2021 22nd IEEE International Conference on Mobile Data Management (MDM) | 978-1-6654-2845-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/MDM52706.2021.00025

Towards Predicting Vehicular Data Consumption

Andi Zang*, Xiaofeng Zhu†, Yuxiang Guo*, Fan Zhou‡ Goce Trajcevski§
*Department of Computer Science
Northwestern University, Evanston, IL/USA
Email: {andi.zang,yuxiangguo2021}@u.northwestern.edu
†Microsoft, Redmond, WA/USA
Email: {xiaofzhu}@microsoft.com
‡ School of Information and SW Engineering
University of Electronic Science and Technology, Chengdu, PR China
Email:{fan.zhou}@uestc.edu.cn
§ Department of Electrical and Computer Engineering
Iowa State University, Ames, IA/USA
Email: {gocet25}@iastate.edu

***Abstract*—Combining in-car multiple sensors measuring parameters that can be used to improve both safety and efficiency with a plethora of external data sources (e.g., traffic conditions, weather) which, if properly used, can significantly improve the overall trip experience. One source that can help the navigation and provide "context awareness", especially for autonomous driving, are the High Definition (HD) maps, which have recently witnessed a tremendous growth of popularity in vehicular technology and use. As they are limited to a particular geographic area with respect to a given point along a trip, different portions need to be downloaded (and processed) on multiple occasions throughout a given trip, along with the other data from internal and external sources. We take a first step towards formalizing the problem of Predicting Map Data Consumption (PMDC) in the future time instants for a given trip, based on a (time) window from its history, and investigate the use of Long Short-Term Memory (LSTM) networks – a special type of Recurrent Neural Networks (RNN). Significant efforts were focused on generating an appropriate dataset for this study, towards which we fused the information available in multiple heterogeneous data sources. We conducted experimental observations demonstrating the benefits of the proposed approach.**

***Index Terms*—High Definition Maps, Data Consumption**

I. Introduction and Motivation

Multiple on-board vehicular sensors, along with external data sources (e.g., traffic, weather), are generating various data types which, in turn, are enabling multiple functionalities (efficiency, re-routing), improving the overall trip experience. One particular source important for the overall navigation in autonomous and assisted driving settings are the High Definition (HD) maps [14]. While crucial for improving self-localization and safety [19], [29], [30] – they are also the largest consumers of bandwidth and processing power. State-of-the-art applications using HD maps are still built on the old chassis – conventional "Standard Definition" (SD) maps, which consist of road networks, topology and limited road features/objects attached to links and nodes. However, HD maps are much richer in terms of the sets (and types) of objects that they provide.

3D data with higher resolution and accuracy can be acquired from either multi-image reconstruction or point cloud along with the improvement of acquisition hardware and algorithms. Multiple objects ("furniture") are used to help the autonomous vehicle to make decisions, such as lane boundaries, curbs/guardrails, and pole-like objects. Combining these two aspects (acquisition-end and application-end) amplifies the impact of (the increase of) the size of the map data, relative to when the concept of HD maps were first introduced a decade ago. Nowadays, the HD maps can easily contain over a thousand voxels (highway scenario) or even tens of thousands of voxels (urban scenario) per road meter at a higher resolution, in contrast to dozens of points per road link in conventional maps [30]. This, in turn, affects the processing that enables decisions made by the drivers and autonomous vehicles, as the complexity of many important algorithms (e.g., points registration [17] and segmentation [16]), brings them on the cusp of being computationally over-expensive for certain application scenarios, given hardware limitations. To tackle the problem of optimizing the use of ever increasing (demands for) map data in real-time applications, existing approaches are mainly focusing on improving data structures and data flow.

Researchers and companies focusing on improving the HD map structures have proposed their own models to shrink the size of the data. In general, hierarchical structures are downloading different resolutions on demand by various use cases, which could potentially decrease the data throughput from the server to the vehicle. Due to the characteristics of HD maps and the applications using them, downloading (or pre-fetching) the data for entire trip at once is practically infeasible under current hardware constrains. This, in turn, implies that data needs to be downloaded and processed multiple times during a particular trip – which requires efficient management, based on various factors related to a particular vehicle and trip. Similar problems are encountered in hybrid electric vehicle energy management [5]. If the energy consumption of each trajectory point can be predicted entirely before the trip, the vehicle itself can have a better energy management plan to improve fuel economy (the combination between conventional energy and renewable energy) and to reduce emissions. However, given different fluctuations in traffic, even in this scenarios, one

978-1-6654-2845-3/21/$31.00 ©2021 IEEE
DOI 10.1109/MDM52706.2021.00025

needs to re-asses the energy consumption in points throughout the trip. What further complicates the matters is that multiple external factors (e.g., traffic fluctuations, weather changes, etc.) may still affect the quality of various predictions in real time.

