Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 2467539, 21 pages
https://doi.org/10.1155/2023/2467539

Research Article

CNFRD: A Few-Shot Rumor Detection Framework via Capsule Network for COVID-19

Danroujing Chen,[1] **Xiaoliang Chen**,[1,2] **Peng Lu**,[2] **Xuyang Wang**,[1] **and Xiang Lan**[3]

[1] *School of Computer and Software Engineering, Xihua University, Chengdu 610039, China*
[2] *Department of Computer Science and Operations Research, University of Montreal, Montreal H3C3J7, Canada*
[3] *Institute of Statistical Science, Sichuan Provincial Bureau of Statistics, Chengdu 610041, China*

Correspondence should be addressed to Xiaoliang Chen; xdxlchen@gmail.com

Received 28 August 2022; Revised 2 November 2022; Accepted 16 November 2022; Published 27 February 2023

Academic Editor: B. B. Gupta

In recent years, COVID-19 has become the hottest topic. Various issues, such as epidemic transmission routes and preventive measures, have "occupied" several online social media platforms. Many rumors about COVID-19 have also arisen, causing public anxiety and seriously affecting normal social order. Identifying a rumor at its very inception is crucial to reducing the potential harm of its evolution to society as a whole. However, epidemic rumors provide limited signal features in the early stage. In order to identify rumors with data sparsity, we propose a few-shot learning rumor detection model based on capsule networks (CNFRD), utilizing the metric learning framework and the capsule network to detect the rumors posted during unexpected epidemic events. Specifically, we constructively use the capsule network neural layer to summarize the historical rumor data and obtain the generalized class representation based on the historical rumor data samples. Besides, we calculate the distance between the epidemic rumor sample and the historical rumor class-wise representation according to the metric module. Finally, epidemic rumors are discriminated against according to the nearest neighbor principle. The experimental results prove that the proposed method can achieve higher accuracy with fewer epidemic rumor samples. This approach provided 88.92% accuracy on the Chinese rumor dataset and 87.07% accuracy on the English rumor dataset, which improved by 7% to 23% over existing approaches. Therefore, the CNFRD model can identify epidemic rumors in COVID-19 as early as possible and effectively improve the performance of rumor detection.

1. Introduction

Online social networks (OSN) play a crucial role in our daily lives [1–3]. They help people gather, communicate, and share their common interests [4]. Multiple social network platforms, such as Facebook, Twitter, Sina Weibo, and a few e-mail systems [5], allow users to access various services simultaneously [6, 7]. The use of online platforms is exponentially amplified, which increases the risk of information leakage and opens the door to several cybercrimes due to the large amount of data and information available on these online platforms [8, 9]. Platform users can create information at a very low cost, which also acts as a bridge of information diffusion regardless of the truth or falsity of the information [10]. Information can easily be accessed, created, and disseminated [11, 12]. These online user dynamics are essential for a wide range of social risk analyses. The spread of rumors has seriously affected the health of the OSN information ecosystem and the operation of social order, causing social unrest, endangering public safety, and harming public interests [13, 14]. Typical examples of the highly destructive power of online rumors can be found in the following three online events. The once-popular event "salt rumor" resulted in a nationwide "salt grabbing storm". Millions of people in the Shanxi Province of China took refuge on the streets because of an earthquake rumor. It has been reported that domestic dairy products have suffered from the "leather milk powder" rumor. Overall, major public crises are almost always accompanied by an explosion of online rumors. Therefore, detecting COVID-19 rumors as early as possible is of great practical significance for the governance of the network environment, the comfort of people's emotions, the stability of social order, and the effective prevention and control of epidemics. Not all events will be predicted in the future, but we can prepare for their potential rumors with our best efforts [15].

