Received: 11 March 2023 | Revised: 15 May 2024 | Accepted: 3 June 2024

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12529

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Deep learning-based location prediction in VANET

Nafiseh Rezazadeh | **Mohammad Ali Amirabadi** | **Mohammad Hossein Kahaei**

School of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran

Correspondence
Mohammad Hossein Kahaei, School of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran 1684613114, Iran.
Email: kahaei@iust.ac.ir

Abstract

In recent years, Vehicular Ad-hoc Network (VANET) has become an essential component of intelligent transportation systems that, along with the previous systems such as traffic condition, accident alert, automatic parking, and cruise control, use the communication of vehicle to vehicle and vehicle to the roadside unit to facilitate road transportation. Several challenges hinder efforts to improve traffic conditions and reduce traffic fatalities through VANET. A critical challenge is achieving highly accurate and reliable vehicle localization within the VANET. Additionally, the frequent unavailability of Global Positioning System (GPS), particularly in tunnels and parking lots, presents another significant obstacle. Traditional methods like Dead Reckoning offer low accuracy and reliability due to accumulating errors. Similarly, GPS positioning, map matching with mobile phone location services, and other existing solutions struggle with accuracy and economic feasibility. In this article, two Kalman filter approaches are used based on signal statistical information and the other learning-based networks, including traditional neural network, deep neural network and LSTM (long short-term memory) to locate the car. The prediction error of car position with root mean square measures. The squared error and distance prediction error are evaluated. It is shown that in terms of prediction time and processing time of vehicle localization, all the vehicle localization methods are efficient in terms of response time for localization, and Kalman filter methods, traditional neural network and deep neural network are faster than LSTM method. Also, in terms of localization error, Kalman filter works better than learning-based methods, and in learning-based methods, both deep neural network and LSTM methods perform better than traditional neural network in terms of localization error.

1 | INTRODUCTION

In recent years, the use of Vehicular Ad hoc Networks (VANETs) as platforms for implementing intelligent transportation systems has been considered by government agencies, industry, and research centers [1, 2]. On the other hand, the significant expansion of vehicular networks to increase road safety, has improved transportation efficiency, road traffic management, and the comfort of vehicle occupants, and has raised concerns about the lack of bandwidth allocated to these networks. The use of localization technology is of great importance in improving the efficiency of vehicular network communications and providing the quality of service expected in safety/non-safety applications [3].

Research in the field of localization for VANET has provided methods to manage and control the efficiency of localization technology in vehicular networks due to the particular characteristics of vehicular networks such as high mobility and variable network topology [4–8]. The challenges of implementing this technology can be explored in the vast and dynamic environment of VANET for new methods in traffic efficiency and access patterns. A VANET is an ad-hoc wireless network of mobile vehicles with infrastructure support. Driver decisions, the continuous movement of vehicles, and high speed have created unique features in these networks. Therefore, localization for data dissemination in these networks is a unique and important issue. Localization in these networks is divided into several categories, one of the most important of which is location-based

localization. General Packet Radio Service (GPRS) localization protocol is one of the leading protocols in location-based networks [9]. In the current transportation systems, most vehicles are equipped with Global Positioning System (GPS) systems. The GPS signal transmitted from satellites can be used to locate a vehicle [10, 11]. The available GPS-based vehicle localization systems employ the Kalman Filter (KF), Neural Network (NN), and Dead Reckoning (DR) algorithms. The KF and DR have better localization accuracies than NN, however, NN outperforms the two others in high noise level [10]. On the other hand, the DR may lead to errors due to information accumulation. Reliability is crucial in GPS-based localization methods. However, KF and DR are dependent on the GPS signal which makes them unreliable since the GPS signal is not constantly available, for example, in tunnels and covered areas the GPS signal is lost. NN is a reliable approach for GPS-based localization, especially in modern cities where GPS information is lost in many places. In this approach, the NN is trained when the GPS signal is available and then employed for localization. Besides, the KF requires the system dynamic equations (a process model and a measurement model) while it is not possible to write the model for all sensors. However, NN inherently extracts the dynamics (model) of the system during the training.

1.1 | Related works

A study by [12] employed a KF to enhance the localization accuracy of a two-wheeled mobile robot. The robot collected data through odometry, an inertial measurement unit, and ultrasonic sensors. The KF processed this data to predict and rectify errors, while the differential stimulus and measurement models accounted for and addressed environmental noise. [13] utilized a KF with repeating data packets. This system dynamically generates location updates based on a predefined threshold and the distance between nearby vehicles. This approach significantly reduces message traffic and improves packet delivery, leading to improved network bandwidth, reduced database load, and enhanced communication reliability. [14] presented a KF-based approach for vehicle localization. Their method avoids storing all previous data by discarding the second-to-last state, making it memory-efficient and suitable for embedded systems that combine GPS and video data. Authors of [15] proposed a system for highly accurate vehicle localization under dynamic and unstable conditions. It combines vehicular kinematic information with localization measurements received from a VANET. The EIAE-KF utilizes an innovative "Adaptive Estimation KF" (IAEKF) structure to partially renew the kinematic information based on the received VANET data. Experiments show that the EIAE-KF achieves the highest level of localization accuracy compared to other methods, even under varying noise levels.

In [16], researchers used an EKF to predict vehicle positions in non-linear motion. Their system leverages RSSI and SINR data and outperforms the ESCL-VNET algorithm in both accuracy and speed. In [17], authors focused on improving localization for vehicles exhibiting nonlinear movement patterns using EKFs. They addressed the issue of speed variability common in urban environments. Their study assumed vehicles equipped with omnidirectional antennas communicating with roadside units and other vehicles using the IEEE 802.11p protocol. Additionally, they worked under the assumption that vehicles could access values such as latitude, longitude, velocity, and acceleration. [18] compared the EKF and the KF for vehicle localization using data from GPS and inertial navigation systems, including steering angle. The EKF proved to be more efficient than the KF, and it also delivered better localization results overall. Notably, simulations revealed that the EKF achieved more accurate localization in highway environments compared to urban settings. Another study [19] explored integrating trace traffic times and a reduced inertial sensor system for vehicle localization. An EKF method was used to limit location errors. Importantly, this system does not require synchronized use of trace traffic. Authors of [20] focused on improving vehicle localization accuracy by combining GPS data, Vehicle-to-Vehicle Infrared Ultrawide Band distance measurements, and turning data from Inertial Measurement Units. They specifically aim to reduce the Geometric Dilution of Precision which can degrade localization. In [21], the authors proposed a subspace algorithm for measuring Time-of-Arrival in localization, assuming direct communication between devices within their transmission range. Their findings suggest that 3D localization methods in VANET setups outperform previous approaches, particularly in high-noise environments with fewer roadside units and a smaller number of autonomous vehicles.

In [11], the authors compared vehicle localization using information integration systems and machine learning. In [22], NN and KF are used for analyzing vehicle location information in four different directions considering real vehicle movement and model-based effects. In [23], authors employed NN as a real-time solution for nonlinear optimization in the localization which is caused by the noise of the distance and angle observations between vehicles.

1.2 | Novelties and contributions

Deep learning (DL) involves many layers, and each layer applies nonlinear processing units to its input to automatically extract the feature. It combines the signals received from the previous layer with the weights and linear transforms and then performs a nonlinear operation on them and produces the output [24–28]. This paper investigates the use of DL for vehicle location prediction. We compare the complexity and performance of our proposed DL methods, specifically Deep Neural Networks (DNNs) and Long Short-Term Memory (LSTM) networks, against traditional KF and NN approaches. DL algorithms, due to efficient hardware generalization and implementation, and parallel distributed architecture, are suitable for solving complex problems such as localization in VANETs. The comparison is conducted on three distinct vehicle movement patterns: circular, helical, and L-shaped, across various time intervals. These trajectories encompass the full spectrum of possible vehicle movements. Our proposed DL methods demonstrate superior localization accuracy compared to NNs, while retaining the

benefits of NNs over KFs as previously discussed. The following outlines the key novelties and contributions of this work,

- We employed both traditional (KF) and learning-based (NN, DNN, and LSTM) methods
- We considered the vehicle movement on three datasets in L, circular, and spiral paths (all possible trajectories)
- We comprehensively analyzed both performance and complexity
- We presented discussions regarding comparison of traditional and learning-based approaches.

