Neural Computing and Applications (2023) 35:12995–13005
https://doi.org/10.1007/s00521-020-05236-4

S.I. : DEEP SOCIAL COMPUTING

Deep spatial–temporal structure learning for rumor detection on Twitter

Qi Huang[1,3] · Chuan Zhou[2,3] · Jia Wu[4] · Luchen Liu[1,3] · Bin Wang[5]

Received: 9 May 2020 / Accepted: 24 July 2020 / Published online: 8 August 2020

Abstract

The widespread of rumors on social media, carrying unreal or even malicious information, brings negative effects on society and individuals, which makes the automatic detection of rumors become particularly important. Most of the previous studies focused on text mining using supervised models based on feature engineering or deep learning models. In recent years, another parallel line of works, which focuses on the spatial structure of message propagation, provides an alternative and promising solution. However, these existing methods in this parallel line largely overlooked the temporal structure information associated with the spatial structure in message propagation. Actually the addition of temporal structure information can make the message propagations be classified from the perspective of spatial–temporal structure, a more fine-grained perspective. Under these observations, this paper proposes a spatial–temporal structure neural network for rumor detection, termed as STS-NN, which treats the spatial structure and the temporal structure as a whole to model the message propagation. All the STS-NN units are parameter shared and consist of three components, including spatial capturer, temporal capturer and integrator, to capture the spatial–temporal information for the message propagation. The results show that our approach obtains better performance than baselines in both rumor classification and early detection.

Keywords Rumor detection · Spatial–temporal structure learning

1 Introduction

A rumor in the social psychology literature is defined as a statement or story whose truth value is unverifiable or deliberately false at the time of publication [1]. Due to the openness and convenience of social media such as Twitter, it is easy to publish and spread rumors on social media. The widespread of rumors on social media, carrying unreal or even malicious information, will have a negative impact on society and individuals. For example, there are more than 500 rumors about Donald Trump and Hillary Clinton on Twitter during the 2016 U.S. presidential election [2]. These rumors spread on Twitter have greatly damaged the reputation of candidates and interfered with the judgement of the voters, thus finally affecting the results of the U.S. presidential election. Therefore, detecting rumors circulated on social media early in its propagation before it gets widely spread is highly desirable and socially beneficial.

However, rumor detection is non-trivial but challenging, since it typically needs the investigative journalism and the check of suspected claims, which is labor-intensive and

✉ Chuan Zhou
zhouchuan@amss.ac.cn

Qi Huang
huangqi@iie.ac.cn

Jia Wu
jia.wu@mq.edu.au

Luchen Liu
liuluchen@iie.ac.cn

Bin Wang
wangbin11@xiaomi.com

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[2] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[4] Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia

[5] Xiaomi AI Lab, Beijing, China

time-consuming. The proliferation of social media makes it worse due to the ever-increasing information load and dynamics [3]. Therefore, it is necessary to develop automatic and assistant approaches to facilitate real-time rumor detection.

For the automatic detection of rumors, most of the previous studies focused on text mining from sequential microblog streams using supervised models based on feature engineering [4–7] or deep learning models [8, 9]. Meanwhile, another parallel line of works, which focuses on the spatial structure of message propagation, provides an alternative and promising solution. Along with this line, for example, the kernel-based method [10, 11] was proposed to model the spatial structure as propagation trees in order to differentiate rumorous and non-rumorous claims by comparing their tree-based similarities. The work [3] tried to model the non-sequential propagation tree structure for learning discriminative features via the recursive neural networks and generate more powerful representations for identifying different types of rumors.

The propagation tree structure (e.g., spatial structure)-based methods obtain more and more attentions in recent years. However, these existing methods largely overlooked the temporal structure information associated with the tree structure in message propagation. Here the temporal structure refers to the sequencing information of all messages in a message propagation. It should be pointed out that the propagation tree structure can be further differentiated by their temporal structures. We use an example in Fig. 1 to explain this point. The three message propagations (a)–(c) are equivalent in terms of spatial structure, i.e., sharing the common tree structure, but their temporal structures are different from each other.

In other words, the three message propagations (a)-(c) are different from the perspective of spatial–temporal structure, a more fine-grained perspective. This observation leads to a natural question: how can we model the message propagation from the spatial–temporal perspective to further improve the performance of automatic rumor detection?

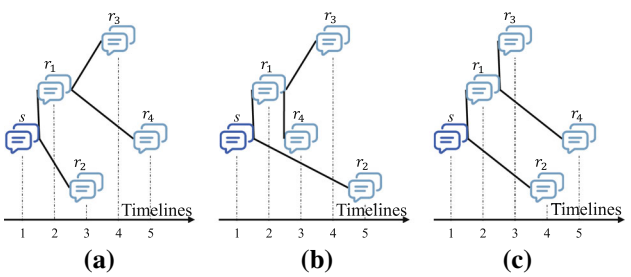


Fig. 1 The three message propagations **a**–**c** share the common tree structure: $s \rightarrow r_1 \rightarrow r_3$, $s \rightarrow r_1 \rightarrow r_4$, $s \rightarrow r_2$. However, their temporal structures are **a** $s \rightarrow r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_4$, **b** $s \rightarrow r_1 \rightarrow r_4 \rightarrow r_3 \rightarrow r_2$, **c** $s \rightarrow r_1 \rightarrow r_3 \rightarrow r_2 \rightarrow r_4$, respectively, and they are totally different. Here letter s denotes the source tweet and r_1, r_2, r_3, r_4 are responsive tweets (e.g., reply or retweet)

To this end, in this paper we propose a **S**patial-**T**emporal **S**tructure **N**eural **N**etwork for rumor detection, termed as STS-NN, which treats the spatial structure and the temporal structure as a whole to model the message propagation. Specifically, the proposed STS-NN model first treats the message propagation as a sequence of messages chronologically, and then applies a STS-NN unit for each message in the sequence. All the STS-NN units are parameter shared and consist of three components, including spatial capturer, temporal capturer and integrator, to capture the spatial–temporal information for each message. The whole STS-NN model can be viewed as an enhanced version of recurrent neural network, and each STS-NN unit captures not only the temporal structure, but also the spatial structure. We evaluate our approach on two public Twitter datasets. The results show that our approach obtains better performance than baselines in both rumor classification and early detection. Our contributions can be summarized in the following threefold:

- We point out that message propagations can be further differentiated by their spatial–temporal structure, a more fine-grained perspective.
- We present a spatial–temporal structure neural network (STS-NN) to model the message propagation for rumor detection, where each STS-NN unit treats the spatial structure and the temporal structure as a whole for feature representation.
- Experiments on two public Twitter datasets show that our STS-NN model (1) achieves better accuracy than the state-of-the-art baselines, and (2) shows a good ability on detecting rumors at a very early stage.

