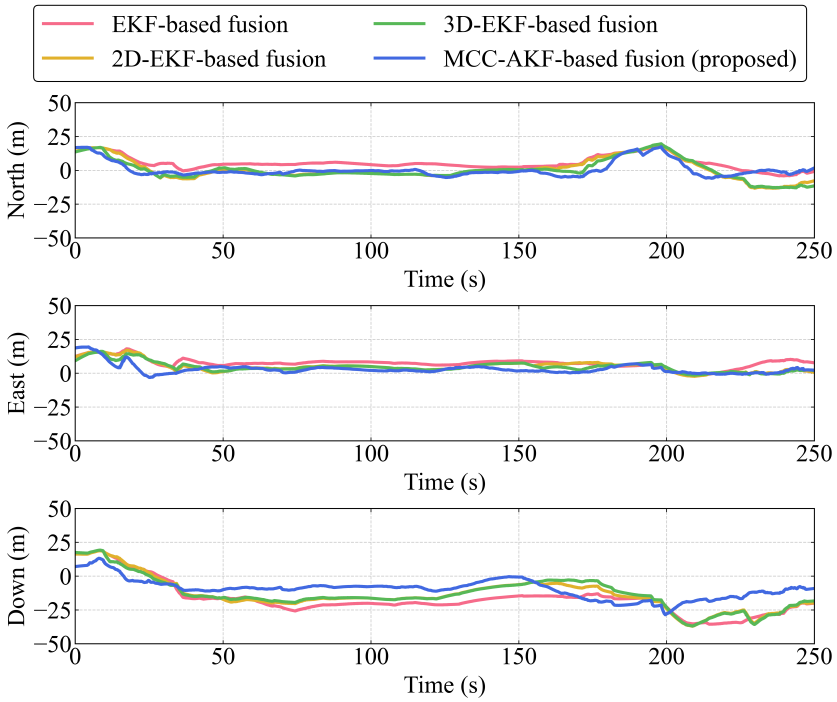

Figure 12. Positioning errors under the viaducts.

Table 5. Comparison of positioning accuracy for different algorithms under the viaducts.

Algorithm	RMSE (m)				
	North	**East**	**Down**	**Horizontal**	**3D**
EKF-based fusion	7.43	8.24	20.72	11.10	23.50
2D-EKF-based fusion	7.80	6.45	18.07	10.12	20.71
3D-EKF-based fusion	7.42	5.87	17.66	9.46	20.04
MCC-AKF-based fusion (proposed)	**5.60**	**4.83**	**11.94**	**7.40**	**14.05**
Improvement (compared with EKF-based fusion)	24.61%	41.41%	42.34%	33.35%	40.22%
Improvement (compared with 2D-EKF-based fusion)	28.12%	25.17%	33.91%	26.90%	32.17%
Improvement (compared with 3D-EKF-based fusion)	24.53%	17.74%	32.38%	21.85%	29.89%

Comparing the results of Scene 1 and Scene 2 reveals that the improvement achieved by the proposed algorithm is even greater in Scene 2 than in Scene 1. Specifically, the 3D

positioning accuracy improvement over the 2D-EKF and 3D-EKF algorithms increases from 16.15% and 15.52% in Scene 1 to 32.17% and 29.89% in Scene 2, respectively. This difference is caused by a series of connected factors driven by the different multipath characteristics of the two environments. Specifically, the overhead bridge decks, support pillars, and nearby buildings under the viaducts in Scene 2 create different signal blockage and reflection conditions compared with the long ramp in Scene 1. These different multipath characteristics lead to different feature values, so measurements in Scene 1 and Scene 2 are often assigned to different IDEC clusters and therefore predicted by different cluster-specific AutoML regression models. Moreover, even when measurements from the two scenes fall into the same cluster, differences in how much they deviate from the cluster training distribution still lead to different regression prediction errors and thus different pseudorange correction results. In addition, both the cross-validation accuracy of the prediction model and the feature deviation measured by the Mahalanobis distance directly determine the PRI value and the adaptive measurement weights in the Kalman filter. Together, these differences in cluster assignment, regression prediction, and PRI-based adaptive weighting lead to a larger accuracy improvement in Scene 2 than in Scene 1.

