Spatio-Temporal GRU for Trajectory Classification

Hong-Bin Liu*, Hao Wu†‡, Weiwei Sun†, Ickjai Lee*
*College of Science and Engineering, James Cook University, Australia
{hongbin.liu, ickjai.lee}@jcu.edu.au
†School of Computer Science, Fudan University, Shanghai, China
{wuhao5688, wwsun}@fudan.edu.cn
‡ Bytedance AI Lab

***Abstract*—Spatio-temporal trajectory classification is a fundamental problem for location-based services with many real-world applications such as travel mode classification, animal mobility detection, and location recommendation. In the literature, many approaches have been proposed to solve this classification task including deep learning models like LSTM recently for sequence classification. However, these approaches fail to consider both spatial and temporal interval information simultaneously, but share some common drawbacks: omitting either the spatial information or the temporal interval information out. Some models like Time-LSTM, have been proposed to handle the temporal interval information for spatio-temporal trajectories, but they do not take into account the spatial information. Note that, considering both spatial and temporal interval information is crucial for spatio-temporal data mining in order not to miss any spatio-temporal pattern. In this study, we propose a trajectory classifier called Spatio-Temporal GRU to better model the spatio-temporal correlations and irregular temporal intervals prevalently present in spatio-temporal trajectories. We introduce a novel segmented convolutional weight mechanism to capture short-term local spatial correlations in trajectories and propose an additional temporal gate to control the information flow related to the temporal interval information. Performance evaluation demonstrates that our proposed model outperforms popular deep learning approaches for the travel model classification problem.**

***Keywords*-trajectory classification, spatio-temporal trajectory, GRU, travel model classification, deep learning**

I. Introduction

Spatio-temporal trajectory classification is one of the fundamental tasks in trajectory data mining. It facilitates many real-world applications such as travel mode classification, animal mobility detection, and location recommendation [1]–[3]. As most trajectories are composed of spatial coordinates and corresponding time stamps, the problem of spatio-temporal trajectory classification is of great interest in the data mining community. A spatio-temporal trajectory is a list of spatio-temporal entries, $(\langle x_1, y_1, t_1\rangle, \langle x_2, y_2, t_2\rangle, \ldots, \langle x_n, y_n, t_n\rangle)$, where $x_i, y_i \in \mathbb{R}^2$ and $t_i \in \mathbb{R}^+$ for $1 \le i \le n$ and $t_1 < t_2 < \ldots < t_n$. A regular spatio-temporal trajectory is when $|t_{j+1} - t_j| = |t_{k+1} - t_k|$ for $\forall\, j, k$ whilst a **irregular spatio-temporal trajectory** is when $|t_{j+1} - t_j| \neq |t_{k+1} - t_k|$ for $\exists\, j, k$ where $1 \le j \neq k < n$. Notably, most spatio-temporal trajectories are irregular due to inconsistent weather conditions, geographical barriers such as tunnels, device malfunctions, battery issues and other environmental conditions where GPS receptions are unavailable [4], thus we focus on irregular spatio-temporal trajectories in this paper. This unique irregular property of spatio-temporal trajectories creates challenges for trajectory classification problems. Figure 1 illustrates two spatio-temporal trajectories with different time intervals.

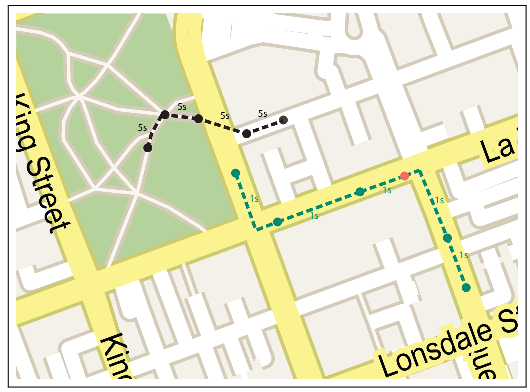


Fig. 1: Two trajectory segments with different sampling rates. Top left (black) is a walking trajectory with a sampling rate of 1 second and bottom right (green) is a driving trajectory with a sampling rate of 5 seconds. The red dot represents a missing point which may result in a high-variance in the estimation of speed.