This method uses time series prediction models to predict the location of the vehicle in the future based on the time series data of the vehicle's current position and speed. A lookback window is used to collect input data to the model. This window contains k+1 samples, which are the first k samples of the input sequence and the next sample of the output sequence. Supervised learning methods are used to train the model.

This method fills the gap in the current literature in two areas:

- Use of time series prediction models: Most existing methods for location in VANET use traditional machine learning models that are not suitable for predicting time series. Time series forecasting models are specifically designed to predict future values in a time sequence and can perform better in this field.
- Using the look-back window: Most existing methods use the entire data history to train the model, which can be inefficient and costly. Using the look-back window helps to reduce the amount of data required to train the model and increase its efficiency.

This method advances the field in two areas:

- Accuracy: Experimental results show that this method can significantly improve the location accuracy compared to existing methods.
- Efficiency: This method can significantly increase the efficiency of the model by using the look-back window.

In this analysis, we examine the sensitivity of different vehicle localization models (KF, traditional NN, DNN, and LSTM) to changes in input parameters. We choose 3 key parameters for sensitivity analysis:

- Car movement path: circular, spiral and L-shaped
- Sampling interval: 0.5, 1, 1.5 and 2 seconds
- Noise: Normal distribution with different mean and variance

These parameters directly affect the performance of car localization models. We use different methods for sensitivity analysis, including, sampling (we collect different samples of data for each combination of input parameters), evaluation (we evaluate the performance of the models on each set of samples and measure the localization error and processing time), visualization (we use graphs and tables to show the results of sensitivity analysis), and findings (performance of all models is highly dependent on the vehicle's trajectory). Results show that:

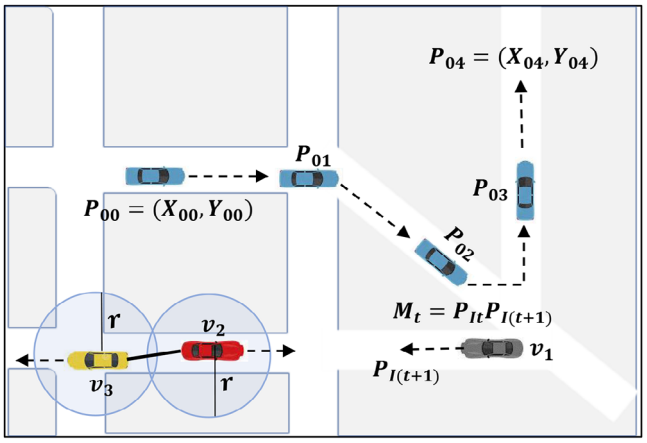


FIGURE 1 Localization and predicting network nodes.

- The KF generally has the lowest localization error, especially in spiral and L-shaped trajectories.
- DNN and LSTM perform better than traditional NN, especially at short sampling intervals.
- The performance of the models strongly depends on the direction of the car. For example, a model trained on a circular path will perform poorly on a spiral path.
- Machine learning-based models have a weaker performance against changes in the environment compared to the KF due to weaker self-organization.
- Choosing the right model for vehicle location depends on various factors such as vehicle movement path, sampling interval and noise level.

This paper proceeds as follows: Section 2 explains the problem statement, Section 3 outlines the proposed methods, Section 4 presents the simulation results, Section 5 offers discussions, and Section 6 concludes with a summary and considerations for future work.

2 | PROBLEM STATEMENT

In this article, we will solve the problem of publishing old location information in VANET by predicting car location. Vehicle localization concerning the vehicle to infrastructure communication in the VANET is shown in Figure 1. The VANET can be expressed through N node Euclidean graph $G = (V, E, r)$ where r is range of vehicles communication, $V = \{v_0, v_1, v_2, ..., v_{N-1}\}$ is a collection of vehicles, if v_i reaches v_j then $i, j \geq \epsilon E$, node v_i is at a distance r from node v_j. $P_{it} = (X_{it}, Y_{it}) \in R^2$ is for calculating the position of nodes v_i based on $v_i \in V$ in a localization system. S_i is its displacement speed, and $L_{it} = (X_{it}, Y_{it})$ is the actual location of the nodes in a discrete-time t.

In the VANET network, car data including car location and speed $P_t = (x_t, y_t, vx_t, vy_t) \in R$ has been calculated by positioning systems where x_t and y_t are the coordinates of the car location and vx_t and vy_t are the speed in the direction x and the speed in the direction in y, respectively. We want to calculate the prediction of vehicle location for discrete time step $t + 1$ based on current knowledge of vehicle location (P_t) at time step t and previous knowledge of vehicle location (P_{t-n}) at time steps $t - n$, where n is the number of previous time steps to calculate $P_{t+1} = (x_{t+1}, y_{t+1})$.

In fact, our input is the position and velocity of the vehicle, the combination of n previous time steps, these values are given to the function F (vehicle location prediction algorithms), and the prediction of the next location of the vehicle is modeled based on this function. This model can be used as a time series regression forecasting problem and also provide a target localization problem. The main goal of localization, detection and continuous estimation is to evolve the target state with respect to time and update the estimation with measurements. Since almost all localization methods are model-based and we considered the speed to be constant. In this way, we can perform the vehicle localization by the discrete time-space model as $x_{t+1} = x_t + vx_t^T$, and $y_{t+1} = y_t + vy_t^T$, where T is the time interval, and x_{t+1} and y_{t+1} are the following coordinates of the vehicle location predicted by the model.

3 | DL-BASED LOCALIZATION

The researches conducted in the field of positioning application in vehicular contingency networks due to the special characteristics of vehicular networks, including high mobility and variable network topology, have provided the methods used to manage and control more efficiency in the positioning technology in vehicular networks. We can examine the challenges in implementing this technology in the wide and dynamic environment of automobile networks in the new methods of using location in vehicular contingency networks, especially in the field of traffic efficiency and access patterns. Based on this, localization for data dissemination in these networks is a special and important issue. Localization in these networks are divided into several categories, one of the most important of which is location-based localization. GPRS location protocol is one of the leading protocols in location-based networks. In fact, one of the most important problems to be solved in the VANET vehicle network is very accurate and reliable localization that we can provide. In the current technology system, most vehicles are equipped with GPS system.

Among the location methods, satellite location using GPS, location using map matching method, location using image and video processing, location using mobile phone etc. can be mentioned. GPS positioning is a global satellite positioning system that can be used to identify the location of a vehicle. Considering that the previous methods had low accuracy, unreliable and high measurement and calculation cost, therefore in our problem, the proposed method to improve the location of the car and also reduce the measurement cost and reduce the calculation and increase the accuracy of one of the methods We use machine learning called DL in the Python software environment considering vehicle-to-infrastructure communication in the VANET network.

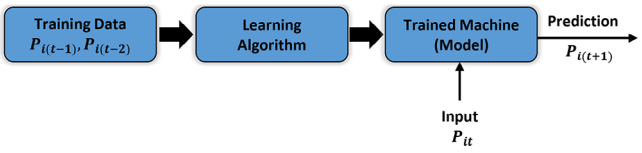


FIGURE 2 Schematic of DL-based localization.

Due to the fact that in places where we do not have GPS information, we have to use machine learning, that is, when the GPS signal is available, the machine learning algorithm is trained, and when the GPS information is lost and we do not have the GPS information, the trained algorithm can locate. to do Also, because we want to locate any system with the KF method, we must write dynamic equations (a process model and a measurement model) for it. Due to the fact that not all sensors have the ability to write a model for them, we have to use machine learning, which does not need to do this, and during the training, they learn the dynamics of the system. In the proposed method, we will use a DNN with a regular layer structure. But we also use the LSTM to check which of these methods is the most suitable solution. Because of the time dependence, the RNN structure may be better.

In this paper, we employ KF, NN, DNN, and LSTM. KF and NN are deployed based on instructions provided by [10]. In the following, we explain the proposed DL-based methods.

