Trajectory Prediction of Neighboring Vehicles via Periodic Beaconing with Inaccurate GPS Data

Jae-Han Lim
Kwangwoon Univ.
ljhar@kw.ac.kr

Katsuhiro Naito
Aichi Institute of Technology
naito@pluslab.org

Yeon-Sup Lim
Sungshin Women's Univ.
ylim@sungshin.ac.kr

***Abstract*—Predicting trajectories of neighboring vehicles accurately is essential for safe driving in Advanced Driving Assistance Systems (ADAS). Conventional approaches for predicting trajectory rely upon an expensive image sensor (i.e., lidar). However, high cost but limited sensing range of lidar makes installing a trajectory prediction system on existing vehicles impractical. Using two functionalities provided by a smartphone, i.e., periodic beaconing and GPS positioning could address issues of conventional approaches. However, modules for wireless communications and GPS are limited in achieving accurate prediction in their current form: transmission error and inaccurate GPS data. In this paper, we propose a novel scheme for predicting neighbors' trajectories using periodic beaconing and GPS, and show that our scheme is feasible to trajectory prediction. To address the limitations of the wireless and GPS modules, our scheme integrates Long Short-Term Memory (LSTM) with two strategies: 1) Tailoring Multi-band Transmission to Prediction (TMTP) and 2) Environment-Aware Positioning (EAP). To improve prediction accuracy even with the imperfect input, our scheme employs interactive LSTM model. To our knowledge, this is the first to enable predicting the neighbors' trajectories using periodic beaconing with inaccurate GPS data. Experimental results using real traces demonstrate that our scheme is accurate in predicting neighbors' trajectories.**

***Index Terms*—trajectory prediction, periodic beaconing, inaccurate GPS data, interactive LSTM, surrounding vehicle**

I. Introduction

Conventional approaches for predicting neighbors' trajectories rely on Deep Neural Network (DNN) that uses data from an expensive image sensor (i.e., lidar) [1], [2]. Specifically, each vehicle periodically obtains kinematic data of its neighbors (e.g., position, velocity, and acceleration) using its own lidar, and then feeds them to a DNN to predict future trajectories of the neighbors. However, the use of lidar has several limitations: 1) high cost[1] and 2) limited sensing range due to obstacles or adverse weather. These limitations motivate finding alternatives to achieve trajectory prediction that facilitates implementation with low cost.

Using resources available in a smartphone would address the limitations of conventional approaches because most people possess a smartphone [4], which has a Global Positioning System (GPS) module for positioning and a wireless module for Inter-Vehicle Communication (IVC) such as LTE and Wi-Fi interfaces. To predict neighbors' trajectories, two primitives of a smartphone can be used: 1) positioning via a GPS module and 2) periodic beaconing via IVC. Specifically, a vehicle obtains its own kinematic data using a GPS module, and then periodically exchanges its beacon including the data through IVC. A receiving vehicle obtains the data of the sending vehicle. Using a prediction module with a series of kinematic data, the receiver predicts the trajectories of the sender.

It is non-trivial to predict trajectories of neighbors using the two primitives because of the limitations of wireless and GPS modules. First, a vehicle sometimes fails to receive beacons of neighbors via IVC. Specifically, a beacon broadcasted through a Wi-Fi interface suffers from network congestion, leading to frequent packet collisions; a packet transmission using an LTE interface is robust to the packet collision, but using an LTE band is not free. Second, kinematic data from a GPS receiver might be inaccurate, particularly in an urban canyon. The incorporation of inaccurate kinematic data into a prediction algorithm like Long Short-Term Model (LSTM) would likely reduce prediction accuracy.

In this paper, we propose a novel scheme that enables accurate trajectory prediction using periodic beaconing with inaccurate GPS data, and investigate its feasibility to trajectory prediction. To enable accurate prediction even with the limitations of wireless and GPS modules, our scheme integrates an LSTM with two strategies. First, we employ Tailoring Multi-band Transmission to Prediction (TMTP) wherein a vehicle exploits a Wi-Fi band for broadcasting a beacon and an LTE band for sharing a neural network and offloading a fraction of beacon traffic. Second, our scheme employs Environment-Aware Positioning (EAP) whereby a vehicle dynamically selects a sensing module depending on its assessment of the accuracy of the module. To further improve the prediction accuracy even with the imperfect input, our scheme employs an interactive LSTM model. The interactive model enables accurate prediction because vehicles in close proximity move in a flow, implying that the movement of one vehicle is associated with that of its neighbor.

We evaluate the proposed scheme using a testbed consisting of TensorFlow [5] and QUALNET [6] with real vehicle traces from NGSIM [7]. The experimental results demonstrate that the proposed scheme is accurate in predicting neighbors' trajectories and is comparable to the previous schemes that adopt lidar and do not employ IVC. To our knowledge, this is the first to enable predicting neighbors' trajectories using periodic beaconing with inaccurate GPS data

The contributions of this paper are multi-fold:

- Propose a novel scheme for predicting neighbors' trajectories via periodic beaconing with inaccurate positioning data.
- Propose an TMTP and an EAP for improving prediction accuracy even with limitations of wireless and GPS modules.
- Propose a simple interactive LSTM improving prediction accuracy even with imperfect inputs.
- Explore the feasibility of predicting neighbors' trajectories using periodic beaconing with inaccurate positioning data.

