mathematics

MDPI

Article

Enhancing Vehicle Location Prediction Accuracy with Road-Aware Rectification for Multi-Access Edge Computing Applications

Asif Mehmood [1], Afaq Muhammad [2], Faisal Mehmood [3,*] and Wang-Cheol Song [4,*]

[1] Department of Biomedical Engineering, College of IT Convergence, Gachon University, Seongnam-si 13120, Republic of Korea; asif@gachon.ac.kr
[2] Computer Engineering Department and Interdisciplinary Research Center for Intelligent Secure Systems, College of Computing and Mathematics, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia
[3] Department of AI and Software, College of IT Convergence, Gachon University, Seongnam-si 13120, Republic of Korea
[4] Department of Computer Engineering, Jeju National University, Jeju-si 63243, Republic of Korea
* Correspondence: faisal89@gachon.ac.kr (F.M.); philo@jejunu.ac.kr (W.-C.S.)

Abstract: In future 6G networks, real-time and accurate vehicular data are key requirements for enhancing the data-driven multi-access edge computing (MEC) applications. Existing estimation techniques to forecast vehicle position aim to meet the real-time data needs but compromise accuracy due to a lack of context awareness. While algorithms such as the Kalman filter improve estimation accuracy by considering certainty-grading and current-state estimate of measurements, they do not include the road context, which is vital for more accurate predictions. Unfortunately, current implementations of linear Kalman filters are not road-aware and struggle to predict a two-dimensional movement accurately. To this end, we propose a significant road-aware rectification-assisted prediction mechanism that enhances the modified Kalman filter predictions by incorporating road awareness. The parameters used for the Kalman filter include vehicle location, angle, speed, and time. In contrast, road-aware location rectification incorporates predicted location and lane shape, increasing the accuracy and precision of vehicle location predictions, reaching up to 99.9%. Performance is evaluated by comparing actual, predicted, and rectified vehicular traces at different speeds. The results demonstrate that the prediction error is approximately 0.005, while the proposed rectification process further reduces the error to 0.001, highlighting the effectiveness of the proposed approach. Overall, results support the idea of provisioning accurate, proactive, and real-time vehicular location data at the edge using a road-aware approach, thereby revolutionizing 6G vehicle location provisioning in MEC.

Keywords: multi-access edge computing; location prediction; road-aware rectification; real-time geo-location data; internet of vehicles

MSC: 90B20; 93E10

check for updates

Citation: Mehmood, A.; Muhammad, A.; Mehmood, F.; Song, W.-C. Enhancing Vehicle Location Prediction Accuracy with Road-Aware Rectification for Multi-Access Edge Computing Applications. *Mathematics* **2024**, *12*, 3980. https://doi.org/10.3390/math12243980

Academic Editors: Huawen Liu, Chengyuan Zhang and Chunwei Tian

Received: 3 October 2024
Revised: 15 December 2024
Accepted: 16 December 2024
Published: 18 December 2024

Copyright: © 2024 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The evolution of wireless communication technologies has paved the way for developing intelligent transportation systems (ITSs) [1]. Traffic efficiency, safe environment assurance, and real-time road information provisioning are the key benefits of ITSs. In vehicular networks, technological advances such as sensors and communication devices embedded into modern vehicles have enabled direct communication between vehicles through vehicular ad hoc networks (VANETs) [2]. Furthermore, with the dawn of 5G mobile technology, cellular networks [3] provide coverage to vehicles in a vehicle-to-network (V2N) manner, including cellular vehicle-to-everything (C-V2X). As we transition

towards 6G, these advancements will revolutionize vehicle location provisioning at the edge, enhancing data-driven MEC applications through road-aware rectification-assisted prediction mechanisms.

MEC plays a significant role in achieving the needs of ITSs. MEC end-to-end use cases for V2X [4] focus on service continuity, application mobility, user context, and data-provisioning applications to nearby vehicles, permitting the reduction in communication delays related to computing, storage, and network resource requirements [5,6]. MEC helps to achieve many specifications of 5G, such as ultra-low latency, increased bandwidth, and network flexibility. As we transition towards 6G, these capabilities will be further enhanced, supporting more advanced applications like real-time, road-aware rectification-assisted prediction mechanisms. Such services are critical for ensuring data accuracy and active device location tracking [7–9].

However, a requirement yet to be fulfilled is the precise location of vehicles, which is crucial in autonomous vehicles (AVs) [10] and electric vehicles (EVs) [11]. As higher precision is significant in active device location tracking services, even a slight difference in measurement can lead to substantial inaccuracies. It is well known that the entire ITS [12] relies on accurate vehicular information and low-latency communication. Therefore, active device/vehicle location tracking applications in the ITS domain require an accurate ultra-low latency framework [13,14]. Transitioning to 6G and addressing these challenges will enhance vehicle location provisioning to meet precision and latency needs.

Data accuracy in the Internet of vehicles (IoV) is crucial for active vehicle location tracking. Various factors can affect data accuracy, including sensor performance and network delays. Relying solely on conventional technologies like global positioning systems (GPSs) may not meet the intrinsic requirements of ITSs. Therefore, advanced techniques and estimation algorithms, such as differential GPS (DGPS) [15] and the Kalman filter, are utilized to improve vehicular trajectory tracking. However, DGPS is not an economic solution for large-scale deployments of the location tracking [16,17]. Next-generation ITSs require transitioning to 6G and meeting the high data accuracy standards. As a result, solutions are needed that are not expensive in terms of investment but are scalable and cost-effective for future MEC applications.

The Kalman filter [18] is one of the most widely used models for estimating and predicting vehicle locations. It enables data provisioning systems to enhance vehicle data accuracy and predict a vehicle's future state, such as location and speed. While discussing active vehicle location tracking systems, road awareness is vital, as it can significantly enhance data accuracy. However, current implementations of the Kalman filter lack road awareness. Incorporating road-aware rectification-assisted prediction mechanisms can enable accurate location provisioning at the edge.

In contrast, particle filters are also utilized for localization and tracking in dynamic systems [19]. These methods rely on probability by representing a distribution of possible states of the system. They are well known for their ability to predict the location trajectory but are computationally expensive [20,21]. Due to the large number of particles used in these filters, the complexity grows to a complexity level of $O(m.n^2)$, where m is the number of particles and n is the state vector size, whereas the Kalman filter has a lower computational complexity $O(n^2)$. The lower complexity allows the Kalman filter to process data faster while maintaining high accuracy for linear and slightly non-linear systems. Both accuracy and computational complexity suggest that the Kalman filter is the most suitable candidate when considering real-time processing.

Road awareness in future networks brings significant benefits for enhancing ITSs, and this comes from the surrounding environment, which most notably includes the roads, lanes, junctions, and traffic [22,23]. Road awareness can be practically leveraged via the relationship between the road environment and vehicular traces. One example is the rectification of vehicular data traces using road information. Similarly, routing decisions based on network awareness [24] can be optimized at the edge over V2X networks using channel quality based on Q-values [25]. The road-awareness approach can also benefit

other use cases, such as travel route optimization [26], virtual Internet of things (IoT) networks [27], underground safety verification [28], and predictive analysis of academic libraries [29]. These enhancements will bring technological innovation and pave the way for 6G enhancements.

Regarding location rectification, vehicle location data can be corrected using road information through geo-referencing [30,31]. In the geo-referencing technique, geo-coordinates are utilized to rectify the location of an object. Three geo-coordinates form a triangle where the first point refers to the vehicle's location, and the other two refer to a road lane's geo-coordinates. Triangular concepts such as rise, run, and slope project each vertex onto the opposite baseline so that the projected point lies on a line between two points. Similarly, the vehicle's location can be projected onto a road lane represented by a line between the two road lane coordinates; this projected point is defined as the rectified point [32]. This rectified point is determined by the association between road and vehicle trace information, hence the term road-aware rectification technique.

Real-time and accurate vehicle location information is crucial for meeting the service requirements of MEC-based operators and third parties [9]. To address this, the work presented in this manuscript focuses on improving the real-time accuracy requirements for ITSs by predicting vehicle locations using road information and applying a rectification process. The proposed solution involves implementing a vTrachea-Store at the edge, a relational database storing map information (roads, lanes, and junctions), and real-time vehicular trajectories to create a road-aware database. Road awareness is embedded into a modified Kalman Filter algorithm using road geo-coordinates for prediction. The modified Kalman filter includes a rectification step that uses road-aware information from the proposed vTrachea-Store to rectify the predicted vehicle locations, improving the accuracy of the geo-coordinates. This work is evaluated using a dataset of vehicular mobility traces from Jeju City, South Korea, in conjunction with the simulation of urban mobility (SUMO) and OpenStreetMap tools. Comparative results between the base and modified Kalman filter demonstrate that incorporating road awareness in the rectification process can significantly enhance the performance of real-time vehicle tracking systems.

The list of research contributions in this work is summarized as follows:

- **vTrachea-Store**: This research contribution enables edge systems to store road information in a relational database with vehicle and road data. The use of this database is not just limited to this proposed system. Indeed, it can be extended to accommodate the ever-evolving requirements described in the conclusive remarks mentioned in Section 4.3. Further details about the vTrachea-Store are provided in the Section 3.1.
- **Modified Kalman filter**: This model predicts the vehicle's following location and integrates road-aware location rectification, which uses geo-coordinates instead of simple coordinates for the prediction, as detailed in Section 3.2.
- **Road-aware location rectification**: Current implementations of the Kalman filter do not incorporate road information, limiting prediction accuracy. To address this, we propose a road-aware location rectification-assisted prediction mechanism for 6G vehicle location provisioning, which rectifies predicted vehicular locations and is detailed in Section 3.3.

