IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 16, 2019, accepted November 28, 2019, date of publication December 3, 2019,
date of current version December 13, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2957296*

# A Fast Lightweight Spatiotemporal Activity Prediction Method

**CHANGXING SHAO, LEI ZHANG, SHAOJIE ZHU, BAILONG LIU, YUN LI, AND SHUMIN CUI**
School of Computer Science, China University of Mining and Technology, Xuzhou 221116, China
Mine Digitalization Department of Engineering Research Center, Xuzhou 221116, China

Corresponding authors: Changxing Shao (shaochangxing@cumt.edu.cn) and Lei Zhang (zhanglei@cumt.edu.cn)

This work was supported by Fundamental Research Funds for Central Universities under Grant 2017XKQY078.

**ABSTRACT** How to predict spatiotemporal activity from geo-tagged social media is an urgent problem. Existing methods don't make full use of spatiotemporal information and text sequence features. In view of above problem, we design a Fast Lightweight Spatiotemporal Activity Prediction method(FLSAP) based on Gated Recurrent Unit(GRU) neural network. While GRU structure can extract text sequence features, the model takes up a lot of space due to the numerous parameters. At the same time, due to the long sequence in the text, the convergence speed of GRU is slow. So, we design a novel GRU neuron, GRU with Tiny and Skip(GTS), which can quickly generate a lightweight model with higher accuracy. In GTS, we add a scalar weighted residual connection to stabilize the training. Furthermore, we extend the residual connection to a gate by reusing the parameter matrices to compress the model size. At last, in order to make the model converge faster, we add a binary gate, which determine whether to skip the current state update. According to the experimental results, compared with ReAct [1] in the spatiotemporal activity prediction task, FLSAP improves the accuracy by 3.3%, reduces the model space by 98.79% and accelerates 74.4% of convergence speed.

**INDEX TERMS** Spatiotemporal activity prediction, accuracy, fast, lightweight.

## I. INTRODUCTION

Recently, big cities face a big challenge when people try to find their desired activities. Imagine if a tourist is in a strange city, how can he/she get information about the popular activity in his/her neighborhood at the time being quickly and accurately? Especially in the age of increasing information, even a local person can hardly answer this question accurately in a short time. However, geo-tagged social media(GTSM) has made it possible to solve this problem. Some studies [2]–[8] have demonstrated that GTSM has great potential in predicting spatiotemporal activity. GTSM includes not only timestamp and geographic coordinates, but also text generated by users using social media. Twitter is a geo-tagged social media, a large number of users use Twitter to generate a large number of messages with time and location tags every day. And these messages are adopted by studies [9]–[15] as data source. These messages contain information about main local activity. For instance, if there are many restaurants in a region, the chances of tweets related to food in this area will be much greater than areas with fewer restaurants. In addition to time and place, text plays a crucial role in the activity prediction process. So, capturing more information from the text will provide more help for activity prediction. In addition, GTSM typically relies on mobile smart terminals. Although the computing power of mobile intelligent terminals is gradually improving, how to quickly get a model that can accurately predict activity while occupying as little space as possible is still an urgent problem to be solved.

There are some researches in spatiotemporal activity prediction. USTAR [16], ReAct [1] and BranchNet [17] embedded all place, time, text and activity into same latent space to capture their correlations, and predict activity by calculating the similarity of embedding. However, they extracted semantic features of text by embedding the mean values of all word vectors, the text sequence features were missed, which play an important role in activity prediction. For example, the meaning of ''is there a restaurant'' is different from the meaning of ''there is a restaurant'', although the

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

words contained in the two sentences are the same. Since the sequence model captures the text sequence features well, Liao D [18] proposed a novel Context Aware Recurrent Neural Network(RNN) to integrate the sequential dependency and spatiotemporal activity, Jung S [19] and Jiang J Y [20] used RNN to extract text sequence features and have achieved good results. Gated RNN [21]–[24] achieved state-of-the-art performance in sequential modeling. Gated Recurrent Unit(GRU) is a kind of Gated RNN, can be a good choice for capturing the sequence features of text. However, The large number of GRU parameters resulting in larger models. Although the computing power of intelligent terminals is getting stronger and stronger, how to reduce the space of GRU model is also an problem to be solved. FastGRNN [25] addressed this limitation by adding a residual connection that does not constrain the range of the singular values explicitly and has only two extra scalar parameters. Residual connections in neural networks have been studied in some literature [26]–[29]. Furthermore, FastGRNN extended the residual connection to a gate by reusing the RNN matrices to get smaller model but with ideal accuracy. Otherwise, the processing of long text sequences in GTSM is also a challenge for GRU. SkipRNN [30] improved the performance of the GRU when dealing with long sequences, which can adaptively determine whether the state of the GRU neuron needs to be updated or copied to the next time step. SkipRNN can be seen as form of conditional computation. Conditional computation has been shown to increase model performance without significantly increasing in computational cost [31]–[35].

In this paper, we propose a Fast Lightweight Spatiotemporal Activity Prediction Method(FLSAP). The main contribution of this work are highlighted as follows:

1) FLSAP uses GRU to extract text sequence features of text and solve the text dependence of spatiotemporal activity. Among them, we use multimodal embedding as background knowledge to generate time embedding, space embedding and text embedding, which as input of GRU.
2) We add Tiny mechanism to Standard GRU to get GT(GRU with Tiny), which uses a scalar weighted residual connection for each and every coordinate of the GRU hidden state, significantly reduces the space occupancy of the model, but without losing the accuracy of the model.
3) We further add Skip mechanism to GT, design a novel GRU neuron, GTS(GRU with Tiny and Skip) for FLSAP, which can quickly generate a model that be able to accurately predict spatiotemporal activity. GTS can be encouraged to upstate state fewer by adding a penalization term during training process. GTS not only reduces the amount of computation of model training, but also speeds up the convergence of the model.

We have evaluated the performance of FLSAP. Compared with ReAct, we find that FLSAP is able to predict spatiotemporal activity faster and more accurately.

