UC Irvine
UC Irvine Previously Published Works

## Title
Measuring Time-Sensitive and Topic-Specific Influence in Social Networks With LSTM and Self-Attention

## Permalink
https://escholarship.org/uc/item/4p52w4z0

## Authors
Zheng, Cheng
Zhang, Qin
Long, Guodong
et al.

## Publication Date
2020

## DOI
10.1109/access.2020.2991683

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

# HHS Public Access

Author manuscript

*IEEE Access*. Author manuscript; available in PMC 2020 June 23.

Published in final edited form as:

*IEEE Access*. 2020 ; 8: 82481–82492. doi:10.1109/access.2020.2991683.

# Measuring Time-Sensitive and Topic-Specific Influence in Social Networks with LSTM and Self-Attention

**CHENG ZHENG**[1], **QIN ZHANG**[2], **GUODONG LONG**[2], **CHENGQI ZHANG**[2], **SEAN D. YOUNG**[3], **WEI WANG**[1]

[1]University of California, Los Angeles, CA 90095 USA.

[2]Center for Artificial Intelligence, FEIT, University of Technology Sydney, NSW 2007, Australia.

[3]University of California, Irvine, CA 92697 USA.

## Abstract

Influence measurement in social networks is vital to various real-world applications, such as online marketing and political campaigns. In this paper, we investigate the problem of measuring time-sensitive and topic-specific influence based on streaming texts and dynamic social networks. A user's influence can change rapidly in response to a new event and vary on different topics. For example, the *political* influence of *Douglas Jones* increased dramatically after winning the Alabama special election, and then rapidly decreased after the election week. During the same period, however, *Douglas Jones*' influence on *sports* remained low. Most existing approaches can only model the influence based on static social network structures and topic distributions. Furthermore, as popular social networking services embody many features to connect their users, multi-typed interactions make it hard to learn the roles that different interactions play when propagating information. To address these challenges, we propose a Time-sensitive and Topic-specific Influence Measurement (TTIM) method, to jointly model the streaming texts and dynamic social networks. We simulate the influence propagation process with a self-attention mechanism to learn the contributions of different interactions and track the influence dynamics with a matrix-adaptive long short-term memory. To the best of our knowledge, this is the first attempt to measure time-sensitive and topic-specific influence. Furthermore, the TTIM model can be easily adapted to supporting online learning which consumes constant training time on newly arrived data for each timestamp. We comprehensively evaluate the proposed TTIM model on five datasets from Twitter and Reddit. The experimental results demonstrate promising performance compared to the state-of-the-art social influence analysis models and the potential of TTIM in visualizing influence dynamics and topic distribution.

### Keywords

Social Influence; Time-sensitive; Topic-specific; LSTM; Self-Attention

---

Corresponding author: Wei Wang (weiwang@cs.ucla.edu).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

## I. Introduction

Social network influence refers to the ability of a user to change the feelings, attitudes, or behaviors of other users within a network [1], [2], [3]. Influence measurement has become an essential task in many fields such as online marketing [4], [5] and political campaigns [6]. Due to its practical importance, measuring social influence has drawn growing research interests [7], [8], [9], [10]. In this paper, we study the problem of influence measurement in temporal social networks: Given streaming posts and interactional activities, the goal is to model the users' influence dynamics and find the influence distribution on different topics.

The influence of a user varies over time [8]. A person can become influential over a certain period due to a particular event. One example is *Douglas Jones*[1], the current United States Senator for Alabama. On December 12, 2017, *Jones* won a special election and became the first Democrat to win a Senate seat in Alabama since 1992. Due to the victory, *Douglas Jones*' political influence increased dramatically during the election period and then vanished rapidly after the election. This is reflected by the influence scores shown in Figure 1. The red star marks the exact date of the election and the peak appears just in the election week (covered with light-grey shadow). Figure 1 also shows the average influence score of all users, which has minor fluctuations. The example demonstrates that a person's influence can vary over time and be driven dramatically by specific events. Measuring the time-varying influence is essential for accurate detection of current influencers, which is critical for applications such as online marketing and political campaigns.

In addition to time, a user's influence depends heavily on the topics [11], [9]. Users who have high global influence scores may not be influencers on a certain topic and vice versa. An example is shown in Figure 2. We plot the influence scores of two Twitter users on distinct topics. *Jeff Dean* has a higher overall influence score than *Vitalik Buterin*[2], especially on the topics of *AI*, *ML* and *Big Data*. However, *Vitalik Buterin* is identified as a potential influencer on the topic of *Blockchain*. The topic-specific influence analysis [9] is of vital importance for many applications.

A vast number of topics in social networks are active and evolve rapidly, which calls for a unified framework to jointly model the influence propagation over time and topics. A question raised from the aforementioned observations is: how can one measure time-sensitive and topic-specific influence? There are three major challenges to approach the problem. First, joint modeling the distribution of influence with respect to time and topics involves combinations of the two types of features, which is impractical to enumerate all possibilities. Recent works [9], [11], [8] focus on using either temporal or topical features but not both. Second, social networks in real-life are composed of multiple types of user interactions. For example, on Twitter [2], [12], [13], we can interact with other users by various features such as *follow*, *retweet* and *mention*, etc. The key question is how to assess the contributions of different interactions when influence propagates. Existing works only consider a single type of interaction or assign equal weights to different types of interactions

[1] https://en.wikipedia.org/wiki/Doug_Jones_(politician)

[2] *Vitalik Buterin* is the co-founder of Ethereum and the co-founder of Bitcoin Magazine

[14]. Third, the nodes and edges in social networks are countless and evolve rapidly [15]. Therefore, supervised models are unable to take full advantage of the large-scale datasets, because only a small fraction of data is labeled with the ground truth

To address the challenges above, we propose an unsupervised model, called **T**ime-sensitive and **T**opic-specific **I**nfluence **M**easurement (**TTIM**) model. TTIM consists of influence attention network and matrix-adaptive long short- term memory (LSTM) [16], which can be jointly trained to automate the feature combinations in the first challenge. The proposed influence attention network aggregates node influence representations with attention to different types of interactions [17], [18]. The unsupervised training objective can drive the learning system without supervision from the ground truth. Our proposed framework can also be naturally adapted for online learning. We evaluate the proposed method with five real-world datasets from Twitter and Reddit. To summarize, the primary contributions of this work are:

- To the best of our knowledge, we are the first to simultaneously measure the time-sensitive and topic-specific influence in social networks.
- We propose a unified computational framework, TTIM, to solve the social influence measurement problem. The two sub-networks, influence attention network and matrix-adaptive LSTM, can jointly learn the contributions of different interactions and the influence dynamics in social networks. The framework supports both standard and online learning.
- We use five datasets crawled from Twitter and Reddit to compare TTIM with the state-of-the-art social influence measurement models. The experimental results demonstrate the effectiveness, efficiency, and scalability of the proposed method.

The rest of this paper is organized as follows. Section II presents the problem formulation. The details of the frame- work are shown in Section III. Section IV presents the datasets and experimental results, comparing our model with state- of-the-art methods. Section V summarizes related work and Section VI concludes this paper.

## II. Preliminaries

In this section, we define the notational conventions used in the paper and formally define the problem statement.

### A. Notations

The notations in this paper are displayed in Table I.

### B. Problem Formulation

As introduced in Section I, we aim to measure time-sensitive and topic-specific influence in social networks. Assume that there are $N$ users, and each user has two types of information: textual and interactional. For example, on Twitter, the textual information consists of the collection of tweets, which can be utilized to model users' affinity to certain topics. The interactional information may be extracted from the activities between users, such as mention and retweet on Twitter. Suppose that we have $T$ time intervals and $L$ types of



Author Manuscript Author Manuscript Author Manuscript Author Manuscript

interactions in total. Then we can formulate the social data as a sequence of temporal attributed graphs,

**Definition 1. Temporal attributed graphs** are denoted as $G_t = (\mathbb{V}, \boldsymbol{A}_t, \boldsymbol{X}_t), t = 1, \ldots, T,$ where $\mathbb{V}$ is the set of user nodes and $|\mathbb{V}| = N$. The interactional information is formulated as the adjacency tensor $\boldsymbol{A}_t \in \mathbb{R}^{N \times N \times L}$ for $L$ types of interactions in the $t$-th time interval. The user-topic affinity tensor $\boldsymbol{X}_t \in \mathbb{R}^{N \times M \times D}$ represents the textual information of $N$ users in the $t$-th interval. $M$ is the number of topics in the entire social network and $D$ is the topic embedding dimension.