Figure 2 shows the predictive steps in DL-based methods. DL-based methods predict the vehicle location based on a target value, $y_{i,t}$ (next vehicle location for the entity i at time t). In its simplest form, time series prediction models can predict one-step-ahead expressed as $\hat{y}_{i,t+1} = f(y_{i,t-k:t}, x_{i,t-k:t}, s_i)$. The vector $\hat{y}_{i,t+1}$ is the prediction of the next vehicle location, $y_{i,t-k:t} = \{y_{i,t-k}, \ldots, y_{i,t}\}$ the vector of observations and measurements of vehicle location and $x_{i,t-k:t} = \{x_{i,t-k}, \ldots, x_{i,t}\}$ input vector, which is based on the so-called lookback window to k windows. The static metadata, s_i, is based on the entity (e.g. location of the sensor) and the prediction function, $f(.)$, is learned by the DL-based methods. The lookback window arranges the input dataset by sliding a window of width $k + 1$. The samples in this window overlap which generally obtain from the $N - k$ window. The k first values in each window is the input sequence, and the next values are the output sequence. For instance, for $N = 8$ samples and the lookback window size $k = 3$, the data set will be arranged as $\{[input_i|output_i]\} = \{[t_1, t_2, t_3|t_4], [t_2, t_3, t_4|t_5], [t_3, t_4, t_5|t_6], [t_4, t_5, t_6|t_7], [t_5, t_6, t_7|t_8]\}$ [27].

To predict with the learning-based methods, we specify the input and target of time intervals data for all trajectories using the lookback window method. We set the window size to 20 for all time interval data in all trajectories, which specify the input and target of the learning-based methods. Also, our target is one-step-ahead (position x and position y). In fact, we use the previous 20 samples of time interval data (lookback

TABLE 1 Architecture of learning-based methods.

Dataset	Method	Epoch	Hidden layers	Neurons	Activation function	Momentum	Optimizer	Batch size
Circular trajectory	NN [10]	400	1	1100	Tangent hyperbolic	0.89	SGD	32
	DNN	400	3	1100	Tangent hyperbolic	0.95	SGD	32
	LSTM	400	1	2048	Tangent hyperbolic	0.95	SGD	32
Helical trajectory	NN [10]	400	1	1100	Tangent hyperbolic	0.89	SGD	32
	DNN	400	3	1024	Tangent hyperbolic	0.95	SGD	32
	LSTM	400	1	2048	Tangent hyperbolic	0.95	SGD	32
L trajectory	NN [10]	400	1	1100	Tangent hyperbolic	0.89	SGD	32
	DNN	400	3	1800	Tangent hyperbolic	0.95	SGD	32
	LSTM	400	1	2048	Tangent hyperbolic	0.95	SGD	32

window = 20) to predict the next vehicle location. We have four values for four features in each sample; as a result, we have 80 values in each input. Hence, we determine 80 neurons in the input layer for DNN, and LSTM. We also need two neurons in the output layer to predict one-step-ahead. After preparing the input and target, we scale the values of this data such that the proposed DL-based methods converge faster. We have used the Min-Max scaler for circular and helical trajectories time intervals data and the Standardizer for L trajectory time intervals data. The architecture of learning-based methods is given in Table 1.

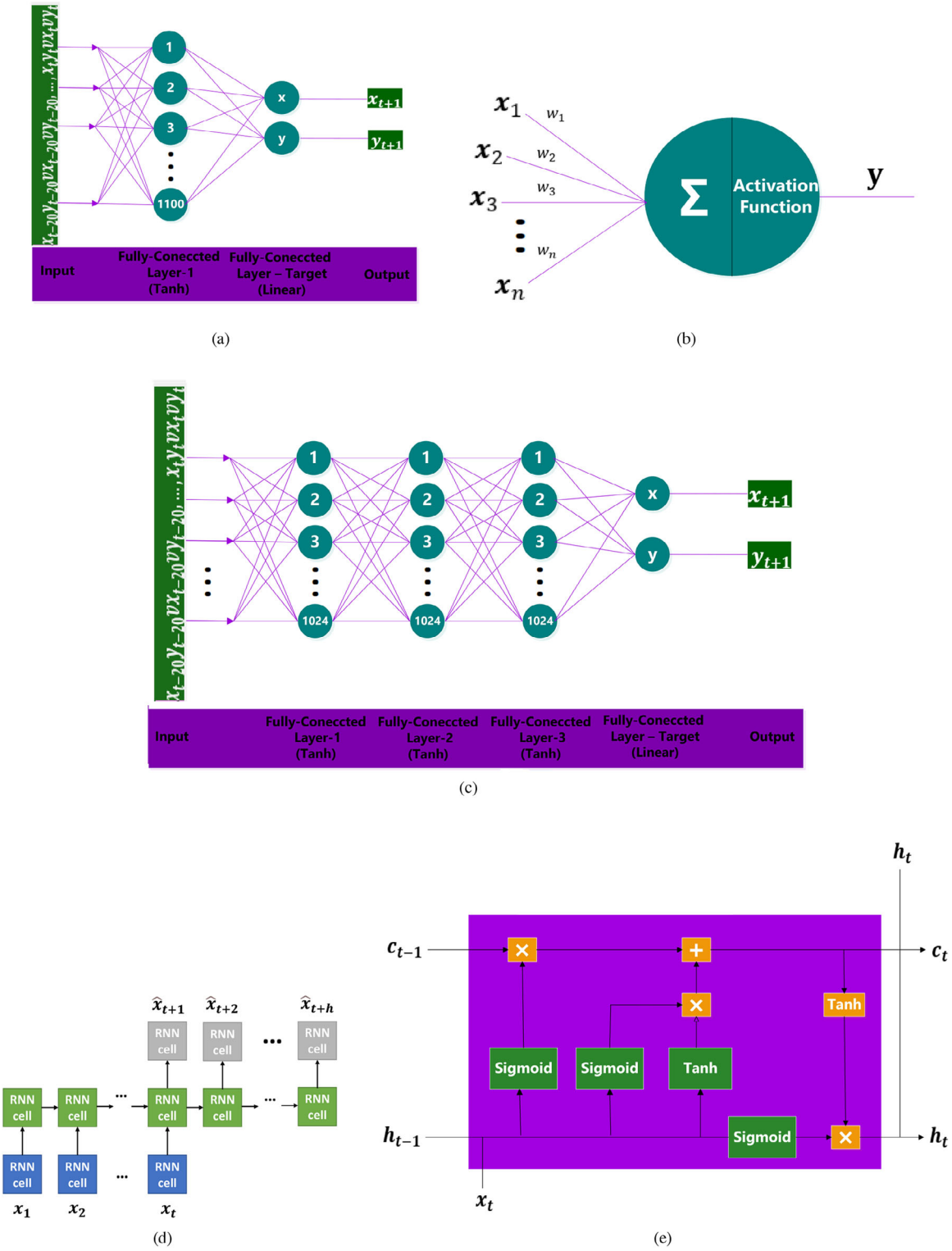
Learning-based algorithms receive environmental conditions as the input data (vehicle location and speed at time steps t-20 to t) and the next vehicle position (t+1) during the training. Using mathematical calculations, they aim to find a mapping function that maps the input to the output with minimal error. The learning process allows the learning-based methods to continuously learn and update their parameters as new data becomes available, adapting to environmental conditions. The values of the mapping function are updated at each time step based on the data to improve the function's understanding of the environmental conditions. At the end of training, the best mapping function is obtained based on the data. This mapping function can predict the vehicle's position based on the environmental conditions (input data) it was trained on. Thus, the more the algorithm is trained on different environmental conditions (more data) during the training phase, the better the results it can provide. The LSTM, due to its ability to learn long-term temporal dependencies on data, utilizes information from the previous neurons in the current layer as well as the previous layers to predict the next time step (next vehicle location). This technique (learning long-term temporal dependencies on data) can achieve better results on time series data.

Figure 3 shows the process of predicting the next position of the car for each of the learning-based algorithms. We determined the input and output data of the time intervals of all three routes with the look-back window method. The input is such that we have considered the window size for time interval data in all types of trajectories (circular, spiral, and L) (20) to specify the input of learning-based methods. Actually, we use the previous 20 samples of time interval data to predict the location of the car, and in each sample we have four values for four features.

In the NN architecture (Figure 3a), the inputs first enter the neurons (Figure 3b) of the first layer, the number of neurons of the first layer is 1100 and its activity function is the hyperbolic tangent, based on their structure, they apply mathematical operations as explained by [29] on the input. Finally, the output of this layer goes to the last layer, and the last layer has 2 neurons that determine the next position of the car by a linear activity function which works best when we want to predict a real number.