## II. RELATED WORK

With the development of network and social media, finding more activity information from GTSM can bring more convenience to our life. The recent studies [16], [36]–[38] use GTSM to predict user activity. Chong and Lim [39], Zhou *et al.* [40] and Zhou *et al.* [41] reveal the relationship between human behavior and temporal location by analyzing GTSM. S Vosoughi *et al.* [42] learns tweet embeddings by CNN-LSTM endcoder-decoder from GTSM. All of them achieve the desired results, rendering the potential of GTSM for spatiotemporal activity modeling.

Some researches have been done on the detection of hot spot activities and events in a region. GeoBurst [43] composes candidate events by detecting all the representative tweets in all tweets, and then selects truly interesting local events from the candidate events. Based on Geoburst, Geoburst+ [44] is capable of updating local event rapidly, thus continuously monitoring events. TrioVecEvent [45] first uses Bayesian mixture model to cluster geographic topics, and uses geographic clusters as candidate events to classify activities. DeLLe [46] first finds abnormal locations that aggregate unexpected numbers of tweets, and then ranks these locations to select the results. Yi Han [47] propose power-low basic and power-low advanced to detect spatiotemporal activity. Social Fusion [48] designs an unsupervised approach that can correlate event signals across multiple social networks. All the methods mentioned above regard tweets as a collection of keywords, without considering the order between words. In addition, the keywords of twitter and main activity in a region are close rather than equal. So it will be more convincing to use semantically sequential texts and spatiotemporal information to predict activity.

As data grows more and more, some useless text data will appear, resulting in text data sparsity. Gmove [49] has the function of text enhancer, which calculates keyword correlation by checking its temporal and spatial distribution. CrossMap [50] can effectively solve the problem of sparse data by detecting spatiotemporal hotspots in people's activities. In addition, BranchNet [17] uses a graphical regularized cross-modal embedding framework to allow external knowledge to guide the cross-modal embedding process in order to learn better spatiotemporal activity models. Zhao *et al.* [51] proposes a novel feature learning framework that solved the problem of sparse social media data by formulating prediction tasks for different locations with different spatial resolutions, allowing the heterogeneous relationships among the tasks to be characterized. By collecting linked data from Twitter and Foursquare, we can not only improve the quality of data, but also obtain complete information of spatiotemporal activity categories marked in Foursquare to guide model training. Splitter [52] excavates fine grained sequential patterns in semantic trajectories. STREAMCUBE [53] can explore spatiotemporal activity with different spatial granularity, Both Chen *et al.* [54] and ReAct [1] consider the continuity of time and space in the prediction of spatiotemporal activity. Although they fully develop the value

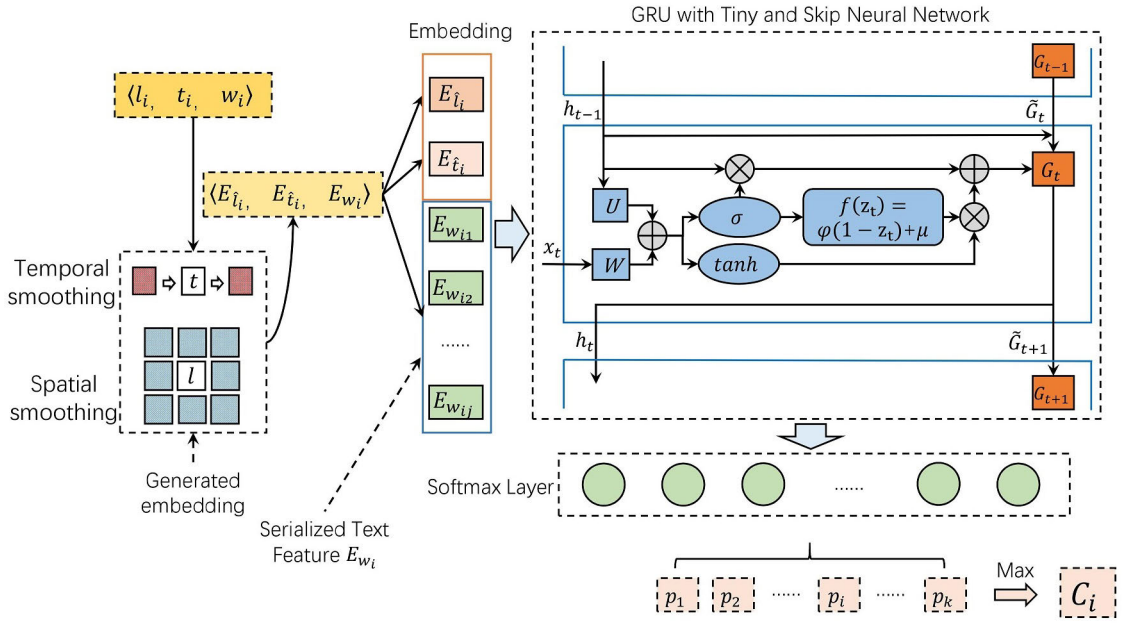



**FIGURE 1.** The overall framework of FLSAP.

of time and space, they do not make full use of text information.

In order to make better use of text information in predicting spatiotemporal activity, some studies use RNN to extract text features. Cui *et al.* [55] propose a hybrid LSTM model for rich contextual learning to recognize user activity not only for the cases where a clear indicator exists in the content, but also for the ones where the activity information is latent. As the same, capturing more information from text provide a great help for Xu *et al.* [56]. However, due to numerous RNN parameters and long sequences in data, the disk storage occupied by the above models will increase and the convergence speed of models will also slow down.

Distinguished from these studies, our goal is to extract text sequential information from tweets and combine it with time and space to predict activity categories, quickly generate model that can accurately predict spatiotemporal activity, rendering our method different from existing methods.

## III. PROBLEM DEFINITION

Spatiotemporal activity prediction uses the latitude, longitude, timestamp and text of the location to be predicted as the input of the neural network to predict what social activities are mainly carried out at that moment.

### A. INITIAL DATA

$T = \{T_1, T_2, \ldots, T_i, \ldots, T_n\}$ is all input data. Each input data $T_i$ is a tuple $\langle l_i, t_i, w_i \rangle$. $l_i$ is a two-dimensional vector representing latitude and longitude. $t_i$ is the timestamp of $T_i$. $w_i$ is a sentence composed of some words.

### B. ACTIVITY CATEGORY

The predicted activity category is the main activity that the model predicts at a specific time and place through text sequence embedding, time embedding, and spatial embedding.