2 Problem statement

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$ be a set of message propagations, where each propagation P_i can be put as

$$P_i := \{s_i, r_{i,1}, r_{i,2}, \ldots, r_{i,|P_i|-1}\}$$

where s_i is the source microblog message and each $r_{i,\cdot}$ is the relevant responsive messages (retweet or reply). Here we assume the relevant responsive messages are in chronological order, i.e., the posting time of message $r_{i,j'}$ is earlier than that of message $r_{i,j''}$ if $j' < j''$. The posting time of message s_i is the earliest one in P_i. Note that although the messages are notated sequentially, there are connections among them based on their retweet or reply relationships. Specifically, this paper models the topological structure of P_i as a *tree structure* $T_i = < P_i, E_i >$ where E_i

denotes a set of directed edges representing the retweet or reply relationship between messages. For example, there are 4 edges, i.e., (s, r_1), (r_1, r_3), (r_1, r_4) and (s, r_2), in the tree structure extracted from any message propagation in Fig. 1.

Here each propagation P_i is associated with a class label from $\mathcal{C}$, which consists of four categories: *non-rumor*, *false rumor*, *true rumor*, and *unverified rumor* [12]. This paper formulates the rumor detection as a supervised classification problem, which aims to learn a classifier f from $\mathcal{P}$ to $\mathcal{C}$.

3 STS-NN model

The STS-NN Model aims to model the message propagation from a spatial–temporal perspective. STS-NN is the abbreviation of **S**patial-**T**emporal **S**tructure **N**eural **N**etwork. The STS-NN model treats the spatial structure and the temporal structure as a whole rather than independent treatment. Follow the recurrent neural network, the STS-NN model treats the message propagation as a sequence of messages chronologically, and then applies a STS-NN unit for each message in the sequence. All the STS-NN units are parameter shared and consist of three components, including spatial capturer, temporal capturer and integrator, to capture the spatial–temporal information for each message.

Specifically, for a given message propagation $P := \{s, r_1, \ldots, r_{|P|-1}\}$ where s is the source message and each r_i is the relevant responsive message (retweet or reply), we assume all the messages in P are in chronological order to represent the temporal structure. For the sake of convenience, we denote s with r_0 afterward. As data preprocessing, for each $t \in \{0, 1, \ldots, |P| - 1\}$, message r_t can be represented as the sum of embeddings of words that r_t contained and we denote this representation as $\boldsymbol{x}_t$. Here the word embeddings are obtained by the SOTA word2vec method [13]. The spatial structure of P is modeled as a *tree* $T = <P, E>$ where E denotes a set of directed edges representing the retweet or reply relationship between messages. Specifically, if message r_t is a retweet or reply to message $r_{t'}$, there will be a directed edge $(r_{t'}, r_t) \in E$ with $0 \le t' < t \le |P| - 1$. To better present the STS-NN model, we first introduce two definitions.

Definition 1 In a given message propagation $P := \{r_0, r_1, \ldots, r_{|P|-1}\}$, message r_{t-1} is defined as the **previous message** of message r_t for $t = 1, 2, \ldots, |P| - 1$.

Definition 2 If a directed edge $(r_{t'}, r_t) \in E$, $r_{t'}$ is defined as the **parent message** of message r_t. We denote the parent message of message r_t as $p(r_t)$.

Note that there is no previous message for source message r_0. Due to the property of tree, the parent message is unique for a non-source message and there is no parent message for source message. Taking ?tic=?>Fig. 1a as an example, the parent of messages r_1 and r_2 is message s, the parent of messages r_3 and r_4 is message r_1 and there is no parent node for source message r_0. In addition, for a message r_t, we can easily conclude that the posting time of its parent message $p(r_t)$ is earlier than that of its previous message r_{t-1}.

Figure 2 illustrates the architecture of the proposed STS-NN unit, where $\boldsymbol{h}_{r_t}$ is the hidden representation of message propagation P up to the occurrence of message r_t, $\boldsymbol{h}'_{r_t}$ is the temporary hidden representation of message propagation P up to the occurrence of message r_t, and o_t is the output classification results based on the hidden representation $\boldsymbol{h}_{r_t}$.

Following the chronological order, the messages in P file in the STS-NN unit one by one, forming a chain with $|P|$ units. In order to maintain the consistency among all STS-NN units, the previous message and the parent message of source message r_0 are both assigned with an empty message $\emptyset$, *i.e.*, $r_{-1} := \emptyset$ and $p(r_0) := \emptyset$. And their corresponding hidden representations are initialized to $\mathbf{0}$, i.e., $\boldsymbol{h}_{r_{-1}} = \boldsymbol{h}_{p(r_0)} = \boldsymbol{0}$.

3.1 Spatial capturer

Given the current message r_t with $0 \le t \le |P| - 1$, the spatial capturer is to collect the hidden representation $\boldsymbol{h}_{p(r_t)}$ of its parent message $p(r_t)$ in order to capture the spatial information of message propagation. Taking Fig. 1a as an example, when the STS-NN unit calculates the hidden representations of messages r_1 and r_2, the spatial capturer will collect the hidden representation of message s (*i.e.* r_0) for them. Similarly, when the hidden representations of r_3 and r_4 are calculated, the spatial capturer collects the hidden representation of message r_1 for them. Here we should point out that the hidden representation $h_{p(r_t)}$ has already been obtained before the current STS-NN unit, since $p(r_t)$ is in front of r_t in the message propagation P.