Given a spatio-temporal trajectory, trajectory classification involves learning a model that can classify trajectories into various categories based on underlying characteristics. Classification of transportation modes from spatio-temporal trajectories is a hot trajectory classification problem [5], [6]. Zheng *et al.* [5] studied to classify human trajectories into four groups based on their travel modes, such as driving, cycling, walking, and running through machine learning approaches. As typical machine learning-based classifiers require certain features to be extracted from the raw data, they extracted nine features including speed, velocity and direction, which were then fed into a classifier, e.g. SVM.

With recent advances in deep neural networks, deep models have been proposed that outperform traditional shallow machine learning models. One of the major advantages of deep neural networks is that they perform well on raw data and there is no need to perform feature engineering procedures. Among these deep models, Recurrent Neural Network (RNN) models have become state-of-the-art methods for sequential data. Most existing RNN-based approaches are designed to

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00152

model the sequential information while assuming regular time intervals thus traditional RNN-based models are unsuitable for irregular spatio-temporal trajectories. In speech recognition, raw acoustics are segmented into equal sized frames, whilst in natural language processing, sentences are segmented into words. These sequential data either omit the temporal interval information out as they are equally segmented where predictions do not rely on the time interval information (as in acoustic data), or ignore the temporal interval information (as in natural language data).

The literature reveals that several attempts have been made to apply deep neural networks to trajectory classification tasks. Shah *et al.* [7] fed raw basketball trajectories directly into an LSTM network in order to predict the successful shot. Similarly, Liu *et al.* [8] tried to classify travel modes of raw human trajectories using a bi-LSTM neural network. Shah [7] and Liu [8] investigated the same problem by concatenating the timespan τ to the (x, y). Shi *et al.* [9] were aware of the importance of temporal interval information in spatio-temporal trajectories, and Zhu *et al.* [10] proposed three Time-LSTM models which are specifically designed to handle the temporal interval information. However, this approach was specific to the problem of recommendation and is not designed to be directly applicable to the spatio-temporal trajectory classification task as it is not structured to capture local spatial variations.

Motivated by the fact that no existing approach can both adequately incorporate the temporal interval information and capture the correlations (variations) of spatial information, we propose a RNN-based model, namely Spatio-Temporal GRU (in short ST-GRU), specifically designed for spatio-temporal trajectory data.

II. Spatio-Temporal GRU

We first briefly review the original GRU model and then introduce the proposed ST-GRU model.

A. Traditional GRU

The GRU model [11] is one of the variances of RNN used to solve the gradient vanishing issue of long range dependencies. The computation formula of sequence $X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_n$ at step t can be formalised as follows:

$$\begin{cases} z_t = \sigma(W_{xz}X_t + U_{hz}h_{t-1} + b_z), \\ r_t = \sigma(W_{xr}X_t + U_{hr}h_{t-1} + b_r), \\ h_t^{'} = f(W_{xh}X_t + r_t \odot U_{hh}h_{t-1} + b_h), \\ h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t^{'}, \end{cases}$$

where $z_t, r_t, h_t^{'}, h_t$ represent the update gate, the reset gate, the memory state and the output, respectively; W_{x*}, U_{h*} and b_* are parameters of the GRU cell; $\odot$ denotes the element-wise product, and σ and f denote $sigmoid$ and $tanh$ activation functions, respectively. It has been shown that the GRU model achieves similar results to LSTM in many tasks while having less parameters and computational complexity [11].

B. Proposed Spatio-Temporal GRU

*1) **Spatial modeling**:* For spatio-temporal trajectories, the shape of trajectory and the local segments of several continuous points usually have a key influence on trajectory classification. If we directly feed one spatial point separately into the network, it will make the spatial information sparse and disjoint across time steps. As a solution, we propose a *segmented convolutional weight* mechanism on all computations corresponding to the spatial information in order to model the local spatial correlations and variations. Thus, instead of using one feature vector $X_t \in \mathbb{R}^C$ as input at each step t, we feed a segment $\mathcal{X}_t \in \mathbb{R}^{L \times C}$ which is composed of L consecutive feature vectors in the trajectory.