## IV. THE FRAMEWORK

The overall framework of FLSAP is shown in FIGURE1.

$\langle l_i, t_i, w_i \rangle$ is the initial location, time and text data. We embed all the regions, hours and text into same latent space to generate embedding. $\langle E_{\hat{l}_i}, E_{\hat{t}_i}, E_{w_i} \rangle$ is the embedding of $\langle l_i, t_i, w_i \rangle$, which can be a good representation of the continuity of the proximity time and the continuity between adjacent locations.

### A. SPATIOTEMPORAL SMOOTHING

Since the location of each piece of data is independent of each other, we smooth the space embedding in the process of getting embedding, as shown in equation(1). $E_l$ represents the initial embedding at location $l$, $\varepsilon$ is the spatial smoothing constant, and $S_l$ is the set of locations around point $l$. Similarly, in equation (2), $E_t$ represents the initial embedding at time $t$, $\theta$ is the time smoothing constant, and $S_t$ is the set of neighboring time points before and after the time point $t$. After getting embedding, we put these sequential embedding into GRU model.

$$E_{\hat{l}} = \frac{E_l + \varepsilon \sum_{l_n \in S_l} E_{l_n}}{\varepsilon |S_l| + 1} \tag{1}$$

$$E_{\hat{t}} = \frac{E_t + \theta \sum_{t_n \in S_t} E_{t_n}}{\theta |S_t| + 1} \tag{2}$$

### B. CAPTURE TEXT SEQUENCE FEATURES

In FIGURE1, text suquence features $E_{w_i} = \{E_{w_{i1}}, E_{w_{i2}}, \ldots, E_{w_{id}}, \ldots, E_{w_{ij}}\}$ are captured by GRU. $i$ represents the $i$-th data and $d$ represents the embedding of the $d$-th word of the data. The structural design of GRU allows the input of previous

neurons to affect the output of the current neuron, thereby capturing the sequence features of the text.

### C. LIGHTWEIGHT MODEL AND FAST TRAINING

We add Tiny mechanism and Skip mechanism based on the GRU, design a novel neuron named GTS in FIGURE1. Tiny mechanism stabilize the training process by using a scalar weighted residual connection for each coordinate of state $h_t$ of GRU, and further compress the model size by using a low-rank and a sparse representation of the parameter matrices $W, U$. Skip mechanism determines whether to skip the state update of the current neuron by adding a binary gate to the neuron, thus speeding up the convergence of the model.

In next section, we describe the structure and calculation method of GTS. As aforementioned, we compress the model size without losing the accuracy of the model as much as possible, and then add a binary gate to the neuron to skip unnecessary state updates, thus speeding up the convergence of the model.

## V. FAST LIGHTWEIGHT SPATIOTEMPORAL ACTIVITY PREDICTION

### A. LIGHTWEGHT MODEL WITH TINY MECHANISM

During the prediction process, GRU has a high hardware requirement due to the large number of parameters and data. Although the computing power of smart terminals is constantly improving, the small smart terminals are still difficult to have strong computing power. This makes it difficult to predict spatiotemporal activity on small smart terminals. How to reduce the GRU model is still an urgent problem to be solved. In order to reduce the model without losing prediction accuracy as much as possible, we did the following work.

#### 1) WEIGHTED RESIDUAL CONNECTION:

GT(GRU with Tiny) uses a weighted residual connection to stabilize training by generating well-conditioned gradients. GT's $h_t$ update rule is shown in formula (3), (4) and (5). $\varphi$ and $\mu$ are the trainable parameters parameterized by the sigmoid function. In theory, when $\mu \approx 0, \varphi \approx 1$, GT will obtain ideal results in processing long sequence. GT inputs $x_t$ and $h_{t-1}$ into a nonlinear function to obtain the gated coordinates $z_t$.

$$z_t = \sigma(Wx_t + Uh_{t-1} + b_z) \tag{3}$$

$$\tilde{h}_t = tanh(Wx_t + Uh_{t-1} + b_h) \tag{4}$$

$$h_t = (\varphi(1 - z_t) + \mu) * \tilde{h}_t + z_t * h_{t-1} \tag{5}$$

This paragraph shows how GT generates well-conditioned gradients. Let $L(X, y; \Gamma)$ be the loss function for the labeled data $(X, y)$ and given parameters $\Gamma = (W, U, b)$. let $\alpha = \varphi(1-z_t)+\mu$ and $\beta = z_t$. The gradient of $L$ w.r.t $W, U, b$ is as formula (6), (7) and (8), where $\nabla_{h_T} L = -c(\Gamma) \cdot y \cdot b$, and $c(\Gamma) = \dfrac{1}{1 + exp(y * exp(-y \cdot b^T h_T))}$. Let $M(U) = \prod_{k=t}^{T-1}(\alpha U^T D_{k+1} + \beta I)$. Then, the condition number $\kappa_{M(U)}$ is defined by formula (9), where $D_k = diag(\sigma'(Wx_k + Uh_{k-1} + b))$ is the Jacobian matrix of the pointwise nonlinearity. if $\alpha = 1$ and $\beta = 0$, which is same as RNN, the condition number of $M(U)$ can be as large as $(max_k \dfrac{\| U^T D_{k+1} \|}{\varpi_{min}(\| U^T D_{k+1} \|)})^{T-t}$. $\varpi_{min}(\psi)$ represents the minimum singular value of $\psi$, so condition number of gradient can be exponential in $T$, leading to unstable training. Different from above architechture, if $\alpha = 0$ and $\beta = 1$, $\kappa_{M(U)}$ is bounded by a small term. For instance, if $\beta = 1 - \alpha$ and $\alpha = \dfrac{1}{T \| U^T D_{k+1} \|}$, then $\kappa_{M(U)} = O(1)$. Therefore, residual connection is able to control the $\kappa_{M(U)}$ and stabilize training.

