

Rumor Detection based on Social Immune Network

Mingrui Liu^{1,2,3*}, Zexian Xie^{1,*}, Jielin Chen^{1,4}, and Binyang Li¹ (✉)

1 University of International Relations, Beijing 100080, China
{liumingrui, zxxie, byli}@uir.edu.cn

2 School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100093, China

3 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China

4 Soochow University, Suzhou 215008, China
jlchen23@stu.suda.edu.cn

Abstract. The dissemination of rumors on social media will severely endanger political, economic, and social security, which highlights the importance of rumor detection. Current studies mainly focus on capturing content information or propagation pattern of message cascade, but most of these methods do not describe precisely the potential impact among tweets and tweet's influence in message cascade. To tackle the above issue, this paper considers the spread of a rumor on social media as the procedure of immune response in organism, where the users as immune cells, and the retweets as antibodies. A rumor detection model based on Social Immune Network is proposed, named SIN, which is able to utilize the instantaneous rate of change in the number of immune cells (users) and antibodies (retweets) with certain stance to describe tweet's influence. In this process, interactions among different retweets and users with different stances can be explored, thereby investigating the potential impact of each tweet. Extensive experiments conducted based on PHEME dataset show that SIN outperforms State-Of-The-Art method, with 2.8% higher in F1 value of 84.7%, and 2.9% higher in accuracy of 86.2%.

Keywords: Rumor Detection, Stance Classification, Dynamic Immune Network, Social Immune Network.

1 Introduction

With the increasing popularity of social media, the issue of rumor dissemination has become increasingly severe. Rumors that are not detected and removed timely could bring significant risks to a country's political, economic, and social security. Thus, the development of automated rumor detection models has become a prevalent research focus globally.

Due to reliance on manually feature engineering, the traditional machine learning methods [30,28,19,31,15] are weak in the generalization ability. Currently, the main

* These authors contributed equally.

✉ Corresponding author.

solutions for rumor detection are applying deep learning methods [34,26,17,32]. [20] developed a tree-structured Recursive Neural Networks, and [2] adopted Graph Convolutional Network, to capture content and propagation feature. Although many studies used stance labels to aid rumor detection [5,7,1,9], they do not consider the potential impact among tweets with different stances which varies with the change of overall public opinion environment in message cascade.

Fig. 1 illustrates an example of a false rumor cascade with four stances on social media. A-E are the users who comment on the message cascade indeed. F and G are the users who have read the message, but do not comment in the message cascade. They may feel tweets are overly intense and this is a poor discussion environment. In addition, user B appears twice in the message cascade. The first comment (also known as retweet) posted earlier, labeled *null*, and the second comment posted at the end of the message cascade, labeled *clarify*. There is a shift in user stance here. Both situation cannot be adequately modeled and represented with existing models and methods.

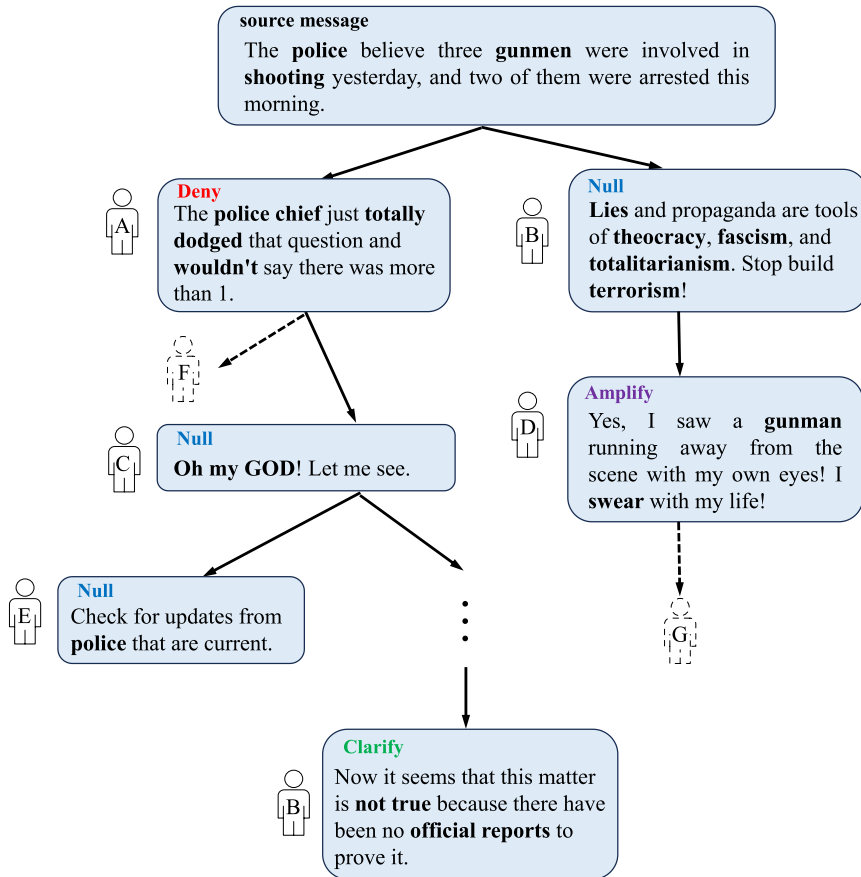


Fig. 1. An instance of a message cascade with four stances on social media. A-E are the users who comment, and all comments are labeled with the stance in color. F and G are users who have read the message, but do not comment on social media. The solid line and directed arrow shows the message propagation path, and dotted line represents potential propagation path.

