Detecting Rumors on Online Social Networks Using Multi-layer Autoencoder

Yan Zhang, Weiling Chen, Chai Kiat Yeo, Chiew Tong Lau, Bu Sung Lee

Abstract—Rumors spread on Online Social Networks sometimes can lead to serious social issues. To accurately identify them from normal posts is proved to be of great value. Users' behaviors are different when they post rumors and normal posts. Since rumors only account for a small percentage of all posts, they can be regarded as anomalies. Therefore, we propose an anomaly detection method based on autoencoder to perform rumor detection. In this paper we propose several self-adapting thresholds which can facilitate rumor detection. In addition, we further discuss how the different number of hidden layers of autoencoder can influence the detection performance. The experimental results show that our model achieves a good F1 and a low false positive rate.

Index Terms—Rumor detection; Autoencoder; Online Social Networks.

1 Introduction

Online social networks (OSN) like Twitter and Sina Weibo have around 313 and 282 million active users respectively. Many active users create and repost others' posts every second making the propagation of information faster and easier than ever before. This is a double-edged sword since useful information can easily reach people while false rumors may cause severe public issues from time to time. Therefore, to make the best use of OSN and mitigate the negative effects, it is of great of value to detect false rumors at an early stage.

Existing works typically apply classification methods on various categories of features to perform rumor detection task. One of the earliest work is [1] which grouped all these features into four categories namely Message, User, Topic, and Propagation. More recent works further studied the content of the microblogs and incorporated the analysis of LDA features [2] and sentiment factors [3].

An obvious limitation of these works is that they just consider the overall differences between rumors and normal posts. They did not consider the behaviors of users can also vary from person to person. For example, According to [4] and [5], false rumors will receive more skepticism than true information. When a celebrity posts a tweet, he or she always receives a lot of replies (more than 100) and many of these replies include question marks. Can we therefore claim that every tweet of this celebrity is suspicious of being a rumor? On the other hand, a normal user seldom receives large number of replies from others but somewhat receives 10 replies for a tweet and most of them contain question marks. Can we say this tweet is less suspicious of being a rumor because the number of question marks in the comments are not that many?

Therefore, we argue that it is more meaningful to observe such differences between rumors and non-rumors on individual user's level. If we ignore the behaviorial differences of different users, the differences between rumors and non-rumors will be diluted when we perform rumor detection. [6] proposed a PCA-based model to profile individual user's behaviors. However, PCA method assumes a linear system which is not always applicable. Inspired by this work, we propose to substitute the PCA model with the autoencoder which can ignore such assumption and thus can learn more features of the original data set. With such a model, we are able to profile the normal behaviors of a user and represent the deviation degree of every post he or she posts.

To determine whether a post is actually a rumor, [6] calculated its rank of deviation degree in its recent posts. This method has a limitation because every recent post of a non-rumor should have similar deviation degree. Thus it is difficult for the rank to reflect whether a post is a rumor or not. To overcome the problem, we propose several self-adapting thresholds to help facilitate rumor detection and discuss the experimental results accordingly.

The contributions of this paper are summarized as follows. For each user, we use autoencoder to build models to profile individual user's behaviors based on the recent microblogs they posted. We discuss the influence of autoencoder modules with different number of hidden layers on the detection performances through experiments. We find out that we can represent the deviation degree of one post from its author's normal behaviors pretty well using a 2-hidden-layer autoencoder module. We further propose several self-adapting thresholds in deviation degree to detect rumors with good F1 score and low false positive rate based on the autoencoder module.

The remainder of the paper is organized as follows. Section 2 reviews the related work on anomaly detection with autoencoder model and existing rumor detection methods. Section 3 introduces the detailed process of our rumor detection method, including the data collection method, autoencoder model and these self-adapting thresholds. Section 4 discusses how the different number of hidden layers influence the performance of autoencoder and indicates the efficiency of our rumor detection method compared with other rumor detection method through experiments. Section

- *Y. Zhang, W. Chen, C. K. Yeo, C. T. Lau, B. S. Lee are with the School of Computer Science and Engineering, Nanyang Technological University, S639798, Singapore.*
E-mail: {zhang_yan, wchen015, asckyeo, asctlau, ebslee}@ntu.edu.sg

978-1-5090-1114-8/17/$31.00 ©2017 IEEE

5 concludes our work.