In this paper we take a fist step towards considering the problem of Map Data Consumption (MDC) and introduce the Prediction of Map Data Consumption (PMDC) for a given trip (cf. Sec. III. Since, to our knowledge, the PMDC problem has not been formally addressed (we review the related works in Sec. II), one of the consequences is that there are no existing datasets built/collected for this task. To this end, we integrated dataobtained from multiple sources and studies related to maps, trips and traffic data. We created a "synthetic city" dataset (SCD) to be used in both training and experiments. and checked the use of an LSTM-based approach towards the PMDC problem (Sec. IV, including additional (internal and external) data sources. We report on experimental observations in Sec. V, and give conclusions and directions for future work in Section VI.

II. Related Works

We now overview the related works and position the paper in that context. We recognize that there are works which have addressed problems related to data use in mobile settings like, for example, data exchange in VANETS [8] and broadcast/indexing in air [9] – but we respectfully note that they are complementary to the current scope of the paper.

HD Maps: Typical HD maps consist of at least lane boundary geometries, road signs and other 3D objects, composed by points and voxels, attached with other descriptive tags and information. Due to extremely high level of detail, their data size is much larger than conventional maps. Real-time vehicle self-localization is the most popular and critical application using HD maps in autonomous driving industry from the broad perspective of decision/action taking. A safe and reliable autonomous driving system should take an action (from observation to decision) no slower than human reaction time – 200 milliseconds to 1 second depends on the specifics of a particular task [10]. Increased computational expenses when capturing specific objects (e.g., pole-like objects, guardrails/curbs, road surface/pavement) or even entire surrounding environment have been addressed in [12], [18], [25], [26].

Map Data Consumption: Similar to the definition of energy consumption in vehicle energy management study, MDC is based on the size of the maps data that a vehicle needs for its semi-autonomous or fully-autonomous driving function(s) at each moment [30]. We note that energy consumption is a well-studied field, and relevant values (e.g., Joule, the weight/volume of the petrol/gas) can be measured/quantified by on-vehicle sensors (and energy density charts) [5].

One possibility to quantify the MDC is by how much map data the vehicle needs to load for executing real time applications. Specifically, the vehicle needs to load surrounding objects, represented in polygons and grids/voxels – also known as vector and raster representations respectively. Even though representing objects in vectors significantly reduces the size of data and has invariance advantages in scale and shift, raster representation is still more in real-time autonomous driving applications since (cf. [30]): (1) sensors (e.g., LIDAR, depth/stereo cameras) equipped on vehicles acquire raster-like information directly. (2) most algorithms for autonomous (and assisted driving) applications, such as the ones used in vehicle self-localization, need to be fed in raster data. If there are higher safety requirements, the data resolution of retrieved voxels per unit area needs to be increased, causing substantial increase in the size of the map data to be downloaded [30].

The size of object information the vehicle needs is also highly dependent on the sensors configuration, such as refresh rate, layout and orientation, angular/spatial resolution, sensing range, and even vehicle's motion.

Linear models, such as Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) have been prediction tasks for decades [21]. In the fields of geospatial and transportation, the trajectories data is not only temporally continuous, but also spatially continuous. Machine Learning (ML) based approaches, such as Long Short-Term Memory (LSTM), RNN and their variants, have provided successful improvements in solutions to multiple mobility-related problems and applications: driver's behavior prediction [6], [15]; trajectory data mining/prediction [1], [7], [27], and vehicle speed/energy consumption prediction [5], [13]; and more broadly, Location Based Services (LBS) [11], [31].

This paper takes a first step towards addressing the PMDC problem from the perspective of broader context awareness. As it has not been studied, one of the tasks was to generate and HD maps dataset along with trips and traffic datasets, since the existing HD maps are proprietary and, typically, confined to smaller geographic areas, and are not integrated with other (e.g., traffic, road-network) data sources. In this spirit, our solution also investigates the benefits of including additional data sources.