DNN architecture is similar to NN and the only difference is in the number of layers and the number of hidden layer neurons. In this architecture, we used 3 hidden layers of each layer with 1024 neurons, and the mathematical operation of the DNN architecture neuron is exactly similar to the NN architecture. Calculations of this type of neuron structure (Figure 3c) in these two architectures can be done in parallel in each layer, and this increases the speed of these architectures.

In the LSTM architecture (Figure 3d), the structure of the neuron, as you can see in (Figure 3e), is completely different from the structure of the neuron in the NN and DNN architecture (Figure 3b). Also, due to the connections of neurons in each layer of the LSTM architecture, parallel calculations of neurons in each layer are not possible. In fact, in addition to being connected to the neuron of the previous layer, each neuron is also connected to the previous neuron in the same layer. One reason LSTM is slower than DNN and NN is the connections of neurons in each layer, and another reason is that each neuron has heavy mathematical calculations based on its structure. In fact, each neuron depends on the input of the previous neuron in the current layer in addition to the input of the previous layer to perform its calculations. The neurons are calculated sequentially with the hyperbolic tangent activity function until the last neuron, the output of the last neuron is connected to two neurons, which is responsible for calculating the next position of the vehicle, and the activity function of these neurons is also linear like the previous two architectures. All the hyperparameters of all three architectures, such as the number of neurons and activity functions, were selected based on trial and error, in fact, they had the best performance.



(a) (b) (c) (d) (e)

FIGURE 3 (a) NN, (b) NN neuron, (c) DNN, (d) LSTM, and (e) LSTM neuron structures.

4 | DATASET GENERATION

We synthetically generated three datasets for vehicle movement on circular, helical, and L trajectories shown in Figure 4. Each dataset has 40000 points, after scaling in all trajectories time intervals data, we considered 80% of the data as train data and 20% of the data as test data. We assume that the sampling is every 0.5 seconds. The feature vector is composed of four features including position x, position y, velocity in the x-direction, and velocity in the y-direction. We also considered the vehicle

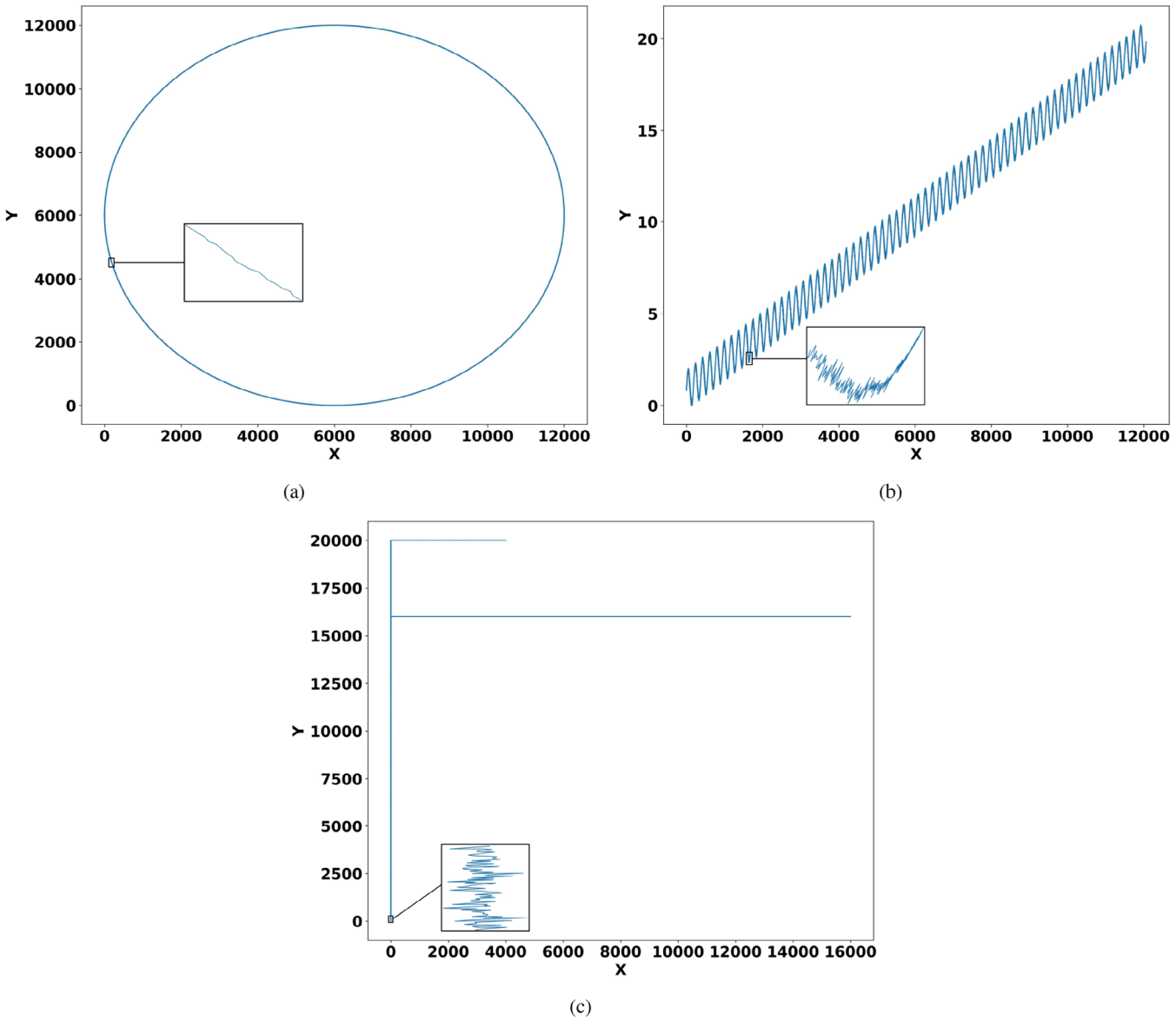


FIGURE 4 (a) circular, (b) helical, (c) and L trajectories.

with constant velocity. To be more practical, we add Gaussian noise to each trajectory, the circular and L trajectories have a mean of 0.5 and variance of 0.5, and the helical trajectory has a mean of 0.1 and a variance of 0.5. The generated datasets are publicly available[1].

5 | SIMULATION RESULTS

We use Keras [30] and TensorFlow [31] Python libraries to implement the proposed learning-based algorithms. TensorFlow is an open source DL framework developed by Google that is used to express the interface of machine learning algorithms and implement the algorithms. Keras is a high-level neural network API and can run on top of TensorFlow. To implement the KF, we used the Python library Filterpy, which has implemented a number of Bayesian filters, especially KFs [32].

We evaluate the performance of localization methods by Root Mean Square Error (RMSE) and Predicted Distance Error (PDE). The PDE and RMSE are common metrics used to evaluate the performance of the KF by assessing the difference between the filter's estimates and the actual measurements. The KF can obtain optimal sequential estimates when the noise is Gaussian and also optimal linear estimation even when the errors are not Gaussian. KF from a linear operator to generate a new state (prediction) at each non-continuous time along with the noise. This filter is a set of state space models and mathematical equations, which is a two-stage estimator in the first stage of prediction and in the second stage of updating based on the current data. At each time step, the KF predicts the next position of

[1] https://drive.google.com/file/d/1u69ZYbmCzquVTOcB9xAC0Dvuw8GBSXf4/view?usp=sharing