Recently, 1D-CNN has been proven to be capable of handling sequence modeling tasks such as machine translation [12] and sentence classification [13], resulting in great advances in both computation efficiency and accuracy. The key benefit of 1D-convolution over the recurrent operations is that the convolution operation explicitly captures the local correlations of input sequence by filters serving as the context window, while recurrent networks can only feed in one element each time and can model the correlations implicitly in the recurrent cell. Note that, for the task of spatio-temporal trajectory classification, the objective is usually related to a general attribute of trajectories such as travel modes. Such general attributes are most likely related to the local shape and correlations of a given trajectory. Thus, directly feeding each point into a recurrent network at each time step can make it hard for the network to focus on such local correlations, whereas such prior knowledge can be naturally incorporated via 1D-CNN. In addition, traditional recurrent networks can only process input data sequentially in one dimension while missing two dimensional correlations, i.e. from left to right or from right to left, but convolutions can capture the two dimensional spatial information by a context window around the input, thus better suited for spatio-temporal trajectory mining.

Based on this observation, we use 1D-convolution to replace all the linear transformation operations. Note that, the input of recurrent cell is a segment consisting of several contiguous points rather than a single point, and 1D-convolution is adopted to capture the local correlations for each time step computation. The computational flow of our ST-GRU modified with the segmented convolutional weight mechanism is formally described as follows:

$$\begin{cases} \mathcal{Z}_T = \sigma(W_{xz} \otimes [\mathcal{X}_T,\ \mathcal{H}_{T-1}]), \\ \mathcal{R}_T = \sigma(W_{xr} \otimes [\mathcal{X}_T,\ \mathcal{H}_{T-1}]), \\ \mathcal{H}_T^{'} = f(W_{xh} \otimes \mathcal{X}_T + \mathcal{R}_T \odot (U_{hh} \otimes \mathcal{H}_{T-1})), \\ \mathcal{H}_T = \mathcal{Z}_T \odot \mathcal{H}_{T-1} + (1 - \mathcal{Z}_T) \odot \mathcal{H}_T^{'}. \end{cases} \tag{1}$$

For clarity, we use the subscript T to indicate the index of the segment and t as the index of the basic step in the original sequence. Here, $\otimes$ denotes the 1D-convolution operation. Note that, the input information $\mathcal{X}_t$ now includes several steps of the sequence, and the linear transformation becomes the 1-D convolution which preserves the step dimension, i.e. L. This results in all gates ($\mathcal{Z}_t$, $\mathcal{R}_t$) become $L \times H$ matrices recording L basic steps. The states $\mathcal{H}_t$ and $\mathcal{H}_t^{'}$ also become $L \times H$ matrices to be compatible with the gate computation.

The segmented convolutional weight mechanism makes the gate and state computation of our ST-GRU become locally sensitive. Namely, for each computation, information before and after the current basic step t can be all inferred during the computation. In contrast to traditional recurrent networks, where there is only one element as input and they tend to be sparse in information, the segmented convolutional weight mechanism makes information in each GRU step rich and useful as input is composed of several points. This maximizes the sequential modeling power of our ST-GRU. Although [14] attempts to extract local sensitive features via 1D-CNN, it is for images ignoring the temporal interval information. Our proposed segmented convolutional weight mechanism is different and more robust to outliers.

Suppose a sequence can be divided into M segments having L basic time steps. The computational flow is illustrated in Figure 2 with arrows indicating the dependencies brought in by the 1D-convolution with a length-3 kernel. We construct L sub-trajectories by subsampling the original trajectory with an interval L, i.e. $\{\mathcal{X}_1^1 \rightarrow \mathcal{X}_2^1 \rightarrow \mathcal{X}_T^1\}$, $\{\mathcal{X}_1^2 \rightarrow \mathcal{X}_2^2 \rightarrow \mathcal{X}_T^2\}, \cdots,$ $\{\mathcal{X}_1^L \rightarrow \mathcal{X}_2^L \rightarrow \mathcal{X}_T^L\}$. The superscript indexes the element in each segment, and the subscript indexes each segment. Then Eq. (1) can be regarded as the computation of L sub-trajectories by referring context information in each time step. For instance, the blue dashed box in Figure 2 represents the computation of the t-th sub-trajectory Tr^t by referring context input features each time step and generates the output h_M^t with respect to Tr^t, and the red dashed box represents the sub-trajectory Tr^{t+1} output and the state h_M^{t+1}. The state $\mathcal{H}_M$ is seen as column-wisely stacking the states of all L sub-trajectories, i.e. $\mathcal{H}_M = [h_M^1, h_M^2, \cdots, h_M^L]$. The final prediction can be produced by performing a global pooling process on these L sub-states. Thus, encoding the whole trajectory can be regarded as encoding L sub-trajectories. Even though some outliers lie in certain sub-trajectories, the final result will be averaged by L predictions in an ensemble way which makes the model more robust to outliers as well as preserving the power to capture the local correlations and variations. We will conduct experiments to demonstrate the superiority of our segmented convolutional weight mechanism compared to the feature convolution strategy.