II. Overview of Proposed Scheme

To predict neighbors' trajectories accurately, a vehicle exploits wireless interfaces and a GPS module of a smartphone. Specifically, each vehicle obtains its own kinematic data using a GPS module and shares the data with its neighbors via periodic beaconing. Here, the kinematic data refers to data representing the movement of a vehicle and two dimensional

[1]The lidar used in google's self-driving car costs $ 75,000 [3].

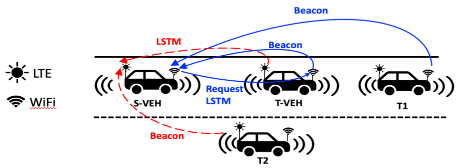


Fig. 1. The proposed strategy in using Wi-Fi and LTE interface (Wi-Fi: solid line, LTE: dotted line)

position, velocity, and acceleration are used in our scheme. When receiving beacons from neighbors, the vehicle trains its LSTM model and predicts their trajectories using the model. However, the innate limitations in wireless communications (i.e., packet delivery error) and a GPS module (i.e., sensing error) hinder the accurate prediction with only a smartphone. These limitations make inputs to an LSTM network imperfect, causing inaccurate prediction of neighbors' trajectories.

For predicting neighbors' trajectories accurately even with these limitations, we propose a novel scheme integrating an interactive LSTM with two strategies. Specifically, a vehicle exploits either a Wi-Fi or LTE interface to improve transmission reliability and resulting prediction accuracy. Moreover, a vehicle selects a sensing module to obtain its kinematic data and opportunistically delivers the data to its neighbors based on the accuracy of the module. To improve the prediction accuracy even with the stated limitations, we adopt an interactive LSTM network. We will elaborate on each strategy in the following paragraphs.

We adopt TMTP for efficient usage of communication module. A vehicle exploits a Wi-Fi interface to periodically broadcast its beacon; the vehicle uses an LTE interface to share its LSTM network with neighbors when receiving a request from them as depicted in Figure.1. To reduce network congestion in a Wi-Fi band, a vehicle offloads a fraction of the beacon traffic from the Wi-Fi band into the LTE band.

Our scheme employs EAP whereby a vehicle obtains its kinematic data from either a GPS module or Dead Reckoning (DR) equation with an Inertia Measurement Unit (IMU). By default, a vehicle gets its kinematic data from a GPS module and broadcasts a beacon including the data every BI; neighbors estimate the trajectory of the vehicle from the received beacon. However, when noticing that the module is likely to be inaccurate, the vehicle gets kinematic data using DR equation and IMU, and broadcasts the DR equation; neighbors derive the trajectory of a sender using the equation.

An interactive LSTM model is adopted to compensate for imperfect input because in reality the movement of one vehicle is significantly affected by those of its neighbors. For example, as shown in Figure 1, S-VEH anticipates that T-VEH will not change its lane due to surrounding vehicles (T1 and T2) even if the S-VEH fails to receive kinematic data from T-VEH.

III. Details of Proposed Scheme

A. Overall Procedure

Figure 2 illustrates the overall procedure of the proposed scheme. When a vehicle starts to move, the vehicle begins TMTP for using Wi-Fi and LTE interfaces intelligently and sets its own sensing mode to a GPS mode and obtains kinematic data from a GPS module. When noticing that data from the module is becoming inaccurate, a vehicle (T-VEH in Figure 2) switches its mode to a DR mode in which the vehicle obtains the data using a DR equation and IMU. So that the neighbors (S-VEH and T1 in Figure 2) can notice the mode

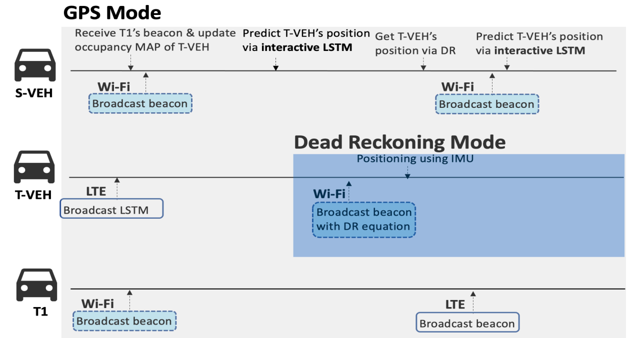


Fig. 2. Overall procedure of the proposed scheme

of T-VEH, T-VEH piggybacks its own mode onto its beacon. To reflect interactions among neighbors, T-VEH divides its surrounding area into multiple grid cells and generates a neighbor vector describing the occupancy of each grid cell (see Figure 5 for neighbor vector). Then, T-VEH piggybacks the vector onto its beacon. Whenever receiving a beacon from a T-VEH, S-VEH updates the vector about T-VEH. Using this vector and a series of kinematic data of T-VEH, S-VEH predicts the best possible trajectory of T-VEH based on an interactive LSTM. We will elaborate on TMTP, EAP, and interactive LSTM in the following three subsections.

