2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

From Retweet to Believability: Utilizing Trust to Identify Rumor Spreaders on Twitter

Bhavtosh Rath[1], Wei Gao[2], Jing Ma[3], and Jaideep Srivastava[4]
[1]Dept. of Computer Science & Engineering, University of Minnesota, Twin Cities, MN, USA
[2]School of Information Management, Victoria University of Wellington, Wellington, New Zealand
[3]Dept. of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Hong Kong
[4]Qatar Computing Research Institute, Doha, Qatar
rathx082@umn.edu, wei.gao@vuw.ac.nz, majing@se.cuhk.edu.hk, jsrivastava@hbku.edu.qa

***Abstract*—Ubiquitous use of social media such as microblogging platforms brings about ample opportunities for the false information to diffuse online. It is very important not just to determine the veracity of information but also the authenticity of the users who spread the information, especially in time-critical situations like real-world emergencies, where urgent measures have to be taken for stopping the spread of fake information. In this work, we propose a novel machine learning based approach for automatic identification of the users spreading rumorous information by leveraging the concept of *believability*, i.e., the extent to which the propagated information is likely to be perceived as truthful, based on the trust measures of users in Twitter's retweet network. We hypothesize that the believability between two users is proportional to the trustingness of the retweeter and the trustworthiness of the tweeter, which are two complementary measures of user trust and can be inferred from retweeting behaviors using a variant of HITS algorithm. With the retweet network edge-weighted by believability scores, we use network representation learning to generate user embeddings, which are then leveraged to classify users into as rumor spreaders or not. Based on experiments on a very large real-world rumor dataset collected from Twitter, we demonstrate that our method can effectively identify rumor spreaders and outperform four strong baselines with large margin.**

I. Introduction

Social media can be characterized as an ideal platform for generating and spreading false or unverified information. For example, Facebook CEO Mark Zuckerberg's offering money to Facebook users who don't share social media hoaxes is itself a parody of social media hoaxes. Some kind of rumors may be potentially detrimental, and even downright dangerous. A rumor circulating on Facebook and Twitter since December 5, 2015 claimed that Muslim residents of Dearborn, Michigan, held a pro-ISIS march, where protesters were carrying ISIS flags. This rumor was circulated following the mass shooting in San Bernardino, California, by a U.S.-born Muslim who became radicalized while living in the U.S. and whose wife was from Pakistan. On a daily basis, such misinformation originates from social media outlets, rendering the quality and credibility of social media content seriously inferior.

Social psychology literature defines rumor as a story or a statement whose truth value is unverified or deliberately false [1]. Differentiating rumor from truth, or measuring the truthfulness of information directly is technically very challenging. One way to address this is by estimating whether the spreader of the information is trusted or not to what extent by their peers so as to identify the "high-risk" users who are more likely to spread false information online.

The concept of *Trust* in Twitter's retweet network can be described as follows: in general, a user which is referred to as **A**, who receives a post tweeted from user **B**, may intend to share the post with his followers with the action of propagating the information. There are two essential factors that can influence the decision of user **A**, who may choose to retweet the post or not: 1) The *trustworthiness* of user **B**, i.e., the willingness of the network to trust **B**; and 2) the *trustingness* of **A**, i.e., the propensity of **A** to trust the other users in the network. According to the prior research of computational trust such as [23], [24], trustingness and trustworthiness are a pair of complementary measures of user trust in social network and both of them are associated with each network user, where a person having higher trustingness contributes to the trustworthiness of its neighbors to a lower degree, while a higher trustworthiness is a result of lots of neighbors linked to the actor having low trustingness.

Intuitively, users with high trustingness are more likely to spread information online than those with low trustingness since they are more likely to believe what someone tweets. When the circulated message is false, such users will be more likely to become rumor spreaders. On the other hand, users with high trustworthiness are generally less likely to inject or spread false information than those who have low trustworthiness in the sense that the tweets of high trustworthy users are historically retweeted more extensively and they

* This work was done when Wei Gao was affiliated with Qatar Computing Research Institute.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ASONAM'17, July 31 - August 03, 2017, Sydney, Australia
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4993-2/17/07?/$15.00
http://dx.doi.org/10.1145/3110025.3110121

are subjectively more cautious on what they tweet for their own reputation. As a result, the properties of users in terms of information veracity they involve in propagating can be inferred somehow based on the nuance of trust relationships among the users. In this paper, we propose a novel approach for the identification of rumor spreaders based on the concept of *believability*, which is a measure defined on the basis of trustingness and trustworthiness measures. Specifically, believability is the strength of a directed edge between a tweeter **B** and its follower **A**, indicating how strong the potential is for information from **B** to be spread through **A**. The basic idea is that the believability of **A** in **B**'s tweeted message is proportional to the trustingness of the retweeter **A** and the trustworthiness of the tweeter **B**. To this end, we construct the trust network among users, using retweets as a proxy of trust relationship, and automatically learn the user representation in a low-dimension space as embeddings inferred from the re-weighted user network using the believability on its edges, for which we employ a state-of-the-art network embedding algorithm called LINE [26]. Finally, based on the generated user embeddings, we apply supervised learning algorithms such as neural networks to classify if the given user spreading the specific information is a rumor spreader or not.