*2) **Temporal modeling**:* As studied in [4], [9], the interval information (the interval between two consecutive points in a trajectory) plays an important role as it contains the temporal information. In this paper, we regard the temporal interval information of step t as the relative time interval between the time stamp at step t and step $t-1$, i.e. $\Delta\tau_t = \tau_t - \tau_{t-1}$. A straightforward way to incorporate the temporal interval information is to directly treat the spatial and the temporal interval information as a whole feature vector, i.e. $X_t = (x_t, y_t, \Delta\tau_t)$, and let the weights W_{x*} automatically learn the correspondence between the spatial features and the temporal

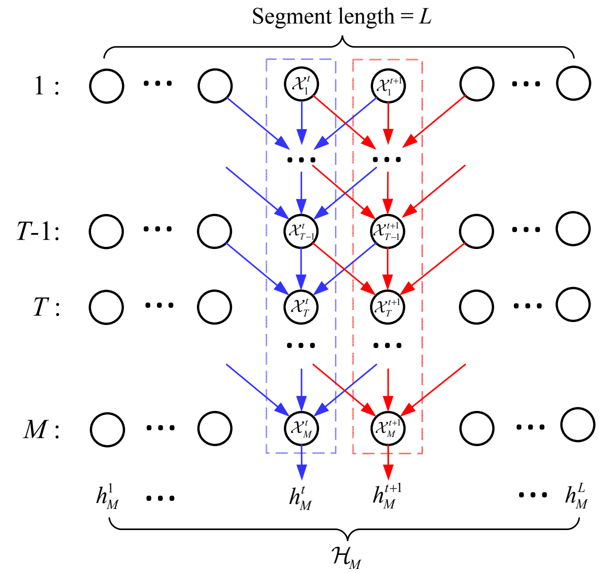
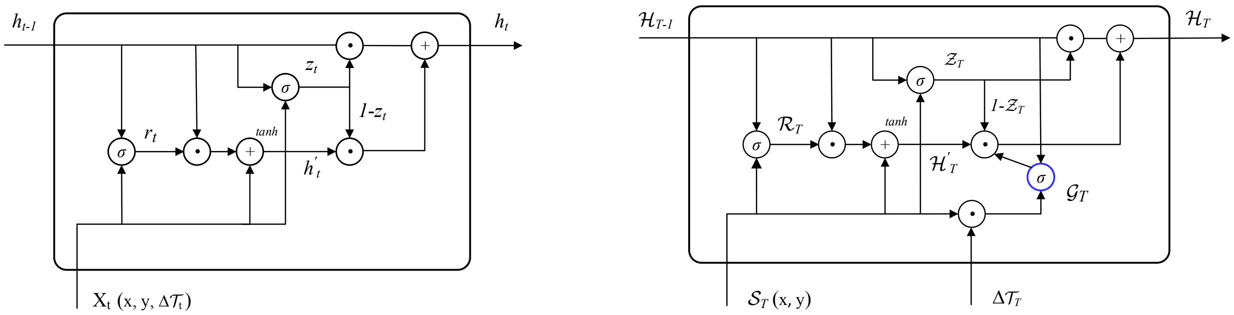


Fig. 2: The computation flow of proposed segmented convolutional weight mechanism with respect to an entire sequence having $M \times L$ time steps. Each row represents a length-L segment $\mathcal{X}_T$ and the superscript (e.g., t in $\mathcal{X}_M^t$) represents the index of the element in each segment.

features. Such a strategy is simple but not optimal as the characteristics of spatial and temporal interval information are totally different (and also measurement units are different) and, therefore, it is inappropriate to use a linear transformation to combine the spatial and the temporal interval information. Inspired by the time gate proposed by Time-LSTM model [10], we extend the GRU model with a temporal gate $\mathcal{G}_T$ with a segmented convolutional weights mechanism. The computation formulas are:

$$\begin{cases} \mathcal{Z}_T = \sigma(W_{xz} \otimes [\mathcal{S}_T,\ \mathcal{H}_{T-1}]), \\ \mathcal{R}_T = \sigma(W_{xr} \otimes [\mathcal{S}_T,\ \mathcal{H}_{T-1}]), \\ \mathcal{G}_T = \sigma(W_{xg} \otimes [\mathcal{S}_T,\ \mathcal{T}_T,\ \mathcal{H}_{T-1}]), \\ \mathcal{H}_T^{'} = f(W_{xh} \otimes \mathcal{S}_T + \mathcal{R}_T \odot (U_{hh} \otimes \mathcal{H}_{T-1})) \odot \mathcal{G}_T, \\ \mathcal{H}_T = \mathcal{Z}_T \odot \mathcal{H}_{T-1} + (1 - \mathcal{Z}_T) \odot \mathcal{H}_T^{'}. \end{cases}$$

The temporal gate $\mathcal{G}_T$ is decided by the current input spatial features, intervals as well as the historical states $\mathcal{H}_t$, and it is plugged onto the input state $\mathcal{H}_T'$. The idea is that we want the temporal gate to control the confidence on the input state of step T, i.e. $\mathcal{H}_T'$ by considering the temporal interval information corresponding to the current position and previous position (which is recorded in $\mathcal{H}_{T-1}$). Take the travel mode classification task as an example, intuitively, we know the moving speed of the object determines the travel mode. However, if the object stops at a crossroad or the interval suddenly becomes very large, both cases will result in an inaccurate speed estimation. In the first case, the object is actually not moving which will confuse the model as it should be regarded as noise and filtered out. Whereas, the second case suffers from a large variance in speed estimation which should lower its information confidence for prediction. If the temporal gate is introduced, the input information for the current step



(a) Traditional GRU

(b) Proposed ST-GRU

Fig. 3: The computational graph of one step in traditional GRU and ST-GRU.

will be further controlled by the temporal gate, which may be set to a low value to filter out the inputs that are likely to confuse the decision.

C. Spatio-Temporal GRU model for Trajectory Classification

Figure 3 illustrates computational graphs of one step in traditional GRU and ST-GRU. In this section, we present the details of adopting the ST-GRU model for the spatio-temporal trajectory classification problem. Given a time interval preprocessed trajectory $Tr = \{(x_1, y_1, \Delta\tau_1), (x_2, y_2, \Delta\tau_2), \cdots, (x_N, y_N, \Delta\tau_N)\}$ containing N points where $\Delta\tau_t = \tau_t - \tau_{t-1}$ for $1 \leq t \leq N$ and $\tau_0 = 0$, we first break them into $M = \lfloor N/L \rfloor$ length-L segments to be compatible as input to our ST-GRU model. We denote $S_T \in \mathbb{R}^{L\times 2}$ as the spatial coordinates of T-th segment and $\mathcal{T}_T \in \mathbb{R}^L$ to be the corresponding interval. Directly regarding the spatial coordinate S_T as a feature may not be an optimal solution as latitude and longitude will then be regarded as independent features. This is not desirable as modeling the correlation between latitude and longitude has to be handled by the ST-GRU's recurrent cell. In the literature, there have been several attempts to adopt embedding techniques into trajectory modeling [15], [16]. However, trajectories are modelled by either a series of categorical Points-of-Interest (POIs) or road segments which are similar to textual data. As a solution, we perform an additional feature transformation layer on the spatial coordinates to increase the feature dimension before feeding them into our ST-GRU model which can be regarded as a *soft-embedding* of the raw position. Formally, the spatial feature $\mathcal{S}_T$ can be computed by $\mathcal{S}_T = ReLU(S_T W_s + b_s) \in \mathbb{R}^{L\times C}$ where $W_s \in \mathbb{R}^{2\times C}$ and the bias $b_s \in \mathbb{R}^C$. C denotes the input spatial feature dimension of ST-GRU.

For the ST-GRU cell, we stack two ST-GRU cells to enlarge the model's capacity. Note that the temporal interval information $\mathcal{T}_T$ will be individually fed into two ST-GRU layers and the states produced by the first layer will be the spatial feature of the second layer. After the whole sequence is traversed, ST-GRU outputs the hidden states of the last segment, i.e. $\mathcal{H}_M \in \mathbb{R}^{L\times H}$ where H denotes the dimension of the hidden states. As introduced in Section II-B1, $\mathcal{H}_M$ is composed of the L predictions $\left(h_M^1, h_M^2, \cdots, h_M^L\right)$ generated by L sub-trajectories. We adopt the average global pooling on these L states to aggregate L states to $h_M \in \mathbb{R}^H$. One fully connected layer with $ReLu$ activation is adopted with a linear Softmax classifier that follows.