B. Tailoring Multi-band Transmission to Prediction (TMTP)

In TMTP, two processes work together in parallel: 1) a beacon process and 2) an LSTM process. In a beacon process, a vehicle exploits a Wi-Fi interface for periodic beaconing in most cases, but sometimes offloads the beacon traffic to an LTE band. In an LSTM process, a vehicle shares an LSTM network using an LTE interface

Figure 3 illustrates a flowchart describing the operations of the two processes in each vehicle. In a beacon process, a vehicle checks whether to broadcast its beacon. If in possession of a beacon, the vehicle checks whether to offload its broadcast to an LTE band. To achieve this, the vehicle chooses its RF interface whenever broadcasting its beacon. Specifically, the vehicle broadcasts its beacon through a Wi-Fi interface by default but broadcasts the beacon through an LTE interface with a probability α (i.e., offloading factor)[2] . In an LSTM process, a vehicle checks whether it has a well-trained network[3] associated with its current driving road[4]. Unless having the network, the vehicle piggybacks a request message onto its beacon via its Wi-Fi interface. When receiving the request, other vehicles having the well-trained network send it via an LTE interface[5].

C. Environment-Aware Positioning (EAP)

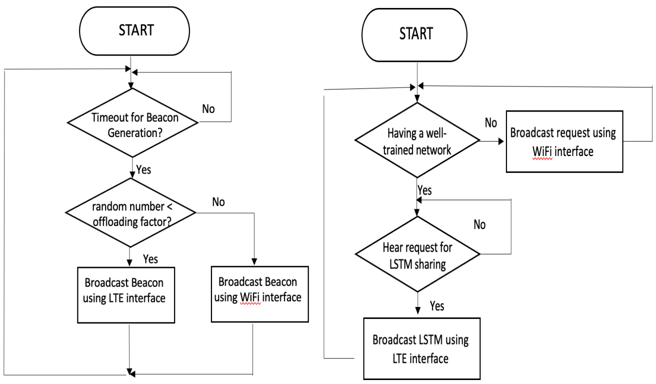
Recall that each vehicle decides its sensing mode between GPS and DR modes based on the measured accuracy of a GPS module. In a GPS mode, a vehicle periodically samples its position from the GPS receiver of a smartphone, and

[2]We decide α based on the level of congestion in Wi-Fi bands. The mechanism for selecting α and an optimization algorithm will be our future work.

[3]A vehicle recognizes whether it has well-trained network based on the number of epochs in training the model.

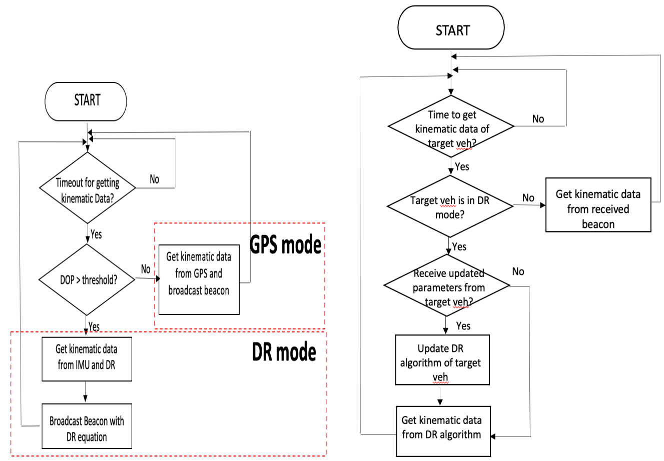
[4]Our scheme uses multiple classes of LSTM networks and each network is related to a type of road (e.g., urban intersection, highway, local freeway)

[5]To reduce the usage of LTE band, a vehicle could download all the LSTM networks associated with its driving path when the vehicle starts.



(a) Beacon process (b) LSTM process

Fig. 3. Flowchart for describing the operation of TMTP



(a) Vehicle measuring kinematic data (b) Neighbors of target vehicle

Fig. 4. Flowcharts for describing (a) operation of EAP at a vehicle measuring a kinematic data and (b) operation of EAP at neighbors of the vehicle

then derives its kinematic data using the sampled positions. Then, the vehicle periodically broadcasts a beacon including the kinematic data and its neighbors obtain the data when receiving the beacon. In a DR mode, each vehicle obtains its kinematic data using a DR equation and data measured from IMU. Our system employs a simple equation [8] as follows[6].

$$p(\vec{t}) = p(\vec{t_0}) + \int_{t_0}^{t} v(\vec{\tau})d\tau \tag{1}$$

where t_0 is the initial time of the DR equation; $p(\vec{t})$ and $v(\vec{t})$ are two-dimensional position and velocity at t, respectively[7]. Each vehicle generates its DR equation and shares the equation with its neighbors. For this, the vehicle piggybacks the equation onto its beacon. When receiving the beacon, the neighbors estimates the kinematic data of the vehicle using the equation in the beacon. Notably, we insert a one-bit flag indicating a sensing mode into a beacon so that receivers recognize the sender's mode.