The manuscript is organized as follows: Section 2 reviews the background research necessary to understand the proposed 6G vehicle location provisioning system and its use cases. Section 3 details the proposed road-aware rectification-assisted prediction mechanism and the interaction of its components. Section 4 evaluates the accuracy of the prediction and rectification processes. Finally, Section 5 provides concluding remarks, highlighting the benefits and future directions for the proposed system.

2. Literature Review

The Kalman filter is a proven algorithm for predicting moving objects' locations. It has been proposed that vehicle locations in VANETs be predicted [33,34]. Comparisons between real mobility traces and model-driven predictions have been made to evaluate accuracy.

However, the predicted traces typically represent unidirectional trajectories due to the limitations of the Kalman gain attribute. In addition, many variations of the Kalman filter exist to address the challenges across different domains. In [35], the authors proposed a multi-model Kalman filter, specifically the extended Kalman filter (EKF), to predict vehicular trajectories. The EKF consisted of five linear models: a constant location model (CLM), constant velocity model (CVM), constant acceleration model (CAM), constant jerk model (CJM), and vehicle turn model (VTM), each representing a different vehicle state or situation. The state vector included *position*, *velocity*, and *distance* to represent the vehicle's state. However, road awareness plays a significant role in prediction, which that model lacked because it only considered *position*, *velocity*, and *distance*. Additionally, the five linear models did not accurately predict all situations due to the absence of an *angle* in the Kalman gain calculation. For a fair analysis and comparison of the proposed system results with existing works [33,34], the existing Kalman filter model was embedded with the features of the five linear models [35]. Considering all five linear model aspects gave ground for an improved and fair analysis as the support of geo-coordinate values and additional features was necessary.

A mobility prediction architecture requires context-awareness at the edge network [36]. In that work, the authors proposed a data analytic framework that provided context-aware real-time and batch services at the edge network, verified through various IoV use cases. It highlighted the importance of considering user preferences in location prediction and finding the best trajectory path. Still, it lacked the integration of road-aware rectification for enhanced accuracy in 6G vehicle location provisioning.

The reviewed literature considers data-driven approaches but does not utilize road awareness, which is vital for providing accurate real-time vehicular data at the edge. The MEC host framework [37], where end-user data-driven services run, can benefit significantly from integrating road-aware rectification to address inaccuracies in various data types, thereby enhancing 6G vehicle location provisioning at the edge.

A comprehensive solution is necessary to address the challenges outlined in the literature [33,35,36,38]. First, a road-aware approach must be embedded into data estimation techniques to enhance accuracy. Second, the support for geo-coordinates and bidirectional movements is essential; adapting algorithms like Haversine [39] or Vincenty [40,41] can facilitate high-precision manipulation of geo-coordinates. Third, incorporating parameters such as the *angle* is crucial for understanding vehicle direction, enabling practical estimation of the Kalman gain. Fourth, positioning Kalman filter predictions at the edge aggregation point (AP) within the MEC host is vital for active vehicle location tracking. Overall, a road-aware rectification-assisted location prediction system is needed to provide real-time and accurate vehicular data at the edge, supporting a versatile range of MEC use cases in the context of 6G vehicle location provisioning.

Overall, the limitations caused by both 5G networks and Kalman filter are overcome in the following ways:

- **Kalman filter**'s inability to adapt to dynamic environments such as the mentioned change in road structure is due to many reasons. Adaptation to such a dynamic environment is addressed through the inclusion of vTrachea-Store. Secondly, considering non-linear movement by including geo-coordinates in the prediction and rectification process improves location accuracy. Thirdly, including road lane shapes further allows the prediction of a vehicle's precise and accurate location.
- **5G networks** cannot scale on demand as they cannot meet the requirements of location provisioning due to speed constraints caused by the higher data density on local networks. Hence, the proposed system addresses this limitation by proposing an MEC-based system for providing vehicle locations. As MEC hosts are close to the edge vehicles, they could synchronize the road structure changes to provide seamless support by the context-aware vehicular applications, particularly in scenarios requiring real-time data based on road awareness.

3. Proposed System

The proposed vTrachea-Store, modified Kalman filter, and road-aware location rectification are explained in detail in this section.

Figure 1 shows the above-mentioned key components of the proposed system architecture. The vehicle sensor data are posted to the modified Kalman filter prediction model over the V2N connection. The predicted output of this model is passed to the road-aware location rectification process along with the *laneshape* taken from vTrachea-Store. Rectification is applied to the predicted outputs to rectify the geo-coordinates. Other sensor data and the actual, predicted, and rectified geo-coordinates are stored in vTrachea-Store, after which the data are exposed to other MEC applications.

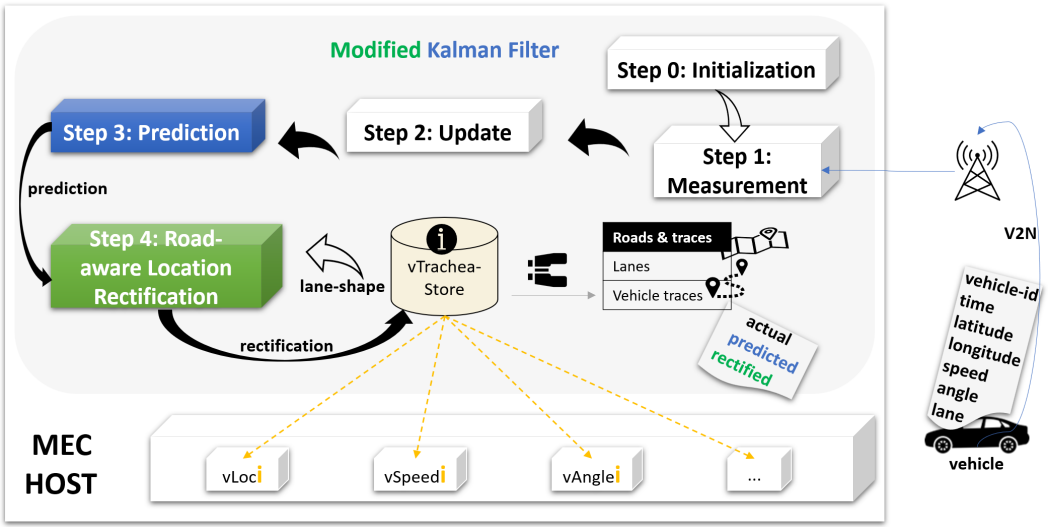


Figure 1. Proposed modified Kalman filter and its interworking. The process includes Step 0: initialization, Step 1: measurement (update), Step 2: state update (Kalman gain calculation), Step 3: prediction (updates in latitude and longitude), and Step 4: rectification (updates in latitude and longitude for enhanced accuracy). The black and white arrows indicate the sequence of steps in the process, while the yellow dashed lines represent the flow of vehicular and road data provision and movement.

The interworking of the proposed model is shown in Figure 1, which is a combination of a modified Kalman filter and road-aware location rectification. The modified Kalman filter takes *lat.*, *long.*, *speed*, and *angle* as inputs and predicts the vehicle geo-coordinates. It outputs the predicted *lat.* and *long.*, which are passed to the rectification process as inputs. The road-aware location rectification process is applied to the predicted geo-coordinates by taking *laneshape* into account. Rectification is applied to improve the precision of predicted geo-coordinates by mapping the predicted location of a vehicle onto a road *lane*. The rectification process is described in Section 3.3.

Symbols and their descriptions are shown in Table 1.

Table 1. Variables, identifiers, and parameters.

Symbol	Description
$lat.$	Latitude
Δt	Time difference, i.e., 1 second
$long.$	Longitude
$angle$	Angle of vehicle ranging between 0 and 360
$lat._{new}$	Predicted latitude
Pt_i	A geo-coordinate point i existing over a $laneshape$
$long._{new}$	Predicted longitude
Pt_P	Predicted geo-coordinate point
$lat._{rectified}$	Rectified latitude
Pt_R	Rectified geo-coordinate point
$long._{rectified}$	Rectified longitude
x	State of a vehicle, i.e., geo-coordinate location
$speed$	Speed of a vehicle
r	Vehicle's rectified state, i.e., geo-coordinate location
i	Geo-coordinate index of a $laneshape$
$lane$	Lane of a road
$radius_{earth}$	Earth's radius value used by Vincenty algorithm
$laneshape$	Shape of a $lane$
$Triangle$	A vector containing three geo-coordinate points
run	Horizontal distance covered by vehicle
$vtrace$	Trace of a vehicle
$Line_{Pt_APt_B}$	Line between two points A and B
n	Denotes a specific time
$slope$	Direct distance covered between Pt_i and Pt_P
z_n	Measurements of a vehicle
$Slope_{Pt_APt_P}$	Slope between two points A and P
pi/π	Value of $22/7$
$rise$	Vertical distance covered by vehicle
vid	Unique id of a vehicle
$distance_{weight}$	An associated weight of $lat.$ and $long.$ distance that needs to be multiplied with it
$Scalar_Product_{Pt_APt_B.Pt_APt_P}$	Scalar product of $Line_{Pt_APt_B}$ and $Slope_{Pt_APt_P}$

3.1. vTrachea-Store

The proposed vTrachea-Store builds relationships between roads, lanes, and vehicular traces. The entity relationship diagram (ERD) of the vTrachea-Store is shown in Figure 2, which contains the information and relationships of vehicles, vehicle traces *vtraces*, and *lanes*. The *vehicle* entity contains the registered vehicle information, and its vehicle traces *vtraces* contain the traces of the registered vehicles. The roads contain *lanes*, which include the structural geo-coordinates of a lane *laneshape*. Storing vehicular traces *vtraces* and the road entities relationally enables the road-aware location rectification to get valuable insights into the road structure while rectifying the Kalman filter predicted locations. The vTrachea-Store also contains the association of each *vtrace* with the *lane*. This association enables the road-aware location rectification to project the predicted location onto a road lane.