2 Related Work

2.1 Anomaly Detection with Autoencoder

An autoencoder neural network is an artificial neural network which is used for unsupervised learning of efficient codings, i.e. setting the target values to be equal to the inputs. By applying non-linear activation functions, an autoencoder can be used for nonlinear dimensionality reduction. Hence the autoencoder can be used for anomaly detection. The assumption behind is that data has variables correlated with one another and these variables can be mapped to a lower dimensional subspace in which normal samples and anomalous samples appear strongly different [7].

[8] used autoencoder to monitor the wireless communication for spectrum anomaly detection. [9] adapted unsupervised ELM algorithms based on the autoencoder to detect anomalies in large aviation data sets. Such applications can be seen in previous research work but to the best of our knowledge, this is a first study using autoencoder to perform rumor detection task on OSN.

2.2 Existing Rumor Detection Work

[10] and [1] are the earliest work in rumor detection on OSN. They observe various features of rumors to facilitate rumor detection. [11] and [12] tried to analyze these features along with time. Some previous work studied the differences between rumor posting and non-rumor posting. [13] concluded that rumors tend to receive more skepticism. [14] [15] and [16] observed the differences in propagation of rumors and non-rumors.

[17] tried to exploit such differences of posting rumors and non-rumors in user behaviors so as to detect rumors. However, their detection method is based on the aggregate of all the users and thus has much room for improvement.

3 Data and Models

This section has four sub-sections. We describe the overview of our rumor detection method in the first sub-section. In the second sub-section, we then introduce the data collection methods and the features about user posting behaviors. The details of autoencoder (AE) in our learning model are shown in the third sub-section. In the last sub-section, we present the self-adapting thresholds and the rumor detection method using autoencoder.

3.1 Process

We illustrate the overall structure of our rumor detection method in Fig 1. For a suspicious original Weibo, we first extract a set of its recent Weibos posted by the same user and the corresponding features. Then we put the data of recent Weibos into our autoencoder module. We finally use the input and output data of the AE module to detect whether the original Weibo is a rumor or not. The detailed information of the AE module and the rumor detection method is introduced in Sections 3.3 and 3.4.

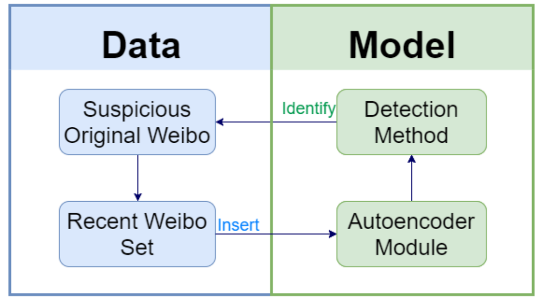

Fig. 1: Rumor Detection Process

3.2 Data Collection

In this paper, we collect data from Sina Weibo. Sina Weibo is one of the most popular social networks in China. Weibo means microblog in Chinese. There is an official Weibo Community Management Center[1] and every user can send a complaint to this center about any suspicious Weibo. Professionals employed by Sina Corporation will seek more information to verify whether the reported Weibo is a rumor or not. The verified false rumors will be published officially at Weibo Community Management Center for everyone to access.

In our experiments, we aim to detect the credibility of n different original Weibos $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$ including both rumors and non-rumors, where the rumors are crawled from the Community Management Center and non-rumors are collected from Sina Weibo's public timeline. For each original Weibo $\mathbf{w}_i$ posted by user u_i at time t_i, we use APIs provided by Sina Weibo[2] to crawl the recent Weibos posted by u_i before t_i, so that we can model the recent posting behavior of the user. Thus we can get its recent weibo set $\mathbf{S}_i = \{\mathbf{w}_{i1}, \mathbf{w}_{i2}, ..., \mathbf{w}_{ik}\}$. Note that $\mathbf{w}_i$ is the first Weibo in $\mathbf{S}_i$ (i.e. $\mathbf{w}_i = \mathbf{w}_{i1}$) and k is the number of Weibos in its recent Weibo set. We manually go through these Weibos collected from Sina Weibo's APIs to make sure they are not rumors. Furthermore, we randomly choose one Weibo in the recent Weibo set as the original non-rumor Weibo for the purpose of avoiding the potential bias of Sina Weibo's public timeline APIs.