III. Problem definition

A trip $l_i = \{\langle x_j^i, y_j^i, t_j\rangle | j \in [1, \ldots, L_i]\}$ is a sequence of time-stamped 2D points, where $\langle x_j^i, y_j^i\rangle$ denotes the geo-location of the j^{th} point of the trip l_i, in a suitable coordinate system, and L_i denotes the number of trajectory points of the trip l_i.

We consider a setting in which at a particular location and time instant $\langle x_k^i, y_k^i, t_k\rangle$ of the trip l_i, the vehicle may need to use map data. We assume that an HD map, denoted by M, is of a format $M = \{m_1, m_2, \ldots, m_p\}$ where each m_j contains the voxels corresponds to single tile in a respective grid partition of the geographical area of interest. The data associated with a given vehicle at a particular location is determined by the tile at which that vehicle is located at the corresponding time instant. We use $I_{(x,y,t)}$ to indicate the quantity of that data with respect to a point/location of the trip l_i at time instant t. We note that in the cases when t does not correspond to a value from the trajectory point (i.e., segment end-point at

some time t_k), we use linear interpolation between $\langle x^i_k, y^i_k, t_k\rangle$ and $\langle x^i_{k+1}, y^i_{k+1}, t_{k+1}\rangle$ where $t \in [t_k, t_{k+1}]$, to determine the corresponding location. We also note that the map dataset is determined by the corresponding tile. Thus, for each tile m_j, we have a pre-determined HD map data $HD(m_j)$ and, whenever $\langle x^i, y^i\rangle$ is $\in m_j$ – essentially, $I_{x^i,y^i,t}$ a function of $|HD(m_j)|$. In addition, however, $I_{x,y,t}$ is dependant on multiple other parameters pertaining to a specific location (e.g., the details of the road segment, lanes, etc.), as well as sensors' configuration which may be relevant to the specific location (e.g., the vehicle's speed, etc.).

When considering a subset of a trip l_i from the available datasets, one often faces the issues that the time instants associated with recorded locations are not uniformly distributed. In other words, the sampling intervals and/or data transmission are not regular in temporal sense (i.e., $t_j - t_{j-1} \neq t_{j+1} - t_j$ in the recorded trip). However, in real time, a particular vehicle may request data at every time instant. Thus, from the perspective of learning and prediction, to capture the location impact at every time instant, one is compelled to consider (portions of) a trip consisting of n *consecutive time instants* starting at some time instant $t = t_{k_1}$ – i.e., $\{\langle x^i_{k_1}, y^i_{k_1}, t_{k_1}\rangle$, $\langle x^i_{k_1}, y^i_{k_1}, t_{k_1+1}\rangle \ldots, \langle x^i_{k_n}, y^i_{k_n}, t_{k_1}+(n-1)\rangle\}$. As mentioned, given a trajectory data, we use linear interpolation for this purpose. However, this setting corresponds to the extreme case where one may intuitively expect that the map data is loaded at every single one of those time instants. In reality, the moving objects need not have downloads at every time instant – i.e., they may need only one download per tile, for as long as the locations are within a single tile. Thus, for a given (subset of the) trip l_i, starting at a time instant t and pertaining to the next n time units, MDC can be defined as:

$$MDC_{(l_i,t_k,n)} = \sum_{j=1}^{j=n} I_{x^i_{k_j}, y^i_{k_j}, t_k+j}$$

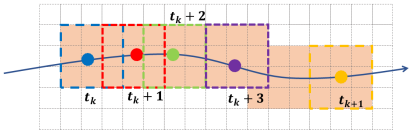

Fig. 1: Illustration of the MDC definition. $\langle x, y\rangle$ and l_i are omitted.

The concepts are illustrated in Figure 1, showing the tile structure of an HD map (light-gray dashed squares) and two consecutive trajectory points $\langle x^i_k, y^i_k, t_k\rangle$ and $\langle x^i_{k+1}, y^i_{k+1}, t_{k+1}\rangle$, along with points in consecutive time-stamps obtained by interpolation, exemplified by the red, green and purple disks (at times t_k, t_k+1 and t_k+2). The PMDC problem can now be defined as:

Given a location $\langle x^i_j, y^i_j\rangle$ of a vehicle with a trip l_i at time instant $t = t_j$, and its previous/historical locations $\{\langle x^i_{j-m}, y^i_{j-m}, t_j - m\rangle |\ m \in [1, s]\}$, where s is the size of the time-window, predict the data usage at a k^{th} time instant in the future – MDC_{l_i,t_j+k}.

