Social Network Analysis and Mining (2022) 12:134
https://doi.org/10.1007/s13278-022-00952-2

ORIGINAL ARTICLE

Adaptive cost-sensitive stance classification model for rumor detection in social networks

Zahra Zojaji[1] · Behrouz Tork Ladani[1]

Received: 10 February 2022 / Revised: 6 August 2022 / Accepted: 8 August 2022 / Published online: 9 September 2022

Abstract

As online social networks are experiencing extreme popularity growth, determining the veracity of online statements denoted by rumors automatically as earliest as possible is essential to prevent the harmful effects of propagating misinformation. Early detection of rumors is facilitated by considering the wisdom of the crowd through analyzing different attitudes expressed toward a rumor (i.e., users' stances). Stance detection is an imbalanced problem as the querying and denying stances against a given rumor are significantly less than supportive and commenting stances. However, the success of stance-based rumor detection significantly depends on the efficient detection of "query" and "deny" classes. The imbalance problem has led the previous stance classifier models to bias toward the majority classes and ignore the minority ones. Consequently, the stance and subsequently rumor classifiers have been faced with the problem of low performance. This paper proposes a novel adaptive cost-sensitive loss function for learning imbalanced stance data using deep neural networks, which improves the performance of stance classifiers in rare classes. The proposed loss function is a cost-sensitive form of cross-entropy loss. In contrast to most of the existing cost-sensitive deep neural network models, the utilized cost matrix is not manually set but adaptively tuned during the learning process. Hence, the contributions of the proposed method are both in the formulation of the loss function and the algorithm for calculating adaptive costs. The experimental results of applying the proposed algorithm to stance classification of real Twitter and Reddit data demonstrate its capability in detecting rare classes while improving the overall performance. The proposed method improves the mean *F*-score of rare classes by about 13% in RumorEval 2017 dataset and about 20% in RumorEval 2019 dataset.

Keywords Stance classification · Rumor detection · Imbalanced data · Cost-sensitive learning · Deep learning · Social networks

1 Introduction

Due to the popularity of smartphones in today's life, mobile instant messengers and online social networks are primary sources of news (Raza and Ding 2022). Nowadays, a large number of social media users do not only learn about recent events but are actively involved in generating and propagating news (Obadă and Dabija 2022). The convenience of publishing news on online social networks, anonymity, and geographical distance may encourage sharing false claims or rumors (Talwar et al. 2020). Several definitions have been proposed for rumors in the literature. In social psychology, rumor is defined as a controversial and fact-checkable statement (DiFonzo and Bordia 2007; Kimmel 2003). As another definition following (Zubiaga et al. 2016a, b) in (Lukasik et al. 2019), the rumor is defined as a circulating story of questionable veracity. Spreading rumors can harmfully affect various aspects of society's perception and behavior. Rumors often cause irreparable damage before being correctly detected (Kumari et al. 2021; Nguyen et al. 2021). With rapid information propagation in online social networks, false news can reach millions of people in a few minutes, significantly amplifying the damage brought about by rumors (Zhou and Zafarani 2020). In the 2016 US election, the spread of fake news on social media such as Facebook and Twitter was extremely high, which caused numerous political repercussions (Tu et al. 2021). Similar examples

✉ Zahra Zojaji
z.zojaji@eng.ui.ac.ir

Behrouz Tork Ladani
ladani@eng.ui.ac.ir

[1] Faculty of Computer Engineering, University of Isfahan, Isfahan 8174673441, Iran

of the harmful impacts of spreading rumors on social networks can be found in (DiFonzo et al. 2013; Takayasu et al. 2015). Determining the veracity of online news is essential to prevent the disruptive impacts of spreading misinformation, including harmful advertising or political astroturfing. Since manual verification of the huge volume of online news in the early stages is infeasible, detecting rumors via automatic mechanisms becomes crucial. This automatic task is called rumor detection.

In the rumor detection literature, the veracity of a message is identified according to different features, such as the message content, structure, propagation pattern, and the user's comments about the message. Based on the achievements of previous researches, automatically determining the users' orientations about the message, namely users' stance, is the most important step toward identifying the veracity of a rumor. For example, Lozano et al. claimed that stance detection provides the basis for veracity detection (Lozano et al. 2017). Again, Lukasik et al. (2016) found that determining user stances can be effectively used for early rumor detection. Zhao et al. (2015) also claimed that raising enquiring responses to a tweet can be utilized for detecting rumors. Similarly, stance classification is introduced as an interesting factor for detecting rumors in (AlDayel and Magdy 2021).

Stance classification is the process of determining the author's attitude toward a comment about a target message (Mohammad et al. 2016). The stance classification focuses on individual messages rather than the whole thread, which is the case in rumor detection. Here, we focus on stance classification of replies to messages containing breaking news. The underlying social networks used for the current research are Twitter and Reddit. Due to the popularity of Twitter among both ordinary people and politicians, this social network continues to be a prevalent platform for detecting rumors. Reddit discussions are usually more focused and deeper that Tweeter and contain longer texts. In addition, its conversations are less time-sensitive meaning that they take place over a longer period of time. Both social networks are employed in this study in order to provide diversity and investigate the generalization of the proposed method over different social networks with different conversation structures. Each conversation thread in the dataset comprises a source post or message (i.e., target) and a sequence of nested replies. The objective of the problem is to assign each message of the conversation to one of the predefined stance classes, including "support", "deny", "query", and "comment", which denotes the orientation of the author toward the target message. Some works, such as (Zeng et al. 2016), merged or neglected two or more of the enumerated categories in their modeling.

Stances somehow show the wisdom of the crowd about a specific claim. As stated in (Zubiaga et al. 2016a, b), investigating the conversation between users arguing about the event provides insights into the veracity of a circulating story. The idea is that users' stances represent how they respond to suspicious posts and what is the orientation of the crowd against that post. They may reply to support the claim, deny the post's veracity by providing counter-evidence or inquiry about the given information. Hence, stance classification as a mechanism for modeling the wisdom of the crowd is an interesting factor for detecting rumors. User stances indicate the supporting or denying attitude of a message and can be used to evaluate the news's emotional posture (Zhang and Ghorbani 2020). As Lukasik et al. (2019) discussed, stance detection has attracted increasing interest in recent years. Various models are proposed for rumor detection based on stance classification via extracting different sets of features categorized from several aspects. For example, features are categorized by (Kwon et al. 2013) into temporal, structural, and linguistic groups and by (Shu et al. 2017) into content-based and social context-based categories. Several studies have shown that it is possible to detect rumors using aggregate analysis of replies (Bondielli and Marcelloni 2019). For example, claims proven to be rumors raise significantly more denying replies than true claims (Zubiaga et al. 2016a, b). Similarly, (Mendoza et al. 2010) indicated that the percentage of supporting replies for a true claim is about 95%, while it is only about 38% in the case of false news. Again, (Mendoza et al. 2010; Ozturk et al. 2015) found that rumors are questioned more than regular news. Furthermore, Shao et al. (2016) revealed that rumors are shared among the top active users much more than typical users.

Although previous works reveal that analyzing the denying and querying stances play a crucial role in detecting rumors (Liu et al. 2015; Qazvinian et al. 2011a; Zhao et al. 2015), the performance of stance classifiers proposed in the literature is very poor in these two important classes. Also, neglecting the veracity status of the rumors, users largely support rumor messages but seldom deny or question rumors (Lukasik et al. 2019; Zubiaga et al. 2016a, b). As a result, the number of replies which are annotated for "query" or "deny" classes is significantly less than "support" and especially "comment" classes. This way, the imbalanced nature of the data leads the stance classifier models to bias toward the majority class of "comment" and ignore two principle classes of "query" and "deny". Since the success of stance-based rumor detection depends on the success of stance classifier in detecting these two rare classes, prior approaches have been faced with the fundamental problem of inefficiency of stance classifier for rumor detection. The importance of rare classes can be explained as follows. "Comment" replies do not have any polarity for or against the veracity of the target post. Due to the natural polarity, these replies cannot help in detecting rumors. On the other hand, the instances of the "support" class have a more influential role in detecting non-rumor claims.

In this paper, we propose a stance classification model that not only has very good overall performance but also performs well on the two essential classes of "deny" and "query". This model can be further efficiently used for detecting rumors according to analyzing the stances of replies. To this end, we propose a learning approach that can deal with highly imbalanced data as well as a large volume of data and the nested structure of the conversations. We experimentally demonstrate the superiority of this model in comparison to the current similar state-of-the-art models. The proposed model utilizes deep neural networks equipped with a novel adaptive cost-sensitive loss function for learning imbalanced data. The main contributions of this paper are summarized as follows:

- A new loss function called CSCE is introduced as a cost-sensitive loss function to handle the data imbalance issue.
- The AdCost cost function was proposed to adaptively adjust the cost matrix values according to the proportion of different classes' performance.
- A deep neural network-based model with the proposed loss function is trained to effectively detect rare classes in rumor stance classification.
- Experimental results on two real-world datasets containing Twitter and Reddit posts verify that the proposed adaptive cost-sensitive learning method outperforms previous stance classification models in detecting rare classes (i.e., "deny" and "query" classes) as well as the overall performance.

The rest of the paper is organized as follows. Section 2 reviews the related work in stance classification and cost-sensitive deep learning aspects. The research objective and the description of the research problem are given in Sects. 3 and 4, followed by an explanation of the proposed method in Sect. 5. Section 6 is devoted to experimental analysis results and discussion. Finally, Sect. 7 concludes the paper.