4. Discussion

The experimental results confirm that grouping GNSS signals by physical multipath features and adaptively tuning measurement noise can overcome the limits of rigid spatial grids and fixed-covariance filtering. Compared with earlier grid methods that mix different multipath conditions inside the same geographic cell, MCC-AKF groups signals by their actual propagation traits. In addition, unlike earlier adaptive filters that rely directly on raw predicted errors, the designed PRI combines cross-validation accuracy with the Mahalanobis distance to reduce the weights of unfamiliar measurements, providing robust sensor fusion for urban navigation.

However, several limitations and sources of uncertainty should be noted. First, the current experiments evaluate positioning performance on a later day at the same test site, while testing in other cities with different building layouts and reflection conditions remains to be explored. Second, remaining atmospheric delays, satellite orbit errors, and IMU drift during long signal blockages can still introduce positioning errors. Future work will collect larger datasets to improve model generalization and integrate additional sensors such as cameras to maintain high positioning accuracy.

5. Conclusions

In this paper, a multipath-characteristic-based clustering algorithm for pseudorange error compensation coupled with a reliability-indicator-based adaptive GNSS/IMU integration framework is proposed to improve positioning accuracy in complex urban environments. Traditional algorithms usually divide urban environments using rigid spatial coordinates. These algorithms often fail because GNSS signals in the same geographic grid can experience completely different multipath conditions. To solve this problem, our proposed algorithm performs multipath-characteristic-based clustering using the IDEC algorithm to group GNSS signals directly based on their physical multipath features. Then, AutoML is used to build a customized pseudorange error prediction model for each cluster. Furthermore, to prevent inaccurate predictions from degrading the navigation solution, we design a PRI-based adaptive fusion strategy. Based on the PRI, an AKF is used to dynamically adjust the sensor weights and provide reliable positioning results. Experimental results demonstrate that, in the overall test, the proposed algorithm achieves a 3D RMSE of 14.53 m. Compared with the traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based fusion, it provides improvements of 40.96%, 20.70%, and 13.38%,

respectively. Particularly in challenging environments such as under the viaducts, the proposed algorithm achieves a 3D RMSE of 14.05 m, with improvements of 40.22%, 32.17%, and 29.89% compared with the traditional EKF-based fusion, 2D-EKF-based fusion, and 3D-EKF-based fusion algorithms, respectively. These results show that by combining multipath-characteristic-based clustering with PRI-based adaptive weighting, the proposed algorithm can better model pseudorange errors and reduce the weights of unreliable predictions, which improves positioning accuracy.

In future work, we will further improve the proposed system in the following ways. First, we will collect more GNSS observation data to build a dataset covering more diverse scenes, so as to evaluate and improve the generalization ability of the regression models. Second, other sensors such as cameras can be integrated into the fusion system to further improve positioning accuracy. These efforts will ensure reliable navigation performance and provide strong support for intelligent transportation systems.