To summary, there are still a few issues that have not been resolved: (1) There is a lack of methods to describe the users who have read the message, but do not comment in the message cascade, influenced by the overall public opinion environment. They do not consider the potential impact among tweets and users with different stance. Especially, in the message cascade with long propagation chains, most existing methods just consider the interaction of directly connected tweets, rather than investigating the overall impact of the whole message cascade on it. (2) The shift in user stance over time is barely focused on with existing methods, and they merely compute the cumulative count of tweets of different stances.

To solve the above problems, we borrow the idea from dynamic immune network theory of Varela et al. [27], which describes the dynamic quantity changes of various immune molecules in organism. In fact, the spread of a rumor on social media is similar to the invasion of a antigen in organism. This paper considers rumor propagation as antigen invasion, analogizing social networks as immune network, source tweet as antigens, users as immune cells, and retweets as antibodies. We present a rumor detection model based on Dynamic Immune Network theory, named Social Immune Network model(SIN), which can utilize the instantaneous rate of change in the number of immune cells (users) and antibodies (retweets) with certain stance to describe tweet's influence. In this process, interactions among different retweets and users with different stances can be explored, thereby investigating the potential impact of each tweet. SIN is composed of the Tweet Representation Module, Social Immune Propagation Network Module, Immune CheckPoint Module, and Rumor Detection Module.

Our main contributions are of three-folds:

- We considers rumor propagation as antigen invasion, analogizing social networks as immune network, source tweet as antigens, users as immune cells, and retweets as antibodies, presenting a rumor detection model based on social immune network.
- We take the shift in user stance into account, capturing the potential impact among users and tweets with different stance, investigating the overall impact of the whole message cascade on each tweet, and modeling the users who have read the message but do not comment, thereby obtaining the influence of each tweet for rumor detection.
- Experiments on real-world dataset collected from Twitter have demonstrated that our proposed model SIN outperforms state-of-the-art method, with 2.8% higher in F1 value of 84.7%, and 2.9% higher in accuracy of 86.2%.

2 Related Work

Traditional rumor detection methods mainly adopt machine learning approaches for feature extraction and classification. [3] used decision trees for the classification of 68

extracted features, including textual, propagation, and user features of the message. [12] suggested that there are different cyclical characteristics and fluctuations over time between rumors and non-rumors, and propose a periodic time series model based on random forests. [18] proposed to capture the temporal characteristics of rumor's lifecycle based on Support Vector Machines (SVM). These machine learning methods require manual extraction of a large number of features, which are often engineering specific and lack generalization ability, making it difficult to provide high performance.

[32] proposed CAMI, which used Convolutional Neural Network (CNN) to extract key features from contents, and could effectively identify misinformation early. [33] used Graph Convolutional Network (GCN) to model the relationships among source tweet, retweets, and users, and combined it with attention to achieve early rumor detection. [24] used BERT to model the common focus among source tweet and retweets, and proposed an attention layer masking strategy.

More and more scholars utilized the classification of each tweet's stance as an auxiliary subtask for rumor detection. [36] modeled the user's stance before the rumor was confirmed as true or false. [13] merged the user credibility information with content to detect tweet stance for final rumor verification. [29] proposed a hierarchical multi-task learning framework, which utilized GCN to classify stance and then exploited the temporal dynamics of stance evolution for rumor verification. There are also some scholars using generalized stance information to assist in rumor verification. [4] proposed the concept of trigger of each tweet in rumor propagation, aggregating each tweets into cascade with their own trigger weights. [8] modeled and predicted social bot behaviors among users based on Graph Neural Network for aiding early rumor detection. The trigger and robot tag of each tweet can be considered as stance of each tweet, or auxiliary information of each tweet, which depict tweets in a more detailed way.

However, they used the stance information of tweets in a simple way and did not explore the underlying relationships among tweets with different stances or their influence. In this paper, we borrow the dynamic immune network theory [27] to model interactions among each tweet with different stance, exploring potential impact and influence among them.

3 Motivation

The immune network theory of biology was first proposed by Jerne in 1974 and has been further developed since then. This theory explains how the immune system develops through the selective pressure exerted by self-antigens [25]. Since rumor detection control and immune system mechanism have similar behavior, our SIN model was inspired by the dynamic immune network theory [27]. When antigens such as viruses invade an organism, various antibodies gradually become active under the control of the immune system, increasing in number and concentration. Antibodies make antigens removed by complexes with them. After achieving immune effect, the activity level of antibodies gradually decreases. Changes in the number of antibodies can effectively describe the process of antigen invasion and the trigger point of the immune response. This is similar to the overall reaction process of the social media network when rumor

is posted. On social media, users usually have different responses after browsing rumor messages, such as questioning, denying, clarifying, or agreeing. The changes in the number of users or tweets with different stance can reveal the verification of the rumor just like revealing the toxicity of antigens.