2 Related work

This section provides an overview of the state-of-the-art methods in the direction of stance classification for rumor detection and cost-sensitive deep learning.

2.1 Stance classification for rumor detection

The automatic classification of rumor stances was originated by (Qazvinian et al. 2011b). The authors introduced a Bayesian classification model to annotate messages to one of the two classes of confirming and denying. After that, there has been a growing interest in stance classification in the scientific community. Zubiaga et al. (2016a, b) focused on tree-structured Twitter conversations started by a rumor. They considered stance detection as a standard 4-category classification problem used in the RumourEval task of SemEval 2017 competitions (Derczynski et al. 2017). In this task, each message should be assigned to one of the four classes of "comment", "deny", "query", or "support". Although some works consider a different number of stance classes, for example, the number of stance categories is assumed to be six in (Hamidian and Diab 2016), three in (Mohammad et al. 2017) and two in (Qazvinian et al. 2011a). A similar rumor detection task was also held in SemEval 2019 competitions (Gorrell et al. 2019). Most of the literature's research proposed a supervised approach for classification. The proposed approaches differ mostly in the feature set and classification algorithms (Zubiaga et al. 2018). These approaches can be grouped into two main categories as suggested by (Chen et al. 2021): (1) feature-engineering-based machine learning methods, and (2) deep learning-based methods.

Among SemEval 2017 participants, Kochkina et al. (2017, 2018) proposed the system with the best classification performance. The authors used LSTM (long short-term memory) to learn tweet replies' sequential relation. Singh et al. (2017) considered each pair of source and reply tweets as an input pair for the learning system and employed support vector machines with the polynomial kernel. Bahuleyan and Vechtomova (2017) used manual feature extraction and combined different classifiers to construct an ensemble learning model. Subsequently, applying deep learning methods to solve the stance classification problem became popular. For example, CNN (Convolutional Neural Networks) was applied in many researches, like (Lozano et al. 2017; Poddar et al. 2018), and RNN units, mainly LSTM was employed for sequential learning in numerous papers such as (Li et al. 2019; Padnekar et al. 2020). Ma et al. (2018) trained a network to jointly detect stance and rumor veracity in which two-complex shared layer architectures are suggested and compared. Again, in more recent work, Yang et al. (2022) suggested a joint learning framework for rumor verification and stance detection. Despite improving the accuracy, their proposed method suffers from the complexity of modeling because of using a single-network architecture to train both tasks of rumor and stance detections simultaneously. Dutta et al. (2022) concentrated on semi-supervised stance detection over their prepared datasets with very few labeled tweets. The authors utilized distant supervision from social network properties to learn from large-scale unlabeled posts together with a few annotated ones. Due to the spreading of a large amount of fake news during the COVID-19 pandemic, several studies (Hou et al. 2022) concentrated on domain-specific rumor stance detection over this trending topic. In this research, a stance dataset containing 2631 tweets is collected and annotated, in which all the target tweets are about COVID-19 stories. Some recent works (Fajcik et al. 2019; Tian et al. 2020; Ruoyao Yang et al. 2019) used auxiliary datasets and pre-trained models like BooksCorpus (Zhu et al. 2015) and BERT (Devlin et al. 2019) for stance classification. While obtaining good performance, these methods are not focused on enriching the models to extract patterns from limited imbalanced data but rather concentrate on providing huge data for learning, whether before or during the training, which is not available for different domains, languages, and social platforms.

Although machine learning techniques are extensively studied for the stance classification task, few works concentrate on addressing the imbalance problem involved in stance detection. Wang et al. (2016) were one of the frontier researchers that proposed a model which tackles the imbalance bias of the stance data toward the majority classes. The authors used a two-phase classification in which belonging or not belonging to the "comment" class is identified. Then non-comment tweets are classified into one of the "support", "deny", or "query" classes. The suggested approach had poor generalization on the test data. Again, Chen et al. (2017) utilized CNN for extracting features from tweets' text as the input layer for a softmax layer that performs classification. They applied balanced mini-batching in the training network to address the imbalance issue. Several models are trained on the same data, and a voting scheme is employed for the final prediction. The suggested approach is a data-level approach for imbalanced data classification that resamples data and neglects some training samples. Moreover, the number of samples of each class in a batch that has a similar effect to the class costs is constant and should be set manually. After that, (Ruoyao Yang et al. 2019) used an inference chain-based system of tweets for training the network while utilizing a pre-trained embedding representation model of OpenAI (Radford et al. 2018). This work expanded the training data in minority classes from other similar datasets to overcome the class imbalance issue. Although the proposed method performs well on the expanded dataset, it does not deal with the imbalanced data, but rather, the dataset itself is changed so that it would not be skewed anymore. Hence, the method cannot be used for imbalanced data classification tasks where external data is unavailable, like most real-world datasets. Baris et al. (2019) implemented a CNN-based neural architecture using a pre-trained contextual embedding model called ELMO (Peters et al. 2018). Considering the class imbalance problem, they suggested a loss function, in which the weights of errors in the "comment" class is one-fifth of the other classes. This weighting is static during the training, and the value is set constantly to one-fifth. Recently, Zhang et al. (2019) investigated the problem of stances' imbalance. The authors proposed a hierarchical model (i.e., a two-layer neural network) for stance detection. In this model, each message is classified under related or unrelated categories; then, the related messages are classified to agree, disagree, and discuss classes. In fact, the role of the first layer is related–unrelated classification, while the role of the second layer is the detailed classification of the related category. The final models were compared with related researches in terms of accuracy, which is not a suitable metric in the presence of class imbalance. The suggested hierarchical structure of classes is completely dependent on the number and distribution of classes and may be required to change in other datasets. In other work (Hamidian and Diab 2019), the authors provide a Bi-LSTM (Bidirectional LSTM) based model for stance prediction. Toward improving the performance on imbalanced data, they applied some predefined rules for detecting each stance class on low-confidence network outputs. These rules include existence of the question mark for detecting "query" stance, high cosine similarity of the source and reply posts despite different sentiments for detecting "deny" stance and presence of URL and pictures for detecting "support" stance. Although these rules can help to detect rare classes they are fixed and cannot adapt by dynamic changes of the social networks and user preferences in expressing different stances. Ghanem et al. (2019) suggested using different constant weights for classes and classifying stance data through classic machine learning approaches. This work suffers from the manual tuning of constant weights that should be retuned for each dataset and cannot adaptively change during the training.

To sum up, although many researches have considered the class imbalance problem in stance classification, they are mostly data-level approaches that neglect or augment some data samples. Hence, in addition to the overhead of resampling, these works may fail to perform well on real unseen data. The suggested methods with algorithm-level approaches, on the other hand, are often dataset-specific and cannot be generalized to the other datasets or are highly dependent on hyperparameters. In the current paper, we suggest an algorithm-level approach for handling imbalanced data in a stance classification that can be applied to any stance dataset with any class distribution, even with a different number of majority classes and without the need for data resampling. Furthermore, the cost parameters can be set adaptively and during training in the suggested method.

2.2 Cost-sensitive deep learning

The class imbalance problem is a common challenge in mining real-world data. It occurs when the number of instances of one (or more) class, called minority class, which is often the class of interest, is significantly less than the others. In this situation, the class distributions are strongly imbalanced, which causes popular classification algorithms to face low classification accuracy for minority classes. *Cost-Sensitive Learning* is a type of learning that aims to minimize the total misclassification cost rather than the number of misclassified samples. Misclassification costs reveal penalties for errors in the classification problem context. Usually, cost matrix C is defined so that $C(i, j)$ represents the cost of classifying an instance of class j as being in class i. Taking different costs for different misclassification errors into consideration distinguishes between cost-sensitive and cost-insensitive learning methods.

Applying deep neural network learning algorithms to the imbalanced problem has also faced the common problem of low predictive accuracy on minority classes. In fact, the gradient component of the minority classes is much smaller than the gradient component of the majority ones. Consequently, the error of the majority classes reduces rapidly throughout early iterations, while the errors of the minority classes increase. Although cost-sensitive learning has gained wide-scale attention in the literature, the study of cost-sensitive deep learning approaches is still in the early stages and involves many open issues yet.

Existing cost-sensitive deep learning approaches can be categorized into data-level, algorithmic level, and hybrid methods (Johnson and Khoshgoftaar 2019). In data-level approaches, Hensman and Masko (2015) utilized random oversampling of the minority classes and revealed improving classification performance by balancing the training data on image classification context. Lee et al. (2016) employed a two-phase learning approach, in which the network is trained with the balanced dataset in the first phase and then fine-tuned with the original imbalanced data in the second phase. The authors utilized random undersampling methods for balancing the available data for the first phase. Havaei et al. (2017) also used this two-phase learning approach for the image segmentation problem. Pouyanfar et al. (2018) proposed a novel dynamic sampling method that adaptively identifies each class's sampling rate according to its performance. They also used two-phase learning and trained the network with both sampled and all training data to overcome the overfitting problem. Finally, random undersampling, random oversampling, and two-phase learning were applied and compared over several imbalanced image datasets in (Buda et al. 2018). The authors concluded that oversampling and undersampling methods usually outperform the two-phase learning approach. Generally, data-level methods involve random undersampling, random oversampling, or the combination of these techniques that alter the training data and require preprocessing steps. Applying oversampling methods in huge datasets may cause duplication of significant volumes of data and will extremely increase the time of the training phase (Johnson and Khoshgoftaar 2019). On the other hand, random undersampling may result in increasing the classifier variance and missing important or determinant information (Fernández et al. 2018).