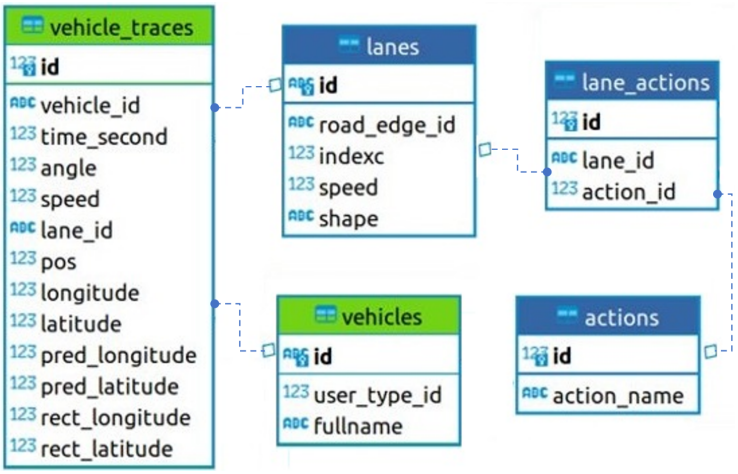


Figure 2. ERD of vTrachea-Store supporting road-awareness into the modified Kalman filter.

The road information is stored in *lanes*, while the vehicular trajectories are stored as vehicular traces. Each road contains multiple *lanes*, whereas a *laneshape* represents the shape of a lane. The vehicle trace *vtrace* includes the measurements z_n, representing the information of a vehicular trace at a specific time n. The basic measurements z_n are $lat.$, $long.$, *lane*, *angle*, and *speed*. The vTrachea-Store also allows the storage of the rectified and predicted $lat.$ and $long.$.

The road-aware location rectification process interacts with the vTrachea-Store to load road awareness into the procedure. While rectifying a vehicular trace *vtrace* from the trajectories, the proposed system loads the *laneshape* of a particular *lane*. A vehicular trace *vtrace* is known to be associated with a specific *lane* at a time n. The *laneshape* is represented in Equation (1) as follows:

$$laneshape = \{Pt_0, Pt_1, Pt_2 \ldots Pt_i\} \tag{1}$$

Each *lane* has a *laneshape* of i geo-coordinates, e.g., Pt_i. The rectification process utilizes two points, Pt_i and Pt_{i+1}, so the predicted point Pt_i lies between them. The criteria for selecting Pt_i and Pt_{i+1} are described in Section 3.3. Thus, with the support of vTrachea-Store, road awareness is embedded into the modified Kalman filter. Furthermore, storing information on all vehicles in the MEC system exposes vehicle information for ultra-low-latency-sensitive applications.

3.2. Modified Kalman Filter

A Kalman filter is utilized to predict the location of moving objects. Location prediction at the MEC enables the supply of real-time information to other applications. The basic steps of the Kalman filter are initialization, measurement, update, and prediction [42]. However, this work added a rectification step to improve the accuracy of the predicted geo-coordinates. The detailed steps of the modified Kalman filter-based system are:

- *Step 0: initialization*
  $x_{0,0}$ refers to the initial state of the vehicle at time $n = 0$. It contains the $lat.$ and $long.$, while $x_{1,0}$ refers to the predicted state of the vehicle at time $n = 0$, both shown in Equations (2) and (3), respectively. In this step, the proposed system initializes the actual data being received from the vehicle. It equates the predicted and rectified geo-coordinates to the actual geo-coordinates, i.e., $lat.$ and $long.$, as follows:

$$x_{0,0} = \begin{bmatrix} lat. = 0 \\ long. = 0 \end{bmatrix} \tag{2}$$

  In step 0, the prediction is initialized as per Equation (3):

$$x_{1,0} = \begin{bmatrix} lat. = x_{0,0}.lat. \\ long. = x_{0,0}.long. \end{bmatrix} \tag{3}$$

The initialization is followed by prediction.

- *Step 1: measurement*
  The measurement data z_n in Equation (4) refer to the actual data received from the vehicle sensor containing the $lat.$, $long.$, $angle$, and $speed$ at time n.

$$z_n = \begin{bmatrix} lat. \\ long. \\ angle \\ speed \end{bmatrix} \tag{4}$$

- *Step 2: state update*
  Referring to Algorithm 1, line 3 retrieves the vehicle direction. Then, based on the vehicle direction, the $distance_{weight}$ shown in Equation (5) is calculated for both geo-coordinates that need to be added to the current actual $lat.$ and $long.$ in step 3.

$$distance = distance_{weight} \times \Delta t \times speed \tag{5}$$

Algorithm 1 Location prediction using modified Kalman filter

1: **procedure** KF_PREDICT($lat.$, $long.$, $angle$, $speed$)
2: $directions \; \epsilon \; \{NE, NW, SE, SW\}$
3: $direction_{vehicle} \leftarrow$ get_direction($angle$)
4: $lat._{weight}$, $long._{weight} \leftarrow$ get_weights($angle$)
5: $distance_{lat.} \leftarrow lat._{weight} \times \Delta t \times speed$
6: $distance_{long.} \leftarrow long._{weight} \times \Delta t \times speed$
7: $lat._{new}$, $long._{new} \leftarrow$ Vincenty($lat.$, $distance_{lat.}$, $long.$, $distance_{long.}$)
8: $return \; lat._{new}$, $long._{new}$
9: **end procedure**

The Kalman gain $distance_{weight}$ is calculated by passing the angle value $z_n.angle$ shown in Equation (4). After calculating the distance, it is applied to the current $lat.$ and $long.$ values as per Equation (6). Current distance vector $distance_{n+1,n}$ is estimated as follows:

$$distance_{n+1,n} = \begin{bmatrix} lat. = lat._{weight} \times \Delta t \times z_1.speed \\ long. = long._{weight} \times \Delta t \times z_1.speed \end{bmatrix} \tag{6}$$

where $distance_{n+1,n}$ refers to the distance that needs to be added to a geo-coordinate point to obtain the prediction.

- *Step 3: prediction*
  In Equation (7), the distance for each $lat.$ and $long.$ is applied to $x_{n,n}$ to obtain the prediction point $x_{n+1,n}$ at time n as follows:

$$x_{n+1,n} = \begin{bmatrix} lat. = x_{n,n}.lat. \pm distance_{n+1,n}.lat. \\ long. = x_{n,n}.long. \pm distance_{n+1,n}.long. \end{bmatrix} \tag{7}$$

- *Step 4: rectification*
  Algorithm 2 is utilized to rectify the predicted geo-coordinate vector $x_{n+1,n}$ in this step. The rectification algorithm takes $x_{n+1,n}$ as an input and returns the rectified geo-coordinate vector $r_{n+1,n}$ as follows:

$$r_{n+1,n} = Rectification(x_{n+1,n}, laneshape, i) \tag{8}$$

where *laneshape* refers to the shape of a *lane* on which the vehicle is running, and i is a value associated with a uniquely identifiable current vehicular trace at time n. A detailed description of this step is provided in Section 3.3.

Algorithm 2 Rectifying predicted location using road information

procedure RECTIFICATION($x_{n+1,n}$, *laneshape*, *index*)
  $Pt_P \leftarrow \{\ lat. \leftarrow x_{n+1,n}.lat.,\ long. \leftarrow x_{n+1,n}.long.\ \}$
  $Pt_A \leftarrow \{\ lat. \leftarrow laneshape[\text{index}].lat.,\ long. \leftarrow laneshape[\text{index}].long.\ \}$
  $Pt_B \leftarrow \{\ lat. \leftarrow laneshape[\text{index+1}].lat.,\ long. \leftarrow laneshape[\text{index+1}].long.\ \}$
  $Line_{Pt_APt_B} \leftarrow \{\ lat. \leftarrow Pt_B.lat. - Pt_A.lat.,\ long. \leftarrow Pt_B.long. - Pt_A.long.\ \}$
  $Slope_{Pt_APt_P} \leftarrow \{\ lat. \leftarrow Pt_P.lat. - Pt_A.lat.,\ long. \leftarrow Pt_P.long. - Pt_A.long.\ \}$
  $Scalar_Product_{Pt_APt_B.Pt_APt_P} \leftarrow ($
    $(\ Line_{Pt_APt_B}.lat. \times Slope_{Pt_APt_P}.lat.,\ Line_{Pt_APt_B}.long. \times Slope_{Pt_APt_P}.long.\ )$
    $/\ (\ Line_{Pt_APt_B}.lat. \times Line_{Pt_APt_B}.lat.,\ Line_{Pt_APt_B}.long. \times Line_{Pt_APt_B}.long.\ )$
  $)$
  $Pt_R \leftarrow \{$
    $lat. \leftarrow Scalar_Product_{Pt_APt_B.Pt_APt_P} \times (\ Pt_A.lat. + Line_{Pt_APt_B}.lat.\ ),$
    $long. \leftarrow Scalar_Product_{Pt_APt_B.Pt_APt_P} \times (\ Pt_A.long. + Line_{Pt_APt_B}.long.\ )$
  $\}$
  return Pt_R
end procedure

The basic Kalman filter was modified to incorporate the changes caused by the road-aware location rectification. The first change was to consider geo-coordinates i.e., *lat.* and *long.*, as the basic Kalman filter works with linear systems and is applied chiefly to systems that do not consider geo-coordinates. The second change is that the modified Kalman considered two directional movements. The third change was reflected in the state update step shown in Equation (5), where the Kalman gain $distance_{weight}$ was calculated based on an *angle*.