3.2 Temporal capturer

For the current message r_t, the temporal capturer is designed to capture the temporal features in message propagation P. Considering that the temporal structure is modeled as a message sequence, our temporal capturer utilizes a gated recurrent unit (GRU) [14] to model the message sequence for obtaining the temporal information of message propagation. The input of the temporal capturer is the message representation $\boldsymbol{x}_t$ of the current node r_t and

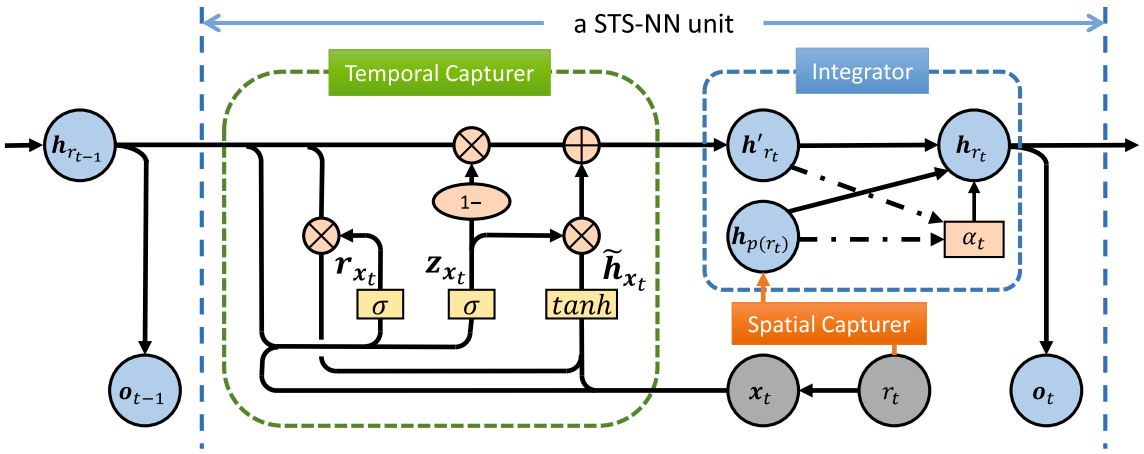


Fig. 2 The architecture of a STS-NN unit, where $\boldsymbol{h}_{r_t}$ is the hidden representation of message propagation P up to the occurrence of message r_t, $\boldsymbol{h}'_{r_t}$ is the temporary hidden representation of message propagation P up to the occurrence of message r_t, and o_t is the output classification results based on the hidden representation $\boldsymbol{h}_{r_t}$. A STS-NN unit consists of three components, including spatial capturer, temporal capturer and integrator, to capture the spatial–temporal information up to the occurrence of current message. For the current massage r_t, the spatial capturer is to collect the information of its parent message $p(r_t)$, the temporal capturer is to process the information of its previous message r_{t-1} with a gated recurrent unit, and the integrator is to assemble the above two kinds of information by the attention mechanism to obtain $\boldsymbol{h}_{r_t}$, which can be used in the subsequent STS-NN unit

the hidden representation $\boldsymbol{h}_{r_{t-1}}$ of previous message r_{t-1}. The output is the temporary hidden representation $\boldsymbol{h}'_{r_t}$ that will be combined with the hidden representation $\boldsymbol{h}_{p(r_t)}$ of parent message r_t. The whole process is shown as follows:

$$\begin{aligned}
\boldsymbol{r}_{x_t} &= \sigma(\mathbf{W}_r \cdot [\boldsymbol{h}_{r_{t-1}}, \boldsymbol{x}_t]) \\
\boldsymbol{z}_{x_t} &= \sigma(\mathbf{W}_z \cdot [\boldsymbol{h}_{r_{t-1}}, \boldsymbol{x}_t]) \\
\tilde{\boldsymbol{h}}_{x_t} &= \tanh(\mathbf{W}_h \cdot [\boldsymbol{r}_{x_t} \odot \boldsymbol{h}_{r_{t-1}}, \boldsymbol{x}_t]) \\
\boldsymbol{h}'_{r_t} &= (1 - \boldsymbol{z}_{x_t}) \odot \boldsymbol{h}_{r_{t-1}} + \boldsymbol{z}_{x_t} \odot \tilde{\boldsymbol{h}}_{x_t}
\end{aligned} \tag{1}$$

where $\boldsymbol{W}_r$, $\boldsymbol{W}_z$ and $\boldsymbol{W}_h$ are weight matrices to be learned. The other symbols are the same with the common GRU. The symbol $\odot$ indicates the element-wise multiplication. The update gate $\boldsymbol{z}_{x_t}$ determines how much memory from the hidden representation $\boldsymbol{h}_{r_{t-1}}$ of previous message r_{t-1} is cascaded to the temporary hidden representation $\boldsymbol{h}'_{r_t}$ of current message r_t. The reset gate $\boldsymbol{r}_{x_t}$ defines how to combine the message representation $\boldsymbol{x}_t$ with the hidden representation $\boldsymbol{h}_{r_{t-1}}$ of previous message r_{t-1}. The representation $\tilde{\boldsymbol{h}}_{x_t}$ is the candidate activation for temporary hidden representation $\boldsymbol{h}'_{r_t}$.

3.3 Integrator

After the above components, we obtain the temporary hidden representation $\boldsymbol{h}'_{r_t}$ and the hidden representation $\boldsymbol{h}_{p(r_t)}$ based on the spatial–temporal structure of message propagation P up to the occurrence of current message r_t. In order to effectively fuse these two kinds of representations, an integrator leveraging the self-attention mechanism [15] is proposed to form a whole hidden representation $\boldsymbol{h}_{r_t}$. The self-attention here is a two-layer perceptron, and the attention coefficients of $\boldsymbol{h}'_{r_t}$ and $\boldsymbol{h}_{p(x_t)}$ are calculated following the formula below:

$$\boldsymbol{\alpha}_{\boldsymbol{h}} = \text{softmax}\left(\frac{\boldsymbol{a} \cdot \tanh(\boldsymbol{W}\boldsymbol{h})}{\sum_{\boldsymbol{h}' \in \{\boldsymbol{h}'_{r_t}, \boldsymbol{h}_{p(r_t)}\}} \boldsymbol{a} \cdot \tanh(\boldsymbol{W}\boldsymbol{h}')}\right) \tag{2}$$

where $\boldsymbol{h} \in \{\boldsymbol{h}'_{r_t}, \boldsymbol{h}_{p(r_t)}\}$, $\boldsymbol{a}$ and $\boldsymbol{W}$ represent the weight vector and weight matrix to be learned, respectively. Then the formalization of aggregation is as follows:

$$\boldsymbol{h}_{r_t} = \sum_{\boldsymbol{h} \in \{\boldsymbol{h}'_{r_t}, \boldsymbol{h}_{p(r_t)}\}} \boldsymbol{\alpha}_{\boldsymbol{h}} \boldsymbol{h} \tag{3}$$

3.4 Output

Given the hidden representation $\boldsymbol{h}_{r_t}$ of message propagation P up to the occurrence of current message r_t, the STS-NN unit uses another softmax function to predict the class of the message propagation P as an output:

$$\boldsymbol{o}_t(P) = \text{softmax}\ (\boldsymbol{V}\boldsymbol{h}_{r_t} + \boldsymbol{b}) \tag{4}$$

where $\boldsymbol{V}$ and $\boldsymbol{b}$ are the weights and bias in output layer that can be learned. The setting of $o_t(P)$ enables the STS-NN model to tell the classification results by only part of the information of message propagation P, i.e., the information up to the occurrence of current message r_t, not necessarily the whole information of P. In other words, this setting enables us to carry out the early rumor detection as we will do in the experimental part.