The contributions of this paper are three-fold:

- To the best of our knowledge, this is the first attempt to identify rumor spreaders on Twitter by exploring the nuance of concepts in computational trust, i.e., trustingness and trustworthiness, for creating a novel measure of believability on the potential of a message being spread from one user to the other.
- We propose a novel technical framework that strengthens the representation of user properties in consideration of information veracity using network feature learning based on a large-scale believability re-weighted trust network. Experimental result demonstrate the superiority of the proposed method over strong baseline approaches.
- We construct a large retweet dataset containing around 2 million users based on a set of rumorous and non-rumorous events gathered from rumor debunking websites, which will be made publicly accessible to research communities.

II. Related Work

The task of rumor detection can be classified into two categories: Rumor information detection (using rumor content) and Rumor spreader detection (using user network). While most research has been done using Natural Language Processing, with some work integrating content and network property [29], no work has been done, to our best knowledge, that takes into consideration only the network property to perform rumor detection.

Automatic detection of false information from social media is based on traditional classifiers stemming from the pioneering study of information credibility on Twitter [4]. In following works [28], [15], [16], [17], [18], [27], [30], different sets of hand-crafted features were proposed and incorporated to determine whether a claim about an event is credible. However, feature engineering is painstakingly labor intensive. Ma et al. [17] proposed a RNN-based method that automatically learns the representations to capture the hidden implications and dependencies of complex signals over time, and achieved better performance due to the effective representation learning capacity of deep neural models.

Various researchers have tried to assign trust scores [2], [12], [20], [21] to nodes in a network to accomplish various tasks. Trust scores can be defined as scores that an algorithm puts on a node in a trust network based on various structural aspects of the node. Eigentrust [12] proposes to rate trust scores of peers in a P2P network. These scores help an ordinary user in the network to identify the trustworthy peers and initiate content download from them. Eigentrust, like Pagerank [14] calculates a single score for each node in the network. However, in this algorithm, one's reputation does not play a part in the weight of the node's trust vote. Other researchers have proposed measures to rank bias and deserve of a node in a network [20]. They used an iterative matrix algorithm to calculate bias and deserve of nodes which reinforce each other.

Roy [23], [24] proposed a pair of complementary measures that can be used to measure trust scores of actors in a social network using involvement of social networks. Based on the proposed measures, an iterative matrix convergence algorithm based on HITS [13] was developed that calculates the trustingness and trustworthiness of each actor in the network. The algorithm runs in $O(k \times |E|)$ time where k denotes the number of iterations and $|E|$ denotes the number of edges in the network. In this paper, we propose a novel measure called *believability* based upon these two complementary measures for assessing the potential of the message being spread from one user to the other, which is used to re-weight the edges of the user trust network. Note that the believability is in essence different from commonly known concept of credibility studied in many papers [4], [9], [11], [19], where credibility is primarily used to measure the quality of content being believed in or that of a user being trusted, but believability here is a measure of "spreadability" of information between *a pair of users* instead of an individual user.

Feature learning has been extensively studied in machine learning. DeepWalk [22] learns node embeddings by exploring local neighborhood of the nodes using truncated random walks. Since the strategy of the random walk is uniform (also DFS-style), it gives no control over the explored neighborhoods. Also, it works only for unweighted, undirected graphs. LINE model [26] proposes a breadth-first strategy to explore neighborhoods. Specifically, it learns a feature representation in two separate phases: first, it learns half of the dimensions by BFS-style simulations over immediate neighbors of nodes, then it learns the other half of dimensions by sampling nodes strictly at a 2-hop distance from the source nodes. This model works for all types of graphs. Node2vec [8] explores diverse network neighborhoods which designs a sampling strategy that smoothly interpolates between BFS and DFS. The assumption is that BFS and DFS are extreme sampling paradigms suited

for structural equivalence (i.e., nodes sharing similar roles) and homophily (i.e., nodes from the sample network community), respectively. Node2vec's sampling strategy accommodates for the fact that these notions of equivalence are not competing or exclusive, and real-world networks commonly exhibit a mixture of both. Considering the weighted, directed nature of our network and the complexity of the learning algorithm, in this paper, we employ LINE algorithm with the 2-hop distance for generating user embeddings from the trust network, where the edges are re-weighted by the believability scores.

III. Trust in Social Media

Trust is an important part of any social interaction, and in the context of social media, researchers have been using social networks widely to understand how trust manifests among users. However, such an abstract concept of trust is generally very hard to compute. In general, trust in a social network is defined as a set of scores assigned to each actor in the network, representing his/her level of trust. Specifically, the level of trust can be manifested by assigning a pair of trust scores to each actor which are termed as trustingness and trustworthiness scores [23]. The former is defined as the propensity of an actor to trust his neighbors in the network, while the latter is defined as the willingness of the network as a whole to trust an individual actor.