Algorithmic level approaches often modify the deep neural network training algorithm to address the class imbalance problem. These methods can be classified into three groups of introducing new loss function, cost-sensitive learning, and category centers (Johnson and Khoshgoftaar 2019). The introduced algorithms in (Wang et al. 2016) and (Lin et al. 2020) contain new loss functions that allow more impact of minority classes samples on the loss value. Wang et al. show the poor performance of MSE (Mean Squared Error) loss function on minority classes and suggest using two proposed measures, namely mean false error and mean squared false error, which are more sensitive to misclassification of minority classes' samples than MSE. Again, Lin et al. proposed a new loss function called “Focal loss”, in which the cross-entropy equation is modified by manually adding class-wise weights. The second group of algorithmic level approaches applied cost-sensitive learning. In another work, Wang et al. (2018) incorporated a manually set predefined cost matrix for different misclassification errors and forced the training algorithm to minimize the total cost. The proposed method was compared with basic classifiers. Khan et al. (2018) simultaneously trained network weights and cost matrix values. The suggested approach is evaluated against multiple imbalanced datasets. As authors stated, the main goal of this study is to improve the overall classification accuracy and most of comparisons and analysis of the paper is performed using this criterion which is not reliable in the context of imbalanced data. Zhang et al. (2016) evolved cost matrix values using an evolutionary algorithm during the training. Each chromosome of the population represents a candidate cost matrix that is set randomly in the initial population. The fitness of each chromosome is considered the performance of a kind of deep neural network on the training dataset while incorporating that cost matrix in the loos function of the network. Applying mutation and crossover operators results in generating the next population of cost matrices, and evolution proceeds until reaching to stop conditions. At the end of training phase, the best evolved cost matrix is used for applying on the deep neural network for evaluation phase. Zhang et al. (2018) extracted the features using CNN, a category center in the feature space was selected for each class, and each new sample was assigned to the nearest class center. This method strongly depends on CNN's ability to generate discriminative features that are not satisfied in the presence of class imbalance. Generally, since algorithmic level methods rarely manipulate the training data and the training time, they can be better applied for big data problems. Except for the misclassification costs, the algorithm-level approaches often need little to no tuning of parameters. Most of these algorithms incorporate predefined misclassification costs in which cost values are set manually, they must be reset from one dataset to another, and the cost values are constant during the training. Therefore, for solving stance classification problem, there is still a need for an algorithmic level approach based on deep neural networks that can automatically adjust cost values during the training, can generalize well to many complex problems in a reasonable time, and can bring in a good performance in terms of imbalanced context measures. That is indeed the objective of this research.

3 Research objective

Our research objective is to propose a cost-sensitive stance detection algorithm that has the following characteristics:

- Can deal with highly imbalanced data as well as a large volume of data and the nested structure of the conversations
- Can obtain a good performance in terms of imbalanced context measures both for rare classes and the overall dataset
- Can adaptively adjust the cost values for different learning stages and even different problems with any class distribution
- Does not need auxiliary datasets for efficient stance detection

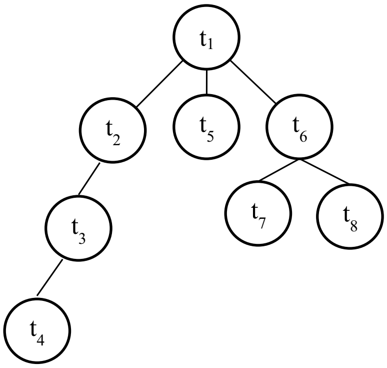


Fig. 1 A sample conversation tree over $T=\{t_1,\ldots,t_8\}$

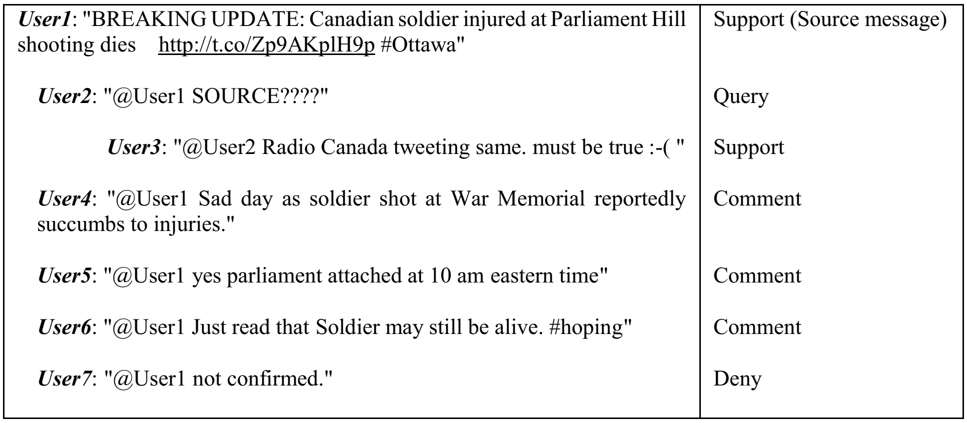
4 Problem definition

Consider a tree-structured conversation over a thread $T = \{t_1, \ldots, y_{|T|}\}$ started by a rumor t_1 followed by a set of direct or indirect replies t_2 to $t_{|T|}$. Thread T obeys a tree structure in which t_1 is the tree's root. Each message response equals an edge in the tree between a source and reply messages. Figure 1 depicts a sample conversation tree over $\{t_1, \ldots, t_8\}$.

The rumor stance classification task determines the type of orientation that each reply t_i expresses toward a target message t_1. The problem typically consists of a four-way classification problem in which each message is assigned to one of the "support", "deny", "query" or "comment" classes. A supporting reply clearly expresses that a rumor is true. A denying message describes disbelief toward a claim. A questioning reply may ask for more information on the claim or challenge its veracity. Messages that do not belong to the mentioned categories are classified as commenting replies. An example of a rumor conversation thread from Twitter is given in Fig. 2.

Detection of messages' stances is often considered preliminary for rumor detection. The most critical stances for detecting rumors are "deny" and "query" ones which are neither adequately represented nor detected in the literature. Identifying "deny" and "query" stances correctly is essential for rumor detection as they are indicators of false and unverified rumors (Wei et al. 2019). As the previous studies (Kochkina et al. 2017, 2018; Lukasik et al. 2019) claimed, users hardly deny or question rumors but mostly support claims even when they are unsure about their veracity.

Fig. 2 An example of a conversation thread with related stance labels from Twitter



User1: "BREAKING UPDATE: Canadian soldier injured at Parliament Hill shooting dies http://t.co/Zp9AKplH9p #Ottawa"	Support (Source message)
User2: "@User1 SOURCE????"	Query
User3: "@User2 Radio Canada tweeting same. must be true :-("	Support
User4: "@User1 Sad day as soldier shot at War Memorial reportedly succumbs to injuries."	Comment
User5: "@User1 yes parliament attached at 10 am eastern time"	Comment
User6: "@User1 Just read that Soldier may still be alive. #hoping"	Comment
User7: "@User1 not confirmed."	Deny

Therefore, the data obtained from the social network is highly imbalanced in the "query" and "deny" classes. Due to severe class imbalance in rumor stance data, current machine learning models often fail to classify instances that fall into "query" and "deny" classes correctly (Lukasik et al. 2019; Zhang et al. 2019). Hence, the stance classification model should effectively detect imbalanced data, especially in the "deny" and "query" classes.

5 The proposed method

In this research, we propose a flexible cost-sensitive model of stance detection. The model is based on a novel adaptive cost-sensitive loss function for deep neural network training. This method aims to effectively address the class imbalance problem in stance detection, which is rarely studied in the literature. Rumor stance detection data is usually obtained through crawling the social networks. Raw messages text, are initially preprocessed, then modeled using the proposed deep neural network in the second step, and finally evaluated.

5.1 Preprocessing

Tree-structured rumor conversations contain a source message (i.e., rumor) and a set of nested replies that should be first preprocessed for use as the training samples of deep neural networks. Since the stance of a message should be identified toward a target message, the conversation tree is modeled as linear sequences already suggested by (Kochkina et al. 2017, 2018; Zubiaga et al. 2016a, b). Each sequence includes a source message followed by some replies to the source and reply to the first reply, and so on. The linear sequences of Fig. 1 are illustrated in Fig. 3. More specifically, each route from the root of the conversation tree (i.e., the source message) to a leaf node (i.e., a reply message that is not replied anymore) composed of a linear sequence. This configuration of conversation threads is due to the sequential nature of the arguments. Note that different sequences may have different sizes depending on the number of replies.