3.5 Model learning

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$ be a set of message propagations, where each propagation P_k is associated with a class label from $\mathcal{C}$, which consists of four finer-grained categories: *non-rumor*, *false rumor*, *true rumor*, and *unverified rumor* [12]. Hereafter we use a 4-dimensional one-hot vector $\boldsymbol{y}_k$ to represent this class label for each $P_k \in \mathcal{P}$.

In order to capture the complete spatial–temporal information of message propagation, we employ the following cross-entropy loss with the regularization term as the optimization objective function (i.e., loss function) for model learning:

$$\mathcal{L} = -\sum_{k=1}^{|\mathcal{P}|} \boldsymbol{y}_k^T \cdot \ln \boldsymbol{o}_{|P_k|-1}(P_k) + \lambda ||\boldsymbol{\Theta}||_2^2 \tag{5}$$

where λ is a trade-off coefficient, $\boldsymbol{\Theta}$ denotes the set of all parameters, *i.e.* $\boldsymbol{\Theta} = \{\boldsymbol{W}_r, \boldsymbol{W}_z, \boldsymbol{W}_h, \boldsymbol{a}, \boldsymbol{W}, \boldsymbol{V}, \boldsymbol{b}\}$, and the notation $||\boldsymbol{\Theta}||_2^2$ is defined as $||\boldsymbol{\Theta}||_2^2 := ||\boldsymbol{W}_r||_2^2 + ||\boldsymbol{W}_z||_2^2 + ||\boldsymbol{W}_h||_2^2 + ||\boldsymbol{a}||_2^2 + ||\boldsymbol{W}||_2^2 + ||\boldsymbol{V}||_2^2 + ||\boldsymbol{b}||_2^2$ to represent regularization term to prevent over-fitting. As an alternative, the optimization objective function in Eq. (5) can also be replaced with following form

$$\mathcal{L} = -\sum_{k=1}^{|\mathcal{P}|} \sum_{t=0}^{|P_k|-1} \boldsymbol{y}_k^T \cdot \ln \boldsymbol{o}_t(P_k) + \lambda ||\boldsymbol{\Theta}||_2^2 \tag{6}$$

to make the STS-NN model to have better early detection ability.

By employing the backpropagation and optimization algorithm, we can obtain the optimal parameters $\boldsymbol{\Theta}^* = \{\boldsymbol{W}_r^*, \boldsymbol{W}_z^*, \boldsymbol{W}_h^*, \boldsymbol{a}^*, \boldsymbol{W}^*, \boldsymbol{V}^*, \boldsymbol{b}^*\}$ with the smallest loss calculated according to Eqs. (5) or (6). The time complexity of training is $O(\sum_{P_i \in \mathcal{P}} |P_i|)$, which is proportional to the total number of STS-NN units.

3.6 Rumor detection with the learned STS-NN model

Given the STS-NN model with the parameters $\boldsymbol{\Theta}^*$ and a newcome message propagation $P = \{r_0, r_1, \ldots, r_{|P|-1}\}$, the rumor detection process can be summarized in Algorithm 1, whose time complexity is $O(|P|)$ which is proportional to the total number of messages in P. When many message propagations occur simultaneously, the detection problems can be dealt through parallel computing, since there is no correlation between different message propagations.

Algorithm 1 Rumor Detection

Input: the model parameters Θ^*, a newcome message propagation $P = \{r_0, r_1, \cdots, r_{|P|-1}\}$ where the messages are in chronological order with the spatial structure $T =< P, E >$

Output: the classification probability of P

1: Initialize each message $r_t \in P$ with a representation $\boldsymbol{x}_t$ which is the sum of embeddings of words r_t contains
2: Initialize $\boldsymbol{h}_{r_{-1}} = \boldsymbol{h}_{p(r_0)} = \boldsymbol{0}$
3: **for** $t = 0, 1, ..., |P| - 1$ **do**
4: Calculate the $\boldsymbol{h}'_{r_t}$ by Eq. (1)
5: Calculate the $\boldsymbol{\alpha}_{\boldsymbol{h}'_{r_t}}$ and $\boldsymbol{\alpha}_{\boldsymbol{h}_{p(r_t)}}$ by Eq. (2)
6: Calculate the $\boldsymbol{h}_{r_t}$ by Eq. (3)
7: **end for**
8: Calculate the $\boldsymbol{o}_{|P|-1}(P)$ by Eq. (4)
9: **return** the classification probability $o_{|P|-1}(P)$ of message propagation P

We should point out that Algorithm 1 can be used to classify not only a finished message propagation, but also an ongoing message propagation. For example, we can observe a message propagation P' is ongoing and total T messages in P' have occurred up to the current time. Under this situation, we can apply Algorithm 1 to $P' = \{r_0, , r_1, \ldots, r_{T-1}\}$ and use the classification probability $o_{T-1}(P')$ to classify P', which is just the key idea of early rumor detection in the experimental part.

4 Experiments

In this section, we conduct extensive experiments to verify the performance of the proposed STS-NN model on rumor classification and early detection tasks. The reproducible codes and datasets used in this paper are available at https://github.com/201518018629031/STS-NN.

4.1 Datasets

We conduct experiments on two publicly available Twitter datasets: Twitter15 and Twitter16, which have been widely adopted as standard data in the field of rumor detection Ma et al. [11], Liu and Wu [16], Ma et al. [3], Yuan et al. [17]. Twitter15 dataset contains 1490 tweets propagations and Twitter16 contains 818 tweets propagations with their more details shown in Table 1. Each tweet propagation is labeled with one of four types, including non-rumor, false rumor, true rumor and unverified rumor. Following the same setting in the original paper [11], we perform fivefold cross-validation on datasets and calculate the micro-average accuracy (or Acc. for short) of four categories and the F_1 measure of each category to evaluate the model performance.