For example, if the $l$-th type of interactions on Twitter is *mention* and $\boldsymbol{A}_{t(ijl)}$ equals to 2, then this represents that user $i$ was mentioned twice by user $j$ in the $t$-th interval. We will detail the generation of tensor $\boldsymbol{X}_t$ in Section III-A. Given the above definition, we introduce the problem formulation,

**Problem 1. Time-sensitive and Topic-specific Influence Measurement:** Given the temporal attributed graphs $G_t = (\mathbb{V}, \boldsymbol{A}_t, \boldsymbol{X}_t), t = 1, \ldots, T$ that represent the textual and interactional information in social networks, the goal is to output the time-sensitive and topic-specific influence tensor $\boldsymbol{B} \in \mathbb{R}^{N \times T \times M}$ for users $\mathbb{V}$.

Several key questions about Problem 1 need to be answered: 1) How do we extract the user-topic affinity tensor $\boldsymbol{X}_t$? 2) How do we assess the contributions of different types of interactions during the influence propagation? 3) How do we aggregate the textual and interactional information together? 4) How do we measure user influences as a function of topic and time over the graph sequence $G_t(t = 1, ..., T)$ in an unsupervised fashion?

## III. The Framework of TTIM

This section introduces the framework of TTIM model. An intuitive illustration is given in Figure 3. At each time interval, there are two types of raw data: textual and interactional. For text data, we utilize the Seeded Latent Dirichlet Allocation (SeededLDA) [19] to perform topic distillation and obtain the user-topic affinity tensor in each time interval. For the temporal graphs of $L$ types of interactions, we design the influence attention network to simulate the influence propagation process and learn the contributions of different interactions. Then the temporal influence is learned by optimizing the unsupervised objective function in a matrix-adaptive LSTM model. We also design an online version of TTIM by slightly altering the pipeline.

### A. Topic Distillation

In social networks, a user usually has interests on multiple topics. The topic distillation aims to learn the $D$-dimensional vector $\boldsymbol{X}_{t(ij)}$ that represents the embedding of user $i$ on topic $j$ at time $t$. Hence, we concatenate the messages posted by the same user in one time interval as one document, resulting in $N \times T$ documents. To obtain the topic focus of users, we utilize the SeededLDA model [19], which can identify latent topics in three fashions,



Author Manuscript Author Manuscript Author Manuscript Author Manuscript

- **Unsupervised:** Similar to the vanilla LDA [20], the document-topic distribution is learned from the probability distribution with the Dirichlet prior.
- **Supervised:** SeededLDA accepts sets of seed words as the representative of the underlying topics. In this way, we can obtain the document-topic distribution in specific domains.
- **Online:** It is not desirable to retrain the topic model from scratch whenever new data arrive. Instead, with online training, we could progressively update the model by utilizing previous topic-word distribution as seed words to feed to SeededLDA. Combining with the online LSTM model presented in Section III-C, we can train the model incrementally as new data arrive.

In each time interval, we distill $M$ topics and obtain the user-topic affinity tensors $\boldsymbol{X}_t \in \mathbb{R}^{N \times M \times D}, t = 1, \ldots, T$ For user $i$, $\boldsymbol{X}_{t(ij:)}$ is the term frequencies of top $D$ words belonging to topic $j$. A larger element in the tensor $\boldsymbol{X}_t$ indicates the more focus that a user puts on the corresponding topic. The unsupervised SeededLDA is suitable for training TTIM from scratch, where it automatically detects the topics in social posts. The supervised and online fashions are adaptive to the online training of TTIM model.

## B. Influence Attention Network

We build the influence attention network to simulate the influence propagation process and learn the contributions of different interactions. Following the formulation in Section II-B, we obtain the adjacency tensor $\boldsymbol{A}_t$, $t = 1, \ldots, T$, corresponding to $L$ types of interactions. Intuitively, different types of interactions play different roles in influence propagation. The majority of existing works only considered a single type of interaction or assigned a weight to interactions [14] according to domain knowledge. Inspired by Graph Attention Networks (GAT) [18], [21] and DeepInf [10], we propose the influence attention network, which can aggregate the node topic distribution with attention on the node's local neighborhood features and edges in multi-typed social networks.

Specifically, without loss of generality, we sketch the influence attention process focusing on a specific user $i$ in graph snapshot at time $t$. Let $\mathbb{N}_{i,t}$ be the set of one-hop neighbors of node $i$ at time $t$. Different from GAT or DeepInf, we introduce the attention coefficients for both user-topics affinities and user-user interactions,

$$e_{i,j} = \mathrm{MLP}_{\phi}(\boldsymbol{X}_{t(i)}, \boldsymbol{X}_{t(j)}, \boldsymbol{A}_{t(ij:)}) \tag{1}$$

where $j \in \mathbb{N}_{i,t}$ and the attention coefficient $e_{i,j}$ measures the relative influence that user $i$ has on user $j$. $\mathrm{MLP}_{\phi}$ is a multilayer neural network with parameters $\phi$. To accommodate users with different neighborhood sizes, we normalize the coefficients with softmax,

$$a_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k \in \mathbb{N}_{i,t}} \exp(e_{i,k})} \tag{2}$$



Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript



In the influence propagation process, the social network community disseminates messages with multiple rounds of propagation. Therefore, we propose to model the phenomena with multiple influence attention layers by aggregating nodes' topic distribution vectors in their neighborhood. The user-topic affinity tensor $\boldsymbol{X}_t$ is utilized as the input node features to the first layer ($\boldsymbol{F}_t^{(0)} = \boldsymbol{X}_t$). The $p$-th influence attention layer performs as follows,

$$\boldsymbol{F}_{t(i)}^{(p)} = \sigma\Big(\sum_{j \in \mathbb{N}_{i,t}} a_{i,j} \boldsymbol{F}_{t(j)}^{(p-1)} \boldsymbol{W}^{(p)}\Big) \tag{3}$$

where $\sigma(\cdot)$ is a non-linear activation function like ReLU, $\boldsymbol{F}_t^{(p)} \in \mathbb{R}^{N \times M \times d_n^{(p)}}$ is the output node representations, and $\boldsymbol{W}^{(p)} \in \mathbb{R}^{d_n^{(p-1)} \times d_n^{(p)}}$ is the parameter matrix for this layer. The aggregated feature tensor $\boldsymbol{F}_t$ from the output of the final influence attention layer represents the user topic distribution after influence propagation.

## C. Matrix-adaptive LSTM

With the sequence of aggregated feature tensors $\boldsymbol{F}_t$, $t = 1, \ldots, T$, we design a matrix-adaptive LSTM network [22], [23] to learn the time-sensitive and topic-specific influence scores for users. We adopt LSTM [24] motivated by its significant capability for learning long-term dependencies that naturally exist in temporal social network data. Shown in the right part of Figure 4, the matrix-adaptive LSTM accepts a sequence of matrices as input and outputs the state matrices of all time points, working as a many-to-many recurrent model.

The equations from Eq. 4 to Eq. 8 describe the operations in a matrix-adaptive LSTM cell, with the dimension $N$ omitted for simplicity,

$$\boldsymbol{I}_t = \sigma(\boldsymbol{F}_t \boldsymbol{W}_{xi} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hi} + \boldsymbol{C}_{t-1} \boldsymbol{W}_{ci} + \boldsymbol{b}_i) \tag{4}$$

$$\boldsymbol{G}_t = \sigma(\boldsymbol{F}_t \boldsymbol{W}_{xf} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hf} + \boldsymbol{C}_{t-1} \boldsymbol{W}_{cf} + \boldsymbol{b}_f) \tag{5}$$

$$\boldsymbol{C}_t = \boldsymbol{G}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{I}_t \odot \tanh(\boldsymbol{F}_t \boldsymbol{W}_{xc} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hc} + \boldsymbol{b}_c) \tag{6}$$

$$\boldsymbol{O}_t = \sigma(\boldsymbol{F}_t \boldsymbol{W}_{\boldsymbol{xo}} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{ho} + \boldsymbol{C}_t \boldsymbol{W}_{co} + \boldsymbol{b}_o) \tag{7}$$

$$\boldsymbol{H}_t = \boldsymbol{O}_t \odot \tanh(\boldsymbol{C}_t) \tag{8}$$

where $\sigma(\cdot)$ denotes the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, and $\boldsymbol{I}_t, \boldsymbol{G}_t \in [0, 1]^{M \times P}$ are the input and forget gates. $P$ is the size of the hidden states of the LSTM model. $\boldsymbol{C}_t \in \mathbb{R}^{M \times P}$ is the cell state, which is the core of an LSTM cell indicated by the longest vertical line in Figure 4. The cell state serves as the information connection between time $t-1$ and time $t$. The input and forget gates having values normalized to [0, 1] help the cell state control how much information it should take from the input (second term in Eq. 6) and how much is

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript



inherited from the previous time interval (first term in Eq. 6). $\boldsymbol{O}_t \in [0,1]^{M \times P}$ is the output gate and $\boldsymbol{H}_t \in \mathbb{R}^{M \times P}$ is the output state. The output gate filters information from the cell state $\boldsymbol{C}_t$ and passes it to the output state, which serves as the output of the LSTM network. In general, the LSTM network operates in a sequential fashion with $\boldsymbol{F}_t$ as the initial input. The cell state $\boldsymbol{C}_t$ at time $t$ and output state $\boldsymbol{H}_t$ will be repeatedly fed into the LSTM cell at time $t$ + 1. The weights $\boldsymbol{W}_{x\cdot} \in \mathbb{R}^{d_n^{(p)} \times P}, \boldsymbol{W}_{h\cdot} \in \mathbb{R}^{P \times P}, \boldsymbol{W}_{c\cdot} \in \mathbb{R}^{P \times P}$ and biases $\boldsymbol{b}_i, \boldsymbol{b}_f, \boldsymbol{b}_c, \boldsymbol{b}_o \in \mathbb{R}^P$ are the model parameters, which are trained by back-propagation with the objective function introduced in Section III-D. The influence tensor $\boldsymbol{B}$ can be obtained from the concatenation of output states $\boldsymbol{H}_t$ after a pooling layer. Possible choices for the pooling operation include max, average, and sum.