In reality, if the successive (interpolated or raw) trajectory points are too close (relative to the download range requested), overlapped of downloaded tiles may occur, as shown in Figure 1, red (at time instant $t_k + 1$) and green ($t_k + 2$) rectangles. This will cause redundant data download as, strictly speaking, one needs a union of the tiles-data along the trip. Such considerations (e.g., caching part of the downloaded data) are left for future work.

We note that, whenever there is no ambiguity, we will omit certain subscript(s) and/or superscripts. Thus, for example, to denote the MDC for a specific time instant t_i for a given trajectory l, we will use $MDC_{(l,t_i)}$, or even MDC_{t_i} when clear from the context. In addition, when we are dealing with consecutive time-stamps with respect to a known starting time of a given trajectory, we will simply use MDC_j to denote the j^{th} time unit after the initial time-stamp value of that trajectory. Similarly, we will use MDC'_{t_j} to denote the predicted value for time instant t_j (or, relative to a known starting point, simply MDC'_j).

IV. PMDC: Challenges and Preliminary Methodology

We now describe the challenges for studying PMDC problem and present an initial LSTM-based approach.

(A) Data Generation: One of the main challenges to investigate PMDC problem is the lack of appropriate data. Towards that, we had to integrate multiple heterogeneous datasets.

– *Traces, traffic and maps:* A trace dataset needed to have the following properties to be used in our experiment: dense and consistent sampling rate, and large number of traces acquired across a wider time window. Some public datasets, such as government released New York Taxi [22] consists of millions of trips collected during a decade. However, the sampling rate is extremely sparse – sometime only start/pick-up and destination timestamps and locations are recorded. Other trajectory datasets, such as Roma Taxi [2] and T-drive [28] collect taxi data during a long time period and consist of plenty of trips. Even though the sampling rate of these datasets is much higher, it is still at the level of a several minutes which again, is not high enough to satisfy real-time requirement. DiDi offers several datasets that meet the demanding requirements of real-time autonomous driving. The datasets were collected during 2016, in greater Xi'an city area and organized in months.

However, to enable experimenting with the other features, traffic information is also important. Fortunately, DiDi also offers traffic information (travel time index, DiDi TTI) collected during certain months of 2017 [4].

– *Data integration and processing:* An SCD is built based on integrating the given datasets as follows. First of all, a road map of Xi'an metropolitan area is retrieved from OSM because, although DiDi TTI comes with a road network map, it is missing too many road links, which render is unusable. The bounding box of this map can be further narrowed down – e.g., in our case, to the downtown area $(34.207°, 108.922°)$, $(34.280°, 109.009°)$, an $8,060$ meters by $8,053$ meters (lateral)

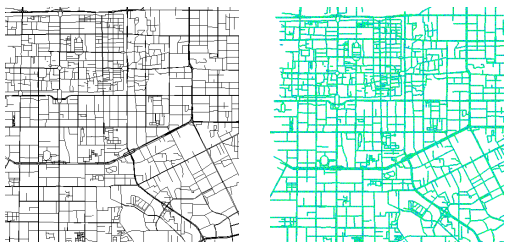

Fig. 2: OSM road map of Xi'an downtown (left) and the heat map of (synthetic) HD map data size per tile/pixel (right).

area to aggregate the density of traces and road network. $1,814$ road links are retrieved in this area, and the visualization is shown in Figure 2 (left). A consistent misalignment between DiDi's coordinate and OSM coordinate is obtained, and manually corrected by aligning several intersections. This enabled having the map, traffic information, and all trajectory traces integrated in SCD.

– *HD maps generation:* There are not many HD map datasets available in either academia or industry, and none of them has the ability to cover (larger regions of) an entire city. Our current work only needs the *size* of HD maps per unit area (a tile) or road length. There is a positive correlation between the size of HD maps per tile, and the number of voxel/occupancy-grids within a tile.

Define the voxel density as the number of voxels per surface unit area. We learn the voxel density distribution (at different resolutions, per unit area) relying on previous work acquired in Chicago area [30], and then apply this distribution to Xi'an's road network to create a synthetic HD maps. The heat map of the size of the HD map is visualized in Figure 2 (right).