For quantifying the trust, we require proxies of trust that can map the social interactions to the original concepts of trust. In the context of network on Twitter, there can be various levels of user interactions acting as the proxies, such as following, retweeting, liking, replying, etc. For example, a user whose tweets are more likely to be retweeted by others is expected to have a high trustworthiness score, while a user who is more likely to retweet others' tweets is expected to have a high trustingness score. Without the loss of generality, in our work, we adopt retweeting interactions as the proxy of trust, and our proposed model is generic and can be straightforwardly extended to accommodate any other kind of proxies. Figure 1 illustrates the trust relationships among the users in a retweet network, where the number of times of retweeting between two users can be used as edge weights.

A. Trustingness and Trustworthiness

To calculate trustingness and trustworthiness scores, we use the TSM algorithm [23] that takes a directed graph as input together with a specified convergence criteria or a maximum permitted number of iterations. In each iteration for every node in the network, trustingness and trustworthiness are computed using the equations below:

$$ti(v) = \sum_{\forall x \in out(v)} \left(\frac{w(v,x)}{1 + (tw(x))^s} \right) \quad (1)$$

$$tw(u) = \sum_{\forall x \in in(u)} \left(\frac{w(x,u)}{1 + (ti(x))^s} \right) \quad (2)$$

where u and v are user nodes, $ti(v)$ and $tw(u)$ are trustingness and trustworthiness scores of v and u, respectively, $w(v,x)$ is the weight of edge from v to x, $out(v)$ is the set of outgoing edges of v, $in(u)$ is the set of incoming edges of u, and s is the involvement score of the network. Involvement is basically the potential risk an actor takes when creating a link in the network, which is set to a constant empirically in [23].

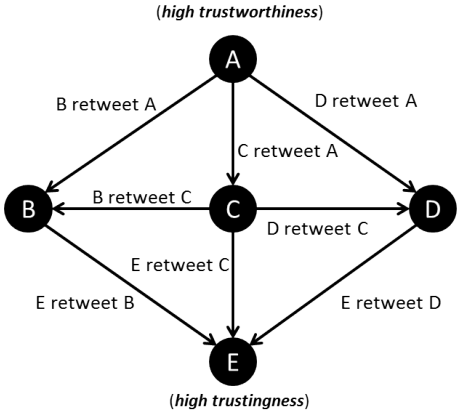

Fig. 1. An illustration of trust in a retweet network, in which the nodes are users and the directed edges indicate the retweet relationship, and each edge can be weighted by the frequency of retweets between two users associated with the edge.

Once the trust scores are calculated for each node in the network, TSM normalizes the scores [23] by adhering to the normalization constraint so that both the sum of trustworthiness and the sum of trustingness of all nodes in the network equal to 1. However, a salient problem of such normalization method lies in that the scale of the scores is dependent of the size of the network. When the network is very large, the resulting scores will become extraordinarily small. To deal with the issue, we perform min-max normalization based on the logarithm of the scores output by TSM to normalize the trustingness and trustworthiness scores into the range of (0,1].

B. Believability

Trustingness and trustworthiness, from different perspectives, are used to measure the strength of trust of each individual user. But they do not quantify the level of trust for two specific users who have retweet relationship, which is nevertheless very important considering the different potential or strength of retweet edges for transmitting messages. When a message is propagated, the intensity of the connection between the tweeter and retweeter would largely determine how fast and how far the message could be transmitted over the network. The original edge weight based on the frequency of retweet interaction of two users cannot satisfy such a need of indication for the "spreadability" of network edges. In this regard, the method of re-weighting the retweet edges properly is very much desirable.

We propose the new concept called *believability*, a quantitative figure that is computed for a directed edge between two nodes used to measure the potential of messages being transmitted through the edge based on the strength of belief between two neighbors on that edge. In the context of retweet,

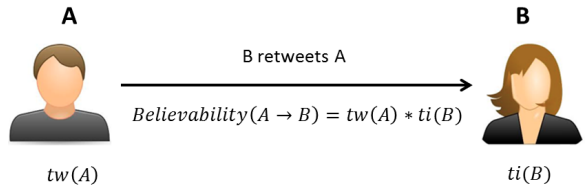

Fig. 2. An illustration of *believability*, which is proportional to the trustworthiness of A and the trustingness of B.