The basic Kalman filter uses the Kalman gain shown in Equation (9) to estimate and predict the next location. In contrast, the proposed system estimates the Kalman gain for *lat.* and *long.* separately using Equation (5). The difference is that the *angle* estimates the Kalman gain $distance_{weight}$ required to update the system. The effectiveness of using the *angle* is analyzed in Section 4.

$$distance = kalman_{gain} \times time \times speed \tag{9}$$

As mentioned earlier, the modified Kalman filter uses an angle to assign a weight factor $distance_{weight}$, which decides the scale of a distance that needs to be added to *lat./long.*. Angles 0 and 180 represent *north* and *south* directions, while angles 90 and 270 refer to a moving vehicle's *west* and *east* directions. The rest of the *angle* values between 0–90, 90–180, 180–270, and 270–360 represent *north-east*, *south-east*, *south-west*, and *north-west* directions, respectively. It is known that the *angle* referring to *north* or *south* directions represents *lat.* movements while the *angle* referring to *east* or *west* directions represents *long.* movements.

Due to the geo-coordinate utilization, a formula for adding the distance was required. The Vincenty formula depicted in Algorithm 3 was developed and incorporated into the modified Kalman filter model to adopt the geo-coordinate-based system.

Algorithm 3 Find the target point with the Vincenty algorithm

procedure VINCENTY($lat.$, $distance_{lat.}$, $long.$, $distance_{long.}$)
    $radius_{earth} \leftarrow 6378$
    $pi \leftarrow \pi$
    $lat._{new} \leftarrow lat. + (distance_{lat.}/radius_{earth}) \times (180/pi)$
    $long._{new} \leftarrow long. + (distance_{long.}/radius_{earth})$
        $\times (180/pi)/\cos(long. \times pi/180)$
    return $lat._{new}$, $long._{new}$
end procedure

Vincenty's formula is best known for its accuracy in calculating geo-coordinates. Pythagoras's Equation (10) was also incorporated into the modified Kalman filter to calculate the *hypotenuse*, i.e., $Slope_{Pt_A Pt_P}$, which is shown to be equal to the square root of the sum of *perpendicular* and *base*.

$$hypotenuse = \sqrt{base^2 + perpendicular^2} \tag{10}$$

Equation (5) calculates the weight for each *north–south* and *east–west* direction, representing the $lat.$ and $long.$ direction movement. The weight calculation details are described in the explanation of Algorithm 1. It detects the vehicle's direction, with the help of which it finds the weights to be assigned to each direction movement, i.e., two-way movement (NS standing for $lat.$ movement and EW standing for $long.$). Then, the Kalman gain $distance_{weight}$ is calculated for both the $lat.$ and $long.$ using Equation (5). Vincenty's formula is required to calculate the distance shown in Equations (5) and (6). Thus, Algorithm 3 is used in the prediction.

3.3. Road-Aware Location Rectification

In Equation (8), the location rectification is shown as a part of the modified Kalman filter. It rectifies the predicted geo-coordinates, i.e., new latitude $lat._{new}$ and new longitude $long._{new}$. For rectification, it is assumed that the road information is already stored in vTrachea-Store, and the modified Kalman filter prediction process described in Section 3.2 has also been applied to $lat.$ and $long.$

The road-aware location rectification process inputs the vehicular trace $vtrace$, $laneshape$, and $index$. The $vtrace$ represents a predicted $lat./long.$. In contrast, the $laneshape$ associated with a particular $lane$ is defined in Equation (1). The $laneshape$ comprises multiple geo-coordinates, e.g., Pt_i, where i is the number of geo-coordinates on a $lane$.

- *Step 1: find two points from laneshape*
  Pt_i and Pt_{i+1} are selected from $laneshape$, where i represents the number of geo-coordinates on a $lane$. The criterion for selecting two points, Pt_i and Pt_{i+1}, is that the predicted point Pt_P must lie between them. Equation (11) is utilized to choose two points having the minimum $distance_i$ between Pt_i and Pt_P. Distances of Pt_P from all points Pt_i are represented in an array as follows:

$$distance_i = \sqrt{(Pt_i.lat. - Pt_P.lat.)^2 + (Pt_i.long. - Pt_P.long.)^2} \tag{11}$$

  Two points, Pt_i and Pt_{i+1}, are selected such that the distance between Pt_i and Pt_P is the least, and the distance between Pt_{i+1} and Pt_P is the second least, as shown in Equation (12). The functions $min()$ and $second_min()$ return a point with the least distance and the second-least distance, respectively.

$$\begin{aligned} Pt_i &= min(distances) \\ Pt_{i+1} &= second_min(distances) \end{aligned} \tag{12}$$

- *Step 2: construct the triangle*
  A triangle $Triangle_{n+1,n}$ is constructed based on three geo-coordinates, as shown in Equation (13). Road-aware location rectification is applied using three points, i.e., Pt_P, Pt_i, and Pt_{i+1}. The two points Pt_i and Pt_{i+1} are on the *lane*, and the predicted point Pt_P lies between them.

$$Triangle_{n+1,n} = \begin{bmatrix} Pt_P \\ Pt_i \\ Pt_{i+1} \end{bmatrix} \tag{13}$$

  The following steps can calculate the *run*, *slope*, and *rise*. The $Line_{Pt_APt_B}$, $Slope_{Pt_APt_P}$, and $Line_{Pt_PPt_R}$ represent the *run*, *slope*, and *rise*, respectively. Some of the mentioned vectors are yet to be calculated in the next steps. Points Pt_i and Pt_{i+1} are referred to as Pt_A and Pt_B to reduce complexity.
- *Step 3: line, slope, and scalar product*
  Line $Line_{Pt_APt_B}$ is calculated as using Equation (14). This line is drawn between both selected geo-coordinates on the *lane*, i.e., Pt_A and Pt_B.

$$Line_{Pt_APt_B} = \begin{bmatrix} lat. = Pt_B.lat. - Pt_A.lat. \\ long. = Pt_B.long. - Pt_A.long. \end{bmatrix} \tag{14}$$

  The slope $Slope_{Pt_APt_P}$ is calculated using Equation (15). That slope is drawn between Pt_A and the predicted location Pt_P.

$$Slope_{Pt_APt_P} = \begin{bmatrix} lat. = Pt_P.lat. - Pt_A.lat. \\ long. = Pt_P.long. - Pt_A.long. \end{bmatrix} \tag{15}$$

  The scalar product $Scalar_Product_{Pt_APt_B.Pt_APt_P}$ is calculated using Equation (16) and $Line_{Pt_APt_B}$ and $Slope_{Pt_APt_P}$.

$$\begin{aligned} &Scalar_Product_{Pt_APt_B.Pt_APt_P} = (\\ &(Line_{Pt_APt_B}.lat. \times Slope_{Pt_APt_P}.lat., Line_{Pt_APt_B}.long. \times Slope_{Pt_APt_P}.long.)/ \\ &(Line_{Pt_APt_B}.lat. \times Line_{Pt_APt_B}.lat., Line_{Pt_APt_B}.long. \times Line_{Pt_APt_B}.long.)) \end{aligned} \tag{16}$$

  Altogether, line $Line_{Pt_APt_B}$, slope $Slope_{Pt_APt_P}$, and scalar product $Scalar_Product_{Pt_APt_B.Pt_APt_P}$ further allow the road-aware location rectification process to find where the predicted point Pt_P meets on $Line_{Pt_APt_B}$, i.e., the rectified vehicle location [32].
- *Step 4: find the rectified location*
  Rectified location Pt_R is a point where the altitude of Pt_P meets the baseline $Line_{Pt_APt_B}$, where $Line_{Pt_APt_B}$ is a straight line representing the shape of a road *lane*. Finding point Pt_R is a challenge, and it is calculated as shown in Equation (17). According to mathematical calculations, it can be proven that point Pt_R lies on a *lane*, i.e., *laneshape*; hence, point Pt_R is rectified accurately with road awareness.

$$Pt_R = \begin{bmatrix} lat. = Scalar_Product_{Pt_APt_B.Pt_APt_P}.lat. \times (Pt_A.lat. + Line_{Pt_APt_B}.lat.) \\ long. = Scalar_Product_{Pt_APt_B.Pt_APt_P}.long. \times (Pt_A.long. + Line_{Pt_APt_B}.long.) \end{bmatrix} \tag{17}$$

The Kalman filter is unaware of the road shape and predicts the next location of a vehicle, which sometimes maps off the lane of a road. Thus, the process of correcting predicted locations is required. For this, the road-aware location rectification process projects predicted geo-coordinates onto a *lane* with the help of the *laneshape* stored in vTrachea-Store.