For each rumor category $c \in \mathcal{C}$, the classification problem can be viewed as a binary one. Let TP_c denote the number of positive samples that are predicted as positive

Table 1 Statistics of Twitter15 and Twitter16

Statistic	Twitter15	Twitter16
# of source tweets	1490	818
# of users	276,663	173,487
# of tweets	331,612	204,820
# of non-rumors	374	205
# of false-rumors	370	205
# of true-rumors	372	207
# of unverified rumors	374	201

ones (i.e., True Positive), FN_c denote the number of positive samples that are predicted as negative ones (i.e., False Negative), and FP_c denote the number of negative samples that are predicted as positive ones (i.e., False Positive). The formulas of Acc. and F_1 measure can be put as below:

$$\text{Acc.} = \frac{\sum_{c\in\mathcal{C}} \mathrm{TP}_c}{\sum_{c\in\mathcal{C}} \mathrm{TP}_c + \sum_{c\in\mathcal{C}} \mathrm{FP}_c} \tag{7}$$

$$F_1(c) = \frac{2 * P_c * R_c}{P_c + R_c} \tag{8}$$

where P_c and R_c denote the accuracy and recall of rumor category c as follows:

$$P_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}. \tag{9}$$

4.2 Comparison methods

We compare our proposed STS-NN model with 5 state-of-the-art baselines on rumor classification and early detection tasks.

- **DTR**: a Decision-Tree-based Ranking model proposed by Zhao et al. [18], which clusters the signal tweets and selects the top k clusters as rumors.
- **RFC**: a Random Forest Classifier proposed by Kown et al. [5], which identifies characteristics of rumors by examining the temporal, structural and linguistic aspects of propagation.
- **SVM-TK**: a SVM classifier using a Tree-based Kernel to compute the similarity between propagation tree structures for rumor detection [11].
- **GRU-RNN**: a RNN with GRU units to model the sequential structure of relevant messages for rumor detection [8].
- **TD-RvNN**: a Recursive Neural Network based on the Top-Down traversal direction of message propagation tree [3].

Note that although there emerge some strong methods recently, such as PPC [16] and GLAN [17], they use user information to guide model learning. Since we focus on, to what extent, the rumor detection can be solved if only the spatial–temporal structure information is available, we do not compare with them in our experiments.

4.3 Experimental setup

We adopt the default optimization settings reported in corresponding papers for all comparison methods. We implement our method by PyTorch. The parameters are optimized using Adam algorithm [19], where the learning rate is initialized at 0.005 and gradually decreases during training. We select the best parameter settings based on the performance on the verification set, which is randomly selected from the training set. The size of the verification set is 10% of the whole dataset. We set the dimension of word embedding as 300, the output dimension of $\boldsymbol{h}_r$ as 100. The batch size of the training set is set to 64. We take Eq. (5) as the optimization objective.

4.4 Rumor classification performance

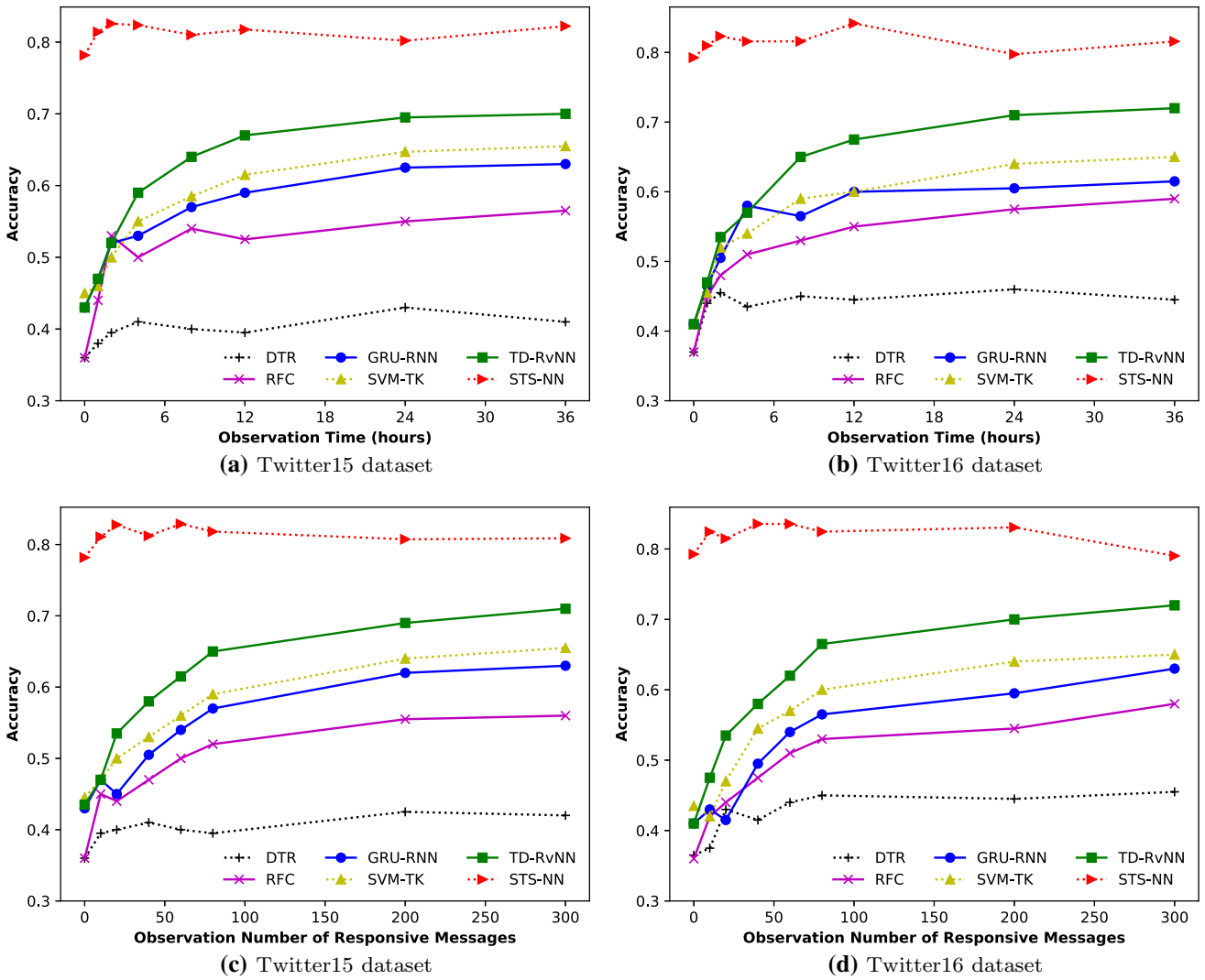
Table 2 shows the comparison of our method with the baselines. We mark the best results in each column on the table. As shown in Table 2, on the whole, our proposed STS-NN model outperforms all the state-of-the-art approaches on both datasets. Specifically, our model