[27] suggested that immune systems are cognitive and presented dynamic immune formulas as shows in formula 1, 2, 3. These formulas describe the dynamic changes in the number of various immune molecules in the immune system over time, modeling several scenes to depict immune system. We introduce each of the several scenes in the order of the formulas below.

$$\frac{df_i}{dt} = (-k_1\sigma_i f_i - k_2 f_i + k_3 \text{Mat}(\sigma_i) b_i) \quad (1)$$

$$\frac{db_i}{dt} = (-k_4 b_i + k_5 \text{Prol}(\sigma_i) b_i + k_6) \quad (2)$$

$$\sigma_i = \sum_{j=1}^n m_{ij} f_j \quad (3)$$

Formula 1 respectively aims to calculate the instantaneous rate of change in the number of antibodies in the immune network within an organism. f_i represents the number of the i -th kind of antibody, and $\frac{df_i}{dt}$ represents the instantaneous rate of change of f_i . There are three terms, respectively representing three scenes related to the number of antibodies. Term k_1 depicts the scene of the death due to interaction among antibodies, and σ_i represents the sensitivity of the immune network to the antibody i , further elaborated introduction of σ_i in the description of formula 3 below. Term k_2 depicts the scene of the natural death of antibodies. Term k_3 depicts the scene of the generation of antibodies by immune B cells, and $\text{Mat}()$ function signifies the maturation function of antibodies, akin to $\text{Prol}()$, both smooth threshold functions resembling the sigmoid function.

Formula 2 aims to calculate the instantaneous rate of change in the number of immune B cells. b_i represents the number of the i -th kind of immune B cell, and $\frac{db_i}{dt}$ represents the instantaneous rate of change of b_i , corresponding one-to-one with the i -th kind of antibody mentioned above. There are also three terms, respectively representing three scenes related to the number of immune B cells. Term k_4 depicts the scene of the death of immune B cells. Term k_5 depicts the scene of the proliferation of immune B cells, and $\text{Prol}()$ signifies the proliferation function of immune B cells. Term k_6 depicts the scene of new immune B cell generation, i.e. the differentiation of hematopoietic stem cells into immune B cells.

Formula 3 aims to calculate the sensitivity of the immune network to the i -th kind of antibody, denoted as σ_i , and m_{ij} represents the sensitivity value of antibody j to antibody i .

We map Formulas 1, 2, 3 to the social media network, employing the same formulas to measure the quantities of users (immune B cells) and retweets (antibodies), named social immune network formula. It represents several basic scenes of rumor spread on social media, and the specific introductions to the meanings of it are as following.

Formula 1 in social immune network formula aims to calculate the instantaneous rate of change in the number of retweets with stance i in the social media network, i.e.

$\frac{df_i}{dt}$. The f_i represents the number of retweets with stance i . There are three terms, respectively representing three scenes related to the number of retweets. Term k_1 depicts the scene that two retweets lost their reference value due to their conflicts of opinion, making it difficult for a third user to obtain valuable information from them. σ_i in the term k_1 represents the sensitivity of the social media network to the retweets with stance i . Term k_2 depicts the scene of the natural demise of a retweet that is not noticed by other users. Term k_3 depicts the scene that users actively post a comment (retweet), and $Mat()$ function represents the degree of willingness of users to retweet.

Formula 2 in social immune network formula aims to calculate the instantaneous rate of change of the number of users with stance i in the social media network, i.e. $\frac{db_i}{dt}$. The b_i represents the number of users with stance i . There are also three terms, respectively representing three scenes related to the number of users. Term k_4 depicts the scene that users no longer participate in the current discussion. Term k_5 depicts the scene of the growth of new users who participate in the current discussion influenced by others, and $Prol()$ function represents the degree to which the current discussion attracts new participants. Term k_6 depicts the scene of the growth of new users who spontaneously participate in current discussion.

Formula 3 in social immune network formula aims to calculate the sensitivity of the social media network to the retweets with stance i , denoted as σ_i , and m_{ij} represents the sensitivity value of retweets with stance j to the retweets with stance i .

4 Proposed Model

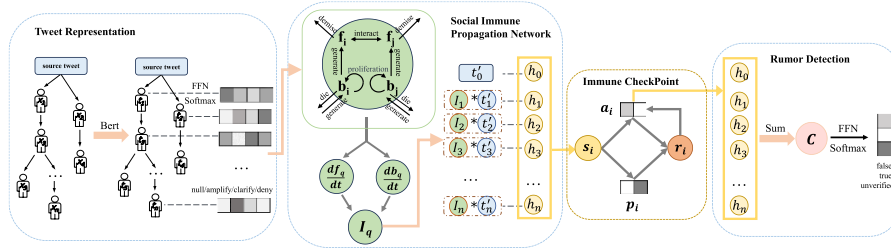


Fig. 2. Overall architecture of our proposed model SIN.