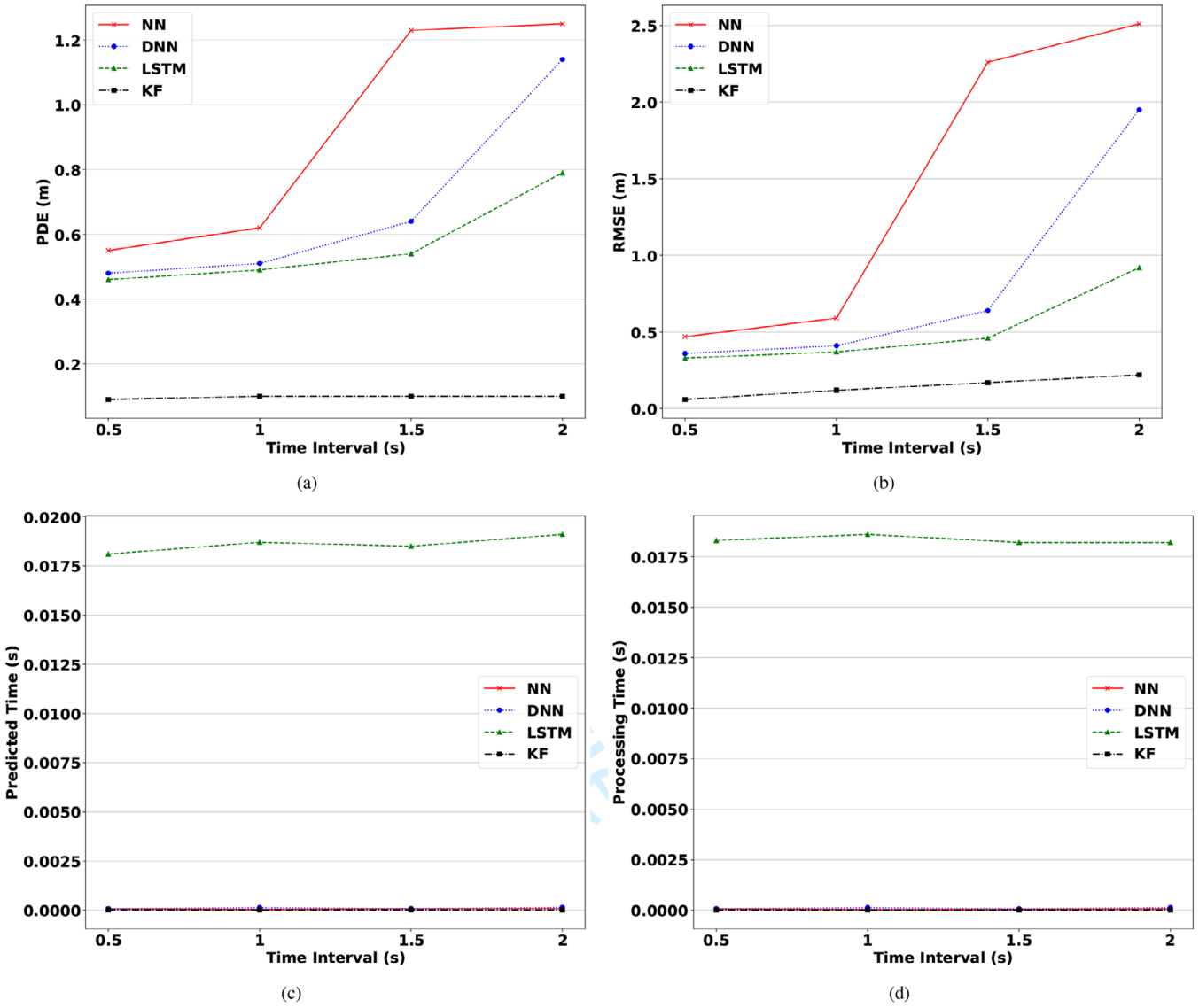


FIGURE 5 (a) PDE, (b) RMSE, (c) predicted time, and (d) processing time for circular trajectory, considering NN, DNN, LSTM, and KF.

the vehicle (prediction step) and these predicted values are measured with the actual values by these criteria for each of the time intervals, and finally the average of these values is presented.

Also, we use predicted time and processing time to evaluate the execution time. Predicted time is the average time that a vehicle predicts its next location and the processing time is the average time for the localization at each time step. We predict vehicle location for time intervals 0.5, 1, 1.5, and 2 seconds.

Figure 5 demonstrates (a) PDE, (b) RMSE, (c) predicted time, and (d) processing time for circular trajectory, considering NN, DNN, LSTM, and KF. As seen in Figures 4a and 4b, all methods predict well. Also, as the time interval of the samples increases, the error of these methods increases. However, in the PDE parameter, with increasing time intervals, the increase of the KF method error is negligible. Among vehicle localization methods, the KF has provided better results than the other methods. The LSTM provides better results among learning-based methods. Also, in the learning-based methods, both our proposed methods LSTM and DNN are better than the NN of reference [10]. As seen in Figures 4c and 4d, the KF is the fastest method, and also the LSTM is the slowest. Of course, all methods are effective for vehicle localization, since the minimum time interval we have in the dataset is 0.5 seconds, and our slowest method is the LSTM which can perform vehicle localization in 0.02 seconds and process each step time to predict.

Figure 6 demonstrates (a) PDE, (b) RMSE, (c) predicted time, and (d) processing time for helical trajectory, considering NN, DNN, LSTM, and KF. As seen in Figures 5a and 5b, the obtained errors are much more than the circular trajectory. In helical trajectory like circular trajectory, the error increases with increasing time intervals. The KF method in this trajectory, like the circular trajectory, has a small error, however, the errors of learning-based methods NN, DNN, and LSTM are higher. Similar to the circular trajectory in the helical trajectory

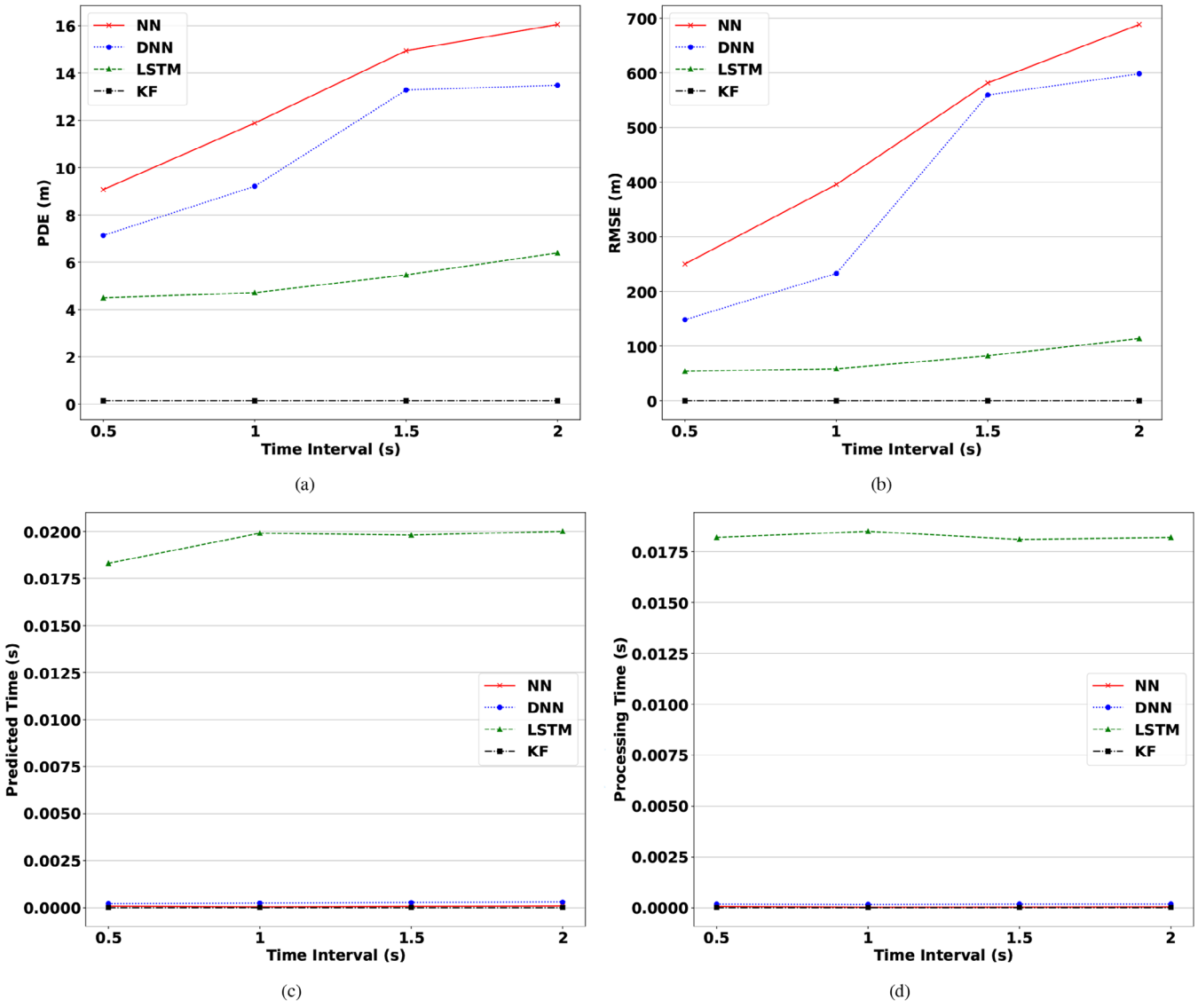


FIGURE 6 (a) PDE, (b) RMSE, (c) predicted time, and (d) processing time for helical trajectory, considering NN, DNN, LSTM, and KF.

among learning-based algorithms, the LSTM has less error than the other two algorithms. In learning-based algorithms, our proposed algorithms LSTM and DNN have less error than the NN of reference [10]. Also, in this trajectory, like the circular trajectory, the KF is the fastest method, and the LSTM is the slowest method. Similarly, all methods are ideal for localization since our slowest method is the LSTM, which can predict the vehicle location in less than 0.02 seconds and process each time step for forecasting.