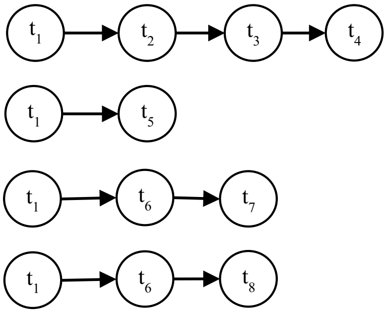


Fig. 3 Linear message sequences extracted from the conversation tree of Fig. 1

Some preprocessing steps, such as removing non-alphabetic characters and tokenizing, are applied to messages' text. The word2vec embedding model (Mikolov et al. 2013) is also applied to text messages to predict the linear context of the words adjacent to the target words. Each message is finally represented in terms of a feature vector. The employed features are as follows:

- Average message text embedding vector
- Frequency of negation words
- Frequency of swearing words
- Presence of punctuations like "?", "!", etc.
- Cosine similarity between embedding vectors of the current and the source messages
- Cosine similarity between embedding vectors of the current and the previous messages
- Cosine similarity between embedding vectors of the current message and the whole thread
- The number of words and characters
- Whether it is a source or a reply message

The features are mostly selected based on (Kochkina et al. 2017, 2018), the winner of the RumorEval 2017 stance detection task.

5.2 Architecture of stance classification network

Since there is a logical relation between subsequent argument messages, the stance of a message may depend on the previous messages in its sequence. Therefore, we employ a recurrent neural network to learn the sequential consequence of messages. The overall architecture of the employed network is shown in Fig. 4.

After the input layer, two LSTM layers with 500 LSTM units in each layer are considered. LSTM units create an internal state that enables the network to capture the dynamic logical relation between subsequent messages. The dense or fully connected layer is added for linear operation on the layer's input. The dense layer is followed by the dropout layer, which enriches the network to prevent the overfitting problem. Finally, the Softmax layer performs multiclass classification and generates predictive probabilities. The loss function generates network feedback based on the error between expected and actual output. An optimization algorithm then minimizes the loss value by adjusting the weight parameters. Previous researches like (Wang et al. 2016) have shown that in class imbalanced scenarios, typical loss functions such as the mean squared error poorly capture the errors from the minority classes. In this case, the majority class dominates the network gradient responsible for updating the weights. This event causes reducing the

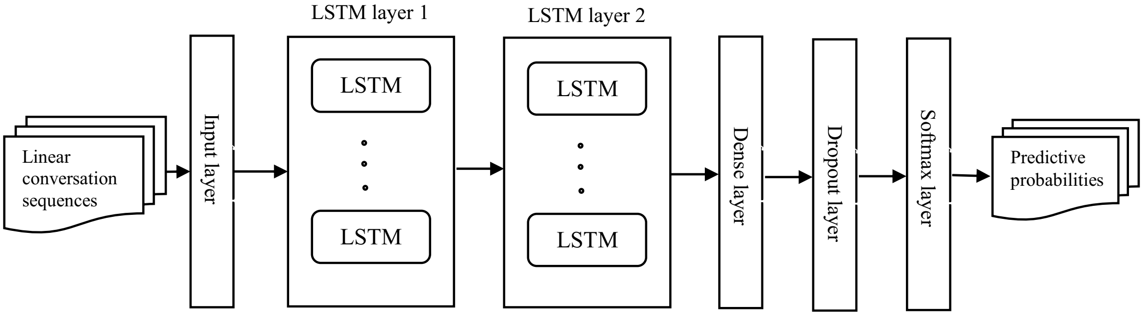


Fig. 4 Network architecture

error of the majority categories rapidly while increasing the error of minority classes. Therefore, we propose an adaptive cost-sensitive loss function that can adequately consider the minority classes' samples.

5.3 Adaptive cost-sensitive loss function

We introduce an adaptive cost-sensitive loss function based on categorical cross-entropy. Equation (1) gives the primary form of categorical cross-entropy loss for multiclass classification problems.

$$\mathrm{CE}(\mathbf{p}, c^*) = -\sum_{c=1}^{M} \delta(c = c^*)\ \log(p_c) \tag{1}$$

where $\delta(x)$ is 1 when x is true and 0 otherwise. c is the class index, M is the number of classes, and c^* is the correct class index for the current observation. The predictive probability vector, $\mathbf{p}$, contains p_c for all classes, which is the prediction probability for belonging the current sample to class c.

Equation (1) infers that the loss value is not sensitive to which class is the correct one, and there is no difference between correct classes when calculating loss. Consider for example two predictive probability vector $\mathbf{p_1}=[0.2\ 0.1\ 0.8\ 0.4]$ and $\mathbf{p_2} = [0.2\ 0.1\ 0.4\ 0.8]$. According to Eq. (1), CE ($\mathbf{p_1}$, 3) = CE ($\mathbf{p_2}$, 4) = 0.22. However, for imbalanced learning detecting one or more classes of interest is usually more important than others, and the cost of misclassification should be higher in this case. To address this problem, we introduce cost-sensitive cross-entropy (CSCE) described in Eq. (2).

$$\mathrm{CSCE}(\mathbf{p}, c^*, \mathbf{cost}) = \sum_{c=1}^{M} \delta(c = c^*)\ \log(p_c)\, \mathbf{cost}(\arg\max(\mathbf{p}), c\ *) \tag{2}$$

where arg max($\mathbf{p}$) implies the predicted class index assigned by the maximum probability. **cost** is the cost matrix that shows how each misclassification should affect the total loss. **cost**(i, j) denotes the cost of predicting the sample of class i as class j. Obviously, **cost**(i, j) = 0 where $i=j$. The cost of classifying the minority classes, which are often the classes of interest, as other classes should be more than the reverse case.

Applying CSCE in training a network increases the length of minority classes' gradient component and emphasizes correctly detecting samples of these classes. Although CSCE is a balanced version of CE and can bias the network weights toward the minority classes, detecting a suitable cost matrix that provides efficient performance is not straightforward. There is no simple rule for setting the proper cost values, and the optimum value depends on problem-specific parameters such as the distribution of classes. Selecting the cost values fewer or greater than optimum can cause neglecting or overestimating the minority classes. Moreover, the cost matrix elements should vary during the training. In the early training iterations, the cost of misclassifying the minority classes as others should be high to improve the corresponding gradient component and bias the network weights to attend the minority classes. In the further iterations, the mentioned cost value should be decreased to prevent overfitting to minority classes and allow the network training algorithm to search locally around the discovered optimum weights. In fact, adjusting the cost values in the training process manages the tradeoff between search exploration and exploitations, which significantly influences the search procedure's efficiency. Consequently, we propose the following adaptive cost function, AdCost, to update the cost matrix during the training iterations. Equation (3) shows the cost matrix values in iteration t.

$$\mathrm{AdCost}^t(i,j) = \left\{ \begin{array}{ll} \lambda + \frac{F\text{-score}^t(c_j)}{F\text{-score}^t(c_i)+\varepsilon} & i \neq j \\ 0 & i = j \end{array} \right\} \tag{3}$$

where λ is a constant to guarantee nonzero value for $i \neq j$ elements of the matrix, ε is a constant that prevents dividing by zero exception, and F-score^t (c_i) refers to the performance measure of samples of class i in iteration t. F-score, expressed in Eq. (4), is the well-known performance metric that is popular in the imbalanced data mining literature and is defined as the harmonic mean of precision and recall metrics:

$$F - \text{score} = \frac{(1 + \beta^2) \times \text{Recall} \times \text{Presicion}}{\beta^2 \times \text{Recall} + \text{Presicion}} \tag{4}$$

where coefficient β indicates the relative importance of recall against precision. Definitions of Recall and Precision are given in Eq. (5). In this formula, *TP*, *FP,* and *FN* stand for true positive, false positive, and false negative classified samples, respectively.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Presicion} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

It can be inferred from Eq. (3) that if class *i* has poor performance and class *j* has fair performance, the cost of detecting the sample of class *i* as class *j* will be high. This formulation is adopted because this type of error causes decreasing the *F*-score of class *i*. Higher cost values for this error type can prevent minority classes from being more and more neglected by proceeding with the training iterations. On the other hand, when the classifier biases enough to minority classes' samples, the performance of these classes improve, and the cost of detecting the minority classes as other classes decreases. Therefore, the CSCE loss function can adaptively balance all classes' attention by updating the cost matrix values according to classes' performance feedback. Figure 5 shows the AdCost function for different *F*-score values.

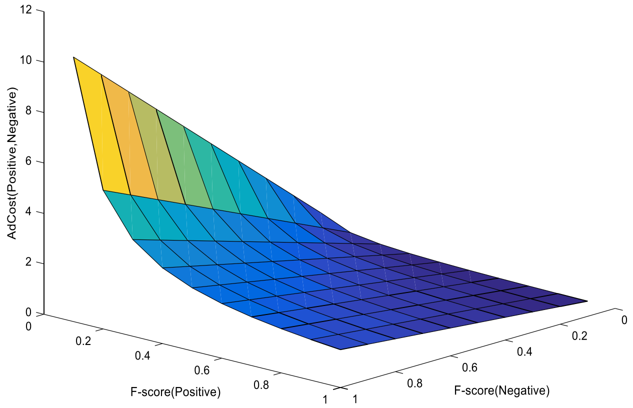


Fig. 5 An intuitive view of AdCost function

The underlying dataset is assumed to be imbalanced and related to two-class problems: positive and negative classes. According to Fig. 5, as the *F*-score of the positive class decreases and *F*-score of the negative class increases, the cost of predicting positive samples as negative ones grow.