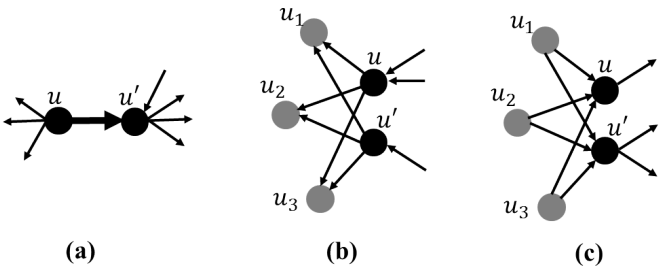

Fig. 3. An illustration of similar users in retweet network, where u and u' are deemed similar in different contexts. Therefore, u and u' should be projected closely in the representation space.

a directed edge from **A** to **B** exists if a tweet of **A** is retweeted by **B**. The believability quantifies the strength that **B** trusts on **A** when **B** decides to retweet **A**. Therefore, **B** is more likely to believe in **A** if:

1) **A** has a high trustworthiness score, i.e., **A** is highly likely to be trusted by other users in the network, or
2) **B** has a high trustingness score, i.e., **B** is highly likely to trust others.

So, the believability score is supposed to be proportional to the two values above, which can be jointly determined and computed as follow:

$$Believability(A \rightarrow B) = ti(A) * tw(B) \tag{3}$$

Figure 2 illustrates the relationship between the believability and the trust measures given a retweet edge in retweet network. The believability score will be used to re-weight the edges so that the representation of users can be reasonably learned with the differentiation of variable spreadability of different edges. The key reason why this can result in better user representation learning is that the inter-user believability score will lead to the random walk being biased to favorably travel towards nodes via high believability edges (see Section IV), thus potentially maximize the transmission of information over the network.

IV. User Representation Learning

In this section, we will discuss how to automatically represent the users based on the re-weighted retweet network using believability scores as the edge weights.

A. Rumorous Users and Retweet Context

We define the retweet network as $G = (V, E)$, where $V = \{u_1, u_2, \ldots, u_n\}$ refers to a set of nodes each representing a user, and $E = \{w_{ij}\}$ is a set of directed edges corresponding to retweet relationship among the nodes in V, which are weighted by believability scores.

Figure 3 illustrates the contexts of network where similar users should be represented closely to each other in the embedding space. Without loss of generality, we illustrate three basic cases of context where two users u and u' reside which should be considered similar and how their similarity is related to rumor propagation:

- **a)** u and u' act as the context of one another and the u-u' weight is strong, suggesting that u' may be a "hardcore fan" of u. If u is a frequent rumor spreader, so potentially very likely is u' because of the generally low veracity of information on the edge; and the other way round, u is more likely being romorous if u' is often rumorous since most of the information u' spreading is from u.
- **b)** u and u' share many common neighbors (like u_1, u_2, u_3) with out-links, implying that they have a large overlapping group of fans. If u often pollute his fans with hearsays while still not losing audience, it is very likely u' being similar to u because otherwise they tends to lose those common followers due to their contrastive styles.
- **c)** Similar but different as (b), u and u' share many neighbors with in-links, indicating that both of them are interested in a common group of sources of messages. If u is a frequent receiver of rumors, u' is inclined to be similar because of substantial overlap of their information source.

As such, considering the commonality of context, similar users need to be projected closely in the representation space for better classification effectiveness.

B. User Embeddings

We adopt the second-order proximity between a pair of nodes in a network-based representation learning method [26] which is called LINE, to learn user embeddings based on the retweet network depicted above. The goal is to embed each user $u_i \in V$ into a lower-dimensional space $\mathbb{R}^d$ by learning a function $f_G : V \rightarrow \mathbb{R}^d$, where d is the dimension of the projected vector. Specifically, for each u_i, let $\overline{v}_i$ denote the embedding of u_i as a node and $\overline{v}'_i$ be the representation of u_i when treated as a specific context of other nodes. For each edge $u_i \rightarrow u_j$, the conditional probability of u_j being generated by u_i as context is defined as follow:

$$p(u_j|u_i) = \frac{exp(\overline{v}'^{\intercal}_j \cdot \overline{v}_i)}{\sum_{k=1}^{|V|} exp(\overline{v}'^{\intercal}_k \cdot \overline{v}_i)} \tag{4}$$

Given this definition, the nodes sharing similar contexts will have similar conditional distributions over the entire set of nodes. To preserve the context proximity, the objective is to make $p(u_j|u_i)$ be close to its empirical distribution $\hat{p}(u_j|u_i)$, where the empirical distribution can be observed from the

weighted social context network. Thus, the objective function is defined as:

$$\min \sum_{(i,j)\in E} \lambda_i * d\big(\hat{p}(u_j|u_i), p(u_j|u_i)\big) \tag{5}$$

where $d(\cdot,\cdot)$ is the distance between two probabilities based on KL-Divergence, λ_i is the prestige of u_i which is set to u_i's out-degree d_i following [26], and the empirical distribution is computed as $\hat{p}(u_j|u_i) = w_{ij}/d_i$.

We use LINE[1] for optimizing equation 5, which provides an efficient solution based on negative sampling of edges and asynchronous stochastic gradient descent over mini-batches of the sampled edges for parameter update.