4.1 Problem Definition

Rumor detection is essentially a supervised classification task. Suppose there is a message cascade C on social media, which contains $N + 1$ tweets. It includes the source tweet x_0 and its relevant retweets x_1, x_2, \dots, x_N , and its corresponding rumor publisher u_0 and users u_1, u_2, \dots, u_V . Our rumor detection mission is to first recognize the stance Y_q^s (s refers to the task of stance classification) of each retweet k , which can be *Null*, *Amplify*, *Deny*, or *Clarify*, and then get the rumor detection result Y_d (d refers to the task of rumor detection) of the source tweet as *True*, *False*, or *Unverified*.

We model a rumor detection model based on Social Immune Network, named SIN. The overall architecture of SIN is shown in Fig. 2. There are four modules in SIN, namely Tweet Representation Module (TR), Social Immune Propagation Network Module (SIPN), Immune CheckPoint Module (ICP) and Rumor Detection Module (RD). We will provide a detailed description of the construction of each module in the following section.

4.2 Tweet Representation Module

We directly utilize pre-trained model BertTweet [22] with fine-tuning to encode the content of each tweet and take the [CLS] as representation of each tweet and output. Then feed the output vector into a simple Feed-Forward neural Network (FFN), and use the softmax function to classify the stance of each tweet.

$$t_q = \text{BertTweet}(x_q) \quad (4)$$

$$Y_q^s = \text{softmax}(W_t t_q + b_t) \quad (5)$$

where x_q is tweet q , t_q is the representation vector of tweet q , W_t is weights, and b_t is bias. We use the cross-entropy function as the loss function of Stance classification.

4.3 Social Immune Propagation Network Module

To model potential impact among tweets and subsequently get tweet's influence, we consider users as immune cells, and the retweets as antibodies, we construct SIPN based on Social Immune Network Formulas (as shown in formula 1, 2, 3). The key point of SIPN lies in the acquisition of the sensitivity matrix M and the calculation of tweet's influence. Their detailed introductions are provided below.

Acquisition of Sensitivity Matrix M . One of the key points of SIPN is acquisition of the sensitivity matrix M , which is a asymmetric square matrix $M \in \mathbb{R}^{S \times S}$ located in formula 3. S is the number of stance in social network. m_{ij} represents the sensitivity value of tweets with stance j towards tweets with stance i , describing the potential impact among tweets with different stances in message cascade. We propose to utilize Hidden Markov Model(HMM) [23] to calculate m_{ij} , using the probability of tweets with stance i triggering tweets with stance j to represent the sensitivity value of tweets with stance j towards tweets with stance i . Accordingly, sensitivity matrix M is transition matrix of HMM. The transition probability m_{ij} from tweet q with stance i to tweet $q + 1$ with stance j can be expressed as follow:

$$m_{ij} = P(s_{q+1} = j | s_q = i) \quad (6)$$

where s_q and s_{q+1} refers to the stance of tweet s_q and tweet s_{q+1} , i and j refer to their stance.

It must be emphasized that the sensitivity matrix M is prior knowledge for our SIN model, which means M is already calculated through HMM before our SIN model start training. In other words, M does not assist our model to classify the stance of each tweet, and it only participates in the calculation of σ_i in formula 3 to model potential impact among tweets.

Notably, different social media platforms will have different discussion atmosphere. Thus, there are different potential impacts among tweets with different stances on different social media platforms, which means every social media platform actually has its own sensitivity matrix M . We calculate M on Twitter, and the specific experimental data are shown in Subsection 5.4: Sensitivity Matrix M .

Calculation of Tweet's Influence. The other key point of SIPN is the calculation of tweet's influence, which is described by the instantaneous rate of change in the number of users and retweets with certain stance. The modeling process is introduced below.

Firstly, we calculate the number of users and retweets with different stance in message cascade at the time when each retweet is posted, based on the predicted stance of each tweet obtained from Tweet Representation Module. Matrix b and matrix f are acquired, $b, f \in \mathbb{R}^{N \times S}$, where N is the number of retweets, representing N time nodes, and S is the number of stances (The same goes for the following). The formula is as below:

$$f_{q+1} = f_q + \max(Y_{q+1}^s) \quad (7)$$

where f_q refers to the number of retweets with each stance at the q -th time node (i.e. the time node when the q -th retweet is posted). Y_{q+1}^s is the predicted stance of the $q + 1$ retweet, and function $\max()$ is to set the maximum value of the vector to 1 and the rest to 0.

Moreover, if the current user has previously posted a retweet, our model will check if the stances of the two retweets are the same. If they are different, user's previous stance will subtract by one and the new stance will be added by one. Thus, the shift of stance of user B in Fig. 1 can be described.

Secondly, at each time node of retweet posting, SIN utilize social immune network formula (as shown in formula 1, 2, 3) to calculate the instantaneous rate of change in the number of users and retweets with stance, $\frac{db}{dt}$ and $\frac{df}{dt}$.