Table 2 Rumor classification results of different methods, where DTR, RFC and GRU-RNN focus on text mining from message streams, while SVM-TK and TD-RvNN focus on tree structure of message propagation

Method	Acc.	NR F_1	FR F_1	TR F_1	UR F_1
(a) Twitter15 dataset					
DTR	0.409	0.501	0.311	0.364	0.473
RFC	0.565	**0.810**	0.422	0.401	0.543
GRU-RNN	0.641	0.684	0.634	0.688	0.571
SVM-TK	0.667	0.619	0.669	0.772	0.645
TD-RvNN	0.723	0.682	0.758	0.821	0.654
STS-NN	**0.809**	0.797	**0.811**	**0.856**	**0.773**
(b) Twitter16 dataset					
DTR	0.414	0.394	0.273	0.630	0.344
RFC	0.585	**0.752**	0.415	0.547	0.563
GRU-RNN	0.633	0.617	0.715	0.577	0.527
SVM-TK	0.662	0.643	0.623	0.783	0.655
TD-RvNN	0.737	0.662	0.743	0.835	0.708
STS-NN	**0.821**	0.739	**0.814**	**0.883**	**0.847**

The best result under each measure is shown in bold

Here NR stands for non-rumor, FR stands for false rumor, TR stands for true rumor, and UR stands for unverified rumor



(a) Twitter15 dataset **(b)** Twitter16 dataset

(c) Twitter15 dataset **(d)** Twitter16 dataset

Fig. 3 Early rumor detection accuracy with the increase in observation time or observation number of responsive messages

achieves an accuracy of 80.9% and 82.1%, respectively, increasing by 8.6% and 8.4% compared with the best baseline. For the non-rumor (NR) category, the baseline RFC slightly outperforms the proposed STS-NN model. The reason behind this is that RFC exploits the number of propagation peaks as temporal features, which is helpful to distinguish the NR category but obviously powerless in other categories.

In addition, we can observe that the performance of deep learning methods is better than that of hand-crafted feature-based methods on the whole. For example, in the first group that focuses on text mining from message streams, GRU-RNN performs better than DTR and RFC except for the F_1 on NR. In the second group that focuses on the tree structure of message propagation, TD-RvNN has better performance than SVM-TK. These results show that hand-crafted feature-based methods lack the power to search for effective features compared with deep learning methods.

We can also obverse that the performance of deep learning methods focusing on the spatial structure of message propagation is superior to that of deep learning methods focusing on text mining. Specifically, TD-RvNN is 8.2% and 10.4% higher than GRU-RNN in the aspect of accuracy on the Twitter15 and Twitter16, respectively. This is because GRU-RNN is a special case of RvNN where each non-leaf node in the tree has only one child node. Meanwhile, the input of GRU-RNN is a sequence of messages, which ignores the spatial structure of message propagation.

The above observations support that the basic idea of STS-NN model is correct and reasonable, since STSNN model is right, a deep learning model than focuses on the spatial–temporal structure of message propagations.

Table 3 Results of ablation study

Method	Acc.	NR F_1	FR F_1	TR F_1	UR F_1
(a) Twitter15 dataset					
STS-NN	0.809	0.797	0.811	**0.856**	**0.773**
w/o Spatial	**0.812**	**0.863**	**0.816**	0.833	0.734
w/o Temporal	0.780	0.737	0.780	0.845	0.757
w/o Attention	0.801	0.831	0.780	0.831	0.760
(b) Twitter16 dataset					
STS-NN	**0.821**	0.739	**0.814**	**0.883**	**0.847**
w/o Spatial	0.772	**0.790**	0.667	0.852	0.779
w/o Temporal	0.779	0.724	0.767	0.832	0.799
w/o Attention	0.799	0.780	0.731	0.870	0.814

The best result under each measure is shown in bold

Here NR stands for non-rumor, FR stands for false rumor, TR stands for true rumor, and UR stands for unverified rumor

4.5 Early rumor detection performance

The early rumor detection can, to a large extent, help to alleviate its harmful impact. In this part, we evaluate the performance of STS-NN model in the aspect of early detection, compared with the baselines in Sect. 4.2. To this end, we design two different scenarios to carry out comparison experiments. One is to classify a message propagation by the information of its first s hours (*i.e.* observing s hours since the source tweet is issued), where $s = 0, 1, 2, 4, 8, 12, 24, 36$. The other one is to classify a message propagation by the information of its first t tweets (*i.e.* observing t retweets or replies after the source tweet is issued), where $t = 1, 10, 20, 40, 60, 80, 200, 300$.

From Fig. 3, we can observe that the proposed STS-NN model consistently outperforms the state-of-the-art baselines in all scenarios, which shows that our model has superior early detection performance than the baselines. In particular, our STS-NN model achieves 82.57% accuracy on Twitter15 and 81.59% accuracy on Twitter16 by observing the information of the first 2 h, and achieves 81.2% accuracy on Twitter15 and 83.5% accuracy on Twitter16 by observing the information of the first 40 tweets. As to the baselines, when observing the information of the first 2 h, DTR, GRU-RNN, TD-RvNN, RFC, and SVM-TK can only achieve 39.5%, 52%, 52%, 53%, and 50% accuracies on Twitter15 and achieve 45.5%, 50.5%, 53.5%, 48%, and 52% accuracies on Twitter16, respectively. Similarly, when observing the information of the first 40 tweets, these baselines can only achieve 41%, 50.5%, 58%, 47%, and 53% accuracies on Twitter15 and achieve 41.5%, 49.5%, 58%, 47.5%, and 54.5% accuracies on Twitter16, respectively.