V. Identifying Spreaders of Misinformation

We use Recurrent Neural Network (RNN) as classification model for two reasons: Firstly, our data is based on time sequence. i.e. retweets are sequential in nature. Secondly, the training data is of variable length. i.e. the source tweets can have different number of retweets. It is important to note that there is no fixed time interval between two successive retweets. So we can safely consider that the data is not a time series.

A. RNN-based User Model

An RNN is a type of feed-forward neural network that can be used to model variable-length sequential information such as sentences or time series. A basic RNN is formalized as follows: given an input sequence $(x_1, \ldots, x_T)$, for each time step, the model updates the hidden states $(h_1, \ldots, h_T)$ and generates the output vector $(o_1, \ldots, o_T)$, where T depends on the length of the input. From $t = 1$ to T, the algorithm iterates over the following equations:

$$\begin{aligned} h_t &= tanh(Ux_t + Wh_{t-1} + b) \\ o_t &= Vh_t + c \end{aligned} \tag{6}$$

where *U*, *W* and *V* are the input-to-hidden, hidden-to-hidden and hidden-to-output weight matrices, respectively, *b* and *c* are the bias vectors, and $tanh(.)$ is a hyperbolic tangent nonlinearity function.

Typically, the gradients of RNNs are computed via back-propagation through time [25]. In practice, because of the vanishing or exploding gradients [3], the basic RNN cannot learn long-distance temporal dependencies with gradient-based optimization. One way to deal with this is to make an extension that includes "memory" units to store information over long time periods, commonly known as Long Short-Term Memory (LSTM) unit [10], [7] and Gated Recurrent unit (GRU) [5].

A GRU has gating units that modulate the flow of the content inside the unit, but a GRU is simpler than LSTM with fewer parameters. The following equations are used for a GRU unit in hidden layer [5]:

$$\begin{aligned} z_t &= \sigma(x_t U_z + h_{t-1} W_z) \\ r_t &= \sigma(x_t U_r + h_{t-1} W_r) \\ \tilde{h}_t &= tanh(x_t U_h + (h_{t-1} \cdot r_t) W_h) \\ h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \end{aligned}$$

where a reset gate r_t determines how to combine the new input with the previous memory, and an update gate z_t defines how much of the previous memory is cascaded into the current time step, and $\tilde{h}_t$ denotes the candidate activation of the hidden state h_t.

We use the recurrent units of GRU to fit the time steps as the basic identification framework. For each source tweet, all of its retweeting users are ordered in terms of the time stamps that indicate when the different users retweet it. In each step, we input the embedding of the user who retweets the message at the time step. Suppose the dimensionality of the generated user embedding is K. The structure of our GRU-RNN model is illustrated in Figure 4. Note that an output unit is associated with each of the time steps, which uses *sigmoid* function for the probabilistic output of the two classes indicating the input user is a rumor spreading user or not.

Let g_c, where c denotes the class label, be the ground-truth 2-dimensional multinomial distribution of a user. Here, the distribution is of the form $[1, 0]$ for rumor spreading users and $[0, 1]$ for non-rumor spreading users. For each training instance (i.e., each source tweet), our goal is to minimize the squared error between the probability distributions of the prediction and ground truth:

$$min \quad \sum_c (g_c - p_c)^2 + \sum_i ||\theta_i||^2$$

where g_c and p_c are the gold and predicted distributions, respectively, θ_i represents the model parameters to be estimated, and the L2-regularization penalty is used for trading off the error and the scale of the problem.

B. Basic User Models

Instead of using RNN-based user model, one might come up with some more straightforward models based upon the property of trust.

1) Trustingness-only model: Intuitively, users with high trustingness, who easily trust others, are more likely to spread rumors. Our trustingness-only model simply learns a threshold based on the correlation between the trustingness score and ground truth of users in the training data. With the threshold, the model can easily predict user class given the trustingness of a new user. The model is described as follows:

$$prediction(u) = \begin{cases} \textbf{true} & \text{if } trustingness(u) \geq \mathcal{T}_{ti}; \\ \textbf{false} & \text{otherwise} \end{cases} \tag{7}$$

where $\mathcal{T}_{ti}$ is the threshold of trustingness score to be learned from training.

[1] github.com/tangjianpku

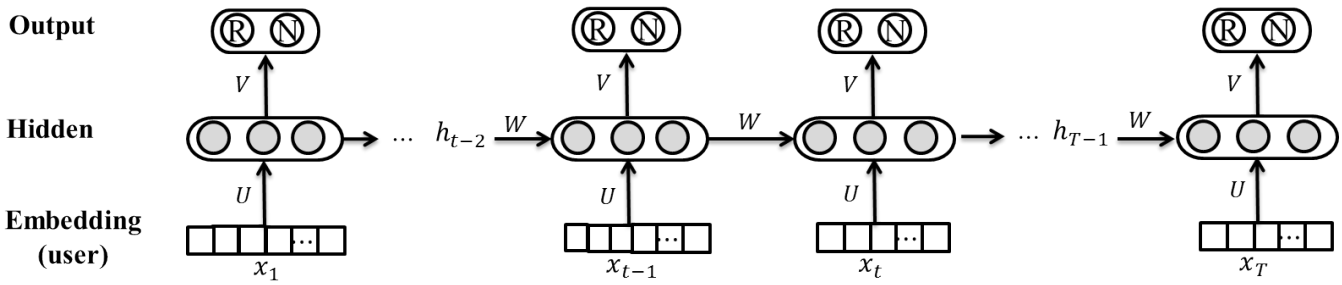

Fig. 4. Our RNN-based model. U, W, V are weight matrices corresponding to the input-to-hidden, hidden-to-hidden and hidden-to-output parameters. R means the user is a rumor spreading user and N means not a rumor spreading user.