In this process, sensitivity matrix M and the number of retweets with different stances at different time nodes of retweet posting are used to calculate matrix σ , $\sigma \in \mathbb{R}^{N \times S}$ and $\sigma_{q,i}$ represents the sensitivity of social immune network to the retweet q with stance i at the time node of the retweet q posting. Next, in term k_3 and k_5 of social immune network formula, σ participates in operation of maturation function $Mat()$ and proliferation function $Prol()$ respectively to calculate the maturity and proliferation rates corresponding to $\sigma_{q,i}$ value, i.e. the probability of users making tweet who have already browsed the message and the probability of attracting more people to browse the message.

Especially, the phenomenon of user F and G probably being afraid to express opposite views in Fig. 1 can be described through term k_3 and k_5 . A poor discussion atmosphere can lead to lower maturity and proliferation rates. Term k_1 , k_2 , k_4 and k_6 also respectively represent the different basic scenes of rumor spread on social media, whose specific meanings can be found in Section 3: Motivation.

It is important to note that all k in the social immune network formula are trainable parameters, and different social media might have their own unique k value due to their unique discussion atmosphere. The experiments of k on Twitter can be found in Subsection 5.5: Ablation Study.

Thirdly, the instantaneous rate of change in the number of users and retweets with stance represents the intensity of change in social immune network at that moment. We consider that user and retweet that can intensively change the social network environment have greater influence. Thus, our model assign coefficient λ_b and λ_f to $\frac{db}{dt}$ and $\frac{df}{dt}$ respectively, to aggregate both to get the influence of each retweet with each implied stance. The calculation formula for the influence vector of q -th retweet is as follows.

$$influence_q = \lambda_b \frac{db_q}{dt} + \lambda_f \frac{df_q}{dt} \quad (8)$$

Fourthly, our model takes the dot product of the influence vector of each retweet $influence_q$ and their respective stance probability prediction vector Y_q^s to perform softmax operation, obtaining the final influence of each retweet in the whole cascade $I_q, I \in R^N$.

$$I_q = influence_q \odot Y_q^s \quad (9)$$

where the symbol \odot represents the function of Hadamard product [6].

Finally, our model multiplies the representations of each retweets in cascade by their own influence, and then add them up with the representation of source tweet.

$$h_q = I_q \times t_q \quad (10)$$

$$C = t_0 + \sum_{q=1}^N h_q \quad (11)$$

where t_q is the representation of the tweet, and C is the representation of the cascade. N is the number of retweets in cascade.

4.4 Immune CheckPoint Module

We mainly refer to the method of zhou et al. [35] to let our model have the ability to detect rumors as early as possible. ICP aims to identify the optimal checkpoint of rumor Detection based on deep Q-learning model [21].

ICP aggregates the representations of all tweets available in the current state, obtaining the state value s_i .

$$s_i = t_0 + \sum_{q=1}^i h_q \quad (12)$$

Then ICP directly using a two-layer Feed-Forward neural Network(FFN) on s_i to calculate action value a_i . a_i represents the action taken by the ICP under state s_i , $a_i \in R^2$, containing *terminate* and *continue*.

$$a_i = \text{FFN}(s_i) \quad (13)$$

$Q^*(s, a)$ is optimal action-value function in ICP, which means the maximum optimal expected return obtained after implementing action a under state s .

$$Q^*(s, a) = E_{s', \varepsilon}(r + \gamma \max_{a'} Q_i(s', a') | s, a) \quad (14)$$

where s refers to the current state, s' refers to the next state, r is the reward value, and γ is the discount rate.

The p_i is the probability of predicted value, $p_i \in R^2$, containing *correct* and *incorrect*. The ICP model calculates the reward value r_i for implementing action a_i under state s_i from the action value a_i and predicted value p_i . Our ICP will update the module parameters based on the reward value r_i , and r_i takes the following value:

$$r_i = \begin{cases} \log M, & \text{terminate with correct prediction} \\ -P, & \text{terminate with incorrect prediction} \\ -\varepsilon, & \text{continue} \end{cases} \quad (15)$$

where M is the cumulative number of correct predictions for reward model making the right choice, P is a large value to punish the model for incorrect predictions, and ε is a small number of punishing for detection delay.

4.5 Rumor Detection Module.

In this module, we directly input the cascade representation C calculated by SIPN to FFN, and then apply a softmax function to get the prediction results of rumor detection.

$$Y^d = \text{softmax}(W_c C + b_c) \quad (16)$$

where W_c is the weights, b_c is the bias, and C is the representation of the cascade. We use cross-entropy function as loss function.

5 Experiment

5.1 Dataset

We validated the effectiveness of our model on the PHEME dataset annotated with trigger information [4], which we consider as stance. PHEME dataset was first established by [36], which includes 5 social events from Twitter. Each event contains a various number of rumor cascades, which has three credibility classifications: *True*, *False*, and *Unverified*. In addition, on the tweet-level, each tweet has its own stance label, including *Null*, *Amplify*, *Deny* and *Clarify*. This dataset has totally 1929 cascades and

26,871 tweets, which is suitable for our exploration of modeling potential impact among tweets and tweet's influence due to its extra special stance annotation and massive conversation propagation structures.