D. Implementation and Hyperparameters

In our implementation, the soft-embedding dimension C is set to 16 and the segment length L is set to 10. In our ST-GRU model, all 1D-convolution operators have the kernel size set to 3. We train the model by optimising the cross-entropy loss function on the labels with a learning rate of $1e^{-2}$. The model is optimised using the Adam optimiser for the first 30 epochs and fine tuned with SGD optimiser for 50 epochs with the learning rate set to $1e^{-4}$.

III. Experiments

A. Data Preparation

We evaluate our model using three spatio-temporal trajectory datasets: the Geolife dataset, the Shanghai taxi dataset and a synthetically generated dataset. We split all the datasets in a ratio of 7:1:2 to get corresponding training, validation and test datasets.

a) Geolife dataset: This is a trajectory dataset commonly used in the data mining community which was collected by Microsoft Asia in 2011 [17]. This dataset contains 17,621 trajectories of which about 8,000 have travel mode labels (4 types, bike, walk, car and bus). For this dataset, we conduct experiments on the travel mode classification task.

b) Shanghai Taxi dataset: We split the traces according to the labels of occupied and available (ones without passengers resulting in about 198k). For this dataset, we conduct evaluations on a binary classification task, i.e., correctly classify trips into having/not-having-passengers.

c) Synthetic dataset: We generate synthetic trajectories according to different speeds, acceleration, and sampling rates. We generate 4 travel modes (the same as Geolife dataset), each mode with following settings: Car (speed of 30-60 km/h, acceleration at 3m/s^2); Bus (30-60 km/h, 1m/s^2); Bike (10-15 km/h); Walk (2-5 km/h). The sampling time interval is randomly drawn from $\{1s, 2s, 5s\}$. We also randomly drop points in order to simulate GPS uncertainties.

B. Overall Evaluation

1) Baselines:

SVM & Random Forest: As proposed in [5], nine features are extracted from the raw trajectory data including speed, velocity, and direction. The extracted features are fed into SVM and Random Forest classifiers as suggested.

RNN: LSTM [18] and GRU [11] are two common variants of RNN. As mentioned in Section II-B1, we extend the traditional RNN by adopting an 1D-CNN directly onto the whole sequence to extract local sensitive features [14]. This approach is denoted by a prefix *FConv*. We compare all RNN models by feeding the spatial information (x, y) only and feeding the spatio-temporal information $(x, y, \Delta\tau)$.

Conv-LSTM: Convolutional LSTM is proposed in [9] for spatio-temporal forecasting, Conv-LSTM replaces the matrix transition with a 2D-convolution. We implement this by replacing 2D-CNN with 1D-CNN, then raw trajectories are fed into the model as we do for the RNN models,

Time-LSTM: Time-LSTM is proposed in [10] to solve the irregularity of time interval problem for recommendation tasks. We implement the first version of Time-LSTM in their series for comparison.

TABLE I: Experimental results for classification accuracy (%).

	Geolife	**Shanghai**	**Synthetic**
SVM $(x, y, \Delta\tau)$	86.11%	87.78%	86.90%
Random Forest $(x, y, \Delta\tau)$	86.89%	88.03%	87.28%
CNN $(x, y, \Delta\tau)$	87.08%	83.88%	81.47%
LSTM (x, y)	71.86%	85.53%	69.45%
LSTM $(x, y, \Delta\tau)$	88.39%	90.94%	88.50%
FConv-LSTM $(x, y, \Delta\tau)$	88.44%	90.91%	88.61%
GRU (x, y)	72.04%	85.78%	70.20%
GRU $(x, y, \Delta\tau)$	89.76%	92.03%	91.49%
FConv-GRU $(x, y, \Delta\tau)$	89.86%	91.98%	91.55%
Conv-LSTM $(x, y, \Delta\tau)$	89.85%	91.52%	91.25%
Time-LSTM $(x, y, \Delta\tau)$	83.92%	90.88%	88.78%
ST-GRU $(x, y, \Delta\tau)$	**91.25%**	**93.89%**	**93.21%**

2) Performance Comparison: We compare our ST-GRU to the baselines listed above including machine learning based approaches and RNN based methods. Table I shows a summary of experimental results. It clearly demonstrates that our ST-GRU outperforms all baselines for three datasets under study.