To assess the accuracy of a GPS module, a vehicle must derive measurement errors from the module. However, it is difficult to measure the errors because a vehicle cannot get the ground-truth of its position. To circumvent this issue, we exploit a well-known method to estimate the accuracy using Dilution of Precision (DOP), provided by a GPS satellite [9]. DOP corresponds to the region where the actual position would likely be located centered around a position measured by a GPS module. If DOP is large, the GPS error is likely to be large and thus we cannot trust the measured data; otherwise, the error would likely be small. Thus, each vehicle switches its mode into a DR mode if the DOP is larger than the threshold[8]; otherwise, the vehicle remains at GPS mode.

Figure 4(a) illustrates a flowchart describing the operation of a vehicle measuring its own kinematic data in EAP. A vehicle periodically compares DOP obtained from its GPS module with a pre-defined threshold. If DOP is less than the threshold, the vehicle obtains its kinematic data from a GPS module and broadcasts a beacon including the data. Otherwise, the vehicle switches its mode to a DR mode. In a DR mode, a vehicle periodically obtains kinematic data using a DR equation and IMU, and then broadcasts its own beacon. Figure 4(b) illustrates a flowchart describing how neighbors of the vehicle recognize the kinematic data of the target vehicle. The neighbors recognize the sensing mode of the target vehicle when receiving a beacon from it. If the sender of the beacon is in a GPS mode, the neighbors obtain kinematic data from the beacon. If the sender is in a DR mode, the neighbors derive the kinematic data of the sender using its DR equation. When receiving the DR equation updated by the sender, the neighbors use the updated equation to obtain the kinematic data.

We should note that a positioning error is large throughout a DR mode if the error measured from GPS is large at the start of a DR mode. This is because the error is accumulated with time in a DR mode. To address this issue, a vehicle establishes several DR equations using GPS data measured at different times before a DR mode starts; the vehicle estimates the positioning error at the start of a DR mode ($err(t_s)$) corresponding to each equation and selects the equation minimizing $err(t_s)$, which is derived by

$$err(t_s) = err_{gps}(t_0) + \delta(t_s - t_0) \tag{2}$$

where $err_{gps}(t_0)$ is GPS measurement error at t_0 and δ is an error in vehicle speedometer.

D. Interactive LSTM

The interactive LSTM model employs the stacked LSTM architecture, which is a deeper model with multiple hidden LSTM and neural network layers. The first and third hidden layers are LSTM layers, each with 256 cells; the last layer generates outputs using the sigmoid function as an activation function. Between these layers, we insert dropout layers to apply the dropout regularization for preventing overfitting [11].

To reduce prediction inaccuracy caused by imperfect input, our interactive LSTM model associates the movements of one vehicle with those of its neighbors by utilizing the trajectory of a target vehicle as well as occupancy information of surrounding vehicles.

Table I lists the features used for the interactive LSTM model. Our model uses the information about the interactions with surrounding vehicles and their kinematic data: i.e., each

[6] The DR equation can be easily replaced with other equations for DR.

[7] Here, initial position ($p(\vec{t_0})$) is derived from GPS measurement and velocity($v(\vec{t})$) is obtained from a speedometer in equation 1.

[8] We should determine this threshold depending on the requirement for positioning accuracy of an application system. For ITS applications, the typical value for the threshold is 1.5, which is comparable to 1.5m error [10].

TABLE I
FEATURES USED IN THE INTERACTIVE LSTM MODEL

Feature	Description
PosX, PosY	Position in X and Y axis
Spd	Speed of vehicle
VelX, VelY	Speed of vehicle in X and Y direction
Acc	Acceleration of vehicle
AccX, AccY	Acceleration of vehicle in X and Y direction
Neighbor vector (Bit_i)	8-bit neighbor vector

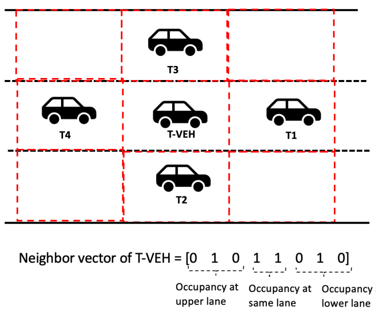


Fig. 5. Neighbor vector of T-VEH describing occupancy around T-VEH

vehicle trains the LSTM model to predict future trajectories of the others given the series of their previous position, velocity, acceleration, and occupancy information. All numeric features such as position and velocity are normalized. The occupancy information is represented as a neighbor vector, which indicates whether any vehicle occupies any of the eight grid cells surrounding the target vehicle (see Figure 5 for illustration). If any vehicle occupies a grid cell, the corresponding element of the vector becomes one; otherwise, zero. By assigning one or zero to each grid cell, each vehicle constructs an 8-bit vector for the neighbor vector, and each bit in the vector is used as a feature for the model. Therefore, our model uses a total of 16 features for training and prediction.

Using the neighbor vector offers several advantages. First, incorporating the neighbor vector into the model is easier to implement in a smartphone than relying on other more complex mechanisms [12]–[14] wherein objects exchange hidden states of their LSTM models with each other. Second, even if this approach is simple and computationally efficient, the accuracy of predicting neighbors' trajectories is comparable to those of the previous interactive models [12]–[14]. Our proposal simply relies on the status of the occupancy of each grid cell being the predominant factor impacting the potential movements of the target vehicle.