Author Contributions: Conceptualization, R.S. and J.Q.; methodology, J.Q. and Y.C.; software, J.Q. and Y.C.; validation, J.Q., Y.C., H.Z., and T.S.; formal analysis, J.Q., Y.C., H.Z., and W.Y.O.; investigation, J.Q.; resources, R.S.; data curation, J.Q., Y.C., and T.S.; writing—original draft preparation, J.Q. and Y.C.; writing—review and editing, R.S. and W.Y.O.; visualization, J.Q. and Y.C.; supervision, R.S.; project administration, R.S.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key R&D Program of China 2023YFE0208400, in part by the National Natural Science Foundation of China under Grant 42561160140, the Special Project for Universities Directly under the Ministry to Serve Jiangsu's High-Quality Development.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhuang, Y.; Sun, X.; Li, Y.; Huai, J.; Hua, L.; Yang, X.; Cao, X.; Zhang, P.; Cao, Y.; Qi, L.; et al. Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches. *Inf. Fusion* **2023**, *95*, 62–90. https://doi.org/10.1016/j.inffus.2023.01.025.
2. Milanés, V.; Naranjo, J.E.; González, C.; Alonso, J.; de Pedro, T. Autonomous vehicle based in cooperative GPS and inertial systems. *Robotica* **2008**, *26*, 627–633. https://doi.org/10.1017/S0263574708004232.
3. Li, S.; Li, X.; Wang, H.; Zhou, Y.; Shen, Z. Multi-GNSS PPP/INS/Vision/LiDAR tightly integrated system for precise navigation in urban environments. *Inf. Fusion* **2023**, *90*, 218–232. https://doi.org/10.1016/j.inffus.2022.09.018.
4. Liu, Y.; Fan, X.; Lv, C.; Wu, J.; Li, L.; Ding, D. An innovative information fusion method with adaptive Kalman filter for integrated INS/GPS navigation of autonomous vehicles. *Mech. Syst. Signal Process.* **2018**, *100*, 605–616. https://doi.org/10.1016/j.ymssp.2017.07.051.
5. Jiao, H.; Dong, W.; Zhou, T.; Chen, R.; Chen, L. 5G/GNSS/INS multi-sources integration navigation based on adaptive-FGO for urban environment. *Inf. Fusion* **2026**, *127*, 103876. https://doi.org/10.1016/j.inffus.2025.103876.
6. Zhu, N.; Marais, J.; Bétaille, D.; Berbineau, M. GNSS Position Integrity in Urban Environments: A Review of Literature. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2762–2778. https://doi.org/10.1109/TITS.2017.2766768.
7. Chiang, K.; Tsai, G.; Chang, H.; Joly, C.; El-Sheimy, N. Seamless navigation and mapping using an INS/GNSS/grid-based SLAM semi-tightly coupled integration scheme. *Inf. Fusion* **2019**, *50*, 181–196. https://doi.org/10.1016/j.inffus.2019.01.004.
8. Dierendonck, A.J.V.; Fenton, P.C.; Ford, T.J. Theory and Performance of Narrow Correlator Spacing in a GPS Receiver. *Navigation* **1992**, *39*, 265–283. https://doi.org/10.1002/j.2161-4296.1992.tb02276.x.
9. Townsend, B.R.; Fenton, P.C.; van Dierendonck, K.J.; van Nee, D.J.R. Performance Evaluation of the Multipath Estimating Delay Lock Loop. *Navigation* **1995**, *42*, 502–514. https://doi.org/10.1002/j.2161-4296.1995.tb01903.x.
10. van Nee, R.; Siereveld, J.; Fenton, P.; Townsend, B. The multipath estimating delay lock loop: Approaching theoretical accuracy limits. In Proceedings of the 1994 IEEE Position, Location and Navigation Symposium (PLANS), Las Vegas, NV, USA, 11–15 April 1994; pp. 246–251.