The matrix-adaptive LSTM network can generalize to support online training. At time $T'$, we may leverage the model trained at time $T' - 1$ to compute the extended user-topic affinity tensor $\boldsymbol{X}_{T'}$ and the aggregated feature matrix $\boldsymbol{F}_{T'}$. We can further train the LSTM model starting with the parameters ($\boldsymbol{W}$) from the previous LSTM model at time $T' - 1$. To capture the temporal dependency, we set a time interval window $T_W$ as a hyper-parameter: only data arrived during $[T' - T_W, T']$ is used to retrain the LSTM model. This allows the model to converge much faster than retraining from scratch.

## D. Objective Function

In order to measure the time-sensitive and topic-specific influence, we consider three criteria when we build the unsupervised objective function. First, the users with a larger neighborhood and higher affinity should have a higher influence score; Second, active users are more likely to have a high influence score than inactive users; Last, the change in the influence matrix should be smooth. Based on the ideas, the final optimization problem is constructed in Eq. 9 to learn the temporal user-topic influence matrix $\boldsymbol{B} \in \mathbb{R}^{N \times T \times M}$,

$$\begin{aligned} \max L(\boldsymbol{W}, \lambda_l) = & \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{A}_{t(ij:)} (1 + \sum_{k=1}^{N} \boldsymbol{A}_{t(jk:)}) \cdot \|\boldsymbol{B}_{(it:)}\|^2 \\ & + \zeta_1 \sum_{t=1}^{T} \sum_{i=1}^{N} \|\boldsymbol{F}_{t(i)}\|^2 \cdot \|\boldsymbol{B}_{(it:)}\|^2 \\ & - \zeta_2 \sum_{t=2}^{T} \|\boldsymbol{B}_{(:t:)} - \boldsymbol{B}_{(:t-1:)}\|_F^2 \end{aligned} \tag{9}$$

where $\boldsymbol{A}_t \in \mathbb{R}^{N \times N \times L}$ is the adjacency tensor for $L$ types of interactions in the $t$-th time interval; $\boldsymbol{F}_t$ is the aggregated user-topic affinity tensor in the time interval $t$ introduced in Section III-B. The larger value of $\boldsymbol{B}_{(itm)}$ represents user $i$ has a higher influence on topic $m$ at time $t$. $\boldsymbol{W}$ contains the weight matrices in the LSTM model and influence attention network. $\xi_i > 0$, $i = 1, 2$ are the trade-off parameters to balance the three components. A constraint is added to normalize user influence scores on a topic for each time interval. We use back-propagation through time (BPTT) algorithm to train the model and learn the user influence scores.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

**Algorithm 1**

TTIM-Online with new data arriving at time $T'$

**Require:** $\boldsymbol{A}_t^{(l)}$, $\boldsymbol{F}_t$, where $t = T' - T_W, \ldots, T'$, documents $d^{T'}$, previous SeededLDA($T' - 1$), previous LSTM($T' - 1$), training epoch $n_{epoch}$, hyperparameters $\alpha$, $\xi_1$, $\xi_2$, $T_W$

1: Load the topic-word distribution from SeededLDA($T' - 1$) as seed distribution for SeededLDA($T'$)

2: Train SeededLDA($T'$) with $d^{T'}$.

3: Compute $\boldsymbol{X}_{T'(ij)}$

4: Compute $\boldsymbol{F}_{T'}$ using Eq. 1, 2, 3

5: Load $\boldsymbol{W}$ from LSTM($T' - 1$) to LSTM($T'$)

6: **for** $epoch = 1$; $epoch \le n_{epoch}$ **do**

7:     **for** $t = T' - T_W$; $t \le T'$ **do**

8:         Compute $\boldsymbol{I}_t$, $\boldsymbol{G}_t$, $\boldsymbol{C}_t$, $\boldsymbol{O}_t$, $\boldsymbol{H}_t$ using Eq. 4 - Eq. 8

9:     **end for**

10:     Compute $L(\boldsymbol{W}, \lambda_l)$ using Eq. 10

11:     Backpropagate and update $\boldsymbol{W}$

12: **end for**

With our proposed influence attention network and matrix-adaptive LSTM, we can extend the TTIM model to online fashion. We depict the pseudocode of the online training of the TTIM model in Algorithm 1. With time $T'$ data arriving, the modified objective function is,

$$\begin{aligned} &\max L(\boldsymbol{W}, \lambda_l) = \\ &\sum_{t = T' - T_W}^{T'} \sum_{i = 1}^{N} \sum_{j = 1}^{N} \boldsymbol{A}_{t(ij)}(1 + \sum_{k = 1}^{N} \boldsymbol{A}_{t(jk)} \cdot \|\boldsymbol{B}_{(it:)}\|^2) \\ &+ \zeta_1 \sum_{t = T' - T_W}^{T'} \sum_{i = 1}^{N} \|\boldsymbol{F}_{t(i)}\|^2 \cdot \|\boldsymbol{B}_{(it:)}\|^2 \\ &- \zeta_2 \sum_{t = T' - T_W + 1}^{T'} \|\boldsymbol{B}_{(:t:)} - \boldsymbol{B}_{(:t-1:)}\|_F^2 \end{aligned} \quad (10)$$

In summary, our TTIM model answers the questions raised in Section II-B with well-designed pipeline: the SeededLDA model learns the user-topic affinity tensor; the adjacency tensors for different interactions are integrated with learnable weights; the influence attention network simulates the influence diffusion; the matrix-adaptive LSTM model captures the long-term dependencies and learns the influence scores following the optimization problem. Streaming texts and dynamic social networks are jointly modeled to measure social influence.

## IV. Experiments

In this section, We evaluate our proposed method with extensive experiments. First, we introduce the labeled datasets to quantitatively evaluate our model with the influencer

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

detection task, shown in Section IV-A. Second, we qualitatively evaluate the time-sensitive and topic-specific property of TTIM model with large unlabeled datasets in Section IV-B. Third, the proposed TTIM-Online method is shown to be efficient in training and achieve competitive results in Section IV-C. Finally, we conduct the parameter sensitivity and scalability analysis in Section IV-D.

### A. Experiments with Labeled Datasets

In this section, we detail the experimental results on the influencer detection task with three labeled datasets. The task aims to identify the top influential individuals from all users in social networks. First, we introduce the datasets and baselines, followed by the influencer detection results.

**1) Datasets:** We created three manually labeled datasets from Twitter (*Politics set* and *Technology set*) and Reddit (*Reddit set*). Dataset statistics are shown in Table II. More preprocessing details can be found in the supplementary materials.

In the two datasets from Twitter, we labeled the influencers by selecting users with a large group of followers, active involvement in the politics/technology topics and top global influence on other users' actions. The labels were selected from the majority votes of three human labelers. The *Politics set* contains 1,031 users who send politics-related tweets, and 64 of them are labeled as influencers. There are 10 one-week intervals and 1,840,552 tweets in total. The *Technology set* contains 1,122 users who send technology-related tweets, and 80 of them are labeled as influencers. There are 7 one-day intervals and 141,835 tweets in total.

Reddit is an online discussion forum where users post and comment on contents in different topical communities. In the Reddit platform, users can upvote posts that they are in favor of, so the number of upvotes can indicate the influence of posts and their senders. We labeled users whose posts received the most upvotes as influencers. The *Reddit set* is from May 2015 Reddit comment dump[3] and it contains 35,267 users, with 100 labeled influencers. We build a user-to-user interaction graph, connecting users if one user comments under another user's post. There are 31 one-day intervals and 126,125 posts/comments in *Reddit set*.