A series of noteworthy worldwide rumors are associated with the novel coronavirus disease (COVID-19), a major global public health event widely publicized since January 2020. Rumors concerning COVID-19 have spread widely through multiple online media such as news reports, Twitter, etc. These can be illustrated briefly by the "pet rumor" that pets could spread the novel coronavirus. In a similar rumor event involving disease treatment, the Chinese medicine "Shuanghuanglian" could prevent the novel coronavirus. Another rumor claims that 80 Chinese citizens were isolated and abused in Moscow. Many news organizations and social media service providers are making efforts to build rumor-reporting platforms. For example, Sina's misinformation management center (https://service.account.weibo.com/?type=5&status=0) as well as the social network management sites Snopes (http://www.snopes.com/) and Factcheck (http://www.factcheck.org/) have contributed to the rumor reporting. Researchers observed that fact-checking efforts on COVID-19 increased by more than 900 percent as rumors spread [16]. In a short period of 74 days from January 18 to March 31, 2020, the three major Chinese rumor-refuting platforms (China Internet Joint Rumor-refuting Platform (https://www.piyao.org.cn/), Baidu rumor-refuting platform, and Tencent's Jiaozhen rumor-refuting platform (https://vp.fact.qq.com/)) verified and refuted 1,491 COVID-19 rumors. Figure 1 shows the distribution of rumor detection results of the three platforms. We can conclude that more than 75% of the information on OSN during this period is rumors. Unfortunately, manual fact-checking is labor-intensive and has difficulty scaling with the volume of emerging rumors [17, 18].

Rumor detection, which aims to recognize the rumors that have appeared on OSN, has recently attracted substantial research attention due to its significant research challenges and practical value. Traditionally, rumor detection has been dominated by retrospective detection models [19, 20]. Such models combine a large number of features and require much time and labor to achieve accuracy. However, rumors do not offer adequate features during the incubation stage. Epidemic rumors exhibit their own particular characteristics. Consequently, existing retrospective detection models do not apply to this severe epidemic scenario. A systematic understanding and process of how to use small-sample learning in rumor detection are still lacking.

This study aims to introduce the few-shot learning (FSL) method into the rumor detection task to overcome the problem of insufficient epidemic rumor data. We observe that the newly emerging epidemic rumors and historical rumors generally share certain similarities in their linguistic expressions, word usages, punctuation, and distinct linguistic features such as "Central notice: XXX can prevent the virus, please spread a lot!" and "Shocked! Doctor's advice that 99.99 percent of people don't know. Be careful!". Therefore, we use the characteristics of historical rumor data to classify a small number of epidemic rumor data since historical rumors are implicitly related to epidemic rumors. Besides, meta-learning is a technique for training a model on a small batch of data in order to develop the ability to generalize across different training sets. The excellent method can be effective in helping the model simulate the low-resource scenario and deal with new rumors during the early stages of an epidemic. Our objective is to develop an epidemic rumor detection model based on meta-learning training and FSL.

This paper proposes a rumor detection model based on the prototype network [21] in FSL. Figure 2 illustrates a comparison between the proposed CNFRD model and the prototype network, in which c_1, c_2, and c_3 represent the class-wise representation of all vectors belonging to the same class, i.e., the prototypes. Figure 2(a) depicts the main steps of the prototype network.

(i) Step 1 (sample-wise vector): The sample-wise vector representation is generated for all samples in the support set.

(ii) Step 2 (class-wise vector): Each class has a prototype created by taking the average of all the vectors in that class.

(iii) Step 3 (measure the distance): The network calculates the distance between the query sample and each prototype representation. As a result, the query sample will then be classified by its nearest prototype.

Figure 2(b) portrays the concept of CNFRD, which is similar to the prototype network in a certain sense. Their main differences include:

(i) Representations of prototypes are acquired differently. The prototypes in the prototype network are calculated based on averaging the vectors of samples belonging to the same class. In contrast, CNFRD prototypes are produced by introducing the capsule network.

(ii) Distances are measured in a variety of ways. The prototype network utilizes Euclidean distances, while the CNFRD model uses a modified cosine similarity function.

The calculation of class-wise features using FSL learning, such as the prototype network [21] and relation network [22], depends heavily on simply summing or averaging representations over samples belonging to the same class. Much critical information in rumors may be lost due to simplistic representations. Therefore, this paper introduces the capsule network (Capsnet) [23] initialed by Sabour et al. to generate the epidemic rumor class representation based on historical rumors. Utilizing the advantages offered by the convolutional neural network (CNN) for extracting spatial features, the Capsnet provides the structure of the capsules