The steps of rectification are as follows. The *laneshape* is retrieved from vTrachea-Store. Based on the geo-coordinates of *laneshape*, the predicted *long./lat.* is projected and mapped onto the road *lane*. For instance, consider the example shown in Figure 3, where point Pt_P refers to the predicted location, and point Pt_P is found to lie between two points, Pt_A and Pt_B.

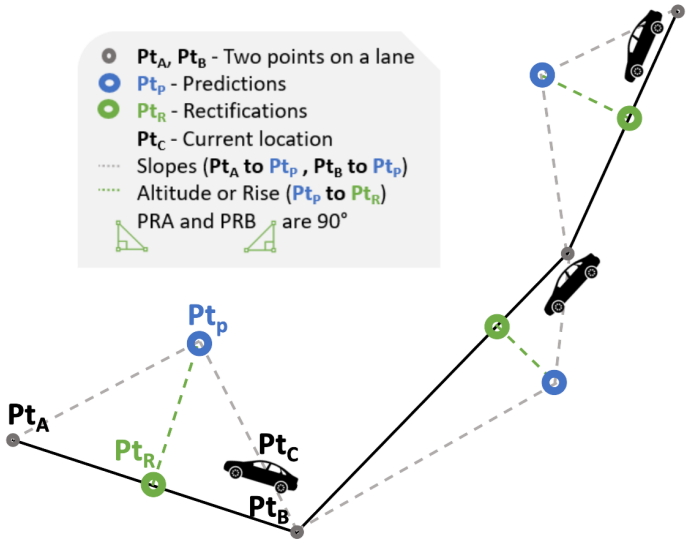


Figure 3. Illustration highlighting the key differences between prediction and rectification.

Points Pt_A and Pt_B are selected based on the criteria. All the points of the current *lane* are iterated and compared with the predicted point Pt_P. For further enhancement, the *angle* is used in the selection process of two points, Pt_A and Pt_B. The vehicle's direction is deduced using the *angle*, which allows the road-aware location rectification process to find the two points Pt_A and Pt_B, with Pt_P lying in between the two points.

The rectification process is applied to find point Pt_R, i.e., rectified geo-coordinate. Point Pt_R is where the altitude of Pt_P meets the base $Line_{Pt_APt_B}$ [32]. The proposed solution combines two algorithms, Algorithms 1 and 2, to find the rectified point Pt_R.

Algorithm 2 is utilized to rectify the predicted locations. The method of rectification is applied after the predictions have been made. The process of location rectification requires road information, for which it takes the predicted vehicular location $x_{n+1,n}$, the shape of the lane *laneshape*, and the trace identification *index* as inputs. $x_{n+1,n}$ provides the predicted vehicle geo-coordinates at time n, which needs to be projected onto the current road lane, i.e., *lane*. A prediction point Pt_P is constructed with $x_{n+1,n}$ values built from lines 2–5. This projected point Pt_P needs to be mapped between the points of a provided *laneshape*, so points Pt_A and Pt_B are constructed from lines 6–13. This process gives three geo-coordinate points, which result in the construction of a triangle with points Pt_A, Pt_B, and Pt_P, where Pt_P needs to be projected onto Pt_A and Pt_B, resulting in a rectified point, i.e., Pt_R. To find the projection, a line $Line_{Pt_APt_B}$ between Pt_A and Pt_B and $Slope_{Pt_APt_P}$ between Pt_A and Pt_P are calculated as shown on lines 14–21, so that the scalar product $Scalar_Product_{Pt_APt_B.Pt_APt_P}$ calculation can be performed as shown in lines 22–28. By utilizing the scalar product $Scalar_Product_{Pt_APt_B.Pt_APt_P}$, the road-aware location rectification process can project point Pt_P over the shape of the provided *laneshape*, which results in the rectified point Pt_R using lines 30–35. Pt_R is the point where the altitude of vertex Pt_P meets $Line_{Pt_APt_B}$, which means that the rectified point Pt_R is proven to lie on the provided *laneshape*.

Rectification is applied to improve the accuracy of the predicted location. The proposed solution can provide accurate and reliable rectified locations as AVs and other future technologies require real-time, correct, and precise information. Thus, if current MEC solutions include road information, the active vehicle location tracking use-case solutions will be able to address the challenge of providing real-time and accurate location information.

The vehicular data were rectified using the road-aware approach and found to be accurate compared with actual, predicted, and rectified locations. The performance of rectifications applied to predicted locations is compared in Figure 4.

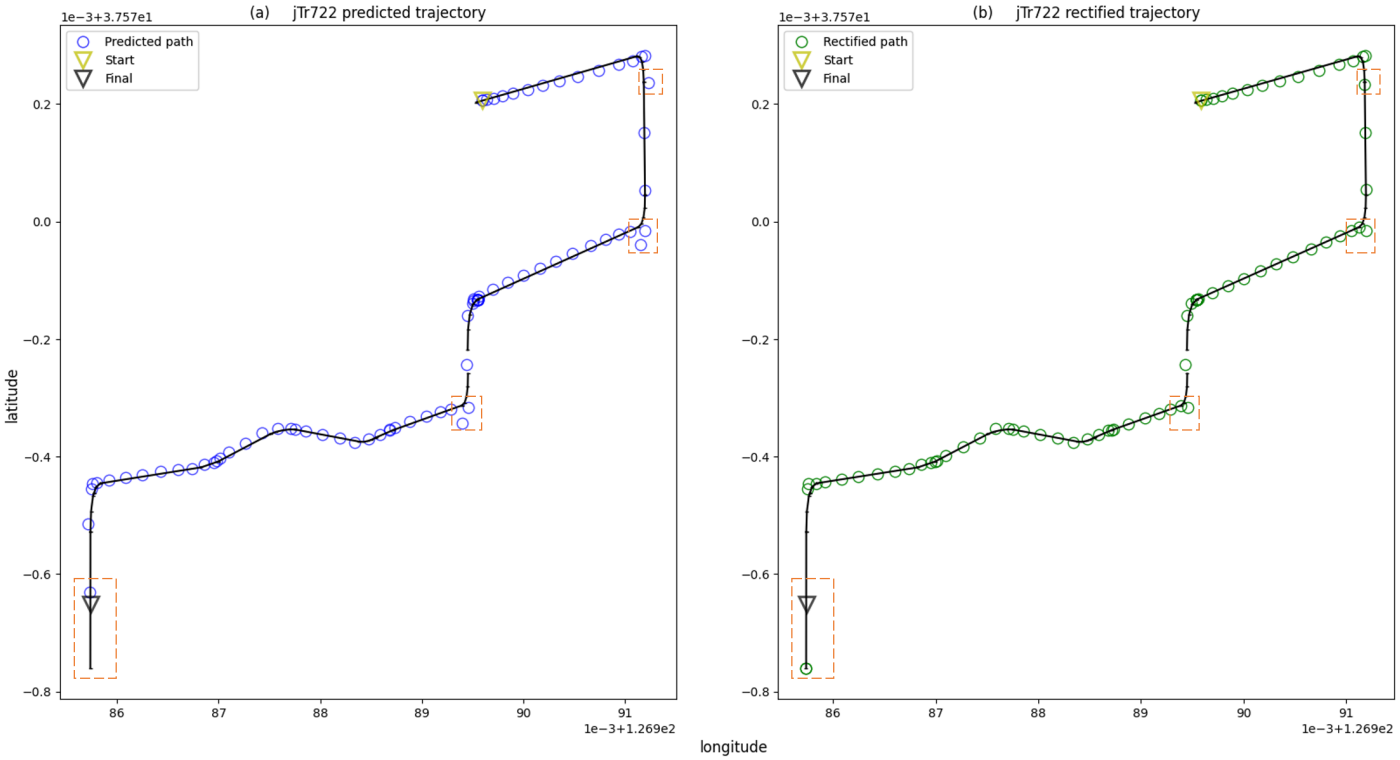


Figure 4. Comparison between predicted and rectified trajectories. The orange dashed frame highlights significant discrepancies in the predicted path and the corresponding corrections applied by the modified Kalman filter for improved accuracy.

4. Experiments and Discussion of Results

Details of the dataset, experimental setup, and discussion of results are described in this section. Results are discussed in the form of comparisons, i.e., predicted against the actual data, rectified against the predicted data, and rectified against the actual data. In addition, the performance of the road-aware location rectification is evaluated on different vehicle speeds and compared based on the root-mean-square error (RMSE) and accuracy for each case. Finally, the benefits of the proposed solution are described comprehensively.

4.1. Dataset and vTrachea-Store

Table 2 represents the dataset used in this work. The fields were *time*, *vid*, *lat.*, *long.*, *angle*, *speed*, and *lane*. All fields were applied to the modified Kalman filter and used as inputs. In the prediction process, *angle* and *speed* populated the predicted *lat.* and *long.*, i.e., $lat._{new}$ and $long._{new}$. Then, the process of road-aware rectification, with the help of a *lane*, was applied to the $lat._{new}$ and $long._{new}$ for the rectification of the predicted location. In the experiment, a dataset consisting of 8,495,739 vehicular traces for 10,000 vehicles was used, and it was generated from the Jeju city map of South Korea [43]. The dataset included different types of vehicles represented by identifiable prefixes denoted by jTx, jTr, jPs, and jBs to represent a taxi, truck, passenger, and bus, respectively. To consider congestion in the traffic, vehicles were categorized into different speed ranges, i.e., less than 18 km/h, between 18 and 36 km/h, between 36 and 60 km/h, and greater than 60 km/h.