**2) Baselines:** We compare the proposed TTIM model with the following seven representative baselines:

- **Followers.** The feature used by this baseline is the number of the user's followers. Note that we only have access to this feature on Twitter, not with Reddit.
- **TwitterRank**[11] is an extension of the PageRank algorithm, which uses LDA to find some topics, and then calculates the rank of users with respect to topics based on their influence on followers and their interests in these topics.

[3]https://www.kaggle.com/reddit/reddit-comments-may-2015

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

- **Topical Affinity Propagation (TAP)** [25] is a topical affinity propagation model built on a factor graph to identify the topic-specific social influence.
- **ReFluence** [26] is a statistical and analytical model based on Edelman's topology of influence to determine the user's role and influence on each other. Here we treat the "Idea Starter" and "Amplifier" defined in this baseline as influencers and the others as normal users.
- **RR-LT** model [8] uses a function of edge weights and the self-weight of nodes to represent influence probabilities under the Linear Threshold model. Polling-based methods and a sample of random reversely reachable sets are used to approximate the influence of nodes.
- **RR-IC** model [8] uses propagation probability, polling, and random reversely reachable sets to track influencers under the Independent Cascade model.
- **CoupledGNN** [27] applies two coupled graph neural networks to iteratively model and predicts the network-aware popularity.

**3) Experimental Settings:** The proposed TTIM is implemented in the Tensorflow framework [28]. The training optimizer is Adam [29] with a learning rate as 0.0005, $\beta_1$ = 0.9, and $\beta_2$ = 0.999. The experiments are conducted on a Linux server with a 16G memory Tesla V100 GPU, 20 Intel Xeon E5–2698 CPUs, and 512 GB memory. We use the grid search to tune hyper-parameters. The topic embedding dimension $D$ is chosen from {5, 10, 20, 30}. The trade-off parameters in equation 9 ($\zeta_1$, $\zeta_2$) are searched from $10^{-4}$ to $10^4$ with a step of $10^1$. We initial the weight matrices in the proposed TTIM model with Xavier initialization [30].

To identify influencers in the labeled datasets, we sum the learned influence score of all topics and max-pool over time to obtain each user's influence score. The output predictions will be the top-$k$ users with the highest influence scores. We evaluate the influencer detection performance by *precision*, *F1-measure*[4], and *AUC* metrics [31]. All the results are the average of 10 repeated runs.

**4) Experimental Results:** Table III shows the detailed results of detecting the top-$k$ influencers, where $k$ is the number of positive samples in the ground truth ($k$ = 64 for *Politics set*, $k$ = 80 for *Technology set*, and $k$ = 100 for *Reddit set*)[5]. The best results are highlighted in bold. The results show that TTIM is effective and outperforms other baselines in precision, F1, and AUC on these datasets. Figure 5 further illustrates the precision at $k$, where $k$ is the number of top influencers identified by each method. We can observe that TTIM always maintains a higher precision level than the baselines and has 100% precision over the top 20 on Politics and Technology sets. These observations verify the outstanding ability of TTIM to detect the top influencers. We attribute the significant improvement to the following two reasons. First, TTIM considers diverse sources including the text contents and

[4] $F1 = 2\frac{Precision \times Recall}{Precision + Recall}$

[5] Here, precision and F1 values are always equal since $k$ equals the number of true labels, making the number of false positives and false negatives equivalent.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

multiple interactions. Compared against baselines like CoupledGNN which treated interactions equally, TTIM automatically learns the different weights of interactions via attention mechanism. Second, TTIM well models the temporal data by using LSTM to learn the influence score with the streaming text and dynamic social networks integrated seamlessly.

We conduct the ablation study by removing the attention mechanism (TTIM w/ Attention) and LSTM (TTIM w/ LSTM) one by one at a time. As the results in the Table III show, each module contributes to the performance improvement and the proposed TTIM benefits from the influence propagation process learned by the influence attention network and the time-sensitive pattern learned by the matrix-adaptive LSTM.

We also highlight the capability of TTIM on retrieving the time-sensitive and topic-specific influence score of users with labeled Twitter datasets, shown in Figure 6(a),(c) for the *Politics set* and in Figure 6(b),(d) for the *Technology set*. We can observe some interesting phenomena. For example, in Figure 6 (a), there is not only a peak for Twitter user *Douglas Jones* but also a similar trend for user *TheDailyEdge* and *TeaPainUSA*. A probable reason for this could be they are in the same political party as *Douglas Jones* and share the influential benefits from the election event. Another finding is that user *Vitalik Buterin*'s influence score is mainly limited to the topic *blockchain*. This could be the reason why his influence trend is similar to the bitcoin price during that period.

## B. Experiments with Unlabeled Datasets

In this section, we discuss the experimental results on influence measurement on the large unlabeled datasets.

**1) Datasets:** We utilize two unlabeled datasets. The *LV-shooting set* contains 1% of all tweets during the period from October 1, 2017 to October 11, 2017. On the night of October 1, 2017, a gunman fired more than 1,100 rounds to a crowd of concertgoers at the Route 91 Harvest music festival in Las Vegas, leaving 58 people dead and 851 injured[6]. This event aroused a huge response on social media platforms, so we crawled the tweets over the following 11 days. After data preprocessing, the dataset contained 2,859,809 users, 17,635,937 tweets and 11 one-day time intervals. Another dataset *General set* contains 1% of tweets in three years (Aug 2016 - Jul 2019). The dataset contained 1,893,174 users and 15,953,165 tweets, with 36 one-month time intervals.

**2) Results:** Table IV shows the top-5 topics with their top keywords and top influencers. We name these topics to simplify the presentation. Intuitively, it is clear that the influencers are very relevant to the corresponding topics. For example, one would expect *Donald Trump*, *Mike Pence*, and *Hillary Clinton* to be influential on *politics*-related topics, just as one would expect the public figures such as *Rihanna* (singer) and *Jake Tapper* (journalist), and online video-sharing platform (*Youtube*) to be influential in the *praying* activities after the Las Vegas shooting tragedy.

[6]https://en.wikipedia.org/wiki/2017_Las_Vegas_shooting

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

We explore the time-sensitive and topic-specific property of the influence score respectively in Figure 7(a), (b) and Figure 7(c), (d). We show the influence scores of the top-5 influencers detected by TTIM in these two datasets. In Figure 7(a), four users had an influence peak on October 1, 2017, just after the *Las Vegas Shooting* happened, except BleacherReport (which is a sports platform). From Figure 7(b), we can observe that *Donald Trump* and two news platforms *Fox News* and *The New York Times* have obvious peaks during the period of the presidential election (October to November 2016) and the presidential inauguration (December 2016 to January 2017), which is reasonable. We can see the account for *Donald J. Trump* has relatively high influence over the period in both datasets, because of his activeness on Twitter. Figure 8 shows the 3D plots of the influence score of *Donald Trump* in the LV-shooting and General datasets, respectively. We can see that the influence score varies significantly along the dimensions of time and topic. In summary, our proposed TTIM model captures the time-sensitive and topic-specific influence on a large scale and can identify influencers with various time granularity.

## C. Online Training

We furnish a comparison between the standard TTIM model and TTIM-Online. When new data arrive at time $T'$:

- **TTIM** is retrained with all data arrive so far, i.e., during the time [1, $T'$]
- **TTIM-Online** starts from the previous model trained based on data of [1, $T' -$ 1], and updates the model only using the recent data, as detailed in Algorithm 1.

Both algorithms run until the models converge. The time interval window $T_W$ in the LSTM model in TTIM-Online is set as 3. We show the TTIM-Online performance in Table III and Figure 5. TTIM-Online still outperforms baseline models measured by precision, F1, and AUC on labeled datasets. Figure 9 shows the detailed precision and training time of TTIM and TTIM-Online in *Politics* and *Technology sets*. In Figure 9(a), we observe that both algorithms achieve similar precision in detecting influencers. In the beginning, the arrival of new data will enhance the precision, until the model saturates and its performance reaches a plateau. However, Figure 9(b) shows the online version of the TTIM model is much more efficient and scalable than the standard TTIM model. The training time of the online TTIM model remains at a constant level for each timestamp, whereas the training time taken by the standard TTIM model at each time stamp grows linearly as the number of timestamps increases.

## D. Parameter Study and Scalability

In this section, we visualize the attention coefficients regarding different interactions in equation 1. The average values on the four Twitter datasets are shown in Table V. We can see that the *quote* interaction has the largest coefficients in all datasets. This demonstrates that people are more deeply influenced by others when they decide to *quote* tweets and write down feelings, The *hashtags* have relatively smaller coefficients, probably because one hashtag is usually mentioned by many users and it is hard to trace the source of the influence. Also, a hashtag has multiple synonyms, which may further complicate the influence propagation. As for parameter sensitivity, we plot the AUC when doing the grid

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

search for $\zeta$ pairs, shown in Figure 10. The model favors a smaller $\zeta_2$ and an optimized $\zeta_1$ value to reach the best performance.