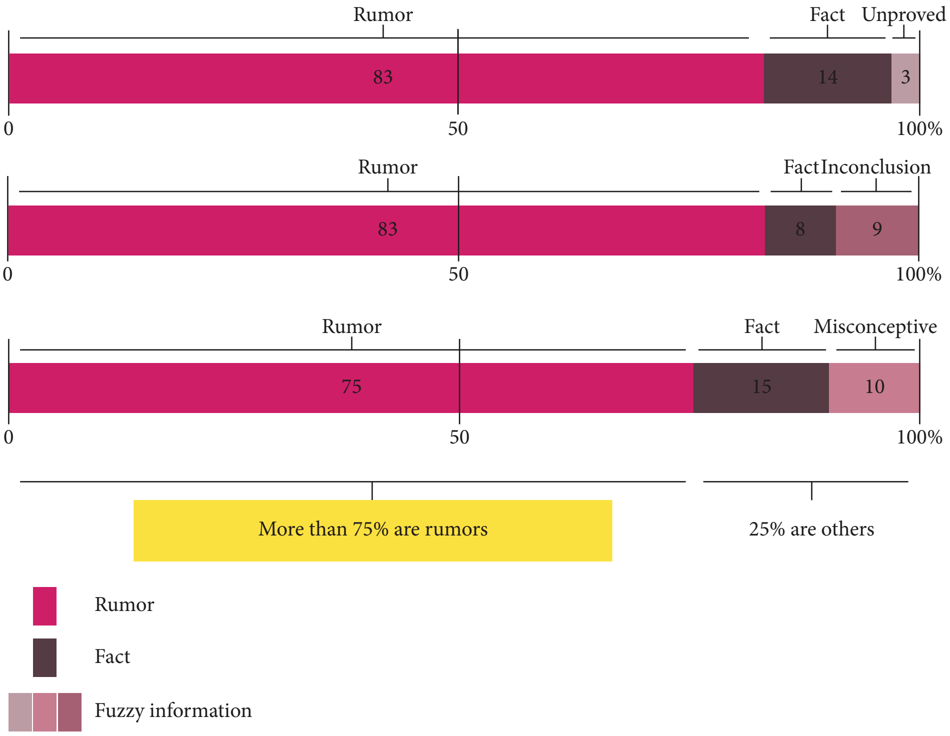


Figure 1: Distribution of rumor identification results on China's three major rumor-refuting platforms (from top to bottom: Baidu rumor-refuting platform, China Internet Joint Rumor-refuting Platform, and Tencent's Jiaozhen rumor-refuting platform).

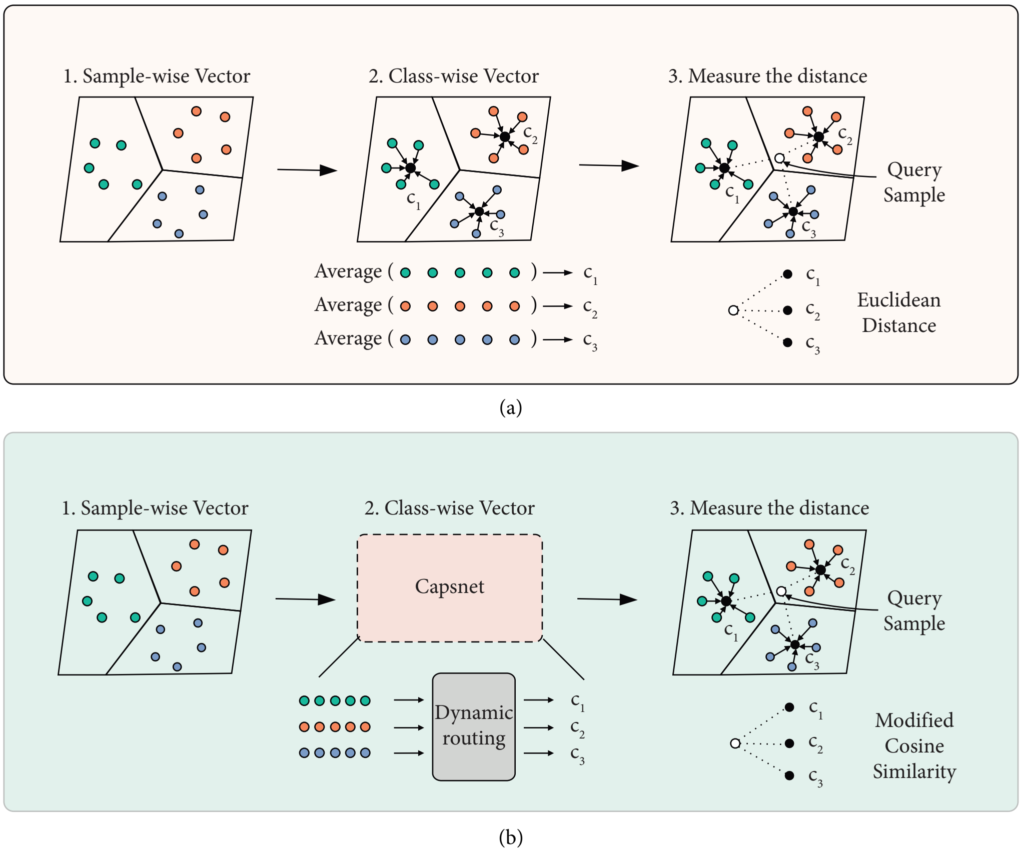


Figure 2: Comparisons of the proposed CNFRD model and the prototype network. (a) Prototype network. (b) CNFRD.