5.4 Adaptive cost-sensitive deep learning

Defining CSCE loss aims to train a deep network that can address the imbalanced issue of data utilizing this loss function. Figure 6 shows the unite diagram of the proposed learning model. The weights of the network are set randomly in the initial iteration. As demonstrated in Fig. 6, for the weight set in iteration *t*, the predictive probability $\boldsymbol{p}^t$ expresses how much each label is probable for each sample. Given $\boldsymbol{p}^t$ and c^* that denote the ground truth labels, we can calculate the *F*-score of each class and then AdCost, accordingly. The value of the CSCE loss function can then be computed. Finally, the loss value is used for updating the network weights. Incrementing the time step and proceeding to the next batch cause the loop to be re-executed, and these steps repeat until the stop conditions are satisfied.

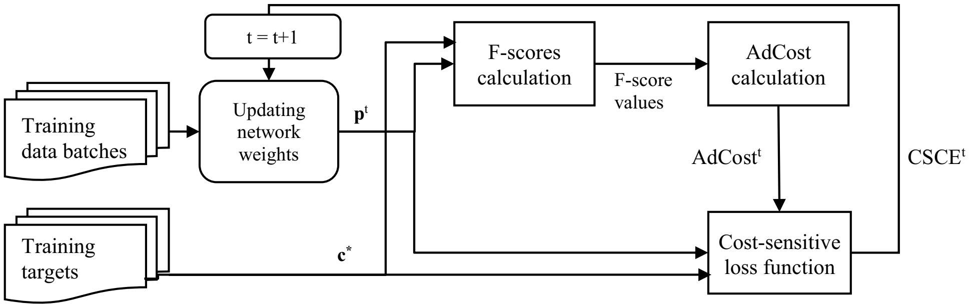
6 Experimental results

The data description is provided in this section, together with the execution setup. Next, we provide a sensitivity analysis of the hyperparameters. We conclude with the comparative results.

6.1 Data description

We used a subset of the PHEME dataset (Derczynski and Bontcheva 2014) published in its current form by (Zubiaga et al. 2016a, b). Then it was used for the RumorEval 2017 task, a large dataset containing multiple breaking news topics. Each topic includes several conversation threads that have their claims and replies. Each thread contains a source message describing the claim and several nested replies that

Fig. 6 The proposed adaptive cost-sensitive neural network model



form a conversation tree. The conversations are captured from Twitter, a popular online social network. Another dataset employed for evaluation is the RumorEval 2019 corpus, an extension of RumorEval 2017. It contains 8529 English posts, namely 6702 from Twitter and 1827 from Reddit.

Table 1 shows the data statistics, including the distribution of classes over training, testing, and development sets for both datasets. As inferred from the table, both datasets are strongly imbalanced. "query" and "deny" classes devote just about 15% of samples together. In contrast, the "comment" class samples are about 66% and 72% in the 2017 and 2019 datasets, respectively. It means that the "comment" class is the majority class and can cause classifiers to be strongly biased toward this class. Although "deny" and "query" classes are underrepresented in these datasets as well as real-world conversations, efficient detection of their sample is a very important step toward resolving the veracity of the rumor.

Therefore, as described in previous sections, this paper is focused on the efficient classification of these two classes while preserving the overall classification performance.

6.2 Setup

The configuration and parameter settings of the conducted experiments are shown in Table 2. The first four parameters are related to the network architecture. The number of epochs, the learning rate, and the batch size are network learning parameters, and λ, ε, and β are cost function parameters whose optimum values are discovered on a trial-and-error basis.

We used a server containing Tesla K80 GPU that has 2496 CUDA cores with 12 GB GDDR5 VRAM. The system CPU is a single core hyper-threaded Xeon Processor @2.3Ghz with 12.6 GB memory.

6.3 Adjusting the model parameters

In this subsection, we conduct several sensitivity analysis experiments to adjust the parameters of our suggested cost function and the underlying neural network. Since classification accuracy is not expressive enough in the context of imbalanced data and cannot represent the classifier's performance on the minority classes, usually, F-score and macro-averaged F-score are employed to evaluate these problems.

Table 2 The parameter settings

Parameter	Value
The number of dense layers	2
The number of LSTM units	100
The number of dense units	500
The number of LSTM layers	2
The number of epochs	30
Learning rate	0.001
Bach size	100
λ (cost function constant)	0.5
ε (cost function constant)	0.0001
β (F-score order)	1

- AdCost function parameters

AdCost automatically learns cost values and does not need significant parameter settings, as discussed in previous sections. In Equation (3), λ is just a constant to guarantee nonzero cost values, and ε is a constant that prevents dividing by zero exceptions that do not play an important role in the resulting costs. However, we study the influence of λ on the overall performance of the stance classifier, as demonstrated in Fig. 7. As λ value grows, the cost values increase, and the misclassification is penalized more. Therefore, classification is biased toward the majority classes. For smaller values of λ, cost values decrease, and misclassified samples increase. As depicted in Fig. 7, setting the value of 0.5 for λ, results in the best values of F-score in "query", "deny", and macro F-score.

Table 1 Dataset statistics

	s	Support	Query	Deny	Comment	Total
RumorEval 2017	Training	841 (19.8%)	330 (7.8%)	333 (7.9%)	2734 (64.5%)	4238
	Development	69 (24.6%)	28 (10.0%)	11 (3.9%)	173 (61.6%)	281
	Test	94 (9%)	106 (10.1%)	71 (6.8%)	778 (74.2%)	1049
	Total	1004 (18%)	464 (8.3%)	415 (7.4%)	3685 (66.1%)	5568
RumorEval 2019	Training	925 (18%)	395 (8%)	378 (7%)	3519 (67%)	5217
	Development	102 (7%)	129 (8%)	82 (6%)	1181 (80%)	1485
	Test	157 (9%)	93 (5%)	101 (6%)	1476 (81%)	1827
	Total	1184 (14%)	608 (7%)	606 (7%)	6176 (72%)	8529

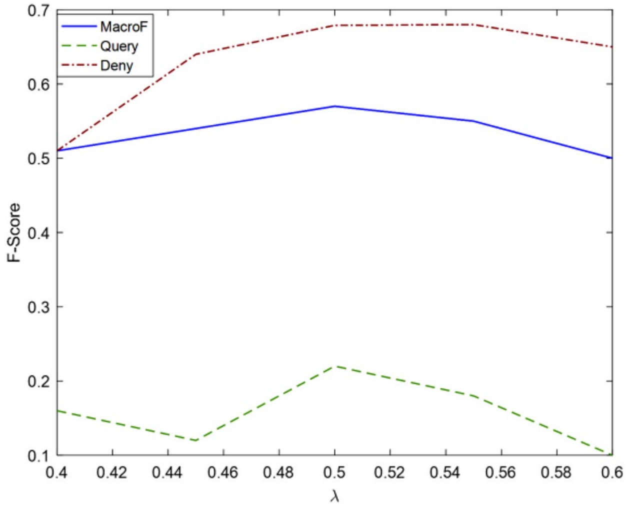


Fig. 7 The impact of different λ values on macro F-score, "query" and "deny" F-scores, respectively

- Deep neural network parameters

The selected values for the architecture and learning parameters of the deep neural network were reported in Table 2. The optimal values are selected based on the performance of the stance classifier on the development set of the data. The sensitivity analysis results for architectural parameters are illustrated in Fig. 8. By reducing the number of LSTM units and layers, the network's capability for capturing the long logical argument in subsequent messages is also reduced. On the other hand, by increasing the number of LSTM units and layers, the complexity of the network and the number of related parameters increase extensively, which needs more data and computational effort for optimization and training of the network. As shown in Fig. 8a and c, the best number of LSTM unities is 100, and the best number of LSTM layers is 2. A similar argument holds for dense layers and unites. According to Fig. 8b and d, the best values for the number of dense units is 500, and for the number of dense layers is 2.

Figure 9 depicts the influence of changing the learning parameters on the model performance. Low learning rate values lead the network not to reach the optimum state in available time and computational facilities. On the other hand, high values may cause oscillating around the optimum network weights. As shown in Fig. 9a, the best value for the learning rate in our model is 0.001. The proposed model with different batch size values is tested on the development set. The results are revealed in Fig. 9b. According to this figure, 100 is our domain's best batch size value.