We evaluate the scalability of TTIM by measuring the training time as a function of the number of users and the convergence rate on two large unlabeled datasets in Figure 11. Figure 11(a) reports the training time of TTIM corresponding to different numbers of users ($N$). We can see that the training time increases approximately linearly as the number of users grows, which verifies that TTIM scales well to large datasets. Figure 11(b) shows the convergence curves on the two large datasets, demonstrating that TTIM converges quickly on both datasets.

## V. Related Work

The problem of influence measurement problem aims to quantify user influence in social networks and identify influencers. Most previous models focused on formulating the user interactions into graphs, and detecting influential nodes based on the formulated graph through PageRank [11], Hyperlink-Induced Topic Search (HITS) [32], probabilistic random walk on expertise graphs [33] or their variants [34]. Deepinf [10] and NNMLinf [35] were trying to model the micro-level social influence and predict the user actions after influenced by the local neighborhood. A two coupled graph neural networks based method, CoupledGNN [27], was proposed to predict the popularity in social networks by capturing the cascading effect in information diffusion. Compared with Deepinf and NNMLinf, our proposed TTIM aims to measure the macro-level influence and model its dynamics, which is vital to global influencer identification. Compared with Coupled-GNNs and its analogs, our TTIM method considers the specific topics during the exploring of influence, not only the cascading effects (i.e. time-sensitive effects).

Besides the vanilla problem, if we ignore the influence of time on the mensuration, **topic-specific influencer detection** has been studied in several previous works [36], [37], [38], [25]. TwitterRank [11] used both network structure and topic similarity in calculating user influence on Twitter. Bi et al. [9] proposed a Bernoulli-multinomial mixture method that jointly modeled text and followship. And if we ignore the influence of specific topics on the mensuration, **influence dynamics analysis** has attracted many interests considering the evolving nature of social networks [39], [40]. Aggarwal et al. [41] proposed the influential node discovery in dynamic networks with the forward and backward trace approach. Yang et al. [8] studied influential node tracking and influence maximization [42], [43] by modeling dynamic changes as a stream of edge weight updates.

In summary, there is no existing work measuring time-sensitive and topic-specific influence in social networks. That motivates us to propose the LSTM and self-attention based TTIM, which integrates streaming texts and multiplex inter- actions to measure the temporal social influence on various topics.

## VI. Conclusion

This paper explores the problem of measuring time-sensitive and topic-specific influence in social networks. A computational framework, Time-sensitive and Topic-specific Influence

Measurement, is proposed based on influence attention network and matrix-adaptive LSTM. With multiple types of interactions and streaming texts, the influence attention network simulates the influence diffusion with self-attention. The matrix-adaptive LSTM captures the long-term dependencies and learns the influence scores following the optimization problem. Comprehensive evaluations of the proposed method are conducted with five datasets from Twitter and Reddit. The experimental results show superior performance of TTIM over the state-of-the-art social influence analysis models. By applying the proposed TTIM model to Twitter data of a large scale, we can visualize the influence dynamics and topic distributions in social networks.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgement

We thank the anonymous reviewers for their careful reading and insightful comments on our manuscript. The work is partially supported by NIH U01HG008488.

## Biography

**Cheng Zheng** received his bachelor's degree in physics from Tsinghua University in 2015. He is currently working towards the doctoral degree in electrical and computer engineering with the University of California, Los Angeles. His research interests include graph mining, social network analysis and deep learning. He has served as a reviewer for WSDM, SIGIR and IEEE Access.

**Qin Zhang** received the master's degree from the University of Chinese Academy of Sciences, China, in 2014. She received her doctoral degree from the Center for Artificial Intelligence (CAI), University of Technology Sydney, Australia in 2019. She is currently with the Data Center of Social Network Group (SNG), Tencent, China. Her main research interests include sequence data learning and network analysis by using various deep learning and optimization methods. She has served as a reviewer (sub-reviewer) for KDD, NIPS, ICDM, IJCAI, AAAI and SDM.



Author Manuscript Author Manuscript Author Manuscript Author Manuscript

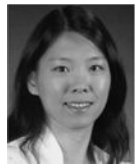
Author Manuscript Author Manuscript Author Manuscript Author Manuscript

**Guodong Long** received the PhD degree from the University of Technology Sydney (UTS), Australia, in 2014. He is currently a senior lecturer with the Research Center for Artificial Intelligence (CAI) in UTS. His research interests include data mining, machine learning, artificial intelligence, social network analytics and healthcare informatics.

**Chengqi Zhang** received the bachelor's degree in computer science from Fudan University, in March 1982, the master's degree in computer science from Jilin University, in March 1985, and the PhD degree in computer science from the University of Queens- land, in October 1991, followed by a doctor of science degree from Deakin University, in October 2002. He has been appointed as a distinguished professor with the University of Technology Sydney from 27 February 2017 to 26/February 2022, an executive director UTS Data Science from 3 January 2017 to 2 January 2021, an honorary professor with the University of Queensland from 1 January 2015 to 31 December 2017, an adjunct professor with the University of New South Wales from 20 March 2017 to 20 March 2020, and a research professor of information technology with the UTS from 14 December 2001. In addition, he has been selected as the chairman of the Australian Computer Society National Committee for Artificial Intelligence since November 2005, and the chairman of the IEEE Computer Society Technical Committee of Intelligent Informatics (TCII) since June 2014.

**Sean D. Young** received his PhD in Psychology and Master's degree in Health Services Research from Stanford University. He is the executive director of the University of California Institute for Prediction Technology, the UCLA Center for Digital Behavior, a Medical School and Informatics Professor with the UCI Departments of Emergency Medicine and Informatics. Before joining UCI, he was a medical school professor in the UCLA Department of Family Medicine, where he continues to hold a joint appointment.

Author Manuscript



**Wei Wang** received the PhD degree in computer science from the University of California, Los Angeles, in 1999. She is the Leonard Kleinrock chair professor in computer science with the University of California, Los Angeles, and the director of the Scalable Analytics Institute (ScAi). Her research in- terests include big data analytics, data mining, bioinformatics and computational biology, and databases. She was a professor in computer science and a member of the Carolina Center for Genomic Sciences and Lineberger Comprehensive Cancer Center, University of North Carolina at Chapel Hill, from 2002 to 2012, and was a research staff member at the IBM T. J. Watson Research Center between 1999 and 2002. She is a member of the IEEE.

## References


[1]. Kelman HC, "Compliance, identification, and internalization three processes of attitude change," Journal of conflict resolution, 1958.

[2]. Cha M, Haddadi H, Benevenuto F, and Gummadi PK, "Measuring user influence in twitter: The million follower fallacy," in ICWSM, 2010.

[3]. Riquelme F. and González-Cantergiani P, "Measuring user influence on twitter: A survey," Information Processing & Management, 2016.

[4]. Richardson M. and Domingos P, "Mining knowledge-sharing sites for viral marketing," in KDD, 2002.

[5]. Domingos P. and Richardson M, "Mining the network value of customers," in KDD, 2001.

[6]. Ranganath S, Hu X, Tang J, and Liu H, "Understanding and identifying advocates for political campaigns on social media," in WSDM, 2016.

[7]. Tang J, Wu S, and Sun J, "Confluence: Conformity influence in large social networks," in KDD, 2013.

[8]. Yang Y, Wang Z, Pei J, and Chen E, "Tracking influential individuals in dynamic networks," TKDE, 2017.

[9]. Bi B, Tian Y, Sismanis Y, Balmin A, and Cho J, "Scalable topic-specific influence analysis on microblogs," in WSDM, 2014.

[10]. Qiu J, Tang J, Ma H, Dong Y, Wang K, and Tang J, "Deepinf: Social influence prediction with deep learning," in KDD, 2018.

[11]. Weng J, Lim E-P, Jiang J, and He Q, "Twitterrank: finding topic- sensitive influential twitterers," in WSDM, 2010.

[12]. Bakshy E, Hofman JM, Mason WA, and Watts DJ, "Everyone's an influencer: quantifying influence on twitter," in WSDM, 2011.

[13]. Wu S, Hofman JM, Mason WA, and Watts DJ, "Who says what to whom on twitter," in WWW, 2011.

[14]. Garimella K, Weber I, and De Choudhury M, "Quote rts on twitter: usage of the new feature for political discourse," in WSDM, 2016.

[15]. Leskovec J, Kleinberg J, and Faloutsos C, "Graphs over time: densification laws, shrinking diameters and possible explanations," in KDD, 2005.