First, RNN based approaches outperform machine learning based approaches, i.e. SVM and Random Forest, as well as CNN. This indicates RNNs have necessary capabilities to model spatio-temporal sequential trajectory data. Random Forest $(x, y, \Delta\tau)$ seems to be the best performer among machine learning approaches under study. RNN based approaches LSTM $(x, y, \Delta\tau)$ and GRU $(x, y, \Delta\tau)$ outperform Random Forest $(x, y, \Delta\tau)$ for all three datasets.

Second, RNN based approaches with the spatio-temporal information produce better accuracies than those with the spatial information while ignoring the temporal interval information. Comparisons of LSTM/GRU (x, y) and LSTM/GRU $(x, y, \Delta\tau)$ prove that the temporal interval information $\Delta\tau$ plays a crucial role for spatio-temporal trajectory classification. Disregarding $\Delta\tau$ significantly decreases the overall performance. For example, classification accuracies for LSTM drops 16.53% on Geolife dataset, 5.41% on Shanghai dataset, and 19.05% on our synthetic dataset without the temporal interval information.

Third, by performing 1-D convolutions on the sequence before loading it into the network (i.e., the FConv version) seems to have a negligible effect. On one hand, the largest improvement is observed for FConv-LSTM $(x, y, \Delta\tau)$ over LSTM $(x, y, \Delta\tau)$ by 0.11% for our synthetic dataset, on the other hand a small decrease is observed for Shanghai dataset. This justifies our claim that the convolution operation should be computed in GRU cells for better modeling the spatial correlations which separates the spatial and temporal computations.

Fourth, Time-LSTM $(x, y, \Delta\tau)$ has a time gating mechanism for modeling the temporal interval information. However, note that their custom time gating cell is specifically designed to model long-term and short-term interests to recommend products that a user might be interested in, which is not suitable for trajectory classification. Time-LSTM achieves only 83.92% on Geolife Dataset which is inferior to LSTM $(x, y, \Delta\tau)$, and notably Time-LSTM $(x, y, \Delta\tau)$ is inferior to GRU $(x, y, \Delta\tau)$ for all three datasets. Conv-LSTM $(x, y, \Delta\tau)$ performs better than Time-LSTM $(x, y, \Delta\tau)$.

Last, ST-GRU performs the best for all three datasets. It performs 91.25% for Geolife dataset, 93.89% for Shanghai dataset, and 93.21% for our synthetic dataset. This clearly demonstrates the outstanding performance of our proposed model over all other approaches under study.

IV. Related Work

A. Travel Mode Classification

There are many existing studies in the literature dealing with the travel mode classification problem. Zheng *et al.* [5], [19], [20] proposed a few machine learning based approach to classify different travel modes based on features extracted from raw GPS logs. More specifically, nine features were extracted including distance, average velocity, speed, and heading change rate. These features were then used to build machine learning based classifiers such as SVM and Random Forest. Reddy *et al.* [21] proposed a Hidden Markov Model (HMM) based approach that works with GPS sensors as well as accelerometer sensors. Xiao *et al.* [6] proposed a tree-based ensemble classifier that combines the global features and the local features extracted from raw trajectories. However, these machine learning based approaches require a complex and time consuming feature engineering process that complicates trajectory travel mode classification.

With the recent success of deep learning in many applications, several researchers have experimented applying deep learning techniques into this field. Endo *et al.* [22] used deep learning to extract features, in particular, they converted raw trajectory data into images and extracted higher-level features from these images by concatenating features like distance and velocity, and then they built classifiers such as SVM and logistic regression. In this approach, deep learning has not really been used directly for classification but for feature engineering process. This approach still requires a complex feature extraction process, and is unsuitable for data-rich environments.

B. Deep Learning for Trajectory Modelling

With recent advances in deep neural networks, several pioneering attempts have been proposed to model trajectories with RNN approaches. Wu and Gao [15], [16] tried to model trajectories with multi-layer LSTM. However, in their study, trajectories are modelled by either a series of categorical POIs or a set of road segments which is similar to text data. Wang *et al.* [23] tried to predict travel time by learning the spatial correlations with a GeoConv layer, unlike their approach, our approach utilizes 1D-CNN for spatial feature extraction in the cell level. Zhao *et al.* [24] proposed a spatio-temporal RNN which consists of a distance gate and a time gate to model the spatio-temporal correlations of the next POI prediction task. Despite the fact we share the similar temporal gating mechanism, our work is completely different in two folds. First, we introduce a segmented convolutional weight mechanism for spatial modeling whereas they use a distance gate. Second, our temporal gate is designed for modeling various time interval information in trajectory classification, whereas they focus on the next POI location prediction.