6.4 Results

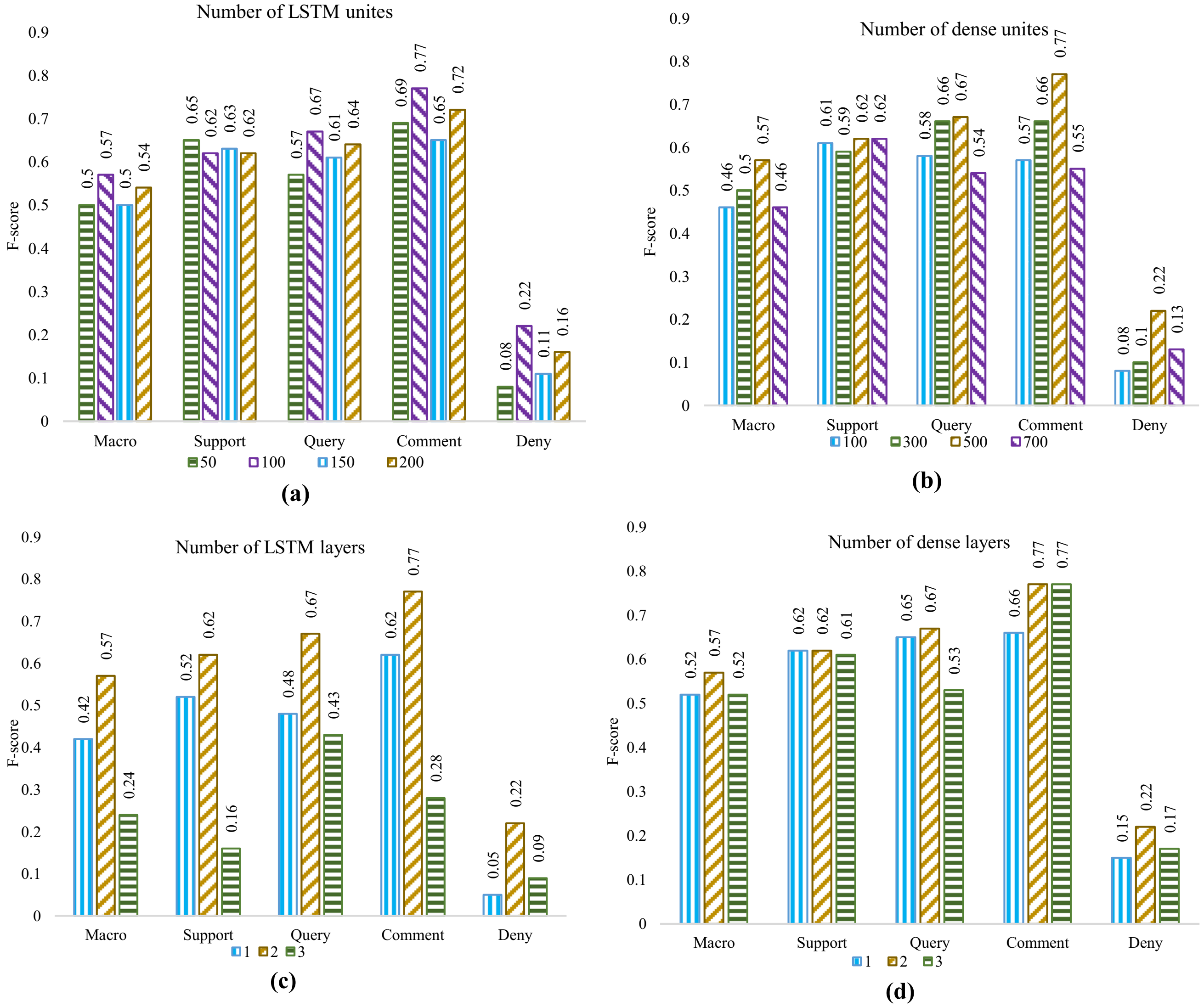
The performance results of the proposed model on testing and development sets are described in Table 3 in terms of precision, recall, and F-score for each class. The accuracy and macro-F-score are also reported for testing and development sets in each dataset.

The effect of cost incorporations into the stance classification model is studied, and a set of experiments are conducted to prove the capability of adaptive cost-sensitive deep neural networks. As demonstrated in Fig. 10, running the stance classification model with no cost results in a good performance on "comment" and "support" classes and poor performance in "deny" and "query" classes. The F-score of this model on the "query" class is zero, highlighting the role of cost-sensitive learning in detecting rare classes.

An alternative approach is to incorporate a static cost matrix into the model. We consider a distribution-based cost matrix that weights each misclassification by the actual and predicted sizes of classes. Although this model does not achieve better performance in the majority classes, it has superior performance to no cost model in predicting "query" and "deny" classes. The adaptive cost-sensitive approach outperforms both methods in "deny" and macro F-score measures. The performance in the "query" and "support" classes is nearly equal to the best method in each class. In conclusion, using the adaptive cost matrix seems crucial for detecting rare classes like "deny" that are very important in stance detection.

In Table 4, the performance of the proposed model is compared with the related work for which mentioned datasets are used as training and testing data, and the evaluation results are reported on each class in addition to the overall test set. Top F-score values for each dataset are represented in bold for all columns.

In this table (Kochkina et al. 2017, 2018) is the baseline provided in RumorEval 2019 contests. Other works are described in Sect. 2.1. The table shows that the proposed model outperforms other models in the "query" and "deny" classes for all entries. Moreover, the classifier's overall performance on all classes reported in terms of macro F-score is superior to the others. For the "comment" class the proposed method achieves the best performance among compared models. For "Support" class, the best results belong to CNNg model of (Lozano et al. 2017) in RumorEval 2017 and (Baris et al. 2019) in RumorEval 2019 datasets. To highlight the performance of the proposed method in rare classes, the mean F-score of "query" and "deny" classes is also reported separately in the table. As it can be calculated from this column, the proposed method improves the



(a)

(b)

(c)

(d)

Fig. 8 Sensitivity analysis for architectural parameters: **a** number of LSTM units, **b** number of dense units, **c** number of LSTM layers, **d** number of dense layers

mean *F*-score of rare classes by about 13% in RumorEval 2017 dataset and about 20% in RumorEval 2019 dataset. The overall macro *F*-score is also improved by about 7% in the former and 12% in the latter. In conclusion, the proposed adaptive cost-sensitive loss function is capable of biasing the classifier toward minority classes while preserving the overall performance.

Each run on the specified server takes, on average 5130 s for preprocessing the whole dataset, 1200 s for building and training the network on the training set, and 2.8 s for evaluating the test set on the trained network.

7 Conclusion

This paper provides a model for addressing the task of stance classification through the proposed cost-sensitive deep learning approach.

7.1 Theoretical and practical implications

In order to develop a cost-sensitive deep learning-based stance classifier, we first introduced the CSCE loss function as a cost-sensitive loss function to handle the data imbalance

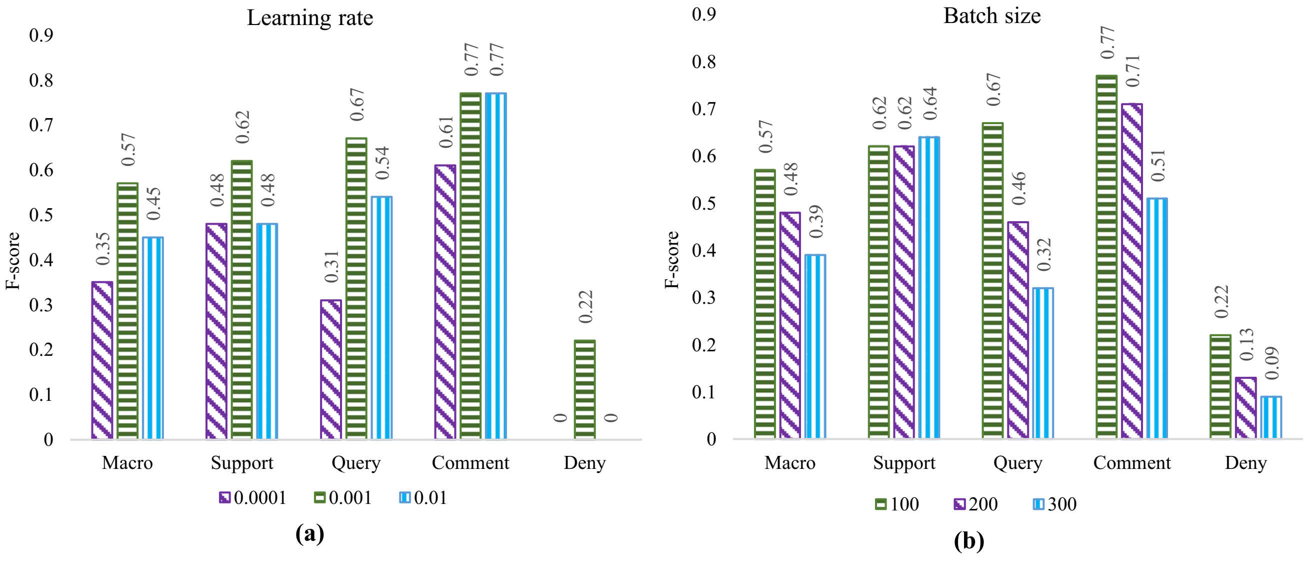


Fig. 9 Sensitivity analysis for learning parameters: **a** learning rate, **b** batch size

Table 3 The results of experimental evaluation on test and development sets

Dataset	Partition		Support	Query	Comment	Deny	Accuracy	Macro *F*-score
RumorEval 2017	Test	*F*-score	0.39	0.55	0.78	0.22	0.65	0.48
		Precision	0.68	0.58	0.84	0.14		
		Recall	0.27	0.53	0.72	0.49		
	Development	*F*-score	0.62	0.67	0.77	0.22	0.67	0.57
		Precision	0.7	0.64	0.82	0.14		
		Recall	0.56	0.71	0.73	0.55		
RumorEval 2019	Test	*F*-score	0.4	0.55	0.9	0.22	0.82	0.52
		Precision	0.8	0.56	0.88	0.22		
		Recall	0.27	0.54	0.93	0.23		
	Development	*F*-score	0.43	0.5	0.9	0.35	0.83	0.54
		Precision	0.75	0.53	0.87	0.38		
		Recall	0.3	0.47	0.94	0.33		

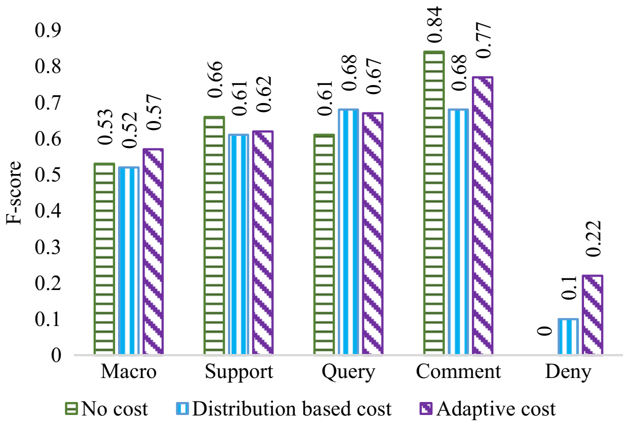


Fig. 10 Different incorporations of cost into the loss function

issue. This loss function is proposed theoretically by embedding cost values to the cross-entropy loss. The AdCost cost function was then defined to adaptively determine the best cost matrix values according to the network performance in each iteration. This cost function is theoretically designed to adjust costs according to the proportion of different classes' performance.