– *TTI integration:* Due to the fact that the traces data and traffic information were collected during different years, it is impossible to fuse them, i.e., assign the TTI information to each trajectory point directly. To address this issue, we select the traces [3] and traffic from the same month but different years, normalize the traffic information (of each link) during a month to a"weekly calendar", and project this traffic information to the traces.

– *Feature embeddings:* Embedding features in graph representation can easily cause the problem of the curse of dimensionality. Since our task focuses on the PMDC on a certain route, while the synthetic HD map data is generated and attached to every unit area on each link, we naturally concatenate the features of following route after local features. Theoretically, we can concatenate all the information of the rest of trajectory points to form an extremely long feature vector for each trajectory point. However, only a certain length of future information will used in our experiment, to ensure that the information is sufficient for the PMDC purpose, while not being redundant.

(B) LSTM based solution to PMDC: We now proceed with explaining a simple (preliminary) solution. Firstly, we note that in our case, the HD maps are represented in higher resolution tiles. When it comes to the auxiliary data, it includes: (a) Vehicle motion measurements, including vehicle velocity (2-D), accelerations (2-D) and location. (b) External and real-time information, not directly a factor to MDC size, including traffic speed and traffic indices of each link. (c) Trip global information, which records the progress (in percentage) of a trip in time-wise and distance-wise.

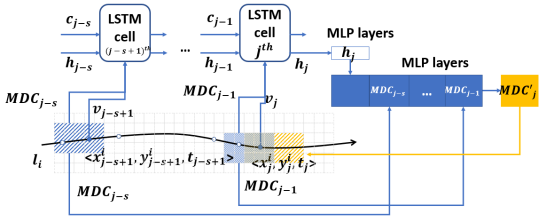

Fig. 3: Vehicle features at j^{th} trajectory point (blue dot) and MDC (blue tiles), combined with previous $j-s+1$ historical features and MDCs are fed into the LSTM, to predict PMDC for the next distance (yellow tiles).

Given a trip $l_i = \{\langle x_j^i, y_j^i, t_j\rangle | j \in [1, \ldots, L_i]\}$, a feature vector at each trajectory point can be represented as $v_i = \{\langle x_j^i, y_j^i, b_j^i, a_j^j, g_j^i, t_j\rangle \| j \in [1, \ldots, L_i]\}$, where b_j^i, a_j^i, and g_j^i denote internal, external and global features at j^{th} trajectory point of trip l_i. More specifically, b_j^i is a combination of vehicle's motion (a function of $\langle x_j^i, y_j^i\rangle$ and $\langle x_{j+1}^i, y_{j+1}^i\rangle$) and surrounding HD maps (m_j). One may argue that, in theory, using b_j^i only is sufficient to predict the MDC (i.e., MDC itself is a function of geo-locations and HD maps), which makes the problem roughly equivalent to a speed prediction or next-location prediction problem. Furthermore, a_j^i is introduced to describe the external and real-time features at j^{th} trajectory point, such as traffic indices, speed and link categories ahead of current geo-location. Last but not least, a trip global information g_j^i indicates the trip status/progress if the trip destination is known. In our experiments (cf. Section V), three feature combinations will be used: $v_j^i = \{b_j^i | \langle b_j^i, a_j^i\rangle | \langle b_j^i, a_j^i, g_j^i\rangle\}$.

Given its effectiveness in summarizing the contextual information from sequential data, we utilize LSTM to encode the trajectory knowledge from historical points, and each trajectory point $j \in L_i$ of a trip l_i is an LSTM time step. Inspired by the usage of word embeddings and sliding windows in natural language processing studies, we generate the trajectory embeddings at each trajectory point using various addressed featurization techniques and use the embeddings in a moving sliding window to forecast the MDC at the future trajectory point continuously.

LSTM cell at time step j (relative to the starting time instant) of one trajectory path, taking trajectory embeddings v_j $LSTM(h_{j-1}, c_{j-1}, v_j, y_{j-1}))$, is defined as follows:

$$
\begin{aligned}
e_j &= \sigma(W_e[h_{j-1}, c_{j-1}, v_j, y_{j-1}] + d_e) \\
f_j &= \sigma(W_f[h_{j-1}, c_{j-1}, v_j, y_{j-1}] + d_f) \\
o_j &= \sigma(W_o[h_{j-1}, c_{j-1}, v_j, y_{j-1}] + d_o) \\
\tilde{c_j} &= tanh(W_c[h_{j-1}, c_{j-1}, v_j, y_{j-1}] + d_c) \\
c_j &= f_j * c_{j-1} + e_j * \tilde{c_j} \\
h_j &= o_j * tanh(c_j)
\end{aligned}
\tag{1}
$$

	Drivers #	Trips #	Median time (s)	Median distance (m)
Raw	523,881	3,069,317	414	2,052
MM	311,228	433,119	397	2,035
Filtered	N/A	113,976	530	2,504

TABLE I: Number of drivers, number of trips, median trip time and trip distance of the raw dataset, map-matched (MM) set and the final filtered set.