5.2 Experimental Setup

Training Details. We randomly divided the dataset into training, validation, and testing sets at a ratio of 8:1:1. The best-performing hyperparameters in the validation set will be recorded for testing. Our model is trained using the AdamW optimizer [14]. We use BertTweet as our pre-train model and the [CLS] as its output with a dimension of 768. The dropout rate is 0.3, the batchsize is set to 8, the maximum number of training epochs is 50, and the learning rate for all parameters is set to $2e-5$.

Model Comparison. We applied the following models for comparison with our proposed SIN model.

SVM: A SVM-based model [18] to capture the temporal characteristics of rumor's lifecycle.

CNN: A CNN-based model [32], which uses convolution kernels to extract key features from tweet contents.

RNN: A RNN-based model [17], which uses time series among tweets to capture dynamic information.

TreeLSTM: A TreeLSTM-based network [11] to encode tweets as binarized constituency trees, learning the pattern of rumor propagation.

GCN: A GCN-based model [16] to represent rumor cascades as graphs, and extract fine-grained features among tweets.

UGRN: A GRN(Graph Recurrent Networks)-based model [4] to bidirectionally model the rumor propagation graph.

SIN: Our proposed model.

5.3 Experimental Result

We conducted stance classification and rumor detection experiments to evaluate the performance of our model. The experimental results are shown in Table 1, the last row of which, SIN-i, is our ablation experiment with the SIPN removed. Column Random represents that the data of each event in the training, validation and testing sets are randomly shuffled. Column LOEO represents that the data implement Leave-One-Event-Out cross validation [10], which means that the rumor detection events in the validation and testing sets do not appear in the training set.

Table 1. Result of stance classification (Stance) and rumor detection (Detection). Both task contain Random and LOEO mode, which are evaluated by Accuracy (Acc.) and Macro-F1 score (MaF1). Bold: highlights the best performance in each column.

Method	Stance				Detection			
	Random		LOEO		Random		LOEO	
	Acc.	MaF	Acc.	MaF	Acc.	MaF	Acc.	MaF
SVM	0.534	0.519	0.527	0.511	0.722	0.708	0.302	0.286
CNN	0.540	0.524	0.516	0.501	0.756	0.741	0.326	0.308
RNN	0.579	0.562	0.574	0.560	0.801	0.785	0.334	0.314
TreeLSTM	0.553	0.538	0.532	0.514	0.768	0.750	0.342	0.317
GCN	0.567	0.548	0.559	0.542	0.794	0.772	0.347	0.322
UGRN	0.593	0.574	0.588	0.570	0.833	0.819	0.361	0.346
SIN	0.594	0.576	0.590	0.571	0.862	0.847	0.401	0.386
SIN-i	0.554	0.537	0.551	0.532	0.811	0.775	0.349	0.323

Overall, our model achieves the best results in both Random and LOEO mode in terms of stance classification and rumor detection. In particular, SIN outperforms State-Of-The-Art method in Random mode rumor detection, UGRN [4], with 2.8% higher in F1 value of 84.7%, and 2.9% higher in accuracy of 86.2%. This undoubtedly proves the effectiveness of SIN model. The model's performance dropped significantly after removing SIPN, which indicates that SIPN can capture potential impact among each tweet from conversational interactions, thereby modeling tweet's influence throughout the whole message cascade.

In addition, in the LOEO mode, all models dropped dramatically in performance. This suggests that there is a large semantic gap among events, and the robustness of these models is still dissatisfied. The imbalanced label distribution of event in the validation or testing set is also one of the important reasons.

5.4 Sensitivity Matrix M

We calculated the sensitivity matrix M utilizing HMM model [23] on PHEME, obtaining the potential impact coefficient among tweets with different stances on Twitter social network, as shown in Table 2.

Table 2. Sensitivity Matrix M .

i \ j	Null	Amplify	Deny	Clarify
Null	0.865	0.068	0.043	0.025
Amplify	0.787	0.100	0.064	0.048
Deny	0.685	0.064	0.125	0.126
Clarify	0.673	0.059	0.097	0.171

It can be seen that *Null* has the highest probability of transition for all four stances. This is because the dataset contains 77.43% of tweets with *Null* stance. Most people comment on social media just to express their emotions and their opinions are often ambiguous. *Amplify* tweets are more likely followed with *Amplify* retweets and it is the same with *Deny* and *Clarify*. This is a common phenomenon on social media. For example, when someone firmly agrees with something, their standpoint often pass on to more people. That is the so-called "post-truth" era. People tend to construct unconsciously information cocoon and get trapped in it, where information tends to develop in the same guided direction.