[16]. Hochreiter S. and Schmidhuber J, "Long short-term memory," Neural computation, 1997.

[17]. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, and Polosukhin I, "Attention is all you need," in NIPS, 2017.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

[18]. Velikovi P, Cucurull G, Casanova A, Romero A, Li P, and Bengio Y, "Graph attention networks," in ICLR, 2018.

[19]. Jagarlamudi J, Daumé H III, and Udupa R, "Incorporating lexical priors into topic models," in EACL, 2012.

[20]. Blei DM, Ng AY, and Jordan MI, "Latent dirichlet allocation," JMLR, 2003.

[21]. Zhang J, Shi X, Xie J, Ma H, King I, and Yeung D-Y, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," arXiv, 2018.

[22]. Gers FA and Schmidhuber J, "Recurrent nets that time and count," in IJCNN, 2000.

[23]. Yu W, Zheng C, Cheng W, Aggarwal C, Song D, Zong B, Chen H, and Wang W, "Learning deep network representations with adversarially regularized autoencoders," in KDD, 2018.

[24]. Graves A, "Generating sequences with recurrent neural networks," arXiv, 2013.

[25]. Tang J, Sun J, Wang C, and Yang Z, "Social influence analysis in large-scale networks," in KDD, 2009.

[26]. Tinati R, Carr L, Hall W, and Bentwood J, "Identifying communicator roles in twitter," in WWW, 2012.

[27]. Cao Q, Shen H, Gao J, Wei B, and Cheng X, "Popularity prediction on social platforms with coupled graph neural networks," in WSDM, 2020.

[28]. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M. et al., "Tensorflow: a system for large- scale machine learning." in OSDI, 2016.

[29]. Kingma DP and Ba J, "Adam: A method for stochastic optimization," in ICLR, 2015.

[30]. Glorot X. and Bengio Y, "Understanding the difficulty of training deep feedforward neural networks," in AISTATS, 2010.

[31]. Huang J. and Ling CX, "Using auc and accuracy in evaluating learning algorithms," TKDE, 2005.

[32]. Campbell CS, Maglio PP, Cozzi A, and Dom B, "Expertise identification using email communications," in CIKM, 2003.

[33]. Serdyukov P, Rode H, and Hiemstra D, "Modeling multi-step relevance propagation for expert finding," in CIKM, 2008.

[34]. Gao B, Liu T-Y, Wei W, Wang T, and Li H, "Semi-supervised ranking on very large graphs with rich metadata," in KDD, 2011.

[35]. Wang X, Guo Z, Wang X, Liu S, Jing W, and Liu Y, "Nnmlinf: social influence prediction with neural network multi-label classification," in Proceedings of the ACM Turing Celebration Conference - China, 2019.

[36]. Chen S, Fan J, Li G, Feng J, Tan K.-l., and Tang J, "Online topic- aware influence maximization," VLDB, 2015.

[37]. Pal A. and Counts S, "Identifying topical authorities in microblogs," in WSDM, 2011.

[38]. Cossu J-V, Dugué N, and Labatut V, "Detecting real-world influence through twitter," in ENIC, 2015.

[39]. Sha M, Li Y, Wang Y, Guo W, and Tan K-L, "River: A real-time influence monitoring system on social media streams," in ICDWM, 2018.

[40]. Zhuang H, Sun Y, Tang J, Zhang J, and Sun X, "Influence maximization in dynamic social networks," in ICDM, 2013.

[41]. Aggarwal CC, Lin S, and Yu PS, "On influential node discovery in dynamic social networks," in SDM, 2012.

[42]. Kempe D, Kleinberg J, and Tardos É, "Maximizing the spread of influence through a social network," in KDD, 2003.

[43]. Chen W, Wang Y, and Yang S, "Efficient influence maximization in social networks," in KDD, 2009.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

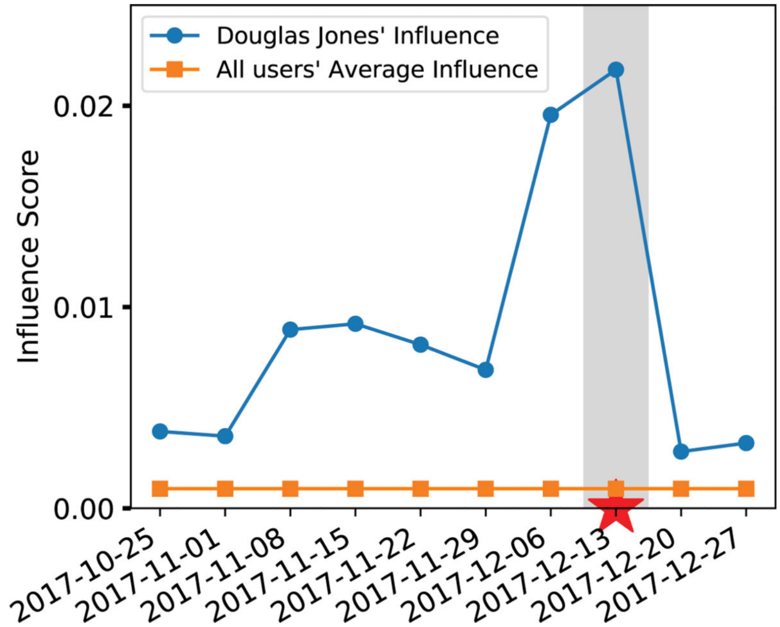



**Fig. 1.**
Time-sensitive influence variation.

Author Manuscript

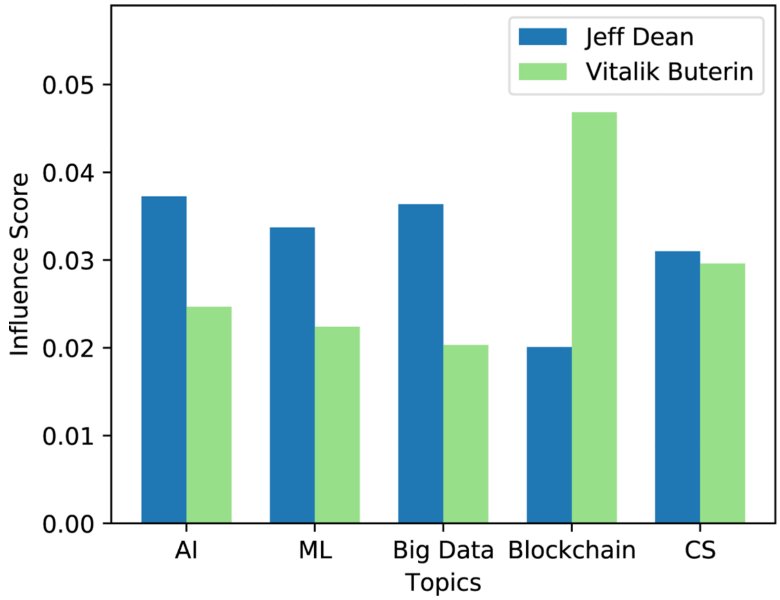



**Fig. 2.**

Topic-specific influence variation.(AI, ML, CS refer to topics *Artificial Intelligence*, *Machine Learning* and *Computer Science* respectively).

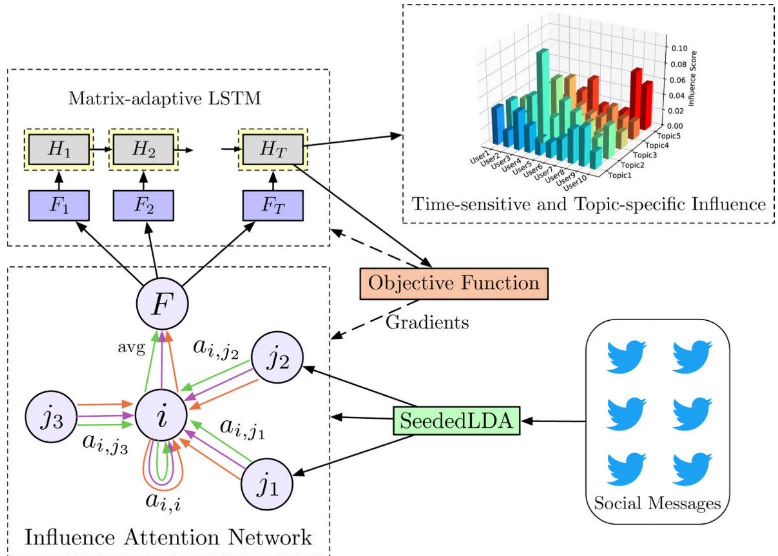



**Fig. 3.**

The workflow of TTIM. In each time interval, we collect two types of data: streaming texts and multiple types of interactions. From raw text data, we distill topics and obtain a user-topic affinity tensor $\boldsymbol{X}_t$ using the SeededLDA model. We combine multiple types of interactions with the influence attention network and obtain user representations $\boldsymbol{F}_t$. The matrix-adaptive LSTM learns the influence dynamics with the sequence of the influence features. After the model training with an objective function in the unsupervised fashion, we obtain time-sensitive and topic-specific influence scores of the users.



Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

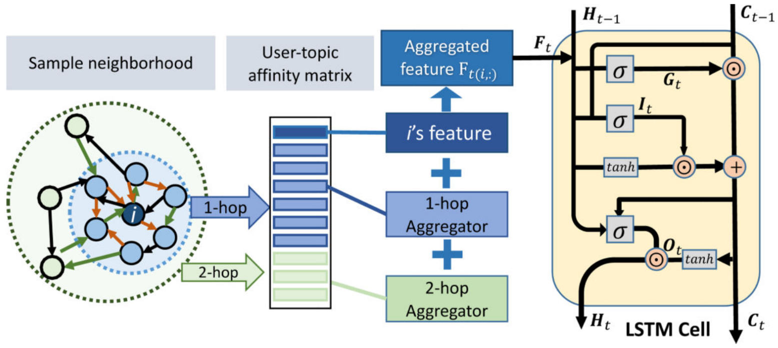



**Fig. 4.**

The aggregation of the user $i$'s textual and interactional information at the $t$-th time interval and the information flow in an LSTM cell. The left part is the sample neighborhood of user $i$ at $t$-th interval, which includes his/her 1-hop and 2-hop neighbors; the middle part illustrates the components of the aggregated feature $\boldsymbol{F}_t$ which is the weighted sum of the affinity features of the neighborhood; the right figure shows the detailed structure of the LSTM cell introduced in Section III-C, with the aggregated features $\boldsymbol{F}_t$ as its input data.

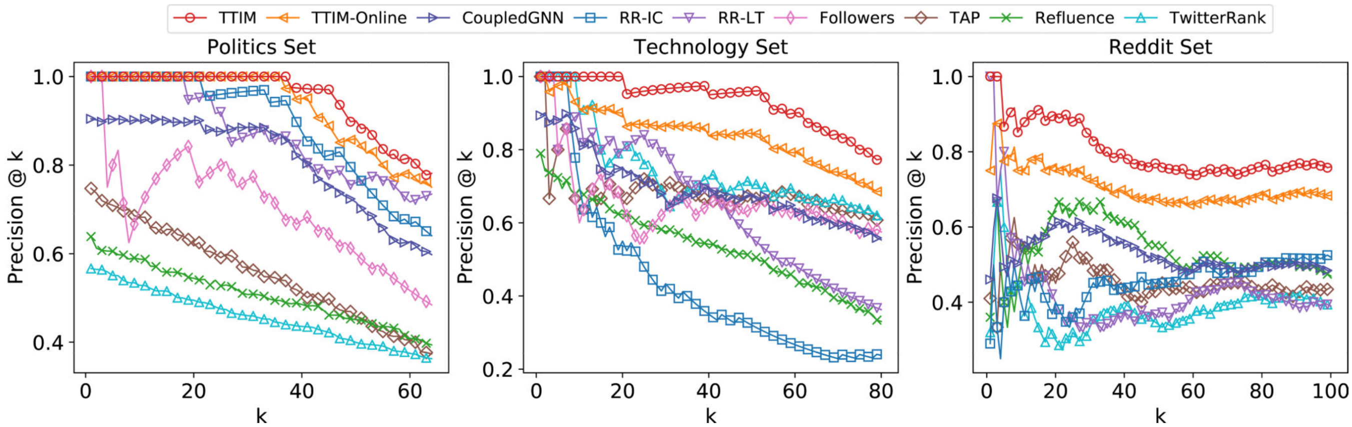



**Fig. 5.**
The precision of top-*k* influencers detection on the labeled datasets.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

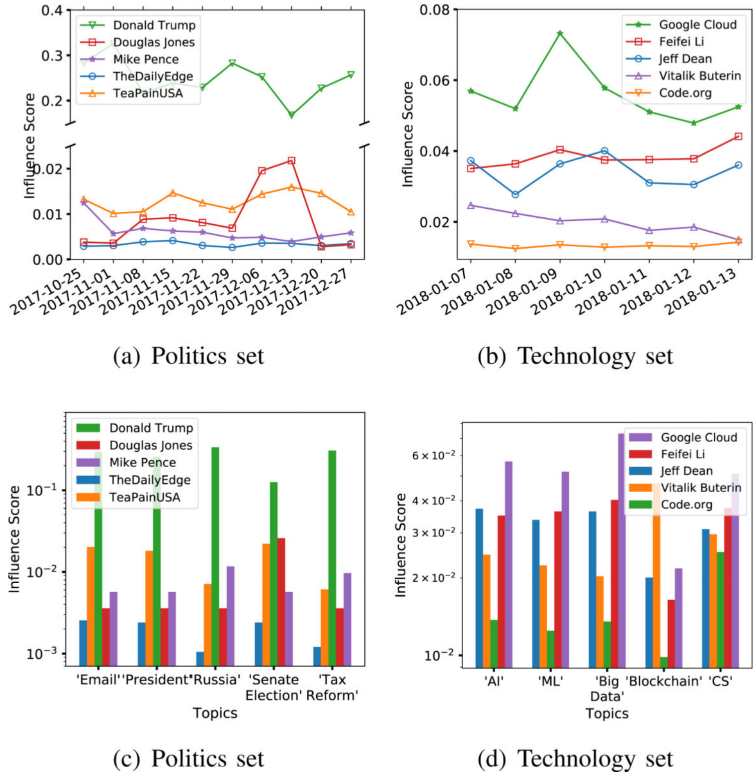



**Fig. 6.**

Time-sensitive influence (a, b) and topic-specific influence (c, d) for the top influencers in *Politics* and *Technology* datasets.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

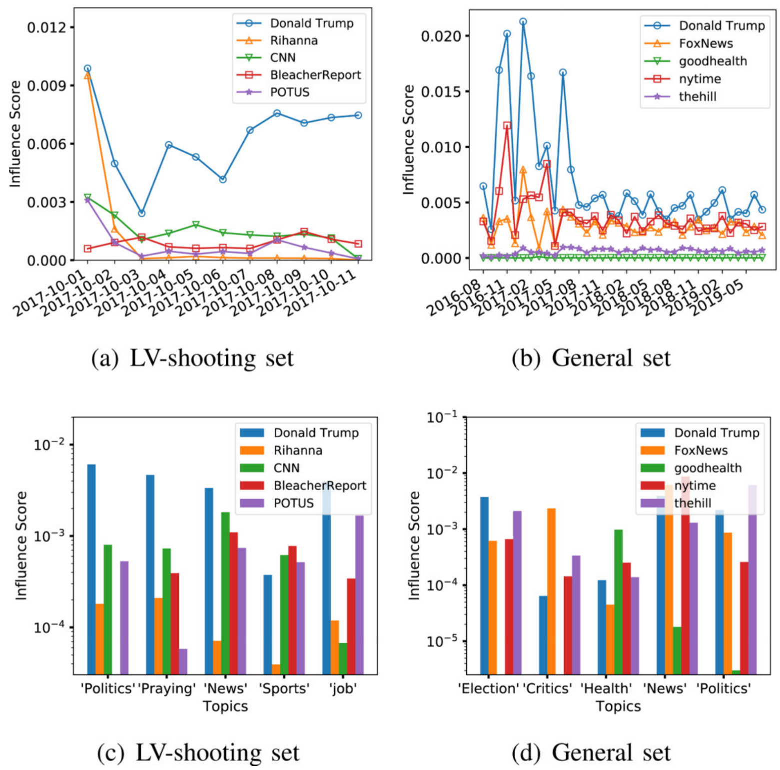
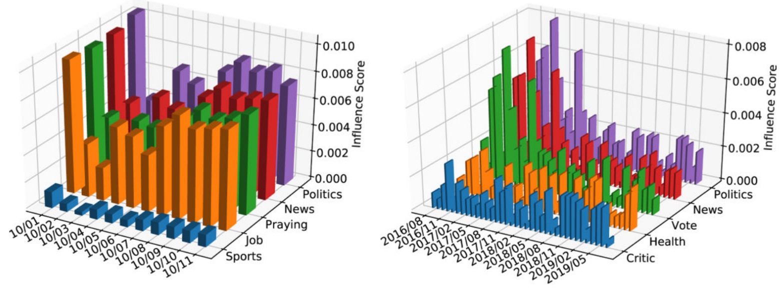



**Fig. 7.**

Time-sensitive influence (a, b) and topic-specific influence (c, d) for the top influencers in *LV-shooting* and *General* datasets.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript



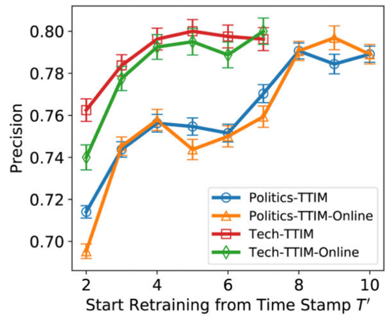



(a) LV-shooting set (b) General set

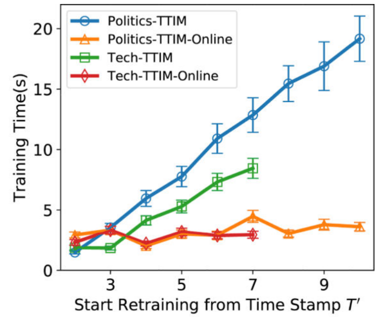
**Fig. 8.**

The variation of influence score for *Donald Trump* in (a) LV-shooting set; and (b) General set.