The input, forget, and output gates are e_j, f_j, and o_j respectively; the hidden state h_j indicates the sequential embeddings and c_j represents the contextual embeddings. $\tilde{c_j}$ denotes the intermediate embeddings carried out from input contexts. Weight matrices W_e, W_f, W_o, W_c and bias vectors d_e, d_f, d_o, d_c are shared across different trajectories $\{l_i\}$.

Suppose that the sliding window is of size s. The hidden state of our vanilla multivariate LSTM with sliding windows $\mathit{LSTM}(j)$ is obtained by the following recursive function:

$$h_j = \mathit{LSTM}(h_{n-1}, c_{n-1}, v_n, y_{n-1}), n \in [j-s+1, j] \quad (2)$$

The initial state h_0 is a tensor padded with 0s. Similarly, when $j < s$ the feature embeddings are padded with 0s. We connect h_j with a stack of fully connected layers $\phi(j)$, for generating the predicted MDC value MDC'_j (noting that the objective function is the l_2 loss between MDC'_j and MDC_j):

$$\mathit{MDC}'_j = ([h_j, \{\mathit{MDC}_n\}]) \quad (3)$$

V. Experimental Results

We now present the empirical results and discuss the impacts of different step lengths and features being used, as well as training models (LSTM, and RNN). Note that, in the sequel we use a 2-character abbreviation to denote experiment setup combination. The first character is numerical and denotes the number of steps; the second character denotes the type of features being used: B for basic internal feature ($v_j = b_j$), A for external features ($v_j = \langle b_j, a_j \rangle$) and T for global features added ($v_j = \langle b_j, a_j, g_j \rangle$), where $\|b_j\|$, $\|a_j\|$ and $\|g_j\|$ are 29, 60 and 2 respectively. For example, **5A** stands for the experiment setup with using 5 steps and external features. The learning models used in our experiments share the same configurations: the dimentionality and batch size are 256 and 32, respectively; the dropout between fully connected layers is 0.2 with using ReLU as the activation function; and the learning rate is $5e^{-6}$. Table I shows the numbers of drivers, trips, and median trip time and distance of each trip processing step. We use 113,976 high quality trips in our experiments.

All of our source code of data generation, processing and learning, as well as a portion of the data are available at https://github.com/zangandi/pmdc. For the "raw" data, please consult with DiDi Gaia Initiative and apply for the TTI and trips data access (to which our code can be directly applied).

A. Empirical Results

In the sequel, we compare the performance among various features and model combinations, and show plots of results in naive metric in Figure 4. The plots are generated using

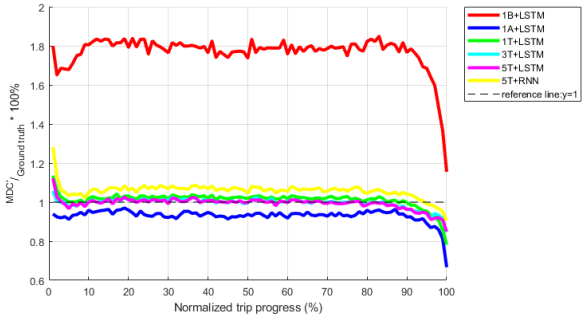

Fig. 4: Prediction performances of selected training setups. Each plot is normalized to 100 poses and presenting the median performance of entire test set of each pose.