$$\frac{\partial L}{\partial W} = \alpha \sum_{t=0}^{T} D_t \Big(\prod_{k=t}^{T-1} (\alpha U^T D_{k+1} + \beta I)\Big)(\nabla_{h_T} L) {x_t}^T \tag{6}$$

$$\frac{\partial L}{\partial U} = \alpha \sum_{t=0}^{T} D_t \Big(\prod_{k=t}^{T-1} (\alpha U^T D_{k+1} + \beta I)\Big)(\nabla_{h_T} L) {h_{t-1}}^T \tag{7}$$

$$\frac{\partial L}{\partial b} = \frac{-y * exp(-y \cdot b^T h_T)}{1 + exp(-y \cdot b^T h_T)} h_T \tag{8}$$

$$\kappa_{M(U)} \leq \frac{(1 + \dfrac{\alpha}{\beta} max_k \| U^T D_{k+1} \|)^{T-t}}{(1 - \dfrac{\alpha}{\beta} max_k \| U^T D_{k+1} \|)^{T-t}} \tag{9}$$

To minimize the number of parameters, GT reuses matrix $W$ and matrix $U$ for the vector-valued gating function. Therefore, the computational complexity of GT is almost the same as that of the standard RNN, but its accuracy and training stability are comparable to GRU. And GT further compresses the model size by using the parameter matrix $W$ and the sparse representation of the matrix $U$.

#### 2) COMPRESS PARAMETER MATRICES

To minimize the number of parameters, GT further compresses the model size by using a low-rank and a sparse representation of the parameter matrices $W, U$, as shown in equation (10).

$$W = \delta^1 (\delta^2)^T, U = \vartheta^1 (\vartheta^2)^T \tag{10}$$

where $\delta^1 \epsilon R^{\hat{D} \times \kappa_w}$, $\delta^2 \epsilon R^{D \times \kappa_w}$, $\vartheta^1$, $\vartheta^2 \epsilon R^{\hat{D} \times \kappa_u}$, $||\delta^i||_0 \leq {\rho_w}^i$, $||\vartheta^i||_0 \leq {\rho_u}^i, i = \{1, 2\}$, among them, $\kappa_w, \rho_w, \kappa_u, \rho_u$are hyper parameters. The parameters of GT include $\Phi_{GT} = \{\delta^i, \vartheta^i, b_h, b_z, \varphi, \mu\}$. If the loss function of the model is $L$, the optimization strategy of the model parameters is as shown in equation (11).

$$\min_{\Phi_{GT}, ||\delta^i||_0 \leq {\rho_w}^i, ||\vartheta^i||_0 \leq {\rho_u}^i} \partial(\Phi_{GT}) = \frac{1}{n} \sum_{j=1}^{n} L(x_j, y_j; \Phi_{GT}) \tag{11}$$

In the model training process, the parameters in equation (11) are optimized by batch stochastic gradient descent(b-SGD). This stage mainly learns the low rank representation of the parameters and does not sparsely constrain the parameters. After the above stages are completed, we map the parameters of the model to the sparse low rank matrix to obtain a model with better performance.

The GT compresses the parameters by quantifying the parameters $\vartheta^i$ and $\delta^i$, so that these parameters occupy only a small storage space and have a byte index of the sparse model. However, the compression process of the parameters is not an equivalent operation but an approximate operation, the accuracy of the model is reduced to some extent. In addition, although this method reduces the size of the model, the training time of the model maybe longer because the nonlinear function requires all hidden states to be floating point. GT overcomes these two shortcomings by staged the nonlinear functions and treats the functions in a linear manner in each sub-phase, thus ensuring that all calculations can be performed by integer operations. GT replaces the nonlinear function in equation (5) with an approximate linear function, and then obtains the parameter $\Phi_{GT}$ through the above training process. GT synthesizes the floating-point representation of the quantization parameters so that the model training can not significantly reduce the model accuracy under the conditions of integer arithmetic.

### B. FAST TRAINING WITH SKIP MECHANISM

GT makes the model lighter. However, due to the long textual suquences in data and the parameter compression in GT, the convergence speed of the model is not satisfactory. We add Skip mechanism to GT. GTS(GT with Skip) adaptively determine whether a state needs to be updated or copied from the last time step. We add penalty items to the model to make the network reduce state updates as much as possible, which allows the model to still train properly when the calculation conditions are not ideal. This is done by the binary gate $G_t \epsilon \{0, 1\}$. The calculation method is as follows.

$$G_t = F_{binarize}(\tilde{G}_t) \tag{12}$$

$$h_t = G_t * ((\varphi(1 - z_t) + \mu) * \tilde{h}_t + z_t * h_{t-1}) + (1 - G_t) * h_{t-1} \tag{13}$$

$$\triangle\tilde{G}_t = \sigma(W * h_t + b) \tag{14}$$

$$\tilde{G}_{t+1} = G_t * \triangle\tilde{G}_t + (1 - G_t) * (\tilde{G}_t + \min(\triangle\tilde{G}_t, 1 - \tilde{G}_t)) \tag{15}$$

The GTS structure is shown in FIGURE2. $\sigma$ is sigmoid function. $F_{binarize}$ converts a floating point number between 0 and 1 to 0 or 1. When $G_t = 1$, GT updates the neuron state according to the original rule. When $G_t = 0$, GT replicates the state of the previous neuron. So how to define $F_{binarize}$ becomes a crucial issue. Here we define $F_{binarize}$ as the rounding function round, when $\tilde{G}_t \epsilon [0, 0.5)$, $F_{binarize} = 0$; when $\tilde{G}_t \epsilon [0.5, 1]$, $F_{binarize} = 1$.

Equations (14) and (15) increase the likelihood of neuronal status updates as the number of consecutively state-updated neurons increases. When the state of a certain neuron is updated, the probability of the next neuron state update becomes $\tilde{G}_{t+1} = \triangle\tilde{G}_t$.

There are several advantages to reducing the number of updates to neurons. From a computational point of view, fewer state updates can reduce the amount of computation and make the computation faster. During model training, the gradient propagates through fewer update time steps, and convergence can be achieved more quickly in some tasks involving long sequences.