The vTrachea-Store, shown in Figure 2, was developed in PostgreSQL. It was normalized and designed to store vehicular traces *vtraces* and road information. The road information was extracted from the extensible markup language (XML) file provided by OpenStreetMap. The extensible markup language (XML) file was processed to extract the road *lanes* to give the edge the concept of awareness about moving vehicles over a road. The OpenStreetMap provided XML files containing all the relevant road entities, which could be used to extract any geographical area's road information. Storing the road entities'

information in the vTrachea-Store enabled the road-aware location rectification process to take advantage of the association between a *vtrace* and the road *lane*.

Table 2. Dataset details.

Time	*Vid*	*Lat.*	*Long.*	*Angle*	*Speed*	*Lane*
0.0	jTx0	33.489513	126.500045	128.413552	0.000000	94703013_0
6.0	jPs1	33.492456	126.493635	94.367646	4.080817	−94703120#3_0
10.0	jBs6	33.497952	126.498469	72.219217	0.000000	−375567393#2_0
508	jTx71	33.49356	126.497616	275.086716	0.0	94703098#5_0
836.0	jPs607	33.494112	126.496768	3.064003	0.138229	94703081#5_0
873.0	jTr629	33.49704	126.495249	248.462859	1.498534	375568473#2_0
–	–	–	–	–	–	–
2044.0	jPs46	37.569478	126.983269	90.569053	12.050666	184443141#0_0
3541.0	jTx67	37.570253	126.989955	87.654345	14.57103	220027347#2_1
4001.0	jPs71	37.571501	126.97736	5.158597	19.983101	616037664#0_1
4327.0	jBs83	37.569574	126.989465	176.629172	5.788782	215843535#3_0
4367.0	jPs85	37.577337	126.989684	91.214774	26.727721	480618375#3_0
4417.0	jTx85	37.576089	127.002668	89.587807	24.595791	218769263#6_0
4433.0	jTr632	37.575929	127.003474	351.048751	9.902973	229026908#2_0
			Vehicular traces: 8,495,739			Vehicles: 10,000

4.2. Experimental Setup

The experimental setup required several software tools and libraries. Generating a dataset was also an initial part of this experimentation, which utilized SUMO [44] on the Jeju, South Korea map, extracting road information from the specified OpenStreetMap file, predicting and rectifying the geo-coordinates with the help of the developed and modified Kalman filter and road-aware location rectification. All experiments were performed on Ubuntu 20.04.3 long-term support (LTS). Virtual environments were created using Conda 4.9.2, and processing the dataset with the source code logic was developed using Python 3.8.10, PostgreSQL 12.9 for the database, Jupyter 4.7.1 for running the notebooks, Pandas 1.3.0 for processing data, and Matplotlib 3.4.2 and Folium 0.12.1 for generating analytical results.

4.3. Results and Discussion

In this paper, results for different vehicles based on their *vid* were compared, e.g., *jTr*722, $jTx36$, *jTr*22, *jPs*91, and *jBs*59. The color scheme is *blue* for predicted and *green* for rectified vehicular traces. Each graph's y and x coordinates represent *lat*. and *long*., respectively.

A total of 8,495,739 vehicular traces were taken, after which the modified Kalman filter for prediction and road-aware location rectification process for the location rectification were applied. The vehicular traces for the predictions and rectifications were stored in vTrachea-Store to analyze the vehicular trajectory accuracy in the form of comparisons and measurements.

A comparison of predicted and rectified trajectories is shown in Figure 4. The black lines represent the *laneshape* of the current roads on which the vehicle is running. Blue circles in Figure 4a represent the predicted traces $x_{n+1,n}$, while green circles in Figure 4b represent rectified traces $r_{n+1,n}$ of a vehicle.

The orange-dotted rectangles are the focal points where the enhancements can be visualized clearly. It can be seen that predictions did not map onto a road *lane* when the vehicle changed its direction. Meanwhile, the rectifications mapped closer to the *lanes* even when the vehicle changed direction. As geo-coordinates were used instead of simple coordinates in the trajectories, even a small difference in the distance between two geo-coordinates referred to a more significant distance separation when translated into meters.

The closeness of a point to the road *lane* refers to the vehicular location accuracy. The predictions of all vehicle trajectories were compared with the measured traces z_n using Equation (18), and the average RMSE was found to be 0.004, representing a 0.04% error in the predicted measurement. Thus, the performance of the road-aware location rectification was improved, especially when the vehicle changed direction, which is considered a more significant enhancement for the AV [10].

$$distance_{error} = \sqrt{(lat._2 - lat._1)^2 + (long._2 - long._1)^2} \quad (18)$$

A sample trace data of actual and Kalman filter-predicted traces are shown in Table 3. To calculate the RMSE value, the $distance_{error}$ between actual and predicted geo-coordinates was computed using Equation (18). Then, the difference among the 8,495,739 vehicular traces shown in Table 2 was accumulated to an RMSE value of 0.00007301996584. This RMSE value reflected *lat*. and *long*. accuracies of 99.99996352575% and 99.99995050356% in the predicted measurement, respectively. The geo-coordinate systems' precision was very high, i.e., a slight difference in the estimation of *lat*. and *long*. values could lead to a more significant error. Also, accuracy in a highly precise location is critical in the domain of AV, as slighter inaccuracy may cause accidents, thus reflecting the urge to improve the accuracy for a system requiring high precision in its measurement. This RMSE value in prediction has a precision of about four decimal places and needs to be improved.

Another comparison was made between the actual and rectified trace data, as shown in Table 4. The RMSE value for these data was 0.0000145329922492. The RMSE value in rectification was less than the RMSE value in prediction, which indicated that significant accuracy was achieved. Consequently, the difference in actual and rectification values also showcased high accuracy and precision. Hence, the proposed road-aware location rectification model is satisfactory for accurate vehicle location tracking systems, which is required for the AV domain.

Table 3. Comparison of actual and Kalman filter-predicted traces.

Time	Actual Traces	Predicted Traces	Error/Distance
304.0	(37.570206, 126.989590)	(37.570205999999999, 126.989590000000007)	0.000000000000000
305.0	(37.570207, 126.989614)	(37.570205999999999, 126.989590000000007)	0.000024020824295
306.0	(37.570209, 126.989659)	(37.570207485599724, 126.989636892742155)	0.000022159067168
307.0	(37.570212, 126.989724)	(37.570209939495051, 126.989703290838236)	0.000020811416613
308.0	(37.570217, 126.989811)	(37.570213345162152, 126.989787415301322)	0.000023866207314
309.0	(37.570222, 126.989926)	(37.570218793941116, 126.989895572200851)	0.000030596237913
310.0	(37.570229, 126.990057)	(37.570224391819401, 126.990038758129245)	0.000018814918995
311.0	(37.570236, 126.990205)	(37.570231711263489, 126.990184817772658)	0.000020632875745
312.0	(37.570244, 126.990371)	(37.570239068281133, 126.990349648758126)	0.000021913406408
313.0	(37.570254, 126.990555)	(37.570247438858772, 126.990533119011374)	0.000022843516311
			(for all vehicles) RMSE: 0.0000689038849152

In Table 5, RMSE values relative to the location prediction and rectification are showcased for vehicles moving at a speed of less than 18 km/h, between 18 and 36 km/h, between 36 and 60 km/h, and greater than 60 km/h. The RMSE values in rectification were low compared to the predictions. Even though the speed ranges varied, the RMSE values of the rectification stayed within a close range. Thus, this indicates the performance of the road-aware location rectification was stable over varying speeds. The maximum error in rectification was 0.00101528173429 and 0.00404954597917 units, less than the maximum prediction error, which means that the RMSE was significantly reduced. In addition, the accuracy of the rectification was greater than that of prediction accuracy. Although the difference in accuracy was also tiny, since the estimation systems were based on a geo-coordinate system, a slight improvement was considered a significant enhancement due

to the high precision requirement. Furthermore, the performance comparison between the RMSE values of the prediction and rectification could also be appreciated. The rectification provided more accurate vehicle location information than the prediction. Hence, it was a better candidate for active vehicle tracking systems. The results show that prediction accuracy can be improved by embedding road-aware location rectification into a modified Kalman filter and can be considered satisfactory for providing real-time and accurate information.

Table 4. Comparison of actual and rectified traces.

Time	Actual Traces	Rectified Traces	Error/Distance
304.0	(37.570206, 126.989590)	(37.570205999999999, 126.989590000000007)	0.000000000000000
305.0	(37.570207, 126.989614)	(37.570205999999999, 126.989590000000007)	0.000024020824283
306.0	(37.570209, 126.989659)	(37.570207485599724, 126.989636892742155)	0.000022159067154
307.0	(37.570212, 126.989724)	(37.570209939495051, 126.989703290838236)	0.000020811416512
308.0	(37.570217, 126.989811)	(37.570213345162152, 126.989787415301322)	0.000023866207314
309.0	(37.570222, 126.989926)	(37.570218793941116, 126.989895572200851)	0.000030596237913
310.0	(37.570229, 126.990057)	(37.570224391819401, 126.990038758129245)	0.000018814918995
311.0	(37.570236, 126.990205)	(37.570231711263489, 126.990184817772658)	0.000020632875745
312.0	(37.570244, 126.990371)	(37.570239068281133, 126.990349648758126)	0.000021913406383
313.0	(37.570254, 126.990555)	(37.570247438858772, 126.990533119011374)	0.000022843516270
			(for all vehicles) RMSE: 0.0000145329922492

Table 5. Prediction and rectification accuracy over different speed ranges.