as well as the feature selection method of dynamic routing. High-level capsules in the Capsnet represent the overall abstract characteristics of samples, while low-level capsules depict their local features. In addition, the dynamic routing mechanism aggregates lower-level capsule information to update high-level capsules. Compared to traditional CNN, the Capsnet performs better in few-shot classification with only a small amount of training data and has greater interpretability and generalization capabilities. Moreover, capsules in the Capsnet contain rich information, including spatial locations and strong correlations between neighboring nodes, which posses the ability to preserve the underlying details in the original data. These properties fit well with the connection and natural ordering of contexts in text data. Consequently, the Capsnet can extract semantic information from words and facilitate the correct categorization of texts.

In summary, this paper proposes a novel few-shot learning rumor detection model based on a capsule network (CNFRD). Specifically, individual historical rumor data samples are regarded as elements, while the historical rumor class is treated as a whole. The capsule encodes the internal spatial relationships between the elements such that a generalized class-level representation of historical rumor data samples can be obtained by performing dynamic routing. Then, we measure the distance between the historical rumor class-wise vector and the epidemic rumor sample-wise vector. Finally, the nearest neighbor principle is applied to determine whether it is a rumor.

To verify the effectiveness of our CNFRD model, we conducted rumor detection experiments on two representative real-world datasets in Chinese and English, respectively. Experiments reveal that our proposed model can identify rumors earlier with higher detection accuracy even with a limited number of epidemic rumor samples. A list of key acronyms used in this paper is summarized in Table 1. The contributions of our work are summarized as follows:

(i) This study applies a metric learning framework of few-shot learning to the rumor detection task, which employs the features from a large amount of historical rumor data to classify a small amount of epidemic rumor information. This approach effectively deals with the problem of small amounts of data in the early stages of epidemic rumor propagation by combining a well-established meta-learning training approach.

(ii) Our study introduces the capsule network as a rumor class generator to enhance the quality of rumor class-wise representation. We propose a few-shot rumor detection model for COVID-19. This is the first study to combine a metric learning framework with the capsule network for social media rumor detection.

(iii) Our experiments on Chinese and English datasets have demonstrated that the proposed CNFRD model has obvious advantages and can detect epidemic rumors equally well with fewer epidemic rumor samples, reaching 88.92% accuracy on the Chinese dataset and 87.07% accuracy on the English dataset. The performance is higher than the existing models by 7% to 23%, and proves the effectiveness of our model.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 elaborates some preliminary knowledge about rumor detection and meta-learning. Section 4 details the developed framework for rumor detection. Section 5 discusses the main findings in light of experimental results. The conclusion and future work are represented in Section 6.

2. Related Work

Three aspects of related research and recent trends are discussed in this section, including rumor detection methods on OSN, few-shot learning methods, and capsule network.

Rumor detection on OSN has gained considerable attention in recent years. Early studies on rumor detection have primarily focused on shallow features of rumors. It is common for rumor mongers to use emotive language and novel words in their messages in order to catch the attention of the public. Hence, rumor messages frequently share similar language expressions, word usages, punctuations, etc. Yang et al. [24] initially developed the rumor detection method based on Sina Weibo. They introduced location features and client features and verified the effectiveness of the features through corresponding quantitative experiments. Takahashi and Igata [19] introduced retweet rate, outbreak points, and word distribution as the main features for analyzing Twitter data related to the event of the "Japanese earthquake-triggered tsunami". The study determined a distinct difference between rumors and nonrumors in terms of word distribution. Kwon et al. [20] used temporal, structural, and linguistic features to improve the detection of rumors. However, hand-crafted feature extraction is time-consuming and labor-intensive, resulting in feature bias and limiting it to specific scenes. The hand-crafted methods for detecting rumors are not effective at identifying unknown rumors [25].