Author Manuscript




(a) Precision




(b) Training Time

**Fig. 9.**
Comparison of TTIM and TTIM-Online

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

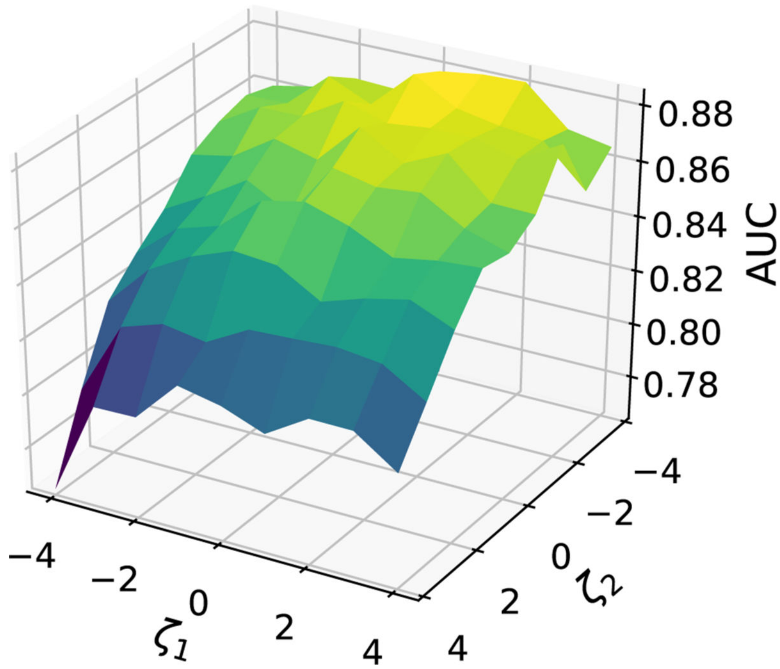
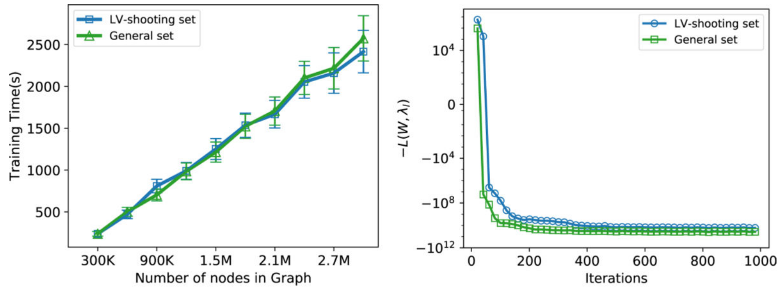



**Fig. 10.**
AUC of *Politics set* with different parameter $\zeta$ pairs




(a) Training Time (b) Convergence Curve

**Fig. 11.**

(a) Training time with respect to data size; (b) Convergence curves on two large datasets.



Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

**TABLE I**

Terms and Notations

| Symbol | Definition |
|---|---|
| $a$, $\alpha$, $B$ | scalars |
| $\boldsymbol{a}$, $\boldsymbol{b}$, . . . | vectors |
| $\mathbb{A}$, $\mathbb{B}$, . . . | sets |
| $\boldsymbol{A}$, $\boldsymbol{B}$, . . . | matrices and tensors |
| $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j} \boldsymbol{A}_{i,j}^2}$ | Frebonius norm of matrix $\boldsymbol{A}$ |
| $\odot$ | Hadamard product |
| $\sigma(\cdot)$ | non-linear activation function |

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**TABLE II**

STATISTICS OF THE DATASETS.

| Dataset | observation window | # time intervals | # users | # posts |
|---|---|---|---|---|
| Politics | 2017.10.22 – 2017.12.30 | 10 | 1,031/64* | 1,840,552 |
| Technology | 2018.01.07–2018.01.13 | 7 | 1,122/80* | 141,835 |
| Reddit | 2015.05.01–2015.05.31 | 31 | 35,267/100* | 126,125 |
| LV-shooting | 2017.10.01–2017.10.11 | 11 | 2,859,809 | 17,635,937 |
| General | 2016.08.01–2019.07.31 | 36 | 1,893,174 | 15,953,165 |

Note:

*refers to the number of influencers we manually labeled.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

## TABLE III

Experimental results on the labeled datasets.

| Methods | Politics set | | Technology set | | Reddit set | |
|---|---|---|---|---|---|---|
| | *Prec, F1* | *AUC* | *Prec, F1* | *AUC* | *Prec, F1* | *AUC* |
| Followers | 0.484 | 0.725 | 0.582 | 0.775 | - | - |
| TAP | 0.374 | 0.693 | 0.613 | 0.791 | 0.445 | 0.711 |
| TwitterRank | 0.363 | 0.635 | 0.620 | 0.796 | 0.392 | 0.684 |
| ReFluence | 0.394 | 0.679 | 0.330 | 0.635 | 0.471 | 0.751 |
| RR-LT | 0.734 | 0.858 | 0.367 | 0.660 | 0.390 | 0.682 |
| RR-IC | 0.641 | 0.808 | 0.241 | 0.592 | 0.527 | 0.763 |
| CoupledGNN | 0.673 | 0.813 | 0.638 | 0.804 | 0.537 | 0.771 |
| TTIM w/ Attention | 0.612 | 0.783 | 0.683 | 0.814 | 0.632 | 0.801 |
| TTIM w/ LSTM | 0.654 | 0.812 | 0.715 | 0.833 | 0.658 | 0.794 |
| TTIM-Online | 0.779 | 0.871 | 0.786 | 0.853 | 0.691 | 0.795 |
| TTIM | **0.789** | **0.883** | **0.805** | **0.899** | **0.761** | **0.838** |

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

## TABLE IV

A SAMPLE OF TOP-5 TOPICS AND THEIR INFLUENCERS ON LV-SHOOTING AND GENERAL DATASETS.

| LV-shooting set | | | General set | | |
|---|---|---|---|---|---|
| **Topic** | **Top-3 keywords** | **Top-3 influencers** | **Topic** | **Top-3 keywords** | **Top-3 influencers** |
| ”Praying” | praying, family, las | Rihanna; Jake Tapper; YouTube | ”Election” | vote, Trump, thank | Donald J. Trump; Fox News; GOP |
| ”News” | prayforvegas, keep, photo | CNN; CNN Breaking News; Fox News | ”Critics” | review, opinion, justice | Fox News; CNN; MSNBC |
| ”Politics” | congrats, Trump, court | Donald J. Trump; Vice President Mike Pence; Hillary Clinton | ”Health” | health, happy, fit | Health; NBC News Health; Health-Care.gov |
| ”Sports” | yankees, living, player | Bleacher Report; NFL; ESPN | ”News” | news, Trump, visit | The New York Times; ABC News; Washington Post |
| ”Job” | job, hiring, new | President Trump; CNN; GOP | ”Politics” | Trump, Clinton, appreciate | Donald J. Trump; The Hill; Hillary Clinton |

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

## TABLE V

VISUALIZATION ON COEFFICIENTS PARAMETERS WITH RESPECT TO THE FIVE INTERACTIONS.

| **Dataset** | **Politic** | **Tech.** | **LV.** | **General** |
|---|---|---|---|---|
| Following * | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Retweet | 0.5881 | 1.0759 | <u>0.3166</u> | <u>0.1195</u> |
| Quote | **1.3260** | **2.1917** | **1.3040** | **1.1488** |
| Mention | 0.1630 | 1.5427 | 0.5993 | 0.8215 |
| Hashtags | <u>0.0116</u> | <u>0.4164</u> | 0.5201 | 0.7114 |

Note

*: The coefficient for the Following relationship was set as 1.0000 as the reference. Others are normalized with the Following coefficient.