The size of a grid cell decides how far vehicles can provide interaction information, affecting the performance of the model. Thus, a vehicle must appropriately decide the size of the cell depending on its movement. To this end, the size increases linearly with the speed of T-VEH in our scheme.This is because the movement of a vehicle far from T-VEH would affect that of T-VEH if those two vehicles are moving fast. In contrast, the movement of the T-VEH would be affected by only nearby vehicles if those vehicles are moving slowly.

IV. EXPERIMENTS

A. Experimental Settings

To assess the proposed scheme, we build a testbed consisting of two components: 1) QUALNET for modeling TMTP and EAP [6] and 2) TensorFlow for implementing an interactive LSTM [5]. Using this testbed, we evaluate the proposed scheme as follows. First, we preprocess a publicly available vehicle trajectory dataset provided by NGSIM [7] [15] to make it suitable to QUALNET. Second, we feed the preprocessed data into QUALNET and analyze the received beacons in QUALNET operation according to the vehicle trajectory. Based on the analysis, we build the traces of features for training and testing the interactive LSTM: we use 85% of the data for training and the rest for testing. Third, we examine the prediction accuracy of the trained model on the test dataset.

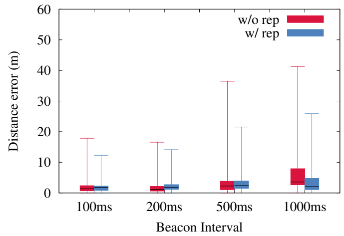
TABLE II
DEFAULT PARAMETER SETTINGS

Default parameters	Values
Generation interval of beacon	100ms
TX rate of Wi-Fi and LTE	6, 12 Mbps
TX power of Wi-Fi and LTE	15, 23 dBm
RX sensitivity of Wi-Fi and LTE	-85, -85 dBm
Channel model	Nakagami
Interval for inputs to LSTM	5 sec
Predicting time of LSTM	1 sec

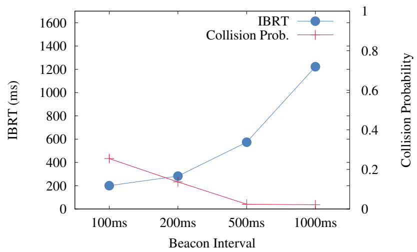
We use the trajectories collected in Lankershim Boulevard and US101 of Los Angeles by NGSIM [7].We regard the original trajectories from NGSIM as Ground-Truth (GT) and generate trajectories measured at vehicles by adding positioning errors to the GT trajectories. We model the error using a normal distribution in a GPS mode [16]. In a DR mode, we use equation 2 with $\delta = 0.08m/s$ [17] by replacing DR starting time t_s with current time t.

We summarize default parameter settings used in our experiment in Table 3. In the MAC layer, each radio interface adopts different methods for multiple access control. More specifically, a Wi-Fi interface adopts Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [18], whereas an LTE interface employs a scheduled channel access control and operates in 10 MHz bandwidth [19]. In the PHY layer, we use the lowest transmission rate for Wi-Fi and LTE interfaces. As a propagation model, we adopt a Nakagami model, which is well-known to be suitable to a channel model for VANET [8], [20]. The threshold of DOP for switching a mode is 1.5, which is suggested for using ITS [10] We set the number of grid cells for neighbor vector to be 8. A vehicle decides the size of each cell according to an equation in [21]. An interactive LSTM takes 5 second trajectory as an input and predicts the trajectory of neighbors in one second ahead. We consider two options in training an LSTM model: a) integration with repetitive training ('w/ rep') and b) integration without repetitive training ('w/o rep'). In the repetitive training, we first train the model using a dataset and then train the model again using data causing worst 25% errors in the first training results.

We get the following performance metrics in this experiment: 1) average distance error, 2) 25% and 75% quartile of distance error, and 3) maximum and minimum of distance error. Here, the distance error refers to the root mean square error (RMSE) between ground-truth of position and predicted position. To understand the experimental results clearly, we derive network-related metrics: 1) IBRT and 2) collision probability in a Wi-Fi band. Here, the collision probability in a Wi-Fi band points out the level of congestion in a Wi-Fi band. All these parameters are derived according to four independent variables: 1) BI, 2) offloading factor, 3) adoption of EAP, and 4) the number of Wi-Fi devices. The typical settings on the variables are 100ms, 0.05, and 'EAP is not adopted', and 'no devices'. To isolate the effect of each variable, we change the variable while fixing others to the typical settings.