	LSTM						RNN	
	MAPE	**MSE**		**MAPE**	**MSE**		**MAPE**	**MSE**
1B	1.07	1.53	**5B**	0.318	0.379	**5B**	0.326	0.381
1A	0.38	0.43	**5A**	0.294	0.372	**5A**	0.331	0.376
1T	0.33	0.399	**5T**	0.288	0.369	**5T**	0.335	0.376

TABLE II: Selected MAPEs and MSEs of different feature, step and model combinations.

the entire test set, and all trips are normalized into a 100-pose trip for a better visualization. We note that in our trip processing step, we have already filtered out all the trips with fewer than 100 poses. As mentioned in Section IV, the MDC generated from our SCD is a function of vehicle's location and HD maps (only). Using the most basic features is sufficient to predict the MDC of each pose. According to Figure 4, with adding external features (a_j), the prediction performance is significantly improved. Another observation is, a huge performance drop appears at the end of a trip, no matter which combination of $v_j = b_j$ or $v_j = \langle b_j, a_j \rangle$, steps and learning model is being used. The feature ($v_j = \langle b_j, a_j, v_j \rangle$) introduced in previous sections can significantly fix the performance drop. The experiment also shows a longer historical steps leads to a higher prediction performance.

B. Metrics and Result Analysis

To systematically evaluate the prediction results of a time series regression problem, assessment criteria such as Mean Absolute Error (MAE), Percentage Error (PE), Scaled Error (SE) and their variants are widely used [20]. Compared to MAE (and its variants) – a scale-dependent error metric, which has advantages in evaluating single time series or multiple time series with the same units in similar orders of magnitudes, and SE-like evaluation metrics – the metrics show advantages in comparing accuracy across series with different units, PE-like evaluations are more suitable in our case [24]. Because the MDCs across the series are with same unit, but the orders of magnitudes are significantly affected by different locations, vehicle motion and configuration – sometime they cover over three orders of magnitude. Define error of the MDC prediction at j^{th} trajectory point (of a trip l_i) as: $e_j = \mathit{MDC}'_j - \mathit{MDC}_j$,

where MDC'_j denotes the MDC prediction value at trajectory point $j \in [1, L_i - 1]$. We note that, ideally, at the end of a trip, $MDC'_j = 0$, and MDC_{j+1} does not exist, so we will ignore this term because we use PE-like measures. Given PE at $p_j = 100 \times e_j / MDC_j$, and a trip $l_i \in T$ where T is our test set, the Mean Absolute Percentage Error (MAPE) can be defined as $\mathbf{MAPE}_j = mean(|p_j|), j \in [1, L_i - 1]$. A detail result of selected experiment setups is presented in Table II.

VI. Concluding Remarks and Future Directions

We identified and addressed a novel problem of vehicular data consumption in the settings of HD maps. Specifically, we formalized the MDC aspect and targeted the solution to the PMDC problem via LSTM. We also considered the potential benefits of including different internal and external parameters in the feature space. Part of the efforts addressed the issue of fusing different types of data from multiple sources in order to create a "synthetic city". Last but not least, we conducted experiments comparing the impact of different feature combinations on the effectiveness of PMDC. As ML based approaches for vehicles data consumption have not been investigated, one of the issues to be addressed in the future is creation of relevant datasets since, in the short term, it is unlikely that one could expect HD maps covering larger portion of cities. Additionally, as mentioned in Section III, one task is to incorporate the aspect of data caching in the overall prediction of data consumption and incorporate uncertainty [23]. Last, but not the least, an interesting problem is to generate a model for speed recommendation based on the limits of data download. Specifically, given a limit on bandwidth availability in particular segments of the trip, for safety, an autonomous vehicle should decrease the speed or, if possible, alert the driver to take the control [32].

Acknowledgement: The work was partially supported by the National Science Foundation grant SWIFT 2030249 and National Natural Science Foundation of China Grant 62072077.

References

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

[2] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). https://crawdad.org/roma/taxi/20140717, July 2014.

[3] DiDi Chuxing GAIA Open Dataset. Oct 2016, Xi'an City Second Ring Road Regional Trajectory Data Set. https://gaia.didichuxing.com.

[4] DiDi Chuxing GAIA Open Dataset. Travel Time Index. https://gaia.didichuxing.com.

[5] Tushar D Gaikwad, Zachary D Asher, Kuan Liu, Mike Huang, and Ilya Kolmanovsky. Vehicle velocity prediction and energy management strategy part 2: Integration of machine learning vehicle velocity prediction with optimal energy management to improve fuel economy. Technical report, SAE Technical Paper, 2019.

[6] Jun Gao, Yi Lu Murphey, and Honghui Zhu. Multivariate time series prediction of lane changing behavior using deep neural network. *Applied Intelligence*, 48(10):3523–3537, 2018.