From Fig. 3, we can also observe that when the observation time is less than 2 h, the accuracy of STS-NN model will increase with the increase in observation time. Similarly, the accuracy will increase with the increase in observation number of responsive messages at the very early stage of message propagation. When the observation time is more than 2 h or the observation number of responsive messages is more than 40, there will be a slight fluctuation in the accuracy trend of STS-NN model. This implies that the spatial–temporal information of the early stage of message propagation is more accurate and valuable for rumor pattern recognition, while the spatial–temporal information of the later stage of message propagation may bring noise for rumor detection.

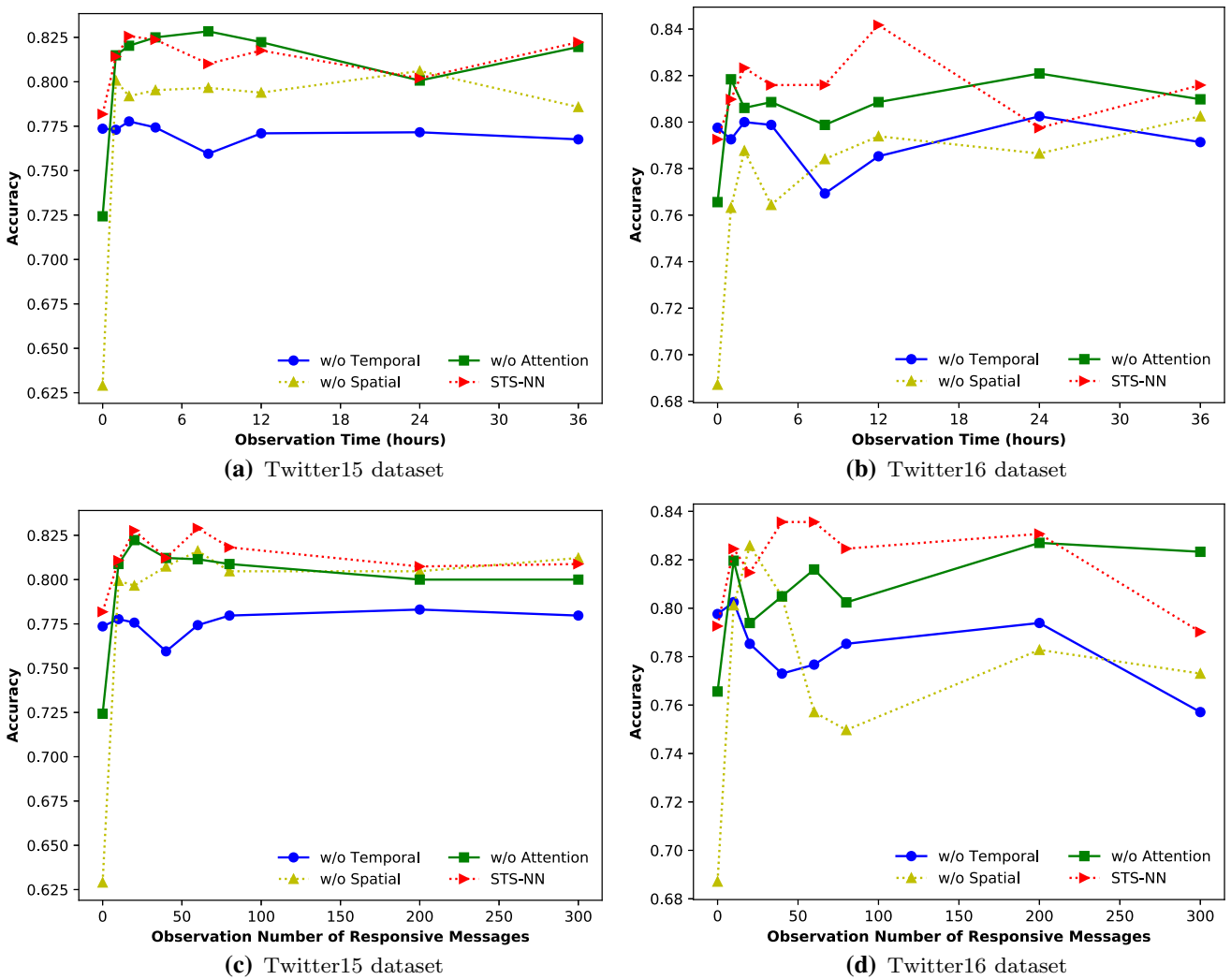
4.6 Ablation study

To study the contribution of each component to the STS-NN unit for rumor detection, we carry out the ablation experiments in this part. The experimental results are shown in Table 3 and Fig. 4. The ablation experiments include the following three variants of STS-NN unit:

- **w/o Spatial**: Removing the spatial capturer component of the STS-NN unit and only exploiting temporal structure information for rumor detection, i.e., replacing Eq. (3) with $\boldsymbol{h}_{r_t} = \boldsymbol{h}'_{r_t}$.
- **w/o Temporal**: Replacing the temporal capturer component of the STS-NN unit with a single layer of perception, *i.e.* replacing Eq. (1) with $\boldsymbol{h}'_{r_t} := \sigma(\mathbf{W}' \cdot \boldsymbol{x}_t)$.
- **w/o Attention**: Replacing the integrator component of the STS-NN unit with an average pooling layer, *i.e.* replacing Eq. (3) with $\boldsymbol{h}_{r_t} = \frac{1}{2}\boldsymbol{h}'_{r_t} + \frac{1}{2}\boldsymbol{h}_{p(r_t)}$.

From Table 3, we can observe that all ablation variants drop some accuracy compared with STS-NN model, except that the w/o Spatial on the Twitter15 dataset is slightly higher than STS-NN model. Specifically, when removing the spatial capturer, the accuracy drops 4.9% on Twitter16. The replacement of the temporal capturer causes the accuracy decreased by 2.9% and 4.2% on two datasets, respectively. And the replacement of the integrator also drops the accuracy from 80.9 to 80.1% and from 82.1 to 79.9% on two datasets. Although the w/o Spatial variant on the Twitter15 dataset is slightly higher than STS-NN model, the accuracy of the w/o Spatial variant drops drastically compared with STS-NN model on the Twitter16 dataset. Overall, the STS-NN model, with all the three components involved, provides a better choice compared with the ablation variants.