11. Tranquilla, J.; Carr, J.; Al-Rizzo, H. Analysis of a choke ring groundplane for multipath control in Global Positioning System (GPS) applications. *IEEE Trans. Antennas Propag.* **1994**, *42*, 905–911. https://doi.org/10.1109/8.299591.
12. Jiang, Z.; Groves, P.D. NLOS GPS signal detection using a dual-polarisation antenna. *GPS Solut.* **2014**, *18*, 15–26. https://doi.org/10.1007/s10291-012-0305-5.
13. Sun, R.; Cheng, Q.; Xie, F.; Zhang, W.; Lin, T.; Ochieng, W.Y. Combining Machine Learning and Dynamic Time Wrapping for Vehicle Driving Event Detection Using Smartphones. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 194–207. https://doi.org/10.1109/TITS.2019.2955760.
14. Phan, Q.H.; Tan, S.L.; McLoughlin, I.; Vu, D.L. A Unified Framework for GPS Code and Carrier-Phase Multipath Mitigation Using Support Vector Regression. *Adv. Artif. Neural Syst.* **2013**, *2013*, 240564. https://doi.org/10.1155/2013/240564.
15. Li, L.; Xu, Z.; Jia, Z.; Lai, L.; Shen, Y. An efficient GNSS NLOS signal identification and processing method using random forest and factor analysis with visual labels. *GPS Solut.* **2024**, *28*, 77. https://doi.org/10.1007/s10291-024-01624-8.
16. Sun, R.; Wang, G.; Zhang, W.; Hsu, L.T.; Ochieng, W.Y. A gradient boosting decision tree based GPS signal reception classification algorithm. *Appl. Soft Comput.* **2020**, *86*, 105942. https://doi.org/10.1016/j.asoc.2019.105942.
17. Yozevitch, R.; Moshe, B.B.; Weissman, A. A Robust GNSS LOS/NLOS Signal Classifier. *Navigation* **2016**, *63*, 429–442. https://doi.org/10.1002/navi.166.
18. Kubo, N.; Kobayashi, K.; Furukawa, R. GNSS Multipath Detection Using Continuous Time-Series C/N0. *Sensors* **2020**, *20*, 4059. https://doi.org/10.3390/s20144059.
19. Suzuki, T.; Amano, Y. NLOS Multipath Classification of GNSS Signal Correlation Output Using Machine Learning. *Sensors* **2021**, *21*, 2503. https://doi.org/10.3390/s21072503.
20. Groves, P.D.; Jiang, Z. Height Aiding, C/N0 Weighting and Consistency Checking for GNSS NLOS and Multipath Mitigation in Urban Areas. *J. Navig.* **2013**, *66*, 653–669. https://doi.org/10.1017/S0373463313000350.
21. Meng, T.; Jing, X.; Yan, Z.; Pedrycz, W. A survey on machine learning for data fusion. *Inf. Fusion* **2020**, *57*, 115–129. https://doi.org/10.1016/j.inffus.2019.12.001.
22. Hsu, L.T. GNSS multipath detection using a machine learning approach. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
23. Sun, R.; Wang, J.; Cheng, Q.; Mao, Y.; Ochieng, W.Y. A new IMU-aided multiple GNSS fault detection and exclusion algorithm for integrated navigation in urban environments. *GPS Solut.* **2021**, *25*, 147. https://doi.org/10.1007/s10291-021-01181-4.
24. Sun, R.; Wang, G.; Cheng, Q.; Fu, L.; Chiang, K.W.; Hsu, L.T.; Ochieng, W.Y. Improving GPS Code Phase Positioning Accuracy in Urban Environments Using Machine Learning. *IEEE Internet Things J.* **2021**, *8*, 7065–7078. https://doi.org/10.1109/JIOT.2020.3037074.
25. Sun, R.; Fu, L.; Wang, G.; Cheng, Q.; Hsu, L.T.; Ochieng, W.Y. Using dual-polarization GPS antenna with optimized adaptive neuro-fuzzy inference system to improve single point positioning accuracy in urban canyons. *Navigation* **2021**, *68*, 41–60. https://doi.org/10.1002/navi.408.
26. Sun, R.; Fu, L.; Cheng, Q.; Chiang, K.W.; Chen, W. Resilient Pseudorange Error Prediction and Correction for GNSS Positioning in Urban Areas. *IEEE Internet Things J.* **2023**, *10*, 9979–9988. https://doi.org/10.1109/JIOT.2023.3235483.
27. Sun, R.; Sheng, Q.; Cheng, Q.; Shang, X.; Ochieng, W.Y. 3-D Grid-Based Resilient Pseudorange Error Prediction for Adaptive GNSS/IMU Integrated Navigation in Urban Areas. *IEEE Internet Things J.* **2025**, *12*, 19264–19279. https://doi.org/10.1109/JIOT.2025.3533077.
28. Zhang, Z.; Li, X.; Yuan, H. Best Integer Equivariant Estimation Based on Unsupervised Machine Learning for GNSS Precise Positioning and Navigation in Complex Environments. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2672–2682. https://doi.org/10.1109/TAES.2023.3320115.
29. Wen, W.; Zhang, G.; Hsu, L.T. Exclusion of GNSS NLOS receptions caused by dynamic objects in heavy traffic urban scenarios using real-time 3D point cloud: An approach without 3D maps. In Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 23–26 April 2018; pp. 158–165.
30. Wen, W.; Pfeifer, T.; Bai, X.; Hsu, L.T. Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *Navigation* **2021**, *68*, 315–331. https://doi.org/10.1002/navi.421.
31. Mohamed, A.H.; Schwarz, K.P. Adaptive Kalman Filtering for INS/GPS. *J. Geod.* **1999**, *73*, 193–203. https://doi.org/10.1007/s001900050236.
32. Hide, C.; Moore, T.; Smith, M. Adaptive Kalman Filtering for Low-cost INS/GPS. *J. Navig.* **2003**, *56*, 143–152. https://doi.org/10.1017/S0373463302002151.
33. Wang, W.; Liu, Z.Y.; Xie, R.R. Quadratic extended Kalman filter approach for GPS/INS integration. *Aerosp. Sci. Technol.* **2006**, *10*, 709–713. https://doi.org/10.1016/j.ast.2006.03.003.
34. Ding, W.; Wang, J.; Rizos, C.; Kinlyside, D. Improving Adaptive Kalman Estimation in GPS/INS Integration. *J. Navig.* **2007**, *60*, 517–529. https://doi.org/10.1017/S0373463307004316.