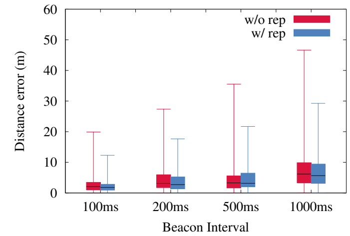


(a) Distance error

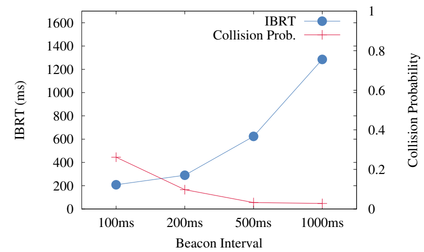


(b) IBRT and collision probability

Fig. 6. Distance error, IBRT, and collision probability according to BI in Lankershim Blvd.: (a) distance error and (b) IBRT and collision probability



(a) Distance error



(b) IBRT and collision probability

Fig. 7. Distance error, IBRT, and collision probability according to BI in US101: (a) distance error and (b) IBRT and collision probability

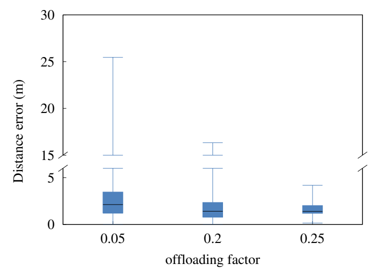
B. In-depth Study of the Proposed Scheme

1) Impact of BI: Figure 6(a) and Figure 7(a) presents Whisker plots showing the minimum, first quartile (Q1), median, third quartile (Q3), and maximum of the distance errors of the proposed scheme according to BI in Lankershim Boulevard and US101, respectively. The figures show that the distance error becomes larger as BI increases. It is because IBRT rises with BI even with fewer beacon collisions as shown in the Figures 6(b) and 7(b); that is, a vehicle less often obtains kinematic data of its neighbors.

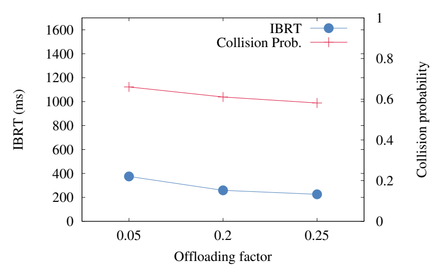
Note that BI of 100ms yields the average prediction error of less than 1.5m, which satisfies the typical requirement for various safety services [17]. It shows that a careful configuration of the parameters such as BI enables a neighbors' trajectory predictor suitable for ADAS even using a smartphone. Also, the distance error of the 75% quartile is relatively smaller than the maximum (see 'w/o rep' option in Figures 6(a) and 7(a)). It implies that our proposed scheme predicts the neighbor's trajectory very accurately in most cases.

From the analysis of the trajectories of ten worst prediction errors in these experiments, we found that those cases fall into three categories: 1) turning left/right at intersections, 2) changing lanes, and 3) stopping suddenly. The fraction of trajectories corresponding to these three cases is very small (less than 1% in Lankershim trace), implying that the LSTM model is not trained well to these trajectories. To reduce the maximum error, we train an LSTM model and then re-train the model using data in which instances with the worst 25% errors, which likely fall into the above three cases, are repeated once. This training method decreases the maximum error by up to 39% in both traces (see 'w/ rep' in figures 6(a) and 7(a)).

2) Impact of offloading factor: A vehicle mitigates network congestion in a Wi-Fi band by offloading a fraction of its beacon traffic to an LTE band. It in turn improves prediction accuracy by decreasing beacon losses. To study the impact of offloading factor, we explore a scenario in which background Wi-Fi traffic coexists with beacon traffic in a Wi-Fi band (e.g., an urban hot spot). In this scenario, 80 users are distributed over the shoulder of the road, generating 200kbps CBR traffic. We examine only Lankershim dataset because Wi-Fi background traffic is rare in highway areas such as US101. Also, we investigate only 'w/o rep' option to focus on the impact of offloading factor.



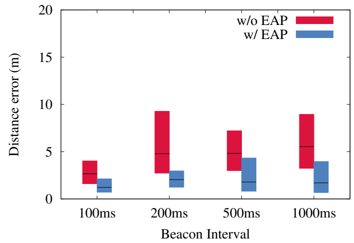
(a) Distance error



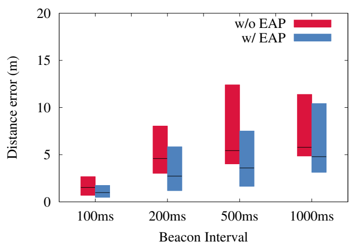
(b) IBRT and collision probability

Fig. 8. Distance error, IBRT, and collision probability according to offloading factor in Lankershim Blvd: (a) distance error and (b) IBRT and collision probability

Figure 8(a) demonstrates that the distance error decreases with an offloading factor. This is because offloading beacon traffic to an LTE band reduces network congestion in Wi-Fi traffic, which leads to the decrease in IBRT. We confirm this by observing that IBRT and the collision probability decreases with the offloading factor in Figure 8(b)

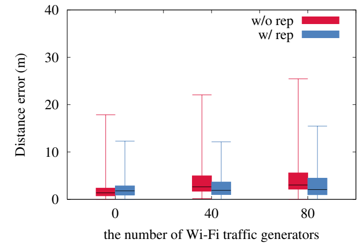


(a) Lankershim

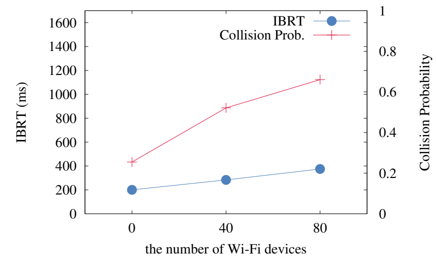


(b) US101

Fig. 9. Comparison between distance error with EAP and that without EAP as a function of BI in Lankershim Blvd. Here, the box shows 25% and 75% quartiles of the error; black line in the box represents an average error.: (a) Lankershim and (b) US101