Speed Range	RMSE		
	Prediction		Rectification
<18 km/h	0.00011216653344	>	0.00000704635149
18 to 36 km/h	0.00005323746171	>	0.00001535886430
36 to 60 km/h	0.00004514223302	>	0.00002240556711
>60 km/h	0.00004923667466	>	0.00002723701492
For all speeds	0.00007301996584	>	0.00001453299224
Maximum error	0.00506482771346	>	0.00101528173429
Accuracy (*lat*.)	99.99996352575%	>	99.9999792298%
Accuracy (*long*.)	99.99995050356%	>	99.9999919943%

The complexities of different algorithms are listed in Table 6.

Considering real-time environments, a few concerns must be mentioned in this discussion as they might impact the proposed model's application scope. One example of this is considering construction zones. The currently proposed system focuses on improving the accuracy of vehicle location prediction by considering several scenarios of speed changes but does not consider road construction in its results. This construction zone aspect is essential for road safety and AV navigation. Future work will involve making slight changes in vTrachea-Store and updating and including the complex and dynamic nature traffic scenarios.

Considerations of essential investigations and their impacts on the proposed system need to be addressed:

- **Speed and shape modifications**: While simulating the dataset, constructing zones can be considered through the speed and shape associated with the traffic assignment zone (TAZ). The proposed vTrachea-Stor database allows the synchronization of speed and shape changes in real time as the database is flexible.
- **Vehicle behavior control**: The speed of vehicles can be controlled in the simulation, and the TAZ can facilitate it using the allow/disallow attributes. This ensures accurate prediction under construction zone conditions.

Table 6. Algorithm complexities in comparison to Kalman filter.

Algorithm	Comparison and Explanation	
Kalman filter [45]	$O(n^2)$	The Kalman filter is more efficient in terms of computation. It is suitable for linear and slightly non-linear systems. It can be computationally inexpensive, making it ideal for real-time systems.
Particle filter [46]	$O(m.n^2)$	The particle filter is not as efficient in computation as the Kalman filter. In this algorithm, m refers to the particles, and n is the state vector size. Its accuracy is good but expensive.
Kalman filter	The Kalman filter is not reliant on expensive infrastructure. It just needs a software execution platform. This property makes it suitable for large-scale deployments where vehicular edge applications must be deployed across regions.	
DGPS/enhanced GPS	Enhanced GPS, for example, DGPS, is considered an expensive solution as it relies on an infrastructure distributed on a large scale [16,17]. Considering the cost and dependency, the Kalman filter is suitable for predicting and estimating vehicular location. Based on this, the proposed system is compared only with the Kalman filter estimation technique to obtain a fair comparison.	

The proposed model demonstrates significant improvements in accuracy over the Kalman filter through the use of vTrachea-Store, which allows runtime updates in road shape, status, and conditions. This allows the handling of construction zones which must be reflected in future systems. Another benefit of using vTrachea-Store is that whenever the maps are updated, the vTrachea-Store's lane shapes are allowed to be updated without changing the model algorithms. In addition to this benefit, lanes can be added to the structure of a lane, providing increased flexibility to the proposed system. Considering the table named as actions from vTrachea-Store, the system allowed the vehicles to perform possible actions such as turning left, right, stop, etc. Using the same methodology, this system can be upgraded such that it incorporates construction zones such as when lanes are not usable, etc. This way, vehicles can be alerted to stop in real time. Using vTrachea-Store facilitates the adoption of dynamic scenarios for various reasons required by real-time systems. Hence, context-aware and real-time changes enable the proposed system to be flexible and to provide accurate vehicle locations.

In future work, we aim to extend this framework to handle situations where road lanes become unusable due to various traffic conditions. Handling this challenge will allow the system to provide real-time alerts to vehicles and enhance vehicle location accuracy.

Conclusively, the performance of the proposed system is significantly better than the Kalman filter's predictions. In addition, vTrachea-Store plays an important role and cannot be neglected as it considers those dynamics required in daily life; for example, it considers the change in road shapes, which are important for accurate vehicle location predictions in real-world dynamic environments.

5. Conclusions and Future Directions

Improving data accuracy will enable MEC to meet the demands of real-time data provisioning and support a versatile set of use cases, such as active vehicle location tracking. This paper focused on overcoming data accuracy issues by embedding road awareness into the data provisioning systems at the edge. An experimental setup was developed to evaluate vehicle data accuracy using the proposed vTrachea-Store database, a modified Kalman filter, and road-aware location rectification models. Comprehensive comparisons of actual/measured, predicted, and rectified vehicular traces were conducted and analyzed against existing systems. Errors in the rectification accuracy were also assessed.

The performance evaluations supported the conclusion that incorporating road-awareness information into the proposed system significantly enhanced trajectory prediction compared to current systems while exhibiting minimal errors in the experimental results.

This manuscript aimed to improve vehicle location accuracy by developing a road-aware rectification framework. Future work may expand this framework to accommodate various types of vehicular data, addressing the growing demand for an autonomous-driving-oriented edge streaming data processing framework [47]. By integrating vehicle environment awareness at the MEC host, these frameworks can enhance the service orchestration platforms [48], enabling timely decision-making based on real-time demands. Furthermore, enhancements to these frameworks could support mobility-aware and migration-enabled edge user allocation within MEC [49].

The GitHub repository "asifgit/vehicle-location-prediction-and-rectification" [50] for this research will be made available after the submission is accepted. This repository includes all the necessary code for rectifying predicted vehicular locations using a modified Kalman filter. Please note that all code is open-source and free to use.

Author Contributions: Conceptualization, A.M. (Asif Mehmood); methodology, A.M. (Asif Mehmood); software, A.M. (Asif Mehmood); validation, F.M. and A.M. (Afaq Muhammad); formal analysis, A.M. (Afaq Muhammad) and F.M.; investigation, A.M. (Asif Mehmood) and A.M. (Afaq Muhammad); resources, W.-C.S.; data curation, A.M. (Asif Mehmood) and F.M.; writing—original draft preparation, A.M. (Asif Mehmood) and F.M.; visualization, A.M. (Asif Mehmood); project administration, A.M. (Afaq Muhammad) and W.-C.S.; funding acquisition, W.-C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2023-RIS009).

Data Availability Statement: The original data presented in the study are openly available at IEEE DataPort at Vehicular Trajectories from Jeju, Republic of Korea [43]: Available online: https://dx.doi.org/10.21227/y8vk-wj40 (accessed on 7 December 2024).

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

References

1. Jha, A.V.; Appasani, B.; Khan, M.S.; Zeadally, S.; Katib, I. 6G for intelligent transportation systems: Standards, technologies, and challenges. *Telecommun. Syst.* **2024**, *86*, 241–268. [CrossRef]
2. Shi, W.; Jiang, H.; Xiong, B.; Chen, X.; Zhang, H.; Chen, Z.; Wu, Q. RIS-empowered V2V communications: Three-dimensional beam domain channel modeling and analysis. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 15844–15857. [CrossRef]
3. Arif, M.; Kim, W.; Qureshi, S. Interference characterization in cellular-assisted vehicular communications with jamming. *IEEE Access* **2022**, *10*, 42469–42480. [CrossRef]
4. Campolo, C.; Iera, A.; Molinaro, A.; Ruggeri, G. MEC support for 5G-V2X use cases through docker containers. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
5. Abbas, K.; Cho, Y.; Nauman, A.; Khan, P.W.; Khan, T.A.; Kondepu, K. Convergence of AI and MEC for autonomous IoT service provisioning and assurance in B5G. *IEEE Open J. Commun. Soc.* **2023**, *4*, 2913–2929. [CrossRef]
6. Mehmood, A.; Muhammad, A.; Khan, T.A.; Rivera, J.J.D.; Iqbal, J.; Islam, I.U.; Song, W.C. Energy-efficient auto-scaling of virtualized network function instances based on resource execution pattern. *Comput. Electr. Eng.* **2020**, *88*, 106814. [CrossRef]
7. Chen, H.; Abbas, R.; Cheng, P.; Shirvanimoghaddam, M.; Hardjawana, W.; Bao, W.; Li, Y.; Vucetic, B. Ultra-reliable low latency cellular networks: Use cases, challenges and approaches. *IEEE Commun. Mag.* **2018**, *56*, 119–125. [CrossRef]
8. Abbas, K.; Nauman, A.; Bilal, M.; Yoo, J.H.; Hong, J.W.K.; Song, W.C. AI-driven data analytics and intent-based networking for orchestration and control of B5G consumer electronics services. *IEEE Trans. Consum. Electron.* **2023**, *70*, 2155–2169. [CrossRef]
9. ETSI GS MEC. *Multi-Access Edge Computing (MEC); Phase 2: Use Cases and Requirements*; European Telecommunications Standards Institute: Sophia Antipolis, France, 2018.
10. Faisal, A.; Yigitcanlar, T.; Kamruzzaman, M.; Paz, A. Mapping two decades of autonomous vehicle research: A systematic scientometric analysis. *J. Urban Technol.* **2021**, *28*, 45–74. [CrossRef]
11. Naqvi, S.S.A.; Jamil, H.; Iqbal, N.; Khan, S.; Khan, M.A.; Qayyum, F.; Kim, D.H. Evolving Electric Mobility: In-Depth Analysis of Integrated Electronic Control Unit Development in Electric Vehicles. *IEEE Access* **2024**, *12*, 15957–15983. [CrossRef]
12. *EN 302 663 V1. 2.1*; Intelligent Transport Systems (ITS); Access Layer Specification for Intelligent Transport Systems Operating in the 5 GHz Frequency Band. European Telecommunications Standards Institute: Sophia Antipolis, France, 2013.