35. Almagbile, A.; Wang, J.; Ding, W. Evaluating the Performances of Adaptive Kalman Filter Methods in GPS/INS Integration. *J. Glob. Position. Syst.* **2010**, *9*, 33–40. https://doi.org/10.5081/jgps.9.1.33.
36. Sun, R.; Qiu, M.; Liu, F.; Wang, Z.; Ochieng, W.Y. A Dual w-Test Based Quality Control Algorithm for Integrated IMU/GNSS Navigation in Urban Areas. *Remote Sens.* **2022**, *14*, 2132. https://doi.org/10.3390/rs14092132.
37. Yang, Y.; He, H.; Xu, G. Adaptively robust filtering for kinematic geodetic positioning. *J. Geod.* **2001**, *75*, 109–116. https://doi.org/10.1007/s001900000157.
38. Li, Y.; Mi, J.; Xu, Y.; Li, B.; Jiang, D.; Liu, W. A Robust Adaptive Filtering Algorithm for GNSS Single-Frequency RTK of Smartphone. *Remote Sens.* **2022**, *14*, 6388. https://doi.org/10.3390/rs14246388.
39. Gao, B.; Hu, G.; Zhong, Y.; Zhu, X. Cubature Kalman Filter With Both Adaptability and Robustness for Tightly-Coupled GNSS/INS Integration. *IEEE Sens. J.* **2021**, *21*, 14997–15011. https://doi.org/10.1109/JSEN.2021.3073963.
40. Sun, R.; Zhang, Z.; Cheng, Q.; Ochieng, W.Y. Pseudorange error prediction for adaptive tightly coupled GNSS/IMU navigation in urban areas. *GPS Solut.* **2021**, *26*, 28. https://doi.org/10.1007/s10291-021-01213-z.
41. Bilich, A.; Larson, K.M. Mapping the GPS multipath environment using the signal-to-noise ratio (SNR). *Radio Sci.* **2007**, *42*, RS6003. https://doi.org/10.1029/2007RS003652.
42. Comp, C.; Axelrad, P. Adaptive SNR-based carrier phase multipath mitigation technique. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 264–276. https://doi.org/10.1109/7.640284.
43. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised Deep Embedding for Clustering Analysis. *arXiv* **2015**, arXiv:1511.06335.
44. Guo, X.; Gao, L.; Liu, X.; Yin, J. Improved Deep Embedded Clustering with Local Structure Preservation. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, 19–25 August 2017; pp. 1753–1759.
45. Wang, C.; Wu, Q.; Weimer, M.; Zhu, E. FLAML: A Fast and Lightweight AutoML Library. *arXiv* **2021**, arXiv:1911.04706.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.