2) Trustworthiness-only model: In contrast, the users with high trustworthiness who are more trustworthy are less likely to spread rumors. The trustworthiness-only model similarly learns a threshold from the training data capturing the relationship between the trustworthiness score and ground truth label of users. Similar to Eq. 7, the trustworthiness-only model is given as below:

$$prediction(u) = \begin{cases} \textbf{false} & \text{if } trustworthiness(u) \geq \mathcal{T}_{tw}; \\ \textbf{true} & \text{otherwise} \end{cases} \quad (8)$$

where $\mathcal{T}_{tw}$ is the threshold of trustworthiness score to be learned from training data.

3) Interpolation model: The interpolation model linearly combines the trustingness and trustworthiness scores in such a way that they are interpolated with the appropriate weights to give an optimal prediction on its trust score. The trust score of a given user can be predicted as:

$$T(u) = \alpha * trustingness(u) + (1-\alpha) * trustworthiness(u)$$

where α is the weight that can be fixed during training stage. With the similar thresholding strategy above, we can obtain the threshold $\mathcal{T}_{tr}$ of the interpolated trust score, and the class of user can be predicted as follows:

$$prediction(u) = \begin{cases} \textbf{true} & \text{if } T(u) \geq \mathcal{T}_{tr}; \\ \textbf{false} & \text{otherwise} \end{cases} \quad (9)$$

VI. Experiments and Results

In this section, we will describe the collection of datasets, comparative experiments and the results achieved.

A. Data Collection

We constructed our datasets based on two reference datasets, namely Twitter15 [15] and Twitter16 [17]. The original datasets were used for binary classification of rumor and non-rumor with respect to a given event that contains its relevant tweets. The two Twitter datasets were originally collected by first gathering a set of rumorous and non-rumorous events from rumor debunking website such as www.snopes.com, and then getting the relevant tweets of each event via keyword search on Twitter.

TABLE I
Statistics of our retweet network

Total # of nodes	1,321,872
Total # of edges	19,645,380
Avg in-degree	14.9
Max in-degree	95,303
Min in-degree	0
Avg out-degree	14.9
Max out-degree	58,274
Min out-degree	0

Based on the users appearing in these events, we constructed a large retweet network in the following three steps: (1) We merged the two datasets into one large corpus; (2) We obtained the follow relationships among the users that have appeared across all the events for getting an initial user network[2]. In particular, we treat follow as the basic form of retweet with frequency of 1 for alleviating the sparsity of the retweet network; (3) From each of the events, we extracted popular source tweets with more than 50 retweets[3] that are highly retweeted, and collected all the retweet users for each source status[4]. These retweet relationships are added as edges into the user network above. The statistics on the retweet network are shown in Table I.

We also built user classification dataset based on the source tweets, where each source tweet is associated with a sequence of retweeters ordered by the time they retweeted the source tweet. The ground-truth label for each user is determined by the nature of the source tweet which is retweeted. If the main claim of event is reported rumorous and the source tweet support it, the ground truth label of the users retweeting that source tweet are given as rumor spreader; if the source tweet denies the claim, the users retweeting it are labeled as non-rumor spreader. If the main claim is reported not a rumor, the ground truth label of the users are assigned the other way

[2]We used Twitter API for getting maximum 5k friends of each user, and obtained more friends by requests via Twitter's Web interface.

[3]Though unpopular tweets could be fake, we ignore them as they do not draw much attention and are hardly impactful

[4]Since Twitter API cannot retrieve over 100 retweets, we gathered the retweet users for a given tweet from Twrench (https://twren.ch)

TABLE II
STATISTICS OF USER CLASSIFICATION DATASET

Total # of users	1,055,299
# of users spreading rumors	426,232
# of users not spreading rumors	629,067
Total # of source tweets	3,098
# of rumor source tweets	1,137
# of non-rumor source tweets	1,961
Avg # of retweets per source tweet	496.8
Max # of retweets per source tweet	4,312
Min # of retweets per source tweet	56

round according to the stance of the source tweet the users are retweeting. This ground truth assignment was done semi-automatically where only the stance of source tweets need to be checked manually. The statistics on the user classification dataset are shown in Table II.