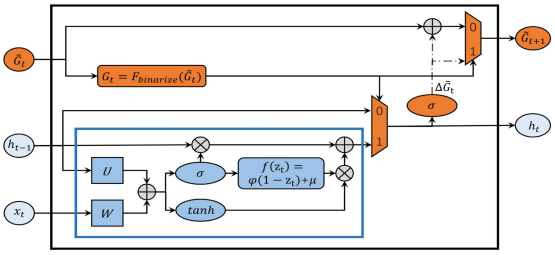



**FIGURE 2.** GTS structure diagram.

In order to make the model run faster on machines with lower computing power, we add an additional loss function $L_{add}$ to the model. $L_{add}$ is defined by equation (16). Where $\lambda$ is the learning rate of $L_{add}$ and $LEN$ represents the length of the entire sequence. Therefore, the final loss function is cross entropy plus $L_{add}$.

$$L_{add} = \lambda * \sum_{t=1}^{LEN} G_t \tag{16}$$

### C. ALGORITHM DESCRIPTION

As aforementioned, we smooth the time embedding and space embedding to obtain the association between adjacent time and place, and merge the text embedding with the spatiotemporal embedding into sequence data as the input of the GRU to capture the text sequence features of the data. Then we design a novel GRU neuron named GTS, which speeds up the convergence of the training while reducing the model space. Algorithm 1 sketches the procedure of FLSAP.

## VI. EXPERIMENT AND ANALYSIS

### A. EXPERIMENTAL SETTING

#### 1) DATASET

Link data of Twitter[a] and Foursquare[b] in Los Angeles and New York from 2014.08.01 to 2014.11.30. Many users link their Foursquare account to their Twitter account and leave check-in information in different geographic locations. There are 158,835 New York data and 143,567 Los Angeles data. Datasets in New York and Los Angeles are randomly divided into training set and test set in a 4:1 ratio.

#### 2) ENVIRONMENT

The experimental program was written in Python 2.7 and tensorflow 1.12.0. CPU: 6 core, Intel (R) Xeon (R) Gold 5117 CPU @ 2.00GHz; Memory 24GB; GPU: Tesla P100, Memory 16GB.

[a]https://dev.twitter.com/streaming/overview
[b]https://developer.foursquare.com

**Algorithm 1** Detail of Fast Lightweight Spatiotemporal Activity Prediction

**Input:** Initial embedding $E = \{E_1, E_2, \ldots, E_i, \ldots, E_n\}$ of initial data $T = \{T_1, T_2, \ldots, T_i, \ldots, T_n\}$

**Output:** Activity category $C = \{C_1, C_2, \ldots, C_i, \ldots, C_n\}$

Initialize $\varepsilon, \theta, \varphi, \mu, \lambda$; $W$ and $U$ of GTS
//generate embedding $E_i$ of initial data $T_i$
**for** $T_i$ in $T$ **do**
  $\langle E_{l_i}, E_{t_i}, E_{w_i}\rangle \longleftarrow \langle l_i, t_i, w_i\rangle$ //generate time embedding, space embedding and text embedding
  $E_{\hat{l}} = (E_l + \varepsilon \sum_{l_n \in S_l} E_{l_n})/(\varepsilon|S_l| + 1)$ //spatial smoothing
  $E_{\hat{t}} = (E_t + \theta \sum_{t_n \in S_t} E_{t_n})/(\theta|S_t| + 1)$ //temporal smoothing
  $E \Leftarrow E_i = \langle E_{\hat{l}_i}, E_{\hat{t}_i}, E_{w_i}\rangle$ //get embedding $E_i$
**end for**
//optimize parameters of the model and predict activity category $C$
**for** $E_i$ in $E$ **do**
  //put sequence $E_i$ into GTS
  **for** $t$ **in** GTS **do**
    $G_t = F_{binarize}(\tilde{G}_t)$ //calculate whether $h_t$ needs to be updated
    $h_t = G_t * ((\varphi(1 - z_t) + \mu) * \tilde{h}_t + z_t * h_{t-1}) + (1 - G_t) * h_{t-1}$ //if $G_t == 0$, $h_t = h_{t-1}$; else, update $h_t$
    $min\partial(\Phi_{GT}) = \frac{1}{n}\sum_{j=1}^{n} L(x_j, y_j; \Phi_{GT})$ //optimize parameters of the model, $L$ is the loss function
    $\triangle\tilde{G}_t = \sigma(W * h_t + b)$
    $\tilde{G}_{t+1} = G_t * \triangle\tilde{G}_t + (1 - G_t) * (\tilde{G}_t + \min(\triangle\tilde{G}_t, 1 - \tilde{G}_t))$ //calculate the value of $\tilde{G}_{t+1}$ for next GTS neuron
  **end for**
  $C \Leftarrow C_i$ //get predicted activity category $C_i$
**end for**
**return** $C$

#### 3) PARAMETERS SETTING

In above experiments, Learning Rate=0.0001, Batch Size=300, $\varepsilon = \theta = 0.1$. The neural network is set up as a three-layer GRU with a fully connected layer, and each layer of the GRU contains 128 neurons. For the second set of experiments, $\varphi = 0.95$, $\mu = 0.3$. For the third set of experiments, $\lambda = 0.0001$.

In order to evaluate the performance of FLSAP in the spatiotemporal activity prediction task, we set up three experiments as follows.

(1) First experiment compares the accuracy of ReAct, FLSAP-GE(GRU with discontinuous embedding) and FLSAP-GCE(GRU with continuous embedding).

(2) Second experiment compares the accuracy and model size of FLSAP-G(GRU), FLSAP-GT(GRU with Tiny) and ReAct.

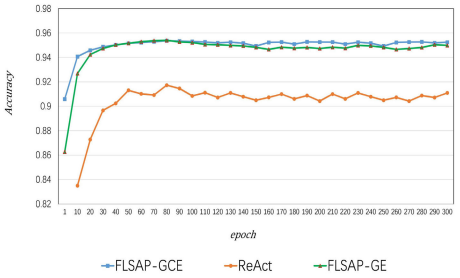



**FIGURE 3.** Accuracy comparison of Los Angeles data set.

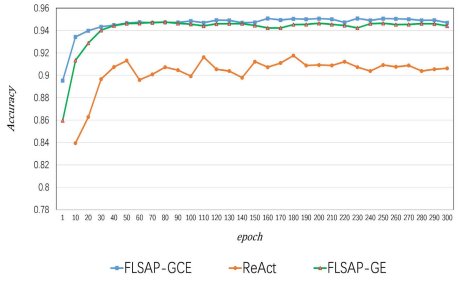



**FIGURE 4.** Accuracy comparison of New York data set.

(3) Third experiment compares the convergence speed and *loss* of FLSAP-GT, FLSAP-GTS(GRU with Tiny and Skip) and ReAct.