Figure 7 plots (a) PDE, (b) RMSE, (c) predicted time, and (d) processing time for L trajectory, considering NN, DNN, LSTM, and KF. As seen in Figures 6a and 6b, due to the breakpoint in L trajectory, the errors of the learning-based methods in the PDE parameter are high, but in the PDE parameter, the KF method has a small error. However, in the RMSE parameter error, the KF method, like the previous two trajectories, cannot provide less error, and also, the breakpoint has a negative effect on the KF method in this parameter but also the results of the KF in this trajectory are still better than other methods. In the L trajectory, as in the circular and helical trajectories, the error increases with increasing time intervals. As with the previous two trajectories in the L trajectory, the LSTM has less error between learning-based methods than the other two methods. Also, our proposed methods LSTM and DNN, have less error in learning-based methods than the NN of reference [10]. Like the previous two trajectories, as seen in Figures 6c and 6d, the LSTM is the slowest, and the KF is the fastest. All methods are ideal for vehicle localization, and they can perform localization in less than 0.02 seconds and also processed each time step for prediction.

6 | DISCUSSIONS

In this article, we evaluate the location of the VANET. The methods that have been used to predict the vehicle position

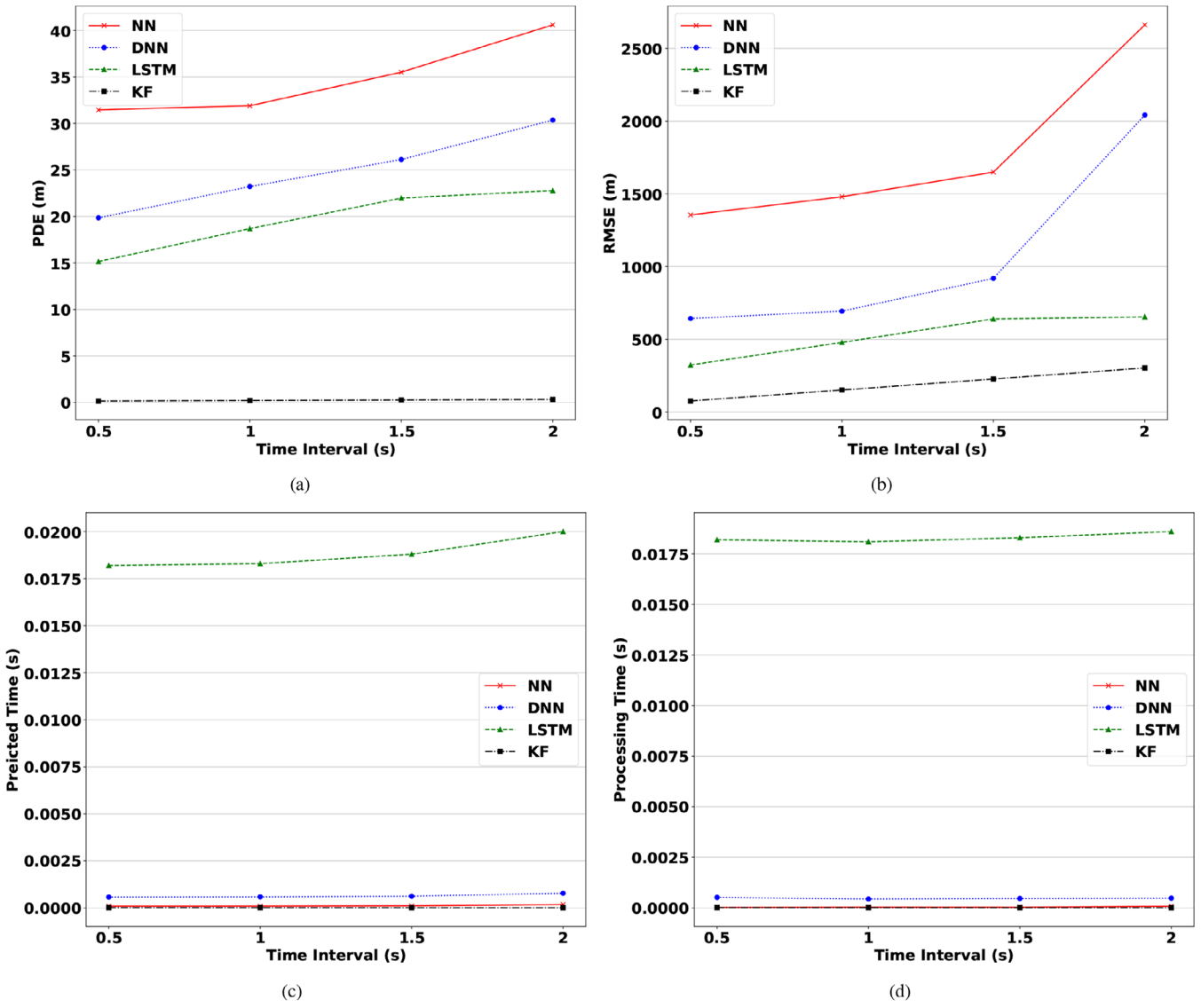


FIGURE 7 (a) PDE, (b) RMSE, (c) predicted time, and (d) processing time for L trajectory, considering NN, DNN, LSTM, and KF.

include KF based on signal statistical information and learning-based methods including traditional NN, DNN and LSTM network. We evaluate these methods on the data of time intervals of 0.5, 1, 1.5 and 2 seconds in three circular, spiral and L paths. In fact, we evaluate the prediction error of the vehicle's next position with two measures of RMSE and PDE, and also used two measures of prediction time and processing time to evaluate the execution time. We fed the data raw as input to the KF, but for the learning-based methods, we first transformed the data into a feedable structure for these methods with a look-back window method, then normalized the data to converge faster and finally, after applying these two approaches, we fed the data as input to the learning-based networks. As the time interval increases, the vehicle position prediction error also increased in all methods and all routes. The lowest localization error among all methods in all directions was obtained by KF. Also, among the learning-based methods, the proposed methods including LSTM and DNN, respectively, due to considering the time dependence of the data and learning specific functions from the data, obtained the least error in all the time intervals of the routes for locating the vehicle compared to the reference NN method. The LSTM method also had the lowest localization error among the learning-based localization methods. Therefore, when the GPS signal is available, the LSTM is trained and in places such as tunnels where the GPS signal is not available, it can perform localization based on the training it has seen. Also, all the computational performance methods are efficient in terms of response time (prediction time and processing time). The KF method was faster than all methods, and the LSTM method, due to its block structure that includes three gates, has a large computational volume compared to the rest of the methods and was therefore slower. The opinion of the response time in localization for all time intervals of the routes was obtained by the KF and when the GPS signal is

available, it can perform the localization with high accuracy and less time.

We compared learning-based methods with the KF which is the optimal in our case. The KF is indeed optimal when the system dynamics (car movement) are linear and the noise affecting the measurements is Gaussian. In these scenarios, the KF minimizes the mean squared error of the state estimate (car position and velocity). Compared to more complex learning-based methods, the KF is relatively simpler to implement and computationally efficient. It also exhibits good robustness to sensor noise within its assumed noise model. We have proposed learning-based methods as a possible alternative to traditional algorithms such as the KF. One of the biggest advantages of the learning-based approach is that we do not need to scale the algorithm parameters for testing and integration, which is a major problem in KF-based approaches. In fact, this is done automatically in the training phase by network-based methods with sufficient training, learning-based methods can learn the patterns needed to perform robustly in diverse test scenarios. In addition, by improving the architecture of the model, increasing the number of its parameters, and increasing the number of examples for training, learning-based algorithms may be able to identify precise and subtle patterns that may be difficult to include in traditional algorithms.