To describe one user's recent posting behavior, we not only need to crawl the recent Weibos, but need to extract the features of all the recent Weibos as well. For each Weibo $\mathbf{w}$, we adopt the features in [6], which are shown in Table 1. Thus $\mathbf{w}$ can be represented as a m-vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, where in our case $m = 14$ and the values in the vector represent the feature values of the Weibo.

[6] showed that the features will remain relatively stable in a period of time and the experiments in their paper demonstrated the significance of the features.

For each recent Weibo set $\mathbf{S}$, every Weibo in the set can be represented as a vector using the behavior posting features and thus $\mathbf{S}$ can be represented as a matrix $\mathbf{X}$ with n rows and m columns, where n is the number of Weibos in $\mathbf{S}$. Then we normalize the data $\mathbf{X}$ with z-score method

1. Weibo Community Management Center for false rumor category: http://service.account.Weibo.com/?type=5
2. Sina Weibo API: http://open.weibo.com/wiki/

TABLE 1: Used Features for Rumor Detection

Feature	Description
atti_cnt	The no. of users who have favored this Weibo.
cmt_cnt	The no. of users who have commented this Weibo.
repo_cnt	The no. of users who have reposted this Weibo.
sent_score	Sentiment score of the Weibo.
pic_cnt	The no. of pictures posted in this Weibo.
tag_cnt	The no. of #topics in this Weibo.
mention_cnt	The no. of @mentions in this Weibo.
smiley_cnt	The no. of smileys in this Weibo.
qm_cnt	The no. of question marks in this Weibo.
fp_cnt	The no. of first person pronouns in this Weibo.
length	The length of the Weibo.
is_rt	Whether the Weibo is a repost
hour	The hour the Weibo was posted.
source	How the Weibo was posted.

and input the normalized data into our learning model for further study. We denote the normalized data as $\mathbf{X}$ in the subsequent sections of this paper.

3.3 Autoencoder

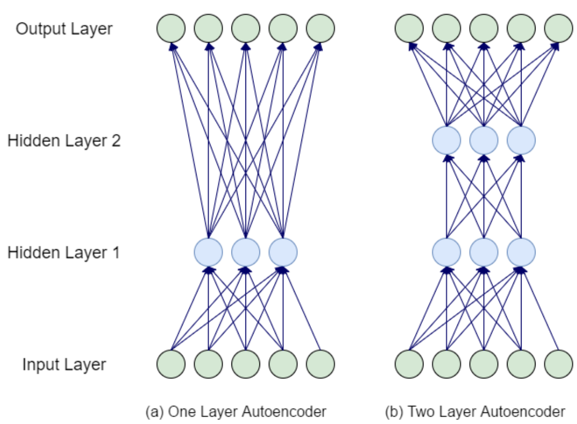

Fig. 2: Proposed autoencoder module.

The artificial neural network we used in our learning model is called autoencoder (AE). Autoencoder is an unsupervised learning method with the purpose of reconstructing its own inputs. It has an input layer, an output layer and one or more hidden layers connecting them. The input layer and output layer have the same number of nodes. In the training process, AE set the target values to be equal to the input value by applying backpropagation to adjust the model parameters.

The input dataset experiences a dimension reduction process in the hidden layer and then the objects in the dataset are reconstructed in the output layer based on the generative characteristics of all objects. Anomalies appear significantly different from their corresponding inputs compared with those normal objects [18]. [7] used AE for anomaly detection considering the above characteristics.