Despite of these several attempts using RNN variants for trajectory classification, none of them is particularly designed to handle both the spatial aspect and the temporal aspect from spatio-temporal trajectories.

V. Conclusion

In this paper, we propose a ST-GRU to model spatial-temporal correlations for trajectory classification. We utilise 1D-CNN in the segmented convolutional weight mechanism to model short-term spatial correlations between neighboring locations, along with the time gating mechanism for spatio-temporal correlations within RNN cells. The experimental results demonstrate the effectiveness of ST-GRU on modeling spatial and temporal correlations, and prove the outstanding performance of our proposed model over existing state-of-the-art approaches.

References

[1] Y. Zheng, “Trajectory data mining: An overview,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 29:1–29:41, May 2015.

[2] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, “Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering,” *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 1081–1094, Aug. 2008.

[3] P. Lei, T. Shen, W. Peng, and I. Su, “Exploring spatial-temporal trajectory model for location prediction,” in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, June 2011, pp. 58–67.

[4] L. Bermingham and I. Lee, “A probabilistic stop and move classifier for noisy gps trajectories,” *Data Mining and Knowledge Discovery*, vol. 32, no. 6, pp. 1634–1662, Nov 2018.

[5] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, “Understanding transportation modes based on GPS data for web applications,” *ACM Transactions on the Web*, vol. 4, no. 1, pp. 1–36, Jan. 2010.

[6] Z. Xiao, Y. Wang, K. Fu, and F. Wu, “Identifying Different Transportation Modes from Trajectory Data Using Tree-Based Ensemble Classifiers,” *ISPRS International Journal of Geo-Information*, vol. 6, no. 3, pp. 57–22, Mar. 2017.

[7] R. Shah and R. Romijnders, “Applying Deep Learning to Basketball Trajectories,” *ArXiv e-prints*, Aug. 2016.

[8] H. Liu and I. Lee, “End-to-end trajectory transportation mode classification using bi-lstm recurrent neural network,” in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Nov 2017, pp. 1–5.

[9] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’15. Cambridge, MA, USA: MIT Press, 2015, pp. 802–810.

[10] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, “What to do next: Modeling user behaviors by time-lstm,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3602–3608.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” *arXiv e-prints*, p. arXiv:1406.1078, Jun. 2014.

[12] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 1243–1252.

[13] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, pp. 1746–1751.

[14] S. Reed, Z. Akata, H. Lee, and B. Schiele, “Learning deep representations of fine-grained visual descriptions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 49–58.

[15] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, “Identifying human mobility via trajectory embeddings,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1689–1695.

[16] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, “Modeling trajectories with recurrent neural networks,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3083–3090.

[17] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, *Geolife GPS trajectory dataset - User Guide*, July 2011.

[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[19] Y. Zheng, L. Liu, L. Wang, and X. Xie, “Learning transportation mode from raw gps data for geographic application on the web,” in *WWW 2008*. WWW 2008, April 2008.

[20] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, “Understanding mobility based on gps data,” in *Proceedings of the 10th International Conference on Ubiquitous Computing*, ser. UbiComp ’08. New York, NY, USA: ACM, 2008, pp. 312–321.

[21] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, “Using mobile phones to determine transportation modes,” *ACM Transactions on Sensor Networks*, vol. 6, no. 2, pp. 1–27, Feb. 2010.

[22] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe, “Deep feature extraction from trajectories for transportation mode estimation,” in *Advances in Knowledge Discovery and Data Mining*, J. Bailey, L. Khan, T. Washio, G. Dobbie, J. Z. Huang, and R. Wang, Eds. Cham: Springer International Publishing, 2016, pp. 54–66.

[23] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, “When will you arrive? estimating travel time based on deep neural networks,” in *AAAI Conference on Artificial Intelligence*, 2018.

[24] P. Zhao, H. Zhu, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, “Where to go next: A spatio--temporal gated network for next poi recommendation,” in *AAAI Conference on Artificial Intelligence*, 2019.