### B. METRICS AND ANALYSIS OF RESULTS

We set three metrics to evaluate the results of the experiment. The metrics are as follows.

(1) *Accuracy*: Assume that $m$ in the $n$ data can be accurately predicted($m \leq n$), *Accuracy* is as shown in equation (17). $p_i$ represents the softmax output value corresponding to the true activity of the $i$th data.

(2) *ModelSize*: The size of the disk occupied by the model.

(3) $\triangle Convergence$: Defined by equation (18). We select $s$ *Accuracy* thresholds, $ep_{A_i}$ is the epoch used by method $A$ to achieve $Accuracy_i$, $ep_{B_i}$ is the epoch used by method $B$ to achieve $Accuracy_i$.

$$Accuracy = \frac{\sum_{i=1}^{m} p_i}{n} \tag{17}$$

$$\triangle Convergence = \frac{\sum_{i=1}^{s} \frac{ep_{A_i} - ep_{B_i}}{ep_{A_i}}}{s} \tag{18}$$

(4) *loss*: The value of loss function, which is described in section *V.B*. Defined by equation (19), where $n$ represents the total number of categories, $y_i$ represents the true classification result, and $\ln \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}}$ represents the $i$th output value of softmax.

$$loss = -\sum_{i=1}^{n} y_i \ln \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}} + L_{add} \tag{19}$$

Results of the first experiment are shown in FIGURE3 and FIGURE4. For Los Angeles data set, the *Accuracy* of the ReAct is best at 91.8%, while the performance of

**TABLE 1. Accuracy and ModelSize comparison of Los Angeles data set.**

| method | *Accuracy* | *ModelSize* |
|---|---|---|
| FLSAP-GT | 95.4% | 1.0Mb |
| FLSAP-G | 95.2% | 3.1Mb |
| ReAct | 91.8% | 82.6Mb |

**TABLE 2. Accuracy and ModelSize comparison of New York data set.**

| method | *Accuracy* | *ModelSize* |
|---|---|---|
| FLSAP-GT | 95.0% | 1.0Mb |
| FLSAP-G | 95.1% | 3.1Mb |
| ReAct | 91.9% | 82.6Mb |

FLSAP-GCE is 95.2%. *Accuracy* increased by 3.4%. And for New York data set, the *Accuracy* of the ReAct is best at 91.9%, while the performance of FLSAP-GCE is 95.1%. *Accuracy* increased by 3.2%. *Accuracy* increased by an average of 3.3%. Compared with ReAct's activity prediction method, which ranks the similarity of input embedding, GRU can better capture the relationship among time, place, text and activity category. In addition, the convergence speed and accuracy of FLSAP-GCE are slightly higher than FLSAP-GE. Because continuous embedding can extract the connection between neighboring data. In the next two sets of experiments, we all used continuous embedding as input.

Results of the second experiment are shown in TABLE 1 and TABLE 2. There is no loss in the *Accuracy* of FLSAP-GT, but the model space is reduced by 98.79% than ReAct. Compared with FLSAP-G, the model space is reduced by 67.7%. Because FLSAP-GT compresses the parameters of the model by adding a residual connection that does not constrain the range of the singular values, and extended the residual connection to a gate by reusing the GRU matrices to match state-of-the-art gated RNN accuracies but with smaller model. Smaller model can increase loading speed and save device space. This makes it possible for the prediction of spatiotemporal activity to run smoothly in a more rigorous environment.

Results of the third experiment are shown in FIGURE5, FIGURE6, FIGURE7 and FIGURE8. To achieve the same *Accuracy*, FLSAP-GTS takes less steps than ReAct and FLSAP-GT. For Los Angelas data set, available by calculation, FLSAP-GTS compared with ReAct, $\triangle Convergence = 76.2\%$, and compared with FLSAP-GT, $\triangle Convergence = 29.08\%$. For New York data set, FLSAP-GTS compared with ReAct, $\triangle Convergence = 72.59\%$, and compared with FLSAP-GT, $\triangle Convergence = 18.46\%$. As can be seen from FIGURE7 and FIGURE8, for the two data sets, the *loss* of FLSAP-GTS is significantly lower than the *loss* of FLSAP-GT after the same epoch. That is to say, the *Accuracy* of FLSAP-GTS is significantly higher than that of FLSAP-GT after the same number of epoch, which proves that FLSAP-GTS performs better in terms of convergence speed. FLSAP-GTS improved convergence speed. Skip mechanism allows GRU to skip over redundant parts in long sequences,

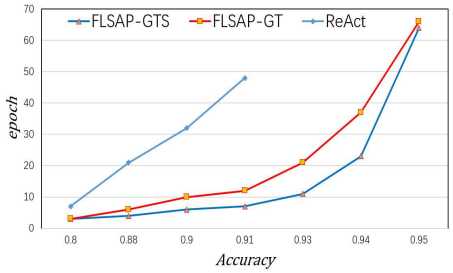



**FIGURE 5. Epoch comparison of Los Angeles data set.**

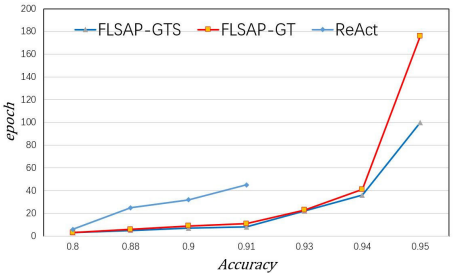



**FIGURE 6. Epoch comparison of New York data set.**

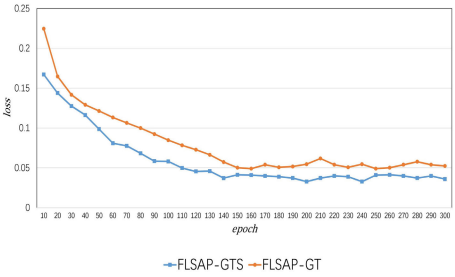



**FIGURE 7. Loss comparison of Los Angeles data set.**

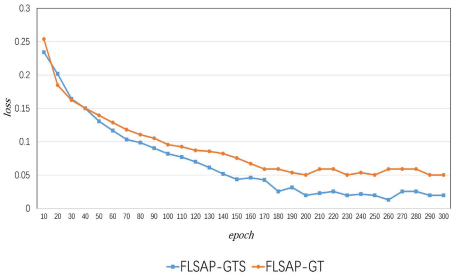



**FIGURE 8. Loss comparison of New York data set.**

which can reduce the impact of low-quality data on model accuracy and make the model converge faster.