Several rumor identification methods relying on deep learning have emerged, and many studies have explored how deep learning can be used to identify rumors based on semantic features [26]. Ma et al. [27] first used a recurrent neural network (RNN) to learn the hidden contextual information of relevant posts and combine it with time-series information to determine the credibility of suspicious statements or events. The effectiveness of RNN in rumor detection was demonstrated through experiments. A study by Yu et al. [28] recommended using CNN for capturing text semantic features. Chen et al. [29] proposed a new algorithm for extracting text features from tweet using the attention mechanism based on the RNN model to achieve better tweet text feature extraction. It is believed that deep neural network methods are capable of more accurately identifying rumors because they use continuous vectors to represent text

Table 1: A list of key acronyms.

Full form	Abbreviation
Bidirectional encoder representation from transformers	BERT
Bidirectional long short-term memory	BiLSTM
Novel coronavirus disease 2019	COVID-19
Online social networks	OSN
Few-shot learning	FSL
Capsule network	Capsnet
Convolutional neural network	CNN
Recurrent neural network	RNN
Graph convolution network	GCN
Natural language processing	NLP
Generative adversarial network	GAN
Variational auto-encoder	VAE
Gated recurrent unit	GRU
Support vector machine	SVM
Euclidean distance	EuD
With	W/
Without	W/O

instead of sparsity features. Yang et al. [30] applied the adversarial graph framework to construct a heterogeneous information network to model the rich information between users, posts, and user comments. Bian et al. [31] applied a graph convolution network (GCN) for the first time to rumor detection on OSN, accounting for propagation and dispersal characteristics. Rao et al. [32] used the stacked BERT model for rumor detection. The method is fed into two similar BERT modules for respective characterization learning after obtaining corresponding posts by both timing sequences and emotion score orders. These approaches are integrated and input into the full connection layer for classification prediction. Srinivasan [33] deployed two parallel CNN networks to minimize the data imbalance and facilitate flexible feature extraction. This automated rumor detection approach provides better results for larger and scale-free networks. Their experimental results on real-life datasets indicated that the model achieved good performance in rumor detection. Sahoo and Gupta [34] proposed an automatic fake news detection approach in a chrome environment. They used multiple features associated with Facebook accounts and some news content to analyze the account's behavior through deep learning. Tembhurne et al. [35] utilized a multichannel deep learning model, leveraging and processing the news headlines and news articles along different channels to differentiate between fake and real news. Recently, researchers have explored external knowledge in rumor detection studies because interpreting news content usually requires sufficient background or professional knowledge [14, 36]. In addition to the above work, researchers have operated adaptive learning [37], fuzzy graph convolutional networks [38], multimodal network [39], and other methods to detect rumors on OSN. Even though all the studies above indicate that detecting rumors on OSN by employing various approaches is effective, certain limitations are identified as follows:

(i) In the early stages of rumor propagation, there are limited data available. As a result, detecting rumors using neural networks is a complicated problem, rendering it less efficient for early-stage rumors [40].

(ii) Some deep learning-based rumor detection methods involve a lot of parameters and require massive corpora for calculation.

During COVID-19, many OSN have sprung up rumors [41]. However, this data remains insufficient compared to thousands of historical rumors. Developing a method for detecting rumors on OSN is challenging. Hence, we propose a model that uses few-shot learning combined with the capsule network to overcome the problems associated with previous studies. Among them, few-shot learning is capable of learning large amounts of data from a given class and then quickly learning new classes with fewer samples to solve the problem of insufficient data for early rumors detection. Experiments show that the parameter number of the proposed model is 8 M, which is 77% less than the 35 M parameters of the CNN deep learning model.

The human brain can recognize new objects with a few examples as opposed to the deep learning models described above, for which large amounts of data are required. A child also does not need to see many pictures to understand what a sheep or tiger are. In order to replicate the rapid learning capacity of human beings, few-shot learning [42] is proposed that allows a computer to learn about new objects with only a few labeled examples, as opposed to the transitional deep learning models which require large amounts of data.

Few-shot learning has emerged in the field of computer vision research [43–45] and more recently in natural language processing (NLP). Some few-shot methods [46–48] aim to fine-tune a pretrained model in a specific way [49]. This means that they train the model on a large-scale data, then fine-tune its parameters (full connection layer or the top layer) on the specific small-sample dataset. The final fine-tuning model is more likely to be accurate. However, these approaches can lead to overfitting of the model on the target dataset since a small number of data do not reflect the true distribution of a large number of data.