13. ERTCO; European Commission; 5G PPP. *"5G Automotive Vision", 5GPPP White Paper, October 2015*; ERTICO: Brussels, Belgium; European Commission: Brussels, Belgium, 2018.
14. Rasouli, A.; Tsotsos, J.K. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 900–918. [CrossRef]
15. Weon, I.S.; Woo, S.H. Lane Departure Detecting with Classification of Roadway Based on Bezier Curve Fitting Using DGPS/GIS. *Teh. Vjesn.* **2021**, *28*, 248–255.
16. Blesing, C.; Finke, J.; Hoose, S.; Schweigert, A.; Stenzel, J. Accuracy Evaluation of a Low-Cost Differential Global Positioning System for Mobile Robotics. In Proceedings of the 2023 IEEE SENSORS, Vienna, Austria, 29 October–1 November 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–4.
17. Glomsvoll, O.; Bonenberg, L.K. GNSS jamming resilience for close to shore navigation in the Northern Sea. *J. Navig.* **2017**, *70*, 33–48. [CrossRef]
18. Barrios, C.; Motai, Y. *Predicting Vehicle Trajectory*; CRC Press: Boca Raton, FL, USA, 2017.
19. Fang, Y.; Wang, C.; Yao, W.; Zhao, X.; Zhao, H.; Zha, H. On-road vehicle tracking using part-based particle filter. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 4538–4552. [CrossRef]
20. Raitoharju, M.; Piché, R. On computational complexity reduction methods for Kalman filter extensions. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 2–19. [CrossRef]
21. Daum, F. Nonlinear filters: Beyond the Kalman filter. *IEEE Aerosp. Electron. Syst. Mag.* **2005**, *20*, 57–69. [CrossRef]
22. Yoon, Y.; Kim, T.; Lee, H.; Park, J. Road-aware trajectory prediction for autonomous driving on highways. *Sensors* **2020**, *20*, 4703. [CrossRef]
23. Sanguino, T.d.J.M.; Domínguez, J.M.L.; González, M.J.R.; Martin, J.M.D. New Approach to Intelligent Pedestrian Detection and Signaling on Crosswalks. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 21416–21428. [CrossRef]
24. Wahid, I.; Tanvir, S.; Ahmad, M.; Ullah, F.; AlGhamdi, A.S.; Khan, M.; Alshamrani, S.S. Vehicular Ad Hoc Networks Routing Strategies for Intelligent Transportation System. *Electronics* **2022**, *11*, 2298. [CrossRef]
25. Mehmood, A.; Khan, T.A.; Rivera, J.J.D.; Muhammad, A.; Song, W.C. A Road-aware Approach for Hierarchical Routing in IoV based on Intents and Q-values. In Proceedings of the 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), Tainan, Taiwan, 8–10 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 13–16.
26. Ahmad, S.; Jamil, F.; Iqbal, N.; Kim, D. Optimal route recommendation for waste carrier vehicles for efficient waste collection: A step forward towards sustainable cities. *IEEE Access* **2020**, *8*, 77875–77887. [CrossRef]
27. Mehmood, F.; Ahmad, S.; Ullah, I.; Jamil, F.; Kim, D. Towards a dynamic virtual iot network based on user requirements. *Comput. Mater. Contin.* **2021**, *69*, 2231–2244. [CrossRef]
28. Iqbal, N.; Rizwan, A.; Khan, A.N.; Ahmad, R.; Kim, B.W.; Kim, K.; Kim, D.H. Boreholes Data Analysis Architecture based on Clustering and Prediction Models for Enhancing Underground Safety Verification. *IEEE Access* **2021**, *9*, 78428–78451. [CrossRef]
29. Iqbal, N.; Jamil, F.; Ahmad, S.; Kim, D. Toward effective planning and management using predictive analytics based on rental book data of academic libraries. *IEEE Access* **2020**, *8*, 81978–81996. [CrossRef]
30. Santos Santana, L.; Araújo E Silva Ferraz, G.; Bedin Marin, D.; Dienevam Souza Barbosa, B.; Mendes Dos Santos, L.; Ferreira Ponciano Ferraz, P.; Conti, L.; Camiciottoli, S.; Rossi, G. Influence of flight altitude and control points in the georeferencing of images obtained by unmanned aerial vehicle. *Eur. J. Remote Sens.* **2021**, *54*, 59–71. [CrossRef]
31. Datta, S.K.; Härri, J.; Bonnet, C. IoT platform for precision positioning service for highly autonomous vehicles. In Proceedings of the 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 21–24 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
32. MBo. Find Point Where Altitude Meets Base (Python); Stack Overflow: Online Resource, 2019. Available online: https://stackoverflow.com/a/55233107/3671997 (accessed on 3 October 2024).
33. Feng, H.; Liu, C.; Shu, Y.; Yang, O.W. Location prediction of vehicles in VANETs using a Kalman filter. *Wirel. Pers. Commun.* **2015**, *80*, 543–559. [CrossRef]
34. Bohner, M.; Wintz, N. The Kalman filter for linear systems on time scales. *J. Math. Anal. Appl.* **2013**, *406*, 419–436. [CrossRef]
35. Abbas, M.T.; Jibran, M.A.; Afaq, M.; Song, W.C. An adaptive approach to vehicle trajectory prediction using multimodel Kalman filter. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3734. [CrossRef]
36. Iqbal, R.; Butt, T.A.; Shafiq, M.O.; Talib, M.W.A.; Umar, T. Context-aware data-driven intelligent framework for fog infrastructures in internet of vehicles. *IEEE Access* **2018**, *6*, 58182–58194. [CrossRef]
37. ETSI GS MEC. *Multi-Access Edge Computing (MEC) Framework and Reference Architecture*; European Telecommunications Standards Institute: Sophia Antipolis, France, 2019; Volume 3, p. V2.
38. Nadeem, L.; Azam, M.A.; Amin, Y.; Al-Ghamdi, M.A.; Chai, K.K.; Khan, M.F.N.; Khan, M.A. Integration of D2D, Network Slicing, and MEC in 5G Cellular Networks: Survey and Challenges. *IEEE Access* **2021**, *9*, 37590–37612. [CrossRef]
39. Winarno, E.; Hadikurniawati, W.; Rosso, R.N. Location based service for presence system using haversine method. In Proceedings of the 2017 International Conference on Innovative and Creative Information Technology (ICITech), Salatiga, Indonesia, 2–4 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–4.
40. Kim, Y.K.; Han, J.W.; Park, H. Trajectory prediction for using real data and real meteorological data. In *Ubiquitous Computing Application and Wireless Sensor*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 89–103.
41. Baskar, A.; Xavior, M.A. A facility location model for marine applications. *Mater. Today Proc.* **2021**, *46*, 8143–8147. [CrossRef]

42. Becker, A. Example 9—Vehicle Location Estimation. *Examples* **2021**.
43. Mehmood, A. *Vehicular Trajectories from Jeju, South Korea*; IEEE: Piscataway, NJ, USA, 2022. [CrossRef]
44. SUMO. *Simulation of Urban MObility*; SUMO: Redwood City, CA, USA, 2022.
45. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
46. Doucet, A.; De Freitas, N.; Gordon, N. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*; Springer: New York, NY, USA, 2001; pp. 3–14.
47. Zhao, H.; Yao, L.; Zeng, Z.; Li, D.; Xie, J.; Zhu, W.; Tang, J. An edge streaming data processing framework for autonomous driving. *Connect. Sci.* **2021**, *33*, 173–200. [CrossRef]
48. Benson, T.A.; Kannan, P.; Gupta, P.; Madhavan, B.; Arora, K.S.; Meng, J.; Lau, M.; Dhamija, A.; Krishnamurthy, R.; Sundaresan, S.; et al. NetEdit: An Orchestration Platform for eBPF Network Functions at Scale. In Proceedings of the ACM SIGCOMM 2024 Conference, Sydney, NSW, Australia, 4–8 August 2024; pp. 721–734.
49. Peng, Q.; Xia, Y.; Feng, Z.; Lee, J.; Wu, C.; Luo, X.; Zheng, W.; Pang, S.; Liu, H.; Qin, Y.; et al. Mobility-aware and migration-enabled online edge user allocation in mobile edge computing. In Proceedings of the 2019 IEEE International Conference on Web Services (ICWS), Milan, Italy, 8–13 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 91–98.
50. Mehmood, A. Rectification of Kalman Filter Based Location Predictions; GitHub: Online Resource, 2022. Available online: https://github.com/asifgit/rectification-of-location-prediction-vehicles (accessed on 7 December 2024).

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.