5.5 Ablation Study

In order to further explore the modeling effects of our SIN in different scenes during rumor propagation on social media, we conducted an ablation study on k values of the social immune network formula (as shown in formula 1, 2) in this subsection. Specifically, We removed each k values from the formula to investigate their meaning in the formula. The results are shown in the following Table 3.

Table 3. Ablation study of $k_1 \sim k_6$. The $-k_i$ represents the removal of the k_i and its term from Social Immune Propagation Network Module.

Method	Acc.	MaF
SIN	0.865	0.847
- k_1	0.789	0.756
- k_2	0.805	0.771
- k_3	0.795	0.764
- k_4	0.807	0.779
- k_5	0.805	0.775
- k_6	0.811	0.782

It is clear that the performance of our SIN model drops most significantly after removing term k_1 . This shows that conflicts among retweets with different stances can indeed render them unreliable, weakening their respective impact. Removing k_1 makes model fail to identify the truly important tweet. The removal of term k_3 also reduces the performance of SIN model by a large amount. The reason seems to be obvious that the removal of it resulted in SIN model losing the ability to describe the potential impact

among retweets and overall public opinion environment. As user F and G in Fig. 1 mentioned above, the phenomenon cannot be described that they dare not to express opposite views because of poor discussion environment. In addition, term k_3 is the only positive term in formula 1. The removal of it will lead to the negative calculation value of formula 1, which means the influence of each retweet continuously diminishes. This is counter-intuitive, so the bad result from removing k_3 is predictable. As for the other k values, the performance of SIN is significantly weakened after they are removed. This suggests that every k term in the social immune networks formula plays a vital role in our SIN model.

5.6 Early Rumor Detection

Early rumor detection holds greater significance due to its increasing societal demand. By setting different detection deadline, we evaluate the performance of early detection of our model, which means there are only tweets that posted before the corresponding deadlines available for rumor detection. Experiments show that most models reached a relatively stable performance when the detection deadline is 12h. In the PHEME dataset, the longest time span of message cascades ranges from 0 to 728 hours. Therefore, we set the detection deadlines as 0, 1, 2, 3, 4, 8, 12, 16, 20, 24, and 728 hours, and their respective proportions of tweets are 77.4%, 86.2%, 89.7%, 91.7%, 94.80%, 96.16%, 96.86%, 97.44%, 98.08%, 100%.

The results are shown in Fig. 3. Overall, deep-learning-based methods are superior to machine-learning-based methods. Specifically, our SIN achieves a macroF1 score of 0.826 when deadline is 1 hour, achieving better early rumor detection performance than all other compared models. This suggests that our SIN has better timeliness. In the early stages of rumor dissemination, SIN can still model potential impact among tweets and tweet’s influence, and be able to accurately detect rumors.

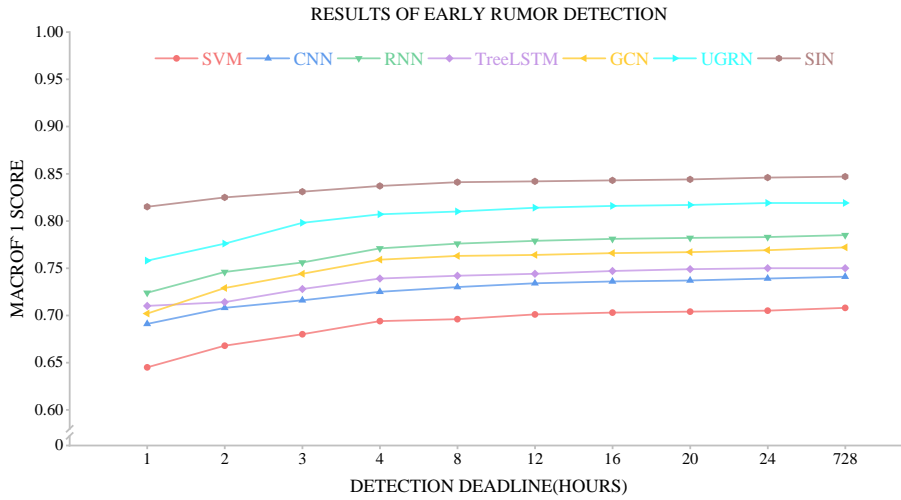


Fig. 3. Experiments results of early rumor detection. We set a restrict to the detection time and evaluate the performance of early rumor detection with macroF1 score.

6 Conclusion

In this paper, we construct a rumor detection model named SIN, based on the social immune network. We analogize users as immune cells, and the retweets as antibodies to describe the potential impact among tweets and tweet's influence with different stances. We get an accuracy of 0.862 which reach the SOTA performance on PHEME dataset.

In the future, we plan to annotate stance information on datasets from other social media platforms such as Weibo and Facebook in order to further explore the sensitivity matrix M and k values, and try to find the similarities and differences of public discussion environment that may influence the feature of rumor propagation on different social media.

7 Acknowledgements

This work was partially done during the author Mingrui Liu's master's degree and Jielin Chen's undergraduate at the University of International Relations.