The detailed architecture of our proposed AE module with one hidden layer and two hidden layers can be seen in Fig 2. An autoencoder always has two parts, the encoder and decoder. Take the one hidden layer autoencoder as an example, it first maps the input data $\mathbf{X}$ to a hidden representation $\mathbf{H}$ through an activation function $f(x)$ in the encoder part. The hidden value $\mathbf{H}$ is then mapped back into a reconstruction $\mathbf{O}$ through the same function. Function $f(x)$ is usually a non-linearity function in practice. Particularly, when $f(x)$ is a linearity function, the encoder part is similar to principal component analysis (PCA). The calculation process is concluded in the following formulas. Note that W, b and V, c are the input-to-hidden and hidden-to-output parameters respectively.

$$\begin{aligned} \mathbf{H} &= f(W\mathbf{X} + b) \\ \mathbf{O} &= f(V\mathbf{H} + c). \end{aligned} \tag{1}$$

In our experiments, the activation function is set as $f(x) = SoftPlus(x) = \ln(1 + e^x)$. Usually $SoftPlus$ is used to avoid the vanishing gradient problem in the training process of deep learning models. We set $V = W^\tau$ in the experiments because the decoder part can be considered as the inverse process of the encoder.

3.4 Rumor Detection Method

As we have mentioned in Section 3.3, autoencoder can be applied in anomaly detection using the differences between the input data and reconstruction data. For a recent Weibo set $\mathbf{S}$ with k different Weibos, the corresponding input and output data of the autoencoder module can be represented as $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_k)^\tau$ and $\mathbf{O}(\mathbf{O}_1, \ldots, \mathbf{O}_k)^\tau$, where $\mathbf{X}_j, \mathbf{O}_j$ are the input and output vector of the jth Weibo in $\mathbf{S}$. The differences between the input and output data (i.e. the reconstructed error) for the jth Weibo can be calculated by the euclidean norm as follows:

$$\mathbf{E}_j = ||\mathbf{O}_j - \mathbf{X}_j||_2. \tag{2}$$

The output data can learn the recent posting behavior habit of a user through AE module. The output of normal Weibos will be similar to the recent habit and thus the reconstructed error of an abnormal Weibo (i.e. rumor) is much higher than the error of a normal Weibo (i.e. non-rumor). If we find out a proper threshold for the reconstructed errors, then we can distinguish the rumors from non-rumors.

It is hard to set a fixed threshold (hard threshold) for all the users, because the proper fixed value will be highly correlated to the data set. Instead, we propose several self-adapting thresholds which are calculated based on the property of each recent Weibo set in our experiments. The thresholds are shown in Table 2.

For a set of reconstructed errors, the term mean (μ) is used synonymously to refer to a central value of the errors and the standard deviation (σ) is used to quantify the dispersion of the set of values. Setting the threshold as the combination of μ and σ, such as $threshold = \mu + \sigma$, we can capture the overall characteristics of the normal behavior habit and the data.

Median (med) may be regarded as the "middle" value of a data set. It might be seen as a better indication of central tendency in describing data than mean because it is not skewed so much by extremely large or small values. The first quartile (Q_1) is defined as the middle value between median and the smallest number while the third quartile (Q_3) is defined as the middle value between median and the highest value of the data set. The value $Q_3 - Q_1$ can describe the dispersion of the data set. For the recent Weibo

TABLE 2: Different Types of Thresholds

	Threshold
1	$\mu + \sigma$
2	$med + \sigma$
3	$Q_3 + 1.5(Q_3 - Q_1)$
4	$\mu + 1.5(Q_3 - Q_1)$
5	$med + 1.5(Q_3 - Q_1)$
6	$\mu + \max(1, \sigma)$
7	$med + \max(1, \sigma)$

set with a rumor, the reconstructed error of the rumor is much larger than other error values. Hence, if the number of recent Weibos k is not large enough, μ and σ could skew a lot. Therefore, the second to seventh thresholds are proposed in consideration that μ and σ may not describe the characteristics of those non-rumors in this case.

The third threshold is proposed to find out the minor outliers of a data set. In statistics, $Q_1 - 1.5(Q_3 - Q_1)$ and $Q_3 + 1.5(Q3 - Q1)$ are called ”inner fences”, a point that falls outside the data set’s inner fences is classified as a minor outlier. However, an outlier is an observation point due to some mistakes in the measurement and should be excluded from the data set. Rumors are some Weibos which deviate from a user’s recent behaviorial habit in our paper and should not be considered as minor outliers. This fact notwithstanding, we still adopt the idea of minor outlier but propose smaller thresholds as $med + 1.5(Q3 - Q1)$ and $\mu + 1.5(Q3 - Q1)$ so as not to have rumors being classified as minor outlier and being ignored.