(a) Distance error



(b) IBRT and collision probability

Fig. 10. Distance error, IBRT, and collision probability according to the number of background Wi-Fi users in Lankershim Blvd..: (a) distance error and (b) IBRT and collision probability

3) Impact of EAP: When adopting EAP, a vehicle dynamically changes its sensing mode based on the accuracy of a GPS module. We investigate the impact of EAP in a scenario where a GPS error is relatively large (e.g., urban canyon, invisible satellite due to a large mountain in a highway area). In this scenario, the change of mode is likely to occur. We construct this scenario by adding GPS errors with a large standard deviation to the NGSIM dataset. Similar to Figure 8, we examine only 'w/o rep' option to focus on the effect of adopting EAP under various BIs. As depicted in Figure 9(a) and 9(b), adopting EAP reduces the distance error by up to 70% and 40% in Lankershim Blvd and US101, respectively. It demonstrates that our scheme improves prediction accuracy even when GPS error is large by adopting EAP.

4) Impact of the number of Wi-Fi users: : In an urban area, background Wi-Fi traffic often hinders the reliable delivery of beacons, which would reduce the accuracy of predicting a neighbor's trajectory. We investigate how the background traffic affects the accuracy by measuring the distance errors of our scheme according to the number of Wi-Fi users in Lankershim Blvd. To this end, each Wi-Fi user generates CBR traffic with 200kbps at the shoulder of the road in our testbed. In Figure 10(a), we observe that the error rises with the number of Wi-Fi users. This is because the background traffic induces congestion in a Wi-Fi band, hindering beacon delivery as shown in Figure 10(b).

C. Comparison

In this subsection, we compare our scheme with two benchmarks [13], [22] as well as two variants of our scheme: 1) vanilla LSTM with accurate sensing data and 2) interactive LSTM with accurate sensing data. In the first variant, vanilla LSTM does not use neighbor vector as its feature to preclude interactions among vehicles. The second variant adopts an interactive LSTM. To mimic the situation in which vehicles obtain accurate data from expensive sensors such as lidar in both benchmarks, we use the ground-truth values of kinematic data for training and prediction.

Evaluations using our testbed demonstrates that our scheme is more accurate than 'vanilla LSTM with GT trajectory' even if it exploits the accurate sensing data while our scheme relies on data containing noise. For example, the average distance errors of our scheme are 1.52m and 1.36m in Lankershim and US101 traces while the errors of the variant are 2.22m and 1.86m. It shows that our scheme improves the prediction accuracy even with imperfect inputs caused by beacon delivery failures and GPS errors. Also, the result presents that the distance error of our scheme is comparable that of 'interactive LSTM with GT trajectory'. For example, the average distance error of 'interactive LSTM with GT trajectory' are 1.12m and 1.26m in Lankershim and US101 traces. The average distance errors of [13] and [22] measured using lidar (i.e., GT trajectories are used) are around 0.9m and 0.8m, which are smaller than the errors of our scheme. However, we emphasize that our scheme is based on devices accessible or existing only in a smartphone. This means that the implementation cost is much lower than those of two benchmarks that require expensive sensor such as lidar.

V. Related Works

There are a number of previous proposals for predicting the maneuvers of a vehicle [23]–[27]. In [23], the intention for changing lanes was estimated based on supervised learning and periodic beaconing. The authors in [24] exploited a recurrent neural network (RNN) to predict a driver's intention at an unsignalized intersection. In [25], the authors proposed an interactive multiple model (IMM) to predict an intention for lane change based on GPS data. The authors in [26] used a Gaussian mixture and a Random Forest models for classifying the braking intensity of an electrified vehicle into multiple levels. In [27], the authors presented a scheme using vehicular communications and radar to estimate the merging potential of a side vehicle. However, the previous proposals have focused on only mechanisms to predict maneuvers, which is much simpler than predicting the neighbors' trajectories.

Several researchers have proposed mechanisms to predict trajectories of vehicles via machine learning [1], [22], [28]. In [22], the authors proposed predicting a trajectory of a vehicle using LSTM. In [1], LSTM consisting of an instant layer and a category layer was used for predicting trajectories of multiple agents in urban areas. In [28], the authors proposed a safety-aware deep learning model to predict trajectories of neighboring vehicles before collisions. However, the schemes in [1], [22], [28] were based on the unrealistic assumption that the sensing data was accurate; they did not consider interactive behavior among vehicles when predicting their trajectories.