This paper was partially supported by the National Natural Science Foundation of China (Grant number: 61976066), Beijing Natural Science Foundation (Grant number: 4212031), and Research Funds for NSD Construction, University of International Relations (Grant number: 2021GA07), Student Academic Research Training Project of University of International Relations (Grant number: 3262022SYJ11), Student Academic Research Training Project of University of International Relations (Grant number: 3262023SYJ19).

References

1. Aker, A., Derczynski, L., Bontcheva, K.: Simple open stance classification for rumour analysis. arXiv preprint arXiv:1708.05286 (2017)
2. Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., Huang, J.: Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 549–556 (2020)
3. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684 (2011)
4. Chen, L., Li, G., Wei, Z., Yang, Y., Zhou, B., Zhang, Q., Huang, X.J.: A progressive framework for role-aware rumor resolution. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2748–2758 (2022)
5. Chen, L., Wei, Z., Li, J., Zhou, B., Zhang, Q., Huang, X.J.: Modeling evolution of message interaction for rumor resolution. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 6377–6387 (2020)
6. Hadamard, J.: Théoreme sur les séries entieres. Acta Mathematica 22(1), 55 (1899)

7. Hamidian, S., Diab, M.T.: Rumor detection and classification for twitter data. arXiv preprint arXiv:1912.08926 (2019)
8. Huang, Z., Lv, Z., Han, X., Li, B., Lu, M., Li, D.: Social bot-aware graph neural network for early rumor detection. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 6680–6690 (2022)
9. Kochkina, E., Liakata, M., Augenstein, I.: Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm. arXiv preprint arXiv:1704.07221 (2017)
10. Kochkina, E., Liakata, M., Zubiaga, A.: All-in-one: Multi-task learning for rumour verification. arXiv preprint arXiv:1806.03713 (2018)
11. Kumar, S., Carley, K.M.: Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In: Proceedings of the 57th annual meeting of the association for computational linguistics. pp. 5047–5058 (2019)
12. Kwon, S., Cha, M., Jung, K., Chen, W., Wang, Y.: Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th international conference on data mining. pp. 1103–1108. IEEE (2013)
13. Li, Q., Zhang, Q., Si, L.: Rumor detection by exploiting user credibility information, attention and multi-task learning. In: Proceedings of the 57th annual meeting of the association for computational linguistics. pp. 1173–1179 (2019)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
15. Luo, Z., Li, P., Qian, Z., Zhu, X.: Teh-gcn: Topic-event based hierarchical graph convolutional networks for rumor detection. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2023)
16. Ma, J., Gao, W., Joty, S., Wong, K.F.: Sentence-level evidence embedding for claim verification with hierarchical attention networks. Association for Computational Linguistics (2019)
17. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks (2016)
18. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: 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 (2015)
19. Ma, J., Gao, W., Wong, K.F.: Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics (2017)
20. Ma, J., Gao, W., Wong, K.F.: Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics (2018)
21. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
22. Nguyen, D.Q., Vu, T., Nguyen, A.T.: Bertweet: A pre-trained language model for english tweets. arXiv preprint arXiv:2005.10200 (2020)
23. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
24. Rao, D., Miao, X., Jiang, Z., Li, R.: Stanker: Stacking network based on level-grained attention-masked bert for rumor detection on social media. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3347–3363 (2021)
25. Richter, P.: A network theory of the immune system. European Journal of Immunology 5(5), 350–354 (1975)

26. Shu, K., Wang, S., Liu, H.: Beyond news contents: The role of social context for fake news detection. In: Proceedings of the twelfth ACM international conference on web search and data mining. pp. 312–320 (2019)
27. Varela, F.J., Stewart, J.: Dynamics of a class of immune networks i. global stability of idio-type interactions. *Journal of Theoretical Biology* 144(1), 93–101 (1990)
28. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. *science* 359(6380), 1146–1151 (2018)
29. Wei, P., Xu, N., Mao, W.: Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. arXiv preprint arXiv:1909.08211 (2019)
30. Wu, K., Yang, S., Zhu, K.Q.: False rumors detection on sina weibo by propagation structures. In: 2015 IEEE 31st international conference on data engineering. pp. 651–662. IEEE (2015)
31. Yang, Z., Wang, C., Zhang, F., Zhang, Y., Zhang, H.: Emerging rumor identification for social media with hot topic detection. In: 2015 12th web information system and application conference (WISA). pp. 53–58. IEEE (2015)
32. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T., et al.: A convolutional approach for misinformation identification. In: IJCAI. pp. 3901–3907 (2017)
33. Yuan, C., Ma, Q., Zhou, W., Han, J., Hu, S.: Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In: 2019 IEEE international conference on data mining (ICDM). pp. 796–805. IEEE (2019)
34. Zhang, J., Cui, L., Fu, Y., Gouza, F.B.: Fake news detection with deep diffusive network model. arXiv preprint arXiv:1805.08751 (2018)
35. Zhou, K., Shu, C., Li, B., Lau, J.H.: Early rumour detection. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 1614–1623 (2019)
36. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., Procter, R.: Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51(2), 1–36 (2018)