The few-shot problem can be solved more intuitively through data augmentation that is a process of expanding or enhancing the original small-sample dataset with additional (auxiliary) information. Xian et al. [50] developed a method for generating zero-shot/few-shot samples for few-shot learning tasks. The method constructed VAE [51] +GAN [52] model, but unfortunately it also introduces noise.

Some researchers believe that metrics-based approaches are effective under resource-constrained conditions. If there is a specific correlation between two domains (source and target), then knowledge can be transferred between them by using the knowledge and features obtained in the source domain. The transformation can help train the classification model in the target domain. Koch et al. [53] first proposed to use the siamese neural network for one-shot image recognition, map input into target space via the embedding function, and calculate similarity through a simple distance calculation. Snell et al. [21] proposed the prototype network that applies a deep neural network to map images into vectors. For each class the model is trying to learn, there is a prototype (a summary vector) representing that class. This vector is created by taking the average vector value of all the samples that belong to that class. As a result, the prototypes for each sample class constitute the vector space. Then, the metric between the class representation and the query embedding can be measured to determine their relationship. The prototype network [21] greatly influenced the subsequent work. Geng et al. [54] used a dynamic routing algorithm to compute the prototype representation. Each cycle of dynamic routing results in a preliminary prototype derived from weighting and summarizing the sentence vectors of the support set, followed by the final output derived from multiple cycles. Li et al. [55] proposed a novel and compact end-to-end covariance metric network that can represent concepts (or categories) with covariance matrices, and then compute the consistency between query samples and categories to construct a covariance metric as a relational measure. Zhao et al. [56] designed the self-guided information convolution model, an improved convolution structure, which utilizes the high-level features to guide the network to extract the required discriminative features for few-shot classification. Xu and Xiang [57] proposed a multiperspective aggregation-based graph neural network that observes through eyes (support and query instance) and speaks by mouth (pair) for few-shot text classification. Pang et al. [58] proposed an adapted bidirectional attention mechanism to exploit the interaction between query and support instances in metric learning to better describe text classification features.

The model we constructed was derived from the prototype network. Still, it introduced the Capsnet and dynamic routing to learn generalized class-level representation based on historical rumor data samples instead of the average of sample vectors. The Capsnet is an emerging neural network that improves classification accuracy in small datasets and achieves excellent results in image classification. However, there are limited studies on the application of the Capsnet in NLP. Wang et al. [59] applied the Capsnet to sentiment analysis. Their model relied on the idea of creating capsules and focusing each capsule on a specific sentiment. Kim et al. [60] followed up by exploring the application of the Capsnet architecture in text classification and obtained certain results in multiple datasets. The improved capsule network model proposed by Yang et al. [61] is suitable for supervised text classification of large labeled datasets. Local information diversity is improved within the capsule by adding shared and unshared matrices. The noise in the network is reduced by introducing isolated categories and updating connection strength. As a model extension of Yang et al. [61], Xia et al. [62] developed the capsule-based architecture to determine the similarity of target and source intents. Inspired by the excellent performance of the Capsnet in image classification, Ren and Hu [63] designed a capsule network combined with a compositional weighted coding method for text classification. They also offered a new routing algorithm based on the k-means clustering theory to thoroughly mine the relationship between capsules. Chen et al. [64] applied the graph convolutional network to learn label embeddings and the correlations between labels, a fusion layer to combine the label information with the contextual semantic information of texts, and the Capsnet to extract the spatial feature information of texts. Kenarang et al. [65] combined the attention mechanism and the Capsnet method to obtain text topics in the Persian news corpus. The results of the comparison show an improvement in the classification performance of the Persian texts. Our study attempts to introduce the idea of the capsule neural network to rumor detection tasks against the background of COVID-19. The effectiveness of the proposed CNFRD model has been validated through the related indicators obtained from a series of experiments.

3. Problem Definition

3.1. Rumor Detection. This paper defines rumors as officially unverified information that is spread around on OSN and causes a significant (normally negative) social impact. Figure 3 illustrates the composition of rumors on OSN. Rumor detection is the process of identifying false or unverified information that is being disseminated on OSN. This can be done by classifying messages as either rumors or not. There are two types of information involved in social media rumor detection: (i) detect rumors based on a single post, and (ii) detect rumors within a set of posts [66]. This paper presents a rumor detection model based on scenario (i).