To address the limitations of [1], [22], [28], researchers have proposed interactive prediction models whereby interactive behaviors among agents were considered for predicting their trajectory [12]–[14], [29], [30]. In [29], the authors proposed a scheme for predicting trajectories of nearby agents using a particle filter. The authors in [30] considered interactions among vehicles in close proximity using a graph convolutional model. In [12], Graph Neural Network (GNN) was used for perceiving an interactive event among vehicles and LSTM was exploited for predicting their trajectories. The authors in [13] proposed a two-layer LSTM model for predicting trajectories of vehicles with interactive behavior. In [14], a social LSTM was presented to predict human movement by considering interactions among people. However, the schemes in [12]–[14], [29], [30] are based on the unrealistic assumption that the sensing data is always accurate and no loss is incurred in obtaining sensing data. Eliminating the unrealistic assumptions distinguishes our work from previous work.

VI. Conclusion

In this paper, we proposed a scheme for predicting neighbors' trajectories accurately using periodic beaconing with inaccurate GPS data, which is often used by a smartphone. For enabling accurate prediction even with communication and positioning errors, our scheme integrated an interactive LSTM with two strategies: 1) TMTP for intelligent use of LTE and Wi-Fi bands and 2) EAP for improving its own positioning. Experimental results demonstrated that the proposed scheme predicted neighbors' trajectories accurately.

References

[1] Yuexin Ma et al. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[2] Sajjad Mozaffari et al. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):33–47, 2022.

[3] Kirsten Korosec. https://techcrunch.com/2019/03/06/waymo-to-start-selling-standalone-lidar-sensors, 2019.

[4] https://www.pewresearch.org/internet/fact-sheet/mobile/, 2021.

[5] https://www.tensorflow.org, 2022.

[6] https://www.scalable-networks.com/products/qualnet-network-simulation-software/, 2022.

[7] J. Colyar and J. Halkias. Lankershim boulevard dataset. Technical report, Federal Highway Administration (FHWA), 2007. TR FHWA-HRT-07-029.

[8] Jae-Han Lim et al. To predict or to relay: Tracking neighbors via beaconing in heterogeneous vehicle conditions. *IEEE Transactions on Mobile Computing*, 21(3):1142–1154, 2022.

[9] Richard B. Langley. Dilution of precision. 1999.

[10] James Chang and Edward Fok. Connected vehicle pilot positioning and timing report: Summary of positioning and timing approaches in cv pilot sites. Technical report, U.S. Department of Transportation, January 2018. FHWA-JPO-18-638.

[11] Ian Goodfellow et al. Deep learning. Technical report, MIT Press, 2006.

[12] Zirui Li et al. A hierarchical framework for interactive behaviour prediction of heterogeneous traffic participants based on graph neural network. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2021.

[13] Lian Hou et al. Interactive trajectory prediction of surrounding road users for autonomous driving using structural-lstm network. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4615–4625, 2020.

[14] Alexandre Alahi et al. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[15] J. Colyar and J. Halkias. Us highway 101 dataset. Technical report, Federal Highway Administration (FHWA), 2007. TR FHWA-HRT-07-030.

[16] Cezary Specht et al. Accuracy of the gps positioning system in the context of increasing the number of satellites in the constellation. Technical report, Polish Maritime Research, 2015.

[17] Novotel. Safety critical positioning – for autonomous applications: Lessons from civil aviation. Technical report, 2016.

[18] IEEE Std. 802.11p. Ieee standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements; part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications; amendment 6: Wireless access in vehicular environments. Technical report, IEEE, July 2010.

[19] Behrad Toghi et al. Multiple access in cellular v2x: Performance analysis in highly congested vehicular networks. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, 2018.

[20] Jae-Han Lim et al. Reliable safety message dissemination in nlos intersections using tv white spectrum. *IEEE Transactions on Mobile Computing*, 17(1):169–182, 2018.

[21] Dayi Qu et al. Modeling of car-following required safe distance based on molecular dynamics. *IEEE Transactions on Intelligent Transportation Systems*, 2014(4):1–7, 2014.

[22] Florent Altché et al. An lstm network for highway trajectory prediction. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 353–359, 2017.

[23] Ahmed Hamdi Sakr et al. Lane change detection using v2v safety messages. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3967–3973, 2018.

[24] Alex Zyner et al. A recurrent neural network solution for predicting driver intention at unsignalized intersections. *IEEE Robotics and Automation Letters*, 3(3):1759–1764, 2018.

[25] Rafael Toledo-Moreo et al. Imm-based lane-change prediction in highways with low-cost gps/ins. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):180–185, 2009.

[26] Chen Lv et al. Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle. *IEEE Transactions on Vehicular Technology*, 67(7):5718–5729, 2018.

[27] Zhikai Ma et al. Safety cruise control of connected vehicles using radar and vehicle-to-vehicle communication. *IEEE Systems Journal*, 14(3):4602–4613, 2020.

[28] Xuanchi Ren et al. Safety-aware motion prediction with unseen vehicles for autonomous driving. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15711–15720, 2021.

[29] Jiachen Li et al. Generic tracking and probabilistic prediction framework and its application in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3634–3649, September 2020.

[30] Xin Li et al. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3960–3966, 2019.

Acknowledgment

This work was partially sponsored by the National Research Foundation of Korea (NRF) (grant no. RS-2023-00241628), and by National Research Foundation of Korea (NRF) (grant no. NRF2022R1G1A1006174).