Loss functions play an important role in implementing and improving the performance of deep learning algorithms. In our problem, for all learning-based methods (NN, DNN, and RNN), we use the RMSE, which is one of the most common loss functions in regression problems, to train machine learning models. This function measures the squared error between the actual value of the vector y (the actual location of the vehicle) and the predicted value f(x) of the sample vector x. Then the gradient of the loss function is calculated, which is a variable value. If the value of the gradient is close to zero, then the length of the gradient is small, and if the length of the gradient is large, then the value of the loss function is large. These properties are important for fast convergence and high accuracy of the model. Scaling methods such as standardization and min-max scaling are used to equalize the range of feature values and accelerate convergence. RMSE, predicted distance error, prediction time and processing time are used to evaluate the performance of localization methods. The proposed method based on DNN and LSTM is superior to other methods in terms of location accuracy and computational efficiency.

The employed LSTM architecture proved to be a good candidate for position estimation even with a limited number of parameters and with only one data sample per path. the results of this method are very close to the KF method, and by increasing the number of training samples and model capacity (improving the model architecture, increasing the number of its parameters), we can get much better results. One of the biggest disadvantages of the KF method is that to predict the next position of the vehicle at each time step, it needs a GPS signal to update, which GPS signal cannot be received due to communication problems in places such as tunnels where there is no communication, hence the prediction error. The KF increases significantly to become virtually unusable. However, in learning-based methods, training is done using the inertial navigation sensor input signal and the GPS target signal, and it does not need the GPS signal in the prediction stage, and in fact, its performance is stable in all location.

We use two measures of prediction time and processing time to evaluate the execution time. The prediction time is the average time required for the vehicle to predict its next location, and the processing time is the average time required to predict the location of the vehicle at each time step. In terms of applicability and importance of predicted time in VANET, the calculation of predicted locations should be done in a time less than the time step in order to be able to use it. In our work, the slowest method is LSTM, which takes 0.02 seconds to process and predict, and our lowest time interval in the data is 0.5 seconds. In this way, the time to predict the next position of the car is much less and optimal. In fact, after predicting the next position of the car, we can perform practical implementations such as safety-related applications, traffic management and productivity-related applications, entertainment and comfort-related applications etc. in real time.

One of the major distinctions of our approach is that it eliminates the need for the tune algorithm parameters for testing and integration. This is a major issue with traditional algorithms like KF-based approaches. This is automatically done during the training phase by learning-based methods. In addition, by improving the model architecture, increasing the number of its parameters, and increasing the number of samples for training, learning-based algorithms can identify accurate and subtle patterns that may be difficult to incorporate into traditional algorithms. One of the major drawbacks of the KF method is its dependency on GPS signals for updating the vehicle's position at each time step. In places like tunnels where communication is unavailable, the GPS signal cannot be received, which significantly increases the prediction error of the KF and renders it practically unusable. On the other hand, learning-based methods are trained beforehand using INS input signals and GPS target signals. Therefore, they do not require GPS signals during the prediction phase, resulting in stable performance in all locations. Note that the learning-based algorithm is trained when the GPS signal is present, allowing it to perform localization when GPS information is lost or unavailable, and we resort to learning-based methods in places where the GPS information is unavailable. When using the KF for localization, we need to write dynamic equations (a process model and a measurement model) for each system. Since not all sensors can be modeled mathematically, we are forced to use learning-based methods, which do not require this step, and learn the system dynamics during training. The employed LSTM architecture showed that it is a good candidate for position estimation even with a limited number of parameters and only one data sample per path, compared to the NN and DNN. A NN is incapable of learning specific functions and complex patterns from data due to its single-layer structure. On the other hand, a DNN can effectively learn nonlinear patterns, unlike the NN, which is supported by the quantitative results presented in Figures 5–7 for the PDE and RMSE evaluation metrics.

7 | FUTURE WORKS

In future works, to increase the accuracy of the prediction of learning-based methods, it is possible to collect the data of a large number of cars moving on a fixed route and make predictions based on them. The more the number of cars, the better the performance of these methods because they can learn more car states. As you can see, LSTM networks performed better than other learning-based methods because, based on its structure, it can also consider long-term dependence between data, and if this method is combined with the attention mechanism, it can perform better.

Measurement errors and sudden changes in speed and traffic density can significantly affect the accuracy of vehicle prediction. Here are some key effects:

- Measurement errors: Positioning errors in GPS and other positioning sensors can lead to inaccurate route and time of arrival predictions. Speed errors in the speedometer and other speed sensors can lead to inaccurate predictions of travel time and fuel consumption. Traffic conditions errors in traffic sensors and traffic data can lead to inaccurate predictions of travel time and traffic congestion.
- Sudden changes in traffic speed and density: Accidents and blockages can suddenly change traffic patterns and lead to inaccurate travel time and arrival time predictions. Weather conditions such as rain and snow can lead to reduced speed and increased traffic congestion and lead to incorrect travel time and fuel consumption predictions. Special events such as concerts and sports games can lead to sudden increases in traffic in an area and lead to inaccurate predictions of travel time and traffic congestion.

To reduce the impact of these factors on the accuracy of vehicle prediction, different methods can be used:

- Using multiple data sources such as GPS, traffic sensors, and traffic data from different sources can increase the accuracy of prediction.
- Using advanced machine learning algorithms can help predict traffic patterns more accurately.
- Data updates in form of regular updates of location, speed and traffic data can help keep forecasts accurate.

Despite these challenges, vehicle prediction remains an active research field and significant progress has been made in recent years. With continued research and development, it can be expected that the accuracy of car prediction will increase significantly in the future.

To improve the accuracy of car prediction, several special machine learning methods can be used:

- DL can learn complex patterns in data that traditional machine learning models are unable to learn. This makes them ideal for vehicle prediction, which requires taking into account multiple factors such as position, speed, traffic and weather conditions.
- Reinforcement learning is a type of machine learning that teaches agents to behave in an environment by performing actions that maximize rewards. This can be done to train predictive vehicle models to learn how to predict traffic patterns in a way that minimizes travel time and fuel consumption.
- Federated learning is a type of machine learning that allows multiple models to simultaneously learn from their local data and share their knowledge. This can be used to train vehicle prediction models to learn from data collected from a large number of vehicles, which can help improve the accuracy of predictions, especially in areas with limited data.
- Blended learning is a type of machine learning that uses multiple machine learning algorithms to improve overall performance. This can be used to train predictive vehicle models to take advantage of the strengths of different algorithms to achieve greater accuracy.

In addition to these methods, current research is focused on developing new machine learning methods specifically for vehicle prediction. These methods include methods for real-time traffic modeling, taking into account driver behavior and predicting traffic events such as accidents and blockages.

With continued research and development, it can be expected that the accuracy of car prediction will increase significantly in the future. This can lead to numerous benefits including reduced travel time, reduced fuel consumption and safer driving.

Research findings on car prediction can specifically help solve the problem of releasing old positioning information in several ways:

- More accurate position prediction: Vehicle prediction models that use advanced machine learning methods such as DL can more accurately estimate the current and future position of the vehicle. This allows navigation systems and ride-sharing apps to provide users with more accurate information about the location of vehicles, even when the positioning information is out of date.
- Faster information update: Using reinforcement learning, predictive vehicle models can actively learn from new positioning data and update their information quickly. This is especially useful in areas with heavy traffic or in situations where the position of vehicles is changing rapidly.
- Position estimation in the absence of data: Machine learning algorithms such as hybrid learning can be used to estimate the position of vehicles in situations where no positioning information is available. This can be useful in areas with poor GPS coverage or when the GPS signal of vehicles is disrupted.
- Identifying and filtering false information: Vehicle predictive models can be used to identify and filter out incorrect or outdated positioning information. This can help improve the quality of positioning data and provide more accurate information to users.
- Prediction of driver behavior: Using machine learning, models can be created to predict driver behavior such as speed, route and future actions. This information can be used to more accurately predict the location of vehicles and provide

users with more useful information such as traffic alerts and arrival time estimates.

Overall, the research findings on car prediction have a significant potential to solve the problem of disseminating old positioning information and provide more accurate and reliable information to users of navigation systems and trip sharing applications.