B. Settings and Protocols

We ran TSM to get the trust scores based on our retweet network, which is then re-weighted by the believability scores. We adopted the generic setting of TSM involvement parameter $s = 0.391$ by referring to [23]. Then, we learned the user embeddings in the retweet network by running LINE algorithm, where we empirically set the size of embeddings as 200 and kept other parameters as the default settings.

For user classification, we fed the sequence of users of each source tweet into GRU-RNN one at a time and trained the RNN model by employing the derivative of the loss via back propagation [6] with respect to all the parameters and stochastic gradient descent for parameter update. The size of the hidden units is set as 100 and the learning rate as 0.5, and the number of epoch as 200 for ensuring the convergence of RNN. In prediction, the probabilities of the same users across different source tweets are averaged for predicting the final class labels.

We made comparisons among the following six models:

- **Trustingness**: The trustingness-only user model (section V-B1);
- **Trustworthiness**: The trustworthiness-only user model (section V-B2);
- **Interpolation**: The interpolation model (section V-B3);
- **RNN-noweight**: The RNN-based user model using user embeddings obtained from the unweighted retweet network whose edge weights are all 1;
- **RNN-retweet**: The RNN-based user model using user embeddings obtained from the original retweet network without considering trust relationship;
- **RNN-trust**: The RNN-based user model using user embeddings obtained from the retweet network whose edges are re-weighted with believability scores.

For evaluation, we used 5-fold cross-validation and four commonly used metrics: Accuracy, Precision, Recall and F1 measure. The Accuracy is defined over the two classes as:

TABLE III
RESULTS COMPARISON OF DIFFERENT IDENTIFICATION MODELS. '+' DENOTES RUMOR SPREADING USER, AND '-' DENOTES NON-RUMOR SPREADING USER

Method	Class	Accu.	Prec.	Rec.	F_1
Trustingness	+	0.564	0.457	0.429	0.443
	-		0.629	0.655	0.641
Trustworthiness	+	0.574	0.482	0.718	0.577
	-		0.714	0.476	0.571
Interpolation	+	0.575	0.483	0.733	0.582
	-		0.721	0.468	0.568
RNN-noweight	+	0.675	0.704	0.634	0.667
	-		0.651	0.719	0.683
RNN-retweet	+	0.686	0.716	0.644	0.678
	-		0.661	0.731	0.694
RNN-trust	+	0.698	0.726	0.662	0.692
	-		0.674	0.736	0.704

$Accuracy = \frac{\text{\# of correctly predicted users}}{\text{Total \# of users}}$. The rest of the three metrics are defined for each class. For the positive class, i.e., rumor spreading users, the Precision is defined as $Precision(+) = \frac{FP}{TP+FP}$, the Recall is defined as $Recall(+) = \frac{TP}{TP+FN}$, and F_1 is defined as $F_1 = \frac{2*Precision*Recall}{Precision+Recall}$, where TP, FP and FN are true positive rate, false positive rate and false negative rate, respectively. The corresponding metrics for the negative class, i.e., non-rumor spreading users, are defined similarly.

C. Results and Analysis

As Table III shows, the **Trustworthiness** model performs slightly better than **Trustingness** model in terms of accuracy with 1.7% improvement, and the two models are basically comparable. This is attributed to the fact that the two scores are complementary measures derived from a global user interaction network which are essentially the reciprocal sides of trust. So, overall they contribute equally to the spreader detection. However, when looking at finer-grained measures on each of the classes, they demonstrate complementary impact on the performance of detection. For example, in terms of F_1 measure, trustingness performs better on detecting non-spreaders than on spreaders while trustworthiness is better on detecting spreaders than non-spreaders, and the interpolation of the two measure achieves better results than using them individually.

RNN-noweight model employs the RNN algorithm for user classification but does not take into consideration the different strength of the retweet edges. The accuracy for the model is 67.5%, which gives around 17.4% improvement over **Interpolation** model. The F_1 scores on both classes also increased considerably by 14.6% and 20.2% on spreaders and non-spreaders, respectively. Except precision of non-spreaders, all other parameters show an improvement. Thus we can conclude that the user presentation learning based on *even unweighted* retweet relationships and RNN classification improve the ability to identify rumor spreaders.

RNN-retweet model takes into account the edge weights of user interactions in terms of retweet frequency. The accuracy is improved over the unweighted counterpart by 1.6% which

achieves 68.6%. Apart from accuracy, all other metrics also show an improvement in prediction.

RNN-trust model considers the believability scores for the retweet edges based on the complementary trust measures derived from the overall topology of the network. The accuracy is further improved over **RNN-retweet** by 1.7% and reaches 69.8%. In addition, it improves over **RNN-noweight** and **interpolation** models by 3.4% and 21.4%, respectively. In terms of the $precison$, $recall$ and F_1 measure, the performances on both classes also demonstrate consistent improvement. This indicates that our model using trust to re-weight retweet network is advantageous for learning the user representations from the network, thus can improve the final classification effectiveness on users.