## VII. CONCLUSION AND FUTURE WORK

We presents A Fast Lightweight Spatiotemporal Activity Prediction Method(FLSAP) which can accurately and quickly predict activity categories through GTSM and a novel GRU neuron called GTS. FLSAP captures the features of text sequence with GRU. Tiny mechanism stabilizes training by adding simple weighted residual connection, and extends the residual connection to a gate by reusing the RNN matrices to get smaller model but with ideal accuracy. Skip mechanism uses a binary gate to determine whether the current neuron

status is updated. Experiments show that FLSAP outperforms ReAct in spatiotemporal activity prediction.

In future work, we can take into account the differences between users when we predict spatio-temporal activity. In addition, the relationship between points can also provide more help for the spatiotemporal activitiy prediction.

## REFERENCES

[1] C. Zhang, K. Zhang, Q. Yuan, F. Tao, L. Zhang, T. Hanratty, and J. Han, "ReAct: Online multimodal embedding for recency-aware spatiotemporal activity modeling," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 245–254.

[2] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis, "Discovering geographical topics in the Twitter stream," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 769–778.

[3] C. C. Kling, J. Kunegis, S. Sizov, and S. Staab, "Detecting non-Gaussian geographical topics in tagged photo collections," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 603–612.

[4] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in foursquare," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, 2011, pp. 570–573.

[5] S. Sizov, "GeoFolk: Latent spatial semantics in Web 2.0 social media," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 281–290.

[6] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical topic discovery and comparison," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 247–256.

[7] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann, "Who, where, when and what: Discover spatio-temporal topics for Twitter users," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 605–613.

[8] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: A survey," 2019, *arXiv:1906.04928*. [Online]. Available: https://arxiv.org/abs/1906.04928

[9] X. Zheng, J. Han, and A. Sun, "A survey of location prediction on Twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1652–1671, Sep. 2018.

[10] A. Kumar and J. P. Singh, "Location reference identification from tweets during emergencies: A deep learning approach," *Int. J. Disaster Risk Reduction*, vol. 33, pp. 365–375, Feb. 2019.

[11] W. Shen, Y. Liu, and J. Wang, "Predicting named entity location using Twitter," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Apr. 2018, pp. 161–172.

[12] P. Li, H. Lu, N. Kanhabua, S. Zhao, and G. Pan, "Location inference for non-geotagged tweets in user timelines," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1150–1165, Jun. 2018.

[13] H. Lu, W. Niu, and J. Caverlee, "Learning geo-social user topical profiles with Bayesian hierarchical user factorization," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 205–214.

[14] M. Ebrahimi, E. ShafieiBavani, R. Wong, and F. Chen, "A unified neural network model for geolocating Twitter users," in *Proc. 22nd Conf. Comput. Natural Lang. Learn.*, 2018, pp. 42–53.

[15] M. Ebrahimi, E. ShafieiBavani, R. K. Wong, and F. Chen, "Twitter user geolocation by filtering of highly mentioned users," *J. Assoc. Inf. Sci. Technol.*, vol. 69, no. 7, pp. 879–889, Jul. 2018.

[16] A. Silva, S. Karunasekera, C. Leckie, and L. Luo, "USTAR: Online multimodal embedding for modeling user-guided spatiotemporal activity," 2019, *arXiv:1910.10335*. [Online]. Available: https://arxiv.org/abs/1910.10335

[17] C. Zhang, M. Liu, Z. Liu, C. Yang, L. Zhang, and J. Han, "Spatiotemporal activity modeling under data scarcity: A graph-regularized cross-modal embedding approach," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 531–538.

[18] D. Liao, W. Liu, Y. Zhong, J. Li, and G. Wang, "Predicting activity and location with multi-task context aware recurrent neural network," in *Proc. IJCAI*, 2018, pp. 3435–3441.

[19] S. Jung, "Semantic vector learning for natural language understanding," *Comput. Speech Lang.*, vol. 56, pp. 130–145, Jul. 2019.

[20] J.-Y. Jiang, M. Zhang, C. Li, M. Bendersky, N. Golbandi, and M. Najork, "Semantic text matching for long-form documents," in *Proc. World Wide Web Conf.*, 2019, pp. 795–806.

[21] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: https://arxiv.org/abs/1409.1259

[22] J. Collins, J. Sohl-Dickstein, and D. Sussillo, "Capacity and trainability in recurrent neural networks," 2016, *arXiv:1611.09913*. [Online]. Available: https://arxiv.org/abs/1611.09913

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

[24] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljacic, and Y. Bengio, "Gated orthogonal recurrent units: On learning to forget," *Neural Comput.*, vol. 31, no. 4, pp. 765–783, 2019.

[25] A. Kusupati, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma, "FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9017–9028.

[26] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8624–8628.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[28] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.*, vol. 20, no. 3, pp. 335–352, Apr. 2007.

[29] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, *arXiv:1505.00387*. [Online]. Available: https://arxiv.org/abs/1505.00387

[30] V. Campos, B. Jou, X. Giró-I-Nieto, J. Torres, and S.-F. Chang, "Skip RNN: Learning to skip state updates in recurrent neural networks," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, Apr./May 2018, pp. 1–17.

[31] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle, and A. Courville, "Dynamic capacity networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2549–2558.

[32] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*. [Online]. Available: https://arxiv.org/abs/1308.3432

[33] L. Liu and J. Deng, "Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3675–3682.

[34] M. McGill and P. Perona, "Deciding how to decide: Dynamic routing in artificial neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2363–2372.

[35] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017, *arXiv:1701.06538*. [Online]. Available: https://arxiv.org/abs/1701.06538

[36] G. Liao, S. Jiang, Z. Zhou, C. Wan, and X. Liu, "Poi recommendation of location-based social networks using tensor factorization," in *Proc. 19th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2018, pp. 116–124.