For the non-rumor (NR) category, the w/o Spatial variant outperforms the proposed STS-NN model by a large margin. The reason behind this is that the spatial



(a) Twitter15 dataset **(b)** Twitter16 dataset **(c)** Twitter15 dataset **(d)** Twitter16 dataset

Fig. 4 Early rumor detection accuracy of ablation study with the increase in observation time or observation number of responsive messages

structure of non-rumor in Twitter is diverse, which leads to the possibility that the STS-NN model will misclassify part of the non-rumor into others when exploiting the spatial structure of message propagation. However, this does not mean the information of spatial structure is useless. The absence of spatial structure leads to a drastic drop of F_1 from 81.4 to 66.7% for the false rumor (FR) category on the Twitter16 dataset. The same argument applies to the F_1 values for the true rumor (TR) and unverified rumor (UR) categories on both Twitter15 and Twitter16 dataset.

In addition, we also compare the performance of STS-NN model and its ablation variants in the aspect of early detection. Here we adopt the same setting with Sect. 4.5 to carry out comparison experiments. The experimental results are shown in Fig. 4, from which we can also observe that the STS-NN model provides an overall better choice, even though this superiority is not comprehensive. In particular, STS-NN model is more accurate in the early stages of message propagation than the ablation variants.

5 Related works

5.1 Traditional machine learning methods

Most of the early works on rumor detection were based on statistical machine learning. They attempted to learn kinds of supervised learning models by exploiting various statistical features including content-based ones, user-based ones and propagation-based ones that are extracted from message propagations [4, 20]. Subsequently, the *temporal structure* and *spatial structure* contained in message propagations are also proved to be able to provide useful features for rumor detection. For example, Kwon et al. [5] introduced a time-series-fitting model to capture the

temporal structure of message propagations for rumor detection. Ma et al. [7] captured the temporal structure of message propagations by modeling the variation of social contextual features in time series. Wu et al. [10] modeled the spatial structure of tweets propagations as propagation trees and extended SVM classifier with hybrid kernel functions, including RBF kernel and random-walk-based graph kernel, to detect rumors in Sina Weibo. Ma et al. [11] exploited a tree-kernel-based approach to capture the high-order patterns of spatial structure in message propagation for rumor detection.

However, these traditional machine learning methods are typically time-consuming and labor-intensive due to the heavy preprocessing and feature engineering. What's worse, some of the features mentioned above are unavailable, inadequate or even impossible to extract.

5.2 Deep learning methods

To address the above shortcomings of traditional statistical machine learning methods, kinds of deep neural networks were proposed successively to capture the patterns of rumor propagation in recent years. For example, Ma et al. [8] explored a recurrent neural network-based method to capture the variation of semantics contained in the temporal structure of message propagations. Liu and Wu [16] modeled the temporal structure of message propagation by combining the recurrent and convolutional networks. Ma et al. [3] presented a tree-based recursive neural network model to capture the content semantics and propagation cues contained in the spatial structure of message propagation. Huang et al. [21] proposed a graph convolutional network-based model to capture the spatial structure of message propagation for rumor detection.

Generally speaking, message propagation contains both temporal and spatial structure characteristics. However, the existing deep learning-based methods typically model temporal structure and spatial structure separately and do not put them together as a whole for comprehensive modeling. To address this issue, in this paper, we propose a spatial–temporal structure neural network (STS-NN) to model the message propagation for rumor detection.

6 Conclusions

In this paper, we proposed a spatial–temporal structure neural network (STS-NN) to treat the spatial structure and the temporal structure as a whole to model the message propagation for rumor detection. The STS-NN model first views the message propagation as a sequence of messages chronologically, and then applies STS-NN unit for each message in the sequence. All the STS-NN units are parameter shared and consist of spatial capturer, temporal capturer and integrator to capture the spatial–temporal information for each message. Experiments on two public Twitter datasets show that STS-NN model performs better than the state-of-the-art baselines. Especially, the spatial–temporal information of early stage of message propagation is more valuable to STS-NN model for rumor pattern recognition.

Acknowledgements This work was supported in part by the NSFC (No. 11688101 and 61872360), the ARC DECRA (No. DE200100964), and the Youth Innovation Promotion Association CAS (No. 2017210).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. DiFonzo N, Bordia P (2007) Rumor psychology: social and organizational approaches, vol 750. American Psychological Association, Washington
2. Jin Z, Cao J, Guo H, Zhang Y, Wang Y, Luo J (2017) Detection and analysis of 2016 us presidential election related rumors on twitter. In: International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation. Springer, pp 14–24
3. Ma J, Gao W, Wong KF (2018) Rumor detection on twitter with tree-structured recursive neural networks. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 1980–1989
4. Castillo C, Mendoza M, Poblete B (2011) Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web, pp 675–684
5. Kwon S, Cha M, Jung K, Chen W, Wang Y (2013) Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th international conference on data mining. IEEE, pp 1103–1108
6. Liu X, Nourbakhsh A, Li Q, Fang R, Shah S (2015) Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 1867–1870
7. Ma J, Gao W, Wei Z, Lu Y, Wong KF (2015) Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 1751–1754
8. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong KF, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 3818–3824
9. Ruchansky N, Seo S, Liu Y (2017) Csi: a hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on conference on information and knowledge management, pp 797–806
10. Wu K, Yang S, Zhu KQ (2015) False rumors detection on sina weibo by propagation structures. In: 2015 IEEE 31st international conference on data engineering. IEEE, pp 651–662
11. Ma J, Gao W, Wong KF (2017) Detect rumors in microblog posts using propagation structure via kernel learning. In: Proceedings

of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 708–717
12. Zubiaga A, Liakata M, Procter R, Hoi GWS, Tolmie P (2016) Analysing how people orient to and spread rumours in social media by looking at conversational threads. PLoS ONE 11(3):0150989
13. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
14. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1724–1734
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
16. Liu Y, Wu YFB (2018) Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Thirty-second AAAI conference on artificial intelligence
17. Yuan C, Ma Q, Zhou W, Han J, Hu S (2019) Jointly embedding the local and global relations of heterogeneous graph for rumor detection. arXiv:190904465
18. Zhao Z, Resnick P, Mei Q (2015) Enquiring minds: early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th international conference on world wide web, pp 1395–1405
19. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:14126980
20. Yang F, Liu Y, Yu X, Yang M (2012) Automatic detection of rumor on sina weibo. In: Proceedings of the ACM SIGKDD workshop on mining data semantics, p 13
21. Huang Q, Zhou C, Wu J, Wang M, Wang B (2019) Deep structure learning for rumor detection on twitter. In: 2019 international joint conference on neural networks (IJCNN). IEEE, pp 1–8