The sixth and seventh threshold are improvements of $\mu+\sigma$ and $med+\sigma$. If a recent Weibo set does not contain one single rumor, the reconstructed errors should remain stable and its σ could be very small. Nearly half of the Weibos in the recent Weibo set shall be wrongly regarded as rumors because $med + \sigma \approx med$ and $\mu + \sigma \approx \mu$.

If the error of the original Weibo in a recent Weibo set is larger than the value of the threshold, we regard this original Weibo as a rumor. We can predict the label of the original Weibo as follows:

$$l_{rumor} = \begin{cases} 1, E_1 > threshold \\ 0, E_1 \leqslant threshold \end{cases} \tag{3}$$

Note that E_1 always denotes the reconstructed error of the original Weibo in each recent Weibo set. The performance of the different types of thresholds set out in Table 2 will be studied and discussed in Section 4.2.

4 Results and Comparisons

In this section we further conduct some experiments to evaluate the performance of our proposed rumor detection method. We have crawled 1,257 rumors and 2,325 non-rumors as original Weibos. In total the experiment data contains 167,731 Weibos including both rumors and non-rumors with their recent Weibo sets. The accuracy (**Acc**), F1 score (**F1**), Precision (**Prec**), Recall (**Recall**) and false positive rate (**FPR**) of the detection results are shown in this section.

4.1 One Hidden Layer vs. Multiple Hidden Layers

Intuitively, we can capture more higher-level features through an autoencoder module with multiple hidden layers. However multiple hidden layer module is computationally intensive and can even cause over-fitting. To compare the performances of AE with different number of hidden layers, we build four AE modules with the number of hidden layers varying from 1 to 4 and set the number of nodes in each hidden layer as 8. We simply give each module a hard threshold (fixed threshold) for all the recent Weibo sets. The hard threshold is calculated in the following way: we traverse from 0 to the maximum of all the error values and the value at which we can achieve the highest F1 is picked as the threshold of that model. The results under the thresholds for each module are shown in Table 3.

TABLE 3: Comparisons of AE with Different Number of Hidden Layers

Layer No.	Acc (%)	F1 (%)	Prec (%)	Recall (%)	FPR (%)
1	88.47	82.88	86.51	79.55	6.71
2	88.97	84.12	85.04	83.21	7.91
3	88.75	83.77	84.83	82.74	8.00
4	88.50	83.14	85.59	80.82	7.35

The AE modules with multiple hidden layers perform better than a single layer. However when the number of hidden layers increase to more than 2, the performance starts to drop. Thus according to our experiments, the 2 hidden layer module performs best. We will adopt the AE structure with 2 hidden layers for the following experiments in the next sub-section.

4.2 Our Model with Different Thresholds vs Other Method

As the best hard threshold to attain the highest F1 score varies depending on the data set, practical feasibility is an issue. Hence we propose 5 self-adapting thresholds as set out in Table 2. We carry out a comparison between the different thresholds in a 2-hidden-layer AE module.

We further compare the performances with the rank method proposed in [6]. We apply the same features in our method as [6] for a fair comparison. The said method ranks the Weibos in a recent Weibo set according to their deviation degree through two strategies and selects the k highest ranked Weibos as rumors. They showed that $k = 3$ always performs well through experiments in their paper thus we set $k = 3$ for comparisons. The rank method is based on a PCA module with 8 principal components (PCs). We have also set the number of nodes in each hidden layer of the AE module to 8 to compare fairly with the rank method. The summary of the results is shown in Table 4.