Furthermore, we studied the influence of the parameters of TSM algorithm (i.e., involvement score) and the LINE algorithm (i.e., embeddings vector length). The resulting accuracies remain in the similar range as the scores given in Table III when we changed these parameters values. This indicates that our model is not sensitive to the setting of these hyper parameters.

VII. Conclusions and Future Work

In this paper, we conducted a pilot study for the identification of rumor spreading users on Twitter based on computational trust measures. We proposed a machine learning framework by using the novel concept of believability between tweeter and retweeter which is defined based upon the trust measures of individual users in a large-scale retweet network. The key hypotheses are that: 1) The believability between two users is proportional to the trustingness of the retweeter and the trustworthiness of the tweeter, where trustingness and trustworthiness are two complementary trust measures inferred from users' retweet behaviors; 2) In return, using the believability for edge re-weighting on the retweet network can help enhance the learning of feature representation of users in the network, whereby the users' structural properties can be better preserved in terms of neighborhood similarity, signaling the distinctive roles different types of users play in spreading messages. We proposed GRU-based RNN model for user classification using user embeddings as input features that are generated from the believability re-weighted retweet network. Experimental results on a large real-world user classification dataset collected from Twitter demonstrate that the proposed method outperformed four baseline systems with large margin.

The research work could be used to build Social Media Reputation framework (similar to how Feedback scores for ebay users is calculated). We can associate a trust score to the users in social media that would let service providers to authenticate the veracity of information. Low trust users when detected can be monitored to prevent any future occurrence of rumor propagation. Thus this research can be used to make social media a more veracious source of information.

Overall, the performance on detecting rumor spreaders is not very high, indicating the task is difficult. In the future, we plan to extend our model by incorporating additional proxies of trust such as liking and replying. We would also like to make a distinction between regular, non-regular users and bots to study their rumor spreading characteristics. We shall enhance our data collection to alleviate the sparsity of user trust networks which seems an important issue. In addition, we propose to study rumor detection based on user trust networks and compare it with state-of-the-art rumor detection systems. Meanwhile, we would be interested to investigate how to perform multiple detection tasks in rumorous environment such as detecting rumors and their spreaders at the same time.

References

[1] G. Allport and L. Postman. *The psychology of rumor*. Russell & Russell, 1965.

[2] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

[3] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157-166, 1994.

[4] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. *WWW*, 2011.

[5] K. Cho, B. van Merri$^{\prime}$enboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[7] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[8] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. *SIGKDD*, pages 855–864, 2016.

[9] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. TweetCred: Real-Time Credibility Assessment of Content on Twitter. *SocInfo*, 2014.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735-1780, 1997.

[11] Z. Jin, J. Cao, Y. Jiang, and Y. Zhang. News credibility evaluation on microblog with a hierarchical propagation model. *ICDM*, pages 230–239, 2014.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. *WWW*, pages 640–651. ACM, 2003.

[13] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5): 604-632, 1999.

[14] P. Lawrence, B. Sergey, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[15] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah. Real-time rumor debunking on twitter. *CIKM*, 2015.

[16] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong. Detect Rumors Using Time Series of Social Context Information on Microblogging Websites. *CIKM*, 2015.

[17] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and C. Meeyoung. Detecting rumors from microblogs with recurrent neural networks. *IJCAI*, 2016.

[18] J. Ma, W. Gao, and K.-F. Wong. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. *ACL*, 2017.

[19] M.J. Metzger and A.J. Flanagin. Credibility and trust of information in online environments: The use of cognitive heuristics. *Journal of Pragmatics*, 59 (2013): 210-220.

[20] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. *WWW*, pages 567–576. ACM, 2011.

[21] K. O'Hara, H. Alani, Y. Kalfoglou, and N. Shadbolt. Trust strategies for the semantic web. *The 2004 International Conference on Trust, Security, and Reputation on the Semantic Web - Volume 127*, pages 42–51, Aachen, Germany, Germany, 2004.

[22] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. *SIGKDD*, pages 701–710, 2014.

[23] A. Roy. Computational Trust at Various Granularities in Social Networks. PhD thesis, University of Minnesota, 2015.

[24] A. Roy, C. Sarkar, J. Srivastava, and J. Huh. Trustingness and trustworthiness: A pair of complementary trust measures in a social network. *ASONAM*, pages 549–554, 2016.

[25] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533-536, 1986.

[26] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. LINE: Large-scale Information Network Embedding. *WWW*, 2015.

[27] K. Wu, S. Yang, and K. Q. Zhu. False rumors detection on sina weibo by propagation structures. *ICDE*, 2015.

[28] F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on sina weibo. *ACM SIGKDD Workshop on Mining Data Semantics*, 2012.

[29] A. Arif, K. Shanahan, F. Chou, Y. Dosouto, K. Starbird, E. Spiro. How information snowballs: Exploring the role of exposure in online rumor propagation. *CSCW*, 2016.

[30] Z. Zhao, P. Resnick, and Q. Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. *WWW*, 2015.