Although imbalanced learning is a relatively mature research direction, configuring deep neural networks for learning imbalanced data is a relatively new trend. One of the most significant advantages of the proposed method is that the suggested loss function can be easily implemented and integrated into the existing models for addressing the class imbalance issue. Furthermore, the proposed model is scalable. It means that, unlike data-level methods, the

Table 4 Comparison of the performance results with related methods

Dataset	Reference	*F*-score				Mean *F*-score of query and deny	Macro *F*-score
		Support	Query	Comment	Deny		
RumorEval 2017 dataset	Kochkina et al. (2017, 2018)	0.40	0.46	0.43	0	0.23	0.43
	Chen et al. (2017)	0.19	0.50	0.40	0.11	0.305	0.40
	Lozano et al. (2017) (CNNg)	**0.56**	0.28	0.43	0.2	0.24	0.43
	Lozano et al. (2017) (CNNr)	0.38	0.01	0.31	0	0.005	0.31
	Lozano et al. (2017) (CNN)	0.40	0.01	0.31	0	0.005	0.31
	Singh et al. (2017)	0.29	0.39	0.39	0.13	0.26	0.39
	Bahuleyan and Vechtomova (2017)	0.39	0.49	0.45	0.05	0.27	0.45
	Ma et al. (2018)	0.31	0.53	0.43	0.15	0.34	0.43
	Yang et al. (2022)	0.33	0.43	0.76	0.17	0.30	0.42
	The proposed approach	0.39	**0.55**	**0.85**	**0.22**	**0.385**	**0.48**
RumorEval 2019 dataset	Kochkina et al. (2017, 2018)	0.43	**0.55**	0.49	0.07	0.31	0.43
	Hamidian and Diab (2019)	0.446	0.40	0.43	0.0	0.2	0.40
	Baris et al. (2019)	**0.46**	0.11	0.37	0.0	0.055	0.43
	Ghanem et al. (2019)	0.40	0.50	0.48	0.14	0.32	0.31
	Yang et al. (2022)	0.35	0.44	0.76	0.17	0.30	0.43
	The proposed approach	0.40	**0.55**	**0.90**	**0.22**	**0.385**	**0.48**

loss function is less likely to increase the training time by increasing the size of the training set. Again, in contrast to existing cost-sensitive deep learning approaches, the cost matrix is adjusted automatically during the learning and does not require manual setting for different problems or different stages of the learning.

The experimental results on the real-world datasets containing Twitter and Reddit demonstrate that the adaptive cost-sensitive deep learning approach outperforms the state of the art in detecting the “deny” and “query” classes and overall performance.

7.2 Limitations and future work

In the presented work, we used word2vec for text embedding and feature extraction of the messages. Contextual word embedding models such as BERT (Devlin et al. 2019) and T5 (Raffel et al. 2020) can be further used to improve the overall performance of the stance classification model. Furthermore, as future work, we consider employing a probabilistic interpretation of *F*-score to incorporate predictive probabilities rather than just predicted classes in *F*-score and cost calculation. The presented stance classifier can be further used for efficient early detection of rumors according to the user's responses, and it will be shown in future research trends. We believe that the proposed model is not limited to stance classification and can equip other classification methods with strength in imbalanced data classification.

Data availability The data used in this study are publicly available in figshare repository with the identifier “https://figshare.com/ndownloader/files/16188500” for RumorEval 2019 and the identifier “https://s3-eu-west-1.amazonaws.com/downloads.gate.ac.uk/pheme/semeval2017-task8-dataset.tar.bz2” for RumorEval 2017 dataset.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article. This article does not contain any studies with human participants or animals performed by any of the authors.

References

AlDayel A, Magdy W (2021) Stance detection on social media: state of the art and trends. Inf Process Manag 58(4):102597. https://doi.org/10.1016/J.IPM.2021.102597

Bahuleyan H, Vechtomova O (2017) UWaterloo at SemEval-2017 Task 8: detecting stance towards rumours with topic independent features. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp. 461–464.

Baris I, Schmelzeisen L, Staab S (2019) CLEARumor at semEval-2019 task 7: Convolving ELMo against rumors. In: Proceedings of the 13th international workshop on semantic evaluation, pp 1105–1109. https://doi.org/10.18653/v1/s19-2193

Bondielli A, Marcelloni F (2019) A survey on fake news and rumour detection techniques. Inf Sci 497:38–55

Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. Neural

Netw 106:249–259. https://doi.org/10.1016/J.NEUNET.2018.07.011

Chen X, Zhou F, Zhang F, Bonsangue M (2021) Catch me if you can: A participant-level rumor detection framework via fine-grained user representation learning. Inf Process Manag 58(5):102678

Chen Y-C, Liu Z-Y, Kao H-Y (2017) Ikm at semeval-2017 task 8: convolutional neural networks for stance detection and rumor verification. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp. 465–469

Derczynski L, Bontcheva K (2014) PHEME: veracity in digital social networks. CEUR Workshop Proceedings, pp 1181

Derczynski L, Bontcheva K, Liakata M, Procter R, Hoi GWS, Zubiaga A (2017) SemEval-2017 task 8: RumourEval: determining rumour veracity and support for rumours. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 69–76

Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. 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 4171–4186

DiFonzo N, Bordia P (2007) Rumor, gossip and urban legends. Diogenes 54(1):19–35. https://doi.org/10.1177/0392192107073433

DiFonzo N, Bourgeois MJ, Suls J, Homan C, Stupak N, Brooks BP, Ross DS, Bordia P (2013) Rumor clustering, consensus, and polarization: dynamic social impact and self-organization of hearsay. J Exp Soc Psychol 49(3):378–399. https://doi.org/10.1016/J.JESP.2012.12.010

Dutta S, Caur S, Chakrabarti S, Chakraborty T (2022) Semi-supervised stance detection of tweets via distant network supervision. In: Proceedings of the fifteenth ACM international conference on web search and data mining, pp 241–251

Fajcik M, Smrz P, Burget L (2019) BUT-FIT at SemEval-2019 task 7: determining the rumour stance with pre-trained deep bidirectional transformers. In: Proceedings of the 13th international workshop on semantic evaluation, pp 1097–1104

Fernández A, García S, Herrera F, Chawla NV (2018) SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 61:863–905. https://doi.org/10.1613/JAIR.1.11192

Ghanem B, Cignarella AT, Bosco C, Rosso P, Pardo FMR (2019) Upv-28-unito at semeval-2019 task 7: exploiting post's nesting and syntax information for rumor stance classification. In: Proceedings of the 13th international workshop on semantic evaluation, pp 1125–1131

Gorrell G, Kochkina E, Liakata M, Aker A, Zubiaga A, Bontcheva K, Derczynski L (2019) SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In: Proceedings of the 13th international workshop on semantic evaluation, pp 845–854. https://doi.org/10.18653/v1/s19-2147

Hamidian S, Diab M (2016) Rumor identification and belief investigation on twitter. In: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp 3–8

Hamidian S, Diab M (2019) GWU NLP at semeval-2019 task 7: Hybrid pipeline for rumour veracity and stance classification on social media. In: Proceedings of the 13th international workshop on semantic evaluation, pp 1115–1119. https://doi.org/10.18653/v1/s19-2195

Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, Pal C, Jodoin PM, Larochelle H (2017) Brain tumor segmentation with deep neural networks. Med Image Anal 35:18–31. https://doi.org/10.1016/J.MEDIA.2016.05.004

Hensman P, Masko D (2015) The impact of imbalanced training data for convolutional neural networks. Degree Project in Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. Available online: https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf

Hou Y, van der Putten P, Verberne S (2022) The COVMis-stance dataset: stance detection on twitter for COVID-19 misinformation. ArXiv preprint. ArXiv:2204.02000

Johnson JM, Khoshgoftaar TM (2019) Survey on deep learning with class imbalance. J Big Data 6(1):1–54

Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Netw Learn Syst 29(8):3573–3587. https://doi.org/10.1109/TNNLS.2017.2732482

Kimmel AJ (2003) Rumors and rum or control: a manager's guide to understanding and combatting rumors. Rumors Rumor Control A Manag Guide Underst Combat Rumors. https://doi.org/10.4324/9781410609502

Kochkina E, Liakata M, Augenstein I (2017) Turing at SemEval-2017 Task 8: sequential approach to rumour stance classification with branch-LSTM. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 475–480. https://doi.org/10.18653/v1/s17-2083

Kochkina E, Liakata M, Zubiaga A (2018) All-in-one: multi-task learning for rumour verification. In: COLING 2018b—27th international conference on computational linguistics, proceedings, pp 3402–3413