8 | CONCLUSION

In this paper, we did positioning of VANET by using traditional methods such as KF, learning-based methods such as NN, DNN, and LSTM. In places such as tunnels or today's busy cities where the GPS signal is frequently lost, KF is not efficient and reliable, and learning-based methods can perform positioning. Learning-based systems excel in situations where GPS is unreliable, such as tunnels or urban canyons. By streamlining calculations and enhancing accuracy, it can rapidly establish driver behavior patterns, even for non-linear movements. Among the learning-based methods, the proposed methods, namely LSTM and DNN, obtained less error than the NN method in all time intervals for vehicle positioning, which is due to learning the long-term dependencies of the data and learning the specific functions of the data. All methods were efficient in terms of computational performance and response time (prediction and processing times).

In future works, to improve the prediction accuracy of learning-based methods, the data from a large number of vehicles moving on a fixed path can be collected and used for prediction. The more vehicles (the more data) there are, the better these methods will perform, as they can be trained on a wider range of vehicle states. In fact, the mapping function can predict the vehicle's position based on the environmental conditions (input data) it was trained on. The more the algorithm is trained on different environmental conditions (more data) during the training phase, the better the results it can provide. As observed, LSTM networks outperformed the other learning-based methods because their structure allows them to consider the long-term dependencies between data samples. Combining this approach with an attention mechanism can further improve performance.

AUTHOR CONTRIBUTIONS

Nafiseh Rezazadeh: Conception and design of study; writing—original draft; writing—review & editing. **Mohammad Ali Amirabadi**: Conception and design of study; writing—original draft; writing—review & editing. **Mohammad Hossein Kahaei**: Conception and design of study; writing—original draft; writing—review & editing.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Research data are shared via link.

ORCID

Mohammad Ali Amirabadi https://orcid.org/0000-0003-4190-3380
Mohammad Hossein Kahaei https://orcid.org/0000-0002-1920-659X

REFERENCES

1. Nokhodberiz, N.S., Nemati, H., Montazeri, A.: Event-triggered based state estimation for autonomous operation of an aerial robotic vehicle. IFAC-PapersOnLine 52(13), 2348–2353 (2019)
2. Nemati, H., Montazeri, A.: Analysis and design of a multi-channel time-varying sliding mode controller and its application in unmanned aerial vehicles. IFAC-PapersOnLine 51(22), 244–249 (2018)
3. Doolan, R., Muntean, G.M.: EcoTrec—A novel VANET-based approach to reducing vehicle emissions. IEEE Trans. Intell. Transp. Syst. 18(3), 608–620 (2016)
4. Soatti, G., Nicoli, M., Garcia, N., Denis, B., Raulefs, R., Wymeersch, H.: Implicit cooperative positioning in vehicular networks. IEEE Trans. Intell. Transp. Syst. 19(12), 3964–3980 (2018)
5. Rohani, M., Gingras, D., Vigneron, V., Gruyer, D.: A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and VANET based inter-vehicle distance measurement. IEEE Intell. Transp. Syst. Mag. 7(2), 85–95 (2015)
6. Lobo, F., Grael, D., Oliveira, H., Villas, L., Almehmadi, A., El-Khatib, K.: Cooperative localization improvement using distance information in vehicular ad hoc networks. Sensors 19(23), 5231 (2019)
7. Howard, A., Mataric, M.J., Sukhatme, G.S.: Putting the 'I' in 'team': An ego-centric approach to cooperative localization. In: 2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422), vol. 1, pp. 868–874. IEEE, Piscataway (2003)
8. Howard, A., Mataric, M.J., Sukhatme, G.S.: Localization for mobile robot teams: A maximum likelihood approach. Technical Report IRIS-01–407. Institute for Robotics and Intelligent Systems, University of Sourthern California (2001)
9. Kim, D., Velasco, Y., Wang, W., Uma, R.N., Hussain, R., Lee, S.: A new comprehensive RSU installation strategy for cost-efficient VANET deployment. IEEE Trans. Veh. Technol. 66(5), 4200–4211 (2016)
10. Liu, W., Shoji, Y.: Edge-assisted vehicle mobility prediction to support V2X communications. IEEE Trans. Veh. Technol. 68(10), 10227–10238 (2019)
11. Balico, L.N., Loureiro, A.A., Nakamura, E.F., Barreto, R.S., Pazzi, R.W., Oliveira, H.A.: Localization prediction in vehicular ad hoc networks. IEEE Commun. Surv. Tutorials 20(4), 2784–2803 (2018)
12. Nasir, N.Z.M., Zakaria, M.A., Razali, S., bin Abu, M.Y.: Autonomous mobile robot localization using Kalman filter. In: MATEC Web of conferences, vol. 90, pp. 01069. EDP Sciences, Les Ulis, France (2017)
13. Mo, Y., Yu, D., Song, J., Zheng, K., Guo, Y.: Vehicle position updating strategy based on Kalman filter prediction in VANET environment. Discrete Dyn. Nat. Soc. 2016, 1404396 (2016)
14. Emami, P., Elefteriadou, L., Ranka, S.: Tracking vehicles equipped with dedicated short-range communication at traffic intersections. In: Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, pp. 9–16. ACM, New York (2017)
15. Ghaleb, F.A., Zainal, A., Rassam, M.A., Abraham, A.: Improved vehicle positioning algorithm using enhanced innovation-based adaptive Kalman filter. Pervasive Mob. Comput. 40, 139–155 (2017)
16. Malki, H.H.A., Moustafa, A.I., Sinky, M.H.: An Improving position method using Extended Kalman filter. Procedia Comput. Sci. 182, 28–37 (2021)
17. Jaiswal, R.K., Jaidhar, C.D.: Location prediction algorithm for a nonlinear vehicular movement in VANET using extended Kalman filter. Wirel. Netw. 23, 2021–2036 (2017)
18. Fox, D., Burgard, W., Thrun, S.: Markov localization for mobile robots in dynamic environments. J. Artif. Intell. Res. 11, 391–427 (1999)
19. Elazab, M., Noureldin, A., Hassanein, H.S.: Integrated cooperative localization for connected vehicles in urban canyons. In: 2015 IEEE Global

Communications Conference (GLOBECOM), pp. 1–6. IEEE, Piscataway (2015)
20. Hoang, G.M., Denis, B., Härri, J., Slock, D.T.: Mitigating unbalanced GDoP effects in range-based vehicular Cooperative Localization. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 659–664. IEEE, Piscataway (2017)
21. Ansari, A.R., Saeed, N., Haq, M.I., Cho, S.: Accurate 3D localization method for public safety applications in vehicular Ad-hoc networks. IEEE Access 6, 20756–20763 (2018)
22. Feng, H., Liu, C., Shu, Y., Yang, O.W.: Location prediction of vehicles in VANETs using a Kalman filter. Wirel. Pers. Commun. 80, 543–559 (2015)
23. Eom, J., Kim, H., Lee, S.H., Kim, S.: DNN-assisted cooperative localization in vehicular networks. Energies 12(14), 2758 (2019)
24. Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Asari, V.K.: A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3), 292 (2019)
25. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M.P., Iyengar, S.S.: A survey on deep learning: Algorithms, techniques, and applications. ACM Comput. Surv. (CSUR) 51(5), 1–36 (2018)
26. Dargan, S., Kumar, M., Ayyagari, M.R., Kumar, G.: A survey of deep learning and its applications: a new paradigm to machine learning. Arch. Comput. Methods Eng. 27, 1071–1092 (2020)
27. Elsworth, S., Güttel, S.: Time series forecasting using LSTM networks: A symbolic approach. arXiv preprint arXiv:2003.05672 (2020)
28. Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A.: Deep learning for time series forecasting: a survey. Big Data 9(1), 3–21 (2021)
29. Amirabadi, M.A., Kahaei, M.H., Nezamalhosseini, S.A.: Novel suboptimal approaches for hyperparameter tuning of deep neural network [under the shelf of optical communication]. Phys. Commun. 41, 101057 (2020)
30. McLeod, C.: A framework for distributed deep learning layer design in python. arXiv preprint arXiv:1510.4303 (2015)
31. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Zheng, X.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283. USENIX Association, Berkeley, CA (2016)
32. Labbe, R.: FilterPy - Kalman and Bayesian Filters in Python (2020). https://filterpy.readthedocs.io/en/latest/

How to cite this article: Rezazadeh, N., Amirabadi, M.A., Kahaei, M.H.: Deep learning-based location prediction in VANET. IET Intell. Transp. Syst. 1–14 (2024). https://doi.org/10.1049/itr2.12529