[37] J. Jiang, C. Wang, Y. Tian, S. Zhang, and Y. Zhao, "Urban activity summarization with geo-tagged social media data," in *Proc. 4th Int. Conf. Comput. Technol. Appl. (ICCTA)*, May 2018, pp. 16–19.

[38] D. Li, Z. Gong, and D. Zhang, "A common topic transfer learning model for crossing city POI recommendations," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4282–4295, Dec. 2019.

[39] W.-H. Chong and E.-P. Lim, "Fine-grained geolocation of tweets in temporal proximity," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, 2019, Art. no. 17.

[40] Y. Zhou, B. P. L. Lau, C. Yuen, B. Tunçer, and E. Wilhelm, "Understanding urban human mobility through crowdsensed data," 2018, *arXiv:1805.00628*. [Online]. Available: https://arxiv.org/abs/1805.00628

[41] F. Zhou, Q. Gao, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Trajectory-user linking via variational AutoEncoder," in *Proc. IJCAI*, 2018, pp. 3212–3218.

[42] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, pp. 1041–1044.

[43] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han, "GeoBurst: Real-time local event detection in geo-tagged tweet streams," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, pp. 513–522.

[44] C. Zhang, D. Lei, Q. Yuan, H. Zhuang, L. Kaplan, S. Wang, and J. Han, "GeoBurst+: Effective and real-time local event detection in geo-tagged tweet streams," *ACM Tran. Intell. Sys. Tech.*, vol. 9, no. 3, p. 34, 2018.

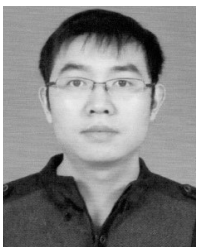
[45] C. Zhang, L. Liu, D. Lei, Q. Yuan, H. Zhuang, T. Hanratty, and J. Han, "TrioVecEvent: Embedding-based online local event detection in geo-tagged tweet streams," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 595–604.
[46] H. Wei, H. Zhou, J. Sankaranarayanan, S. Sengupta, and H. Samet, "Detecting latest local events from geotagged tweet streams," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2018, pp. 520–523.
[47] Y. Han, S. Karunasekera, C. Leckie, and A. Harwood, "Multi-spatial scale event detection from geo-tagged tweet streams via power-law verification," 2019, *arXiv:1906.05063*. [Online]. Available: https://arxiv.org/abs/1906.05063
[48] P. Giridhar, S. Wang, T. Abdelzaher, T. Al Amin, and L. Kaplan, "Social fusion: Integrating Twitter and instagram for event monitoring," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2017, pp. 1–10.
[49] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "GMove: Group-level mobility modeling using geo-tagged social media," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1305–1314.
[50] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, and J. Han, "Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 361–370.
[51] L. Zhao, J. Wang, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Spatial event forecasting in social media with geographically hierarchical regularization," *Proc. IEEE*, vol. 105, no. 10, pp. 1953–1970, Oct. 2017.
[52] C. Zhang, J. Han, L. Shou, J. Lu, and T. La Porta, "Splitter: Mining fine-grained sequential patterns in semantic trajectories," *Proc. VLDB Endowment*, vol. 7, no. 9, pp. 769–780, 2014.
[53] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang, "STREAMCUBE: Hierarchical spatio-temporal hashtag clustering for event exploration over the Twitter stream," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 1561–1572.
[54] J. Chen, N. Gao, C. Xue, C. Tu, and D. Zha, "Perceiving topic bubbles: Local topic detection in spatio-temporal tweet stream," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2019, pp. 730–747.
[55] R. Cui, G. Agrawal, and R. Ramnath, "Tweets can tell: Activity recognition using hybrid long short-term memory model," 2019, *arXiv:1908.02551*. [Online]. Available: https://arxiv.org/abs/1908.02551
[56] H. Xu, P. Wu, J. Wei, Z. Yang, and J. Wang, "A meta-path-based recurrent model for next POI prediction with spatial and temporal contexts," in *Proc. Asia–Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data*. Springer, 2019, pp. 219–235.

**CHANGXING SHAO** received the B.E. degree in computer science and technology from the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China, in 2018, where he is currently pursuing the M.E. degree with the Department of Computer Application Technology. He is also working with the Mine Digitalization Department of Engineering Research Center, Xuzhou. His research interests include spatiotemporal activity prediction and trajectory prediction.

**LEI ZHANG** was born in China, in 1977. He received the Ph.D. degree in computer application technology from Nanjing University of Aeronautics and Astronautics, in 2006.

Since 2008, he has been an Associate Professor with the School of Computer Science and Technology, China University of Mining and Technology. He is currently working with the Mine Digitalization Department of Engineering Research Center, Xuzhou, China. His research interests include spatiotemporal data mining and trajectory data analysis.

**SHAOJIE ZHU** received the B.E. degree in computer science and technology from the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China, in 2018, where he is currently pursuing the M.E. degree with the Department of Computer Application Technology. He is also working with the Mine Digitalization Department of Engineering Research Center, Xuzhou. His research interests include spatiotemporal activity prediction and trajectory prediction.



**BAILONG LIU** was born in China, in 1983. He received the Ph.D. degree in computer application technology from Harbin Engineering University, in 2009.

Since 2011, he has been a Lecturer with the School of Computer Science and Technology, China University of Mining and Technology. He is currently working with the Mine Digitalization Department of Engineering Research Center, Xuzhou, China. His research interests include spatiotemporal data mining and trajectory data analysis.

**YUN LI** received the B.E. degree in computer science and technology from the School of Computer and Communication Engineering, Liaoning University of Petroleum and Chemical Technology, Fushun, China, in 2017, where she is currently pursuing the M.A. degree with the Department of Computer Application Technology. She is also working with the Mine Digitalization Department of Engineering Research Center, Xuzhou, China. Her research interests include spatiotemporal activity prediction and trajectory prediction.

**SHUMIN CUI** received the B.M. degree in computer science and technology from the School of Management Science and Engineering, Shandong Normal University, Jinan, China, in 2018, where she is currently pursuing the M.E. degree with the Department of Computer Application Technology. She is also working with the Mine Digitalization Department of Engineering Research Center, Xuzhou, China. Her research interests include spatiotemporal activity prediction and trajectory prediction.