Kumari R, Ashok N, Ghosal T, Ekbal A (2021) Misinformation detection using multitask learning with mutual learning for novelty detection and emotion recognition. Inf Process Manag 58(5):102631. https://doi.org/10.1016/J.IPM.2021.102631

Kwon S, Cha M, Jung K, Chen W, Wang Y (2013) Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th international conference on data mining, pp 1103–1108

Lee H, Park M, Kim J (2016) Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. IEEE Int Conf Image Process (ICIP) 2016:3713–3717

Li Q, Zhang Q, Si L (2019) EventAI at SemEval-2019 task 7: rumor detection on social media by exploiting content, user credibility and propagation information. In: Proceedings of the 13th international workshop on semantic evaluation, pp 855–859. https://doi.org/10.18653/v1/s19-2148

Lin T-Y, Goyal P, Girshick R, He K, Dollar P (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell 42(2):318–327. https://doi.org/10.1109/tpami.2018.2858826

Liu X, Nourbakhsh A, Li Q, Fang R, Shah S (2015) Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 1867–1870

Lozano MG, Lilja H, Tjörnhammar E, Karasalo M (2017) Mama Edha at SemEval-2017 task 8: stance classification with CNN and rules. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 481–485

Lukasik M, Bontcheva K, Cohn T, Zubiaga A, Liakata M, Procter R (2019) Gaussian processes for rumour stance classification in social media. ACM Trans Inf Syst. https://doi.org/10.1145/3295823

Lukasik M, Srijith PK, Vu D, Bontcheva K, Zubiaga A, Cohn T (2016) Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In: Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 2: Short Papers), pp 393–398. https://doi.org/10.18653/v1/p16-2064

Ma J, Gao W, Wong K-F (2018) Detect rumor and stance jointly by neural multi-task learning. Companion Proc Web Conf 2018:585–593

Mendoza M, Poblete B, Castillo C (2010) Twitter under crisis: Can we trust what we RT?. In: SOMA 2010 - proceedings of the 1st workshop on social media analytics. https://doi.org/10.1145/1964858.1964869

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: 1st international conference on learning representations, ICLR 2013 - workshop track proceedings

Mohammad SM, Sobhani P, Kiritchenko S (2017) Stance and sentiment in tweets. ACM Trans Internet Technol (TOIT) 17(3):1–23

Mohammad S, Kiritchenko S, Sobhani P, Zhu X, Cherry C (2016) Semeval-2016 task 6: detecting stance in tweets. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 31–41

Nguyen TT, Nguyen TT, Nguyen TT, Vo B, Jo J, Nguyen QVH (2021) JUDO: just-in-time rumour detection in streaming social platforms. Inf Sci 570:70–93. https://doi.org/10.1016/J.INS.2021.04.018

Obadă D-R, Dabija D-C (2022) "In flow"! why do users share fake news about environmentally friendly brands on social media? Int J Environ Res Public Health 19(8):4861

Ozturk P, Li H, Sakamoto Y (2015) Combating rumor spread on social media: the effectiveness of refutation and warning. In: 2015 48th Hawaii international conference on system sciences, pp 2406–2414

Padnekar SM, Kumar GS, Deepak P (2020) Bilstm-autoencoder architecture for stance prediction. Int Conf Data Sci Eng (ICDSE) 2020:1–5

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long papers), pp 2227–2237

Poddar L, Hsu W, Lee ML, Subramaniyam S (2018) Predicting stances in twitter conversations for detecting veracity of rumors: a neural approach. In: 2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI), pp 65–72

Pouyanfar S, Tao Y, Mohan A, Tian H, Kaseb AS, Gauen K, Dailey R, Aghajanzadeh S, Lu YH, Chen SC, Shyu ML (2018) Dynamic sampling in convolutional neural networks for imbalanced data classification. In: Proceedings - IEEE 1st conference on multimedia information processing and retrieval, MIPR 2018, pp 112–117. https://doi.org/10.1109/MIPR.2018.00027

Qazvinian V, Rosengren E, Radev DR, Mei Q (2011a) Rumor has it identifying misinformation in microblogs(2). In: Conference on Empirical Methods in Natural Language Processing, pp 1589–1599

Qazvinian V, Rosengren E, Radev DR, Mei Q (2011b) Rumor has it: identifying misinformation in microblogs. In: EMNLP 2011b - conference on empirical methods in natural language processing, proceedings of the conference

Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding with unsupervised learning. Technical report, OpenAI

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(140):1–67

Raza S, Ding C (2022) Fake news detection based on news content and social contexts: a transformer-based approach. Int J Data Sci Anal 13(4):335–362

Shao C, Ciampaglia G, Flammini A, Menczer F (2016) Hoaxy: a platform for tracking online misinformation. In: Proceedings of the 25th international conference companion on world wide web, pp 745–750

Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Fake news detection on social media: a data mining perspective. ACM SIGKDD Explor Newslett 19(1):22–36

Singh V, Narayan S, Akhtar MS, Ekbal A, Bhattacharyya P (2017) IITP at SemEval-2017 task 8: a supervised approach for rumour evaluation. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 497–501

Takayasu M, Sato K, Sano Y, Yamada K, Miura W, Takayasu H (2015) Rumor diffusion and convergence during the 3.11 earthquake: a twitter case study. PLoS ONE 10(4):e0121443. https://doi.org/10.1371/JOURNAL.PONE.0121443

Talwar S, Dhir A, Singh D, Virk GS, Salo J (2020) Sharing of fake news on social media: application of the honeycomb framework and the third-person effect hypothesis. J Retail Consum Serv 57:102197. https://doi.org/10.1016/J.JRETCONSER.2020.102197

Tian L, Zhang X, Wang Y, Liu H (2020) Early detection of rumours on twitter via stance transfer learning. Adv Inf Retr 12035:575

Tu K, Chen C, Hou C, Yuan J, Li J, Yuan X (2021) Rumor2vec: a rumor detection framework with joint text and propagation structure representation learning. Inf Sci 560:137–151. https://doi.org/10.1016/J.INS.2020.12.080

Wang S, Liu W, Wu J, Cao L, Meng Q, Kennedy PJ (2016) Training deep neural networks on imbalanced data sets. Int Jt Conf Neural Netw (IJCNN) 2016:4368–4374

Wang H, Cui Z, Chen Y, Avidan M, Abdallah AB, Kronzer A (2018) Predicting hospital readmission via cost-sensitive deep learning. IEEE/ACM Trans Comput Biol Bioinf 15(6):1968–1978. https://doi.org/10.1109/TCBB.2018.2827029

Wei P, Xu N, Mao W (2019) Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 4787–4798. https://doi.org/10.18653/v1/D19-1485

Yang R, Xie W, Liu C, Yu D (2019) BLCU_NLP at SemEval-2019 task 7: an inference chain-based GPT model for rumour evaluation. In: Proceedings of the 13th international workshop on semantic evaluation, pp 1090–1096. https://doi.org/10.18653/v1/s19-2191

Yang R, Ma J, Lin H, Gao W (2022) A weakly supervised propagation model for rumor verification and stance detection with multiple instance learning. ArXiv preprint ArXiv:2204.02626. http://arxiv.org/abs/2204.02626

Zeng L, Starbird K, Spiro ES (2016) # unconfirmed: classifying rumor stance in crisis-related social media messages. In: tenth international AAAI conference on web and social media.

Zhang X, Ghorbani AA (2020) An overview of online fake news: characterization, detection, and discussion. Inf Process Manag 57(2):102025

Zhang C, Tan KC, Ren R (2016) Training cost-sensitive deep belief networks on imbalance data problems. Int Jt Conf Neural Netw (IJCNN) 2016:4362–4367. https://doi.org/10.1109/IJCNN.2016.7727769

Zhang Y, Shuai L, Ren Y, Chen H (2018) Image classification with category centers in class imbalance situation. In: 2018 33rd youth academic annual conference of Chinese association of automation (YAC), pp 359–363. https://doi.org/10.1109/YAC.2018.8406400

Zhang Q, Liang S, Lipani A, Ren Z, Yilmaz E (2019) From stances' imbalance to their hierarchical representation and detection. In: The world wide web conference, pp 2323–2332

Zhao Z, Resnick P, Mei Q (2015) Enquiring minds: early detection of rumors in social media from enquiry posts. In: WWW 2015 - proceedings of the 24th international conference on world wide web, pp 1395–1405. https://doi.org/10.1145/2736277.2741637

Zhou X, Zafarani R (2020) A survey of fake news. ACM Comput Surv (CSUR). https://doi.org/10.1145/3395046

Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, Fidler S (2015) Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision, pp 19–27

Zubiaga A, Liakata M, Procter R, Wong Sak Hoi G, Tolmie P (2016b) Analysing how people orient to and spread rumours in social media by looking at conversational threads. PLoS ONE 11(3):e0150989. https://doi.org/10.1371/JOURNAL.PONE.0150989

Zubiaga A, Aker A, Bontcheva K, Liakata M, Procter R (2018) Detection and resolution of rumours in social media: a survey. ACM Comput Surv (CSUR) 51(2):1–36

Zubiaga A, Kochkina E, Liakata M, Procter R, Lukasik M (2016a) Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In: COLING 2016a - 26th international conference on computational linguistics, proceedings of COLING 2016a: technical papers, pp 2438–2448

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.