TABLE 4: Comparisons of AE with Different Thresholds and other Method

Method/Threshold	Acc (%)	F1 (%)	Prec (%)	Recall (%)	FPR (%)
Rank Method	85.68	79.00	81.37	76.77	9.51
$\mu + \sigma$	73.62	70.62	57.96	90.37	35.44
$med + \sigma$	86.40	80.94	79.66	82.26	11.35
$Q_3 + 1.5(Q_3 - Q_1)$	84.48	79.22	74.70	84.33	15.44
$\mu + 1.5(Q_3 - Q_1)$	86.35	81.11	78.83	83.53	12.13
$med + 1.5(Q_3 - Q_1)$	87.86	81.96	85.62	78.60	7.13
$\mu + \max(1, \sigma)$	86.35	81.05	79.00	83.21	11.95
$med + \max(1, \sigma)$	87.07	81.23	82.81	79.71	8.94

From this table we can see that the performances under most of the thresholds in our AE module have higher F1 scores than the Rank method. The Rank method always selects 3 highest ranked Weibo in a recent Weibo set as rumors. However, there is a high possibility that some user will never post rumors and their method will mistakenly set some normal Weibos as rumors. Instead, the self-adapting thresholds in our method only depend on the characteristics of the recent Weibo set and we can thus avoid these mistakes with a proper threshold. From the table, we can also see that the thresholds containing med perform better than those similar thresholds containing μ, which shows that med can be a better indication of central tendency in our case. The threshold $med + \sqrt{3}(Q_3 - Q_1)$ and $med + \max(1, \sigma)$ yield better results than other thresholds in Table 4. The FPR under the threshold $med + \sqrt{3}(Q_3 - Q_1)$ is lower than the other methods.

5 Conclusion

Rumors spread on OSN sometimes can cause serious negative social effects and it is usually impossible for humans to manually inspect millions of posts created everyday. Thus an automated technique to detect rumors on OSN is of high practical value.

In this paper, we model users' normal behaviors through their recent Weibo sets and view rumors as anomalies. A multi-layer structured autoencoder based model is proposed to detect rumors on OSN automatically. We vary the number of hidden layers and observe the corresponding performance and conclude that an autoencoder with 2 hidden layers performs best.

The reconstructed error of the autoencoder is used to represent the deviation degree of Weibos in recent Weibo set. We further propose and discuss several self-adapting thresholds which can be used to distinguish rumors from non-rumors based on deviation degree. According to our experiments, the threshold $med + 1.5(Q_3 - Q_1)$ can achieve the accuracy of 88%, f1 of 82% and FPR of 7%.

Acknowledgments

This research work is partially supported by NTU Grant No. M4081329.020, Nanyang Technological University.

References

[1] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Aspects of rumor spreading on a microblog network," in *Social Informatics*. Springer, 2013, pp. 299–308.

[2] J. Ito, J. Song, H. Toda, Y. Koike, and S. Oyama, "Assessment of tweet credibility with lda features," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 953–958.

[3] T. Kawabe, Y. Namihira, K. Suzuki, M. Nara, Y. Sakurai, S. Tsuruta, and R. Knauf, "Tweet credibility analysis evaluation by improving sentiment dictionary," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 2354–2361.

[4] N. DiFonzo and P. Bordia, *Rumor psychology: Social and organizational approaches.* American Psychological Association, 2007.

[5] A. J. Kimmel, *Rumors and rumor control: A Manager's Guide to Understanding and Combatting Rumors*. Taylor and Francis, 2004.

[6] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, "Behavior deviation: An anomaly detection view of rumor preemption," in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*. IEEE, 2016, pp. 1–7.

[7] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, p. 4.

[8] Q. Feng, Z. Dou, C. Li, and G. Si, "Anomaly detection of spectrum in wireless communication via deep autoencoder," in *International Conference on Computer Science and its Applications*. Springer, 2016, pp. 259–265.

[9] V. M. Janakiraman and D. Nielsen, "Anomaly detection in aviation data using extreme learning machines," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 1993–2000.

[10] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013, pp. 1103–1108.

[11] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "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*. ACM, 2015, pp. 1751–1754.

[12] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of IJCAI*, 2016.

[13] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1395–1405.

[14] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng, "Rumor cascades." in *ICWSM*, 2014.

[15] Y. Yang, K. Niu, and Z. He, "Exploiting the topology property of social network for rumor detection," in *Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on*. IEEE, 2015, pp. 41–46.

[16] S. Wang and T. Terano, "Detecting rumor patterns in streaming social media," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2709–2715.

[17] G. Liang, W. He, C. Xu, L. Chen, and J. Zeng, "Rumor identification in microblogging systems based on users behavior," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 3, pp. 99–108, 2015.

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.