[7] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. Identifying human mobility via trajectory embeddings. In *IJCAI*, volume 17, pages 1689–1695, 2017.

[8] Hannes Hartenstein and Kenneth P. Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.*, 46(6):164–171, 2008.

[9] Qinglong Hu, Wang-Chien Lee, and Dik Lun Lee. A hybrid index technique for power efficient data broadcast. *Distributed Parallel Databases*, 9(2), 2001.

[10] Arne Kesting and Martin Treiber. How reaction time, update time, and adaptation time influence the stability of traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, 23(2):125–137, 2008.

[11] Dejiang Kong and Fei Wu. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *IJCAI*, volume 18, pages 2341–2347, 2018.

[12] Liang Li, Ming Yang, Lindong Guo, Chunxiang Wang, and Bing Wang. Precise and reliable localization of intelligent vehicles for safe driving. In *International Conference on Intelligent Autonomous Systems*, 2016.

[13] Kuan Liu, Zachary Asher, Xun Gong, Mike Huang, and Ilya Kolmanovsky. Vehicle velocity prediction and energy management strategy part 1: Deterministic and stochastic vehicle velocity prediction using machine learning. Technical report, SAE Technical Paper, 2019.

[14] Rong Liu, Jinling Wang, and Bingqi Zhang. High definition map for automated driving: Overview and analysis. *Journal of Navigation*, 73(2):324–341, 2020.

[15] Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Transactions on Intelligent Transportation Systems*, 18(5), 2016.

[16] Anh Nguyen and Bac Le. 3D point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pages 225–230. IEEE, 2013.

[17] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3D registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[18] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[19] Heiko G Seif and Xiaolong Hu. Autonomous driving in the iCity–HD maps as a key challenge of the automotive industry. *Engineering*, 2(2), 2016.

[20] Maxim Vladimirovich Shcherbakov, Adriaan Brebels, Nataliya Lvovna Shcherbakova, Anton Pavlovich Tyukov, Timur Alexandrovich Janovsky, and Valeriy Anatol'evich Kamaev. A survey of forecast error measures. *World Applied Sciences Journal*, 24(24):171–176, 2013.

[21] Kamilya Smagulova and Alex Pappachen James. A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228(10):2313–2324, 2019.

[22] Taxi and Limousine Commission. New York city taxi trip data, 2009-2018. Inter-university Cons. for Political and Social Research, 2019.

[23] Goce Trajcevski. Uncertainty in spatial trajectories. In *Computing with Spatial Trajectories*. 2011.

[24] Dongjie Wang, Yan Yang, and Shangming Ning. Deepstcl: A deep spatio-temporal convlstm for travel demand prediction. In *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018.

[25] Thorsten Weiss, Nico Kaempchen, and Klaus Dietmayer. Precise ego-localization in urban areas using laser scanner and high accuracy feature maps. In *Intelligent Vehicles Symposium*, 2005.

[26] Lihong Weng, Ming Yang, Lindong Guo, Bing Wang, and Chunxiang Wang. Pole-based real-time localization for autonomous driving in congested urban scenarios. In *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 96–101. IEEE, 2018.

[27] Hao Xue, Du Q Huynh, and Mark Reynolds. SS–LSTM: A hierarchical lstm model for pedestrian trajectory prediction. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[28] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: Driving directions based on taxi trajectories. In *Proceedings of 18th ACM SIGSPATIAL Conference on Advances in Geographical Information Systems*, 2010.

[29] Andi Zang, Zichen Li, David Doria, and Goce Trajcevski. Accurate vehicle self-localization in high definition map dataset. In *1st ACM SIGSPATIAL Workshop on High-Precision Maps and Intelligent Applications for Autonomous Vehicles*, page 2. ACM, 2017.

[30] Andi Zang, Shiyu Luo, Xin Chen, and Goce Trajcevski. Real-time applications using high resolution 3D objects in high definition maps (systems paper). In *ACM SIGSPATIAL*, pages 229–238, 2019.

[31] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Zhixu Li, Jiajie Xu, and Victor S Sheng. Where to go next: A spatio-temporal LSTM model for next poi recommendation. *arXiv preprint arXiv:1806.06671*, 2018.

[32] Martin Zimmermann and Franz Wotawa. An adaptive system for autonomous driving. *Software Quality Journal*, 28, 2020.