Let $\mathscr{M} = (a_1, a_2, \ldots, a_N)$ denote a post from an online social media platform, where a_i indicates a single word, number, or symbol, and N indicates the number of characters contained in the post. Our objective is to learn the prediction function $\mathscr{F}\colon \mathscr{M} \longrightarrow e^f \longrightarrow \{1, 0\}$. More Specifically, we convert the post $\mathscr{M}$ into an f -dimensional matrix-vector e^f and finally turn it into a credibility binary classification problem, i.e., distinguish between nonrumor (0) and rumor (1), as shown in equation (1). It is important to note that all vectors e^f have f -dimensions. Hence, this study abbreviates e^f with e for simplicity.

$$\mathscr{F} = \begin{cases} 1, & \text{if } \mathscr{M} \text{ is a piece of rumor,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

3.2. Meta-Learning in Few-Shot Learning. Contrary to traditional data partitioning, meta-learning divides data into three separate components: meta-training, meta-validation, and meta-testing. The three components are also known as episodes in meta-learning, which simulate the few-shot situation by using a specially divided small set of data. Typically, an episode contains C classes of data involved in the current phase of the data pool. Each class of data has $K + Q$ randomly selected instances, where K of the instances are labeled and the remaining Q unlabelled. In an episode, the $C \times K$ labeled data is referred to as the support set, while the $C \times Q$ unlabelled data is known as the query set. Several few-shot problems with K labeled examples for each C class in the support set are known as C -way K -shot problems.

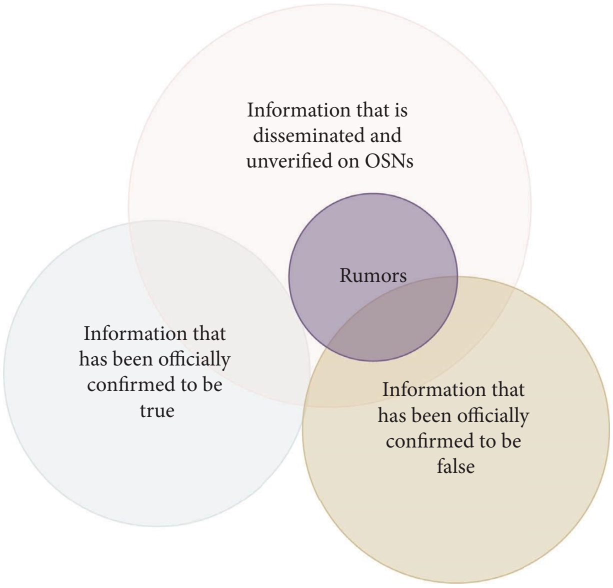

Information that is disseminated and unverified on OSNs
Rumors
Information that has been officially confirmed to be true
Information that has been officially confirmed to be false

Figure 3: Rumor composition on OSN.

The model is supervised and trained on the support set for each episode, and its performance is verified on the query set. Consequently, we aim to perform meta-learning in the training set in order to extract transferable knowledge from multiple episodes and generalize a small set of labeled data to a large set of unlabelled data.

4. Methodology

The objective of this section is to establish the framework of a particular CNFRD model that can determine whether a post is a rumor or not. The model is explained in detail, including each module and the definition of the objective function. The notations we will use throughout the article are summarized in Table 2.

4.1. Model Overall Framework. We are seeking to determine whether a post is a rumor. This goal can be achieved by establishing a particular CNFRD model. Figure 4 depicts the three main components of the model. The first component is the embedding module based on BERT + BiLSTM. This module is responsible for generating sample-wise vector representations of the input datasets, including the epidemic rumor dataset and historical rumor dataset with labels. The second component is the rumor class-wise representations ("Rumor" and "Nonrumor") based on the capsule network. This component generates class-wise vector representations of the historical rumor data. The third component is the metric module based on the modified cosine similarity. This component is responsible for comparing two rumor class-wise representations with the previous sample-wise vector of epidemic rumors. When epidemic rumor class-wise vectors are similar to historical rumor class-wise vectors, their distance will be closer, and vice versa, the distance between them will be "pushed." The result will be output in accordance with the nearest neighbor principle after constant iteration and updating.

Table 2: Notations and explanations.

Notation	Explanation
$\mathscr{M}$	A post from an online social media platform
a_i	i-th character in a post
$\mathscr{F}$	Prediction function
e^f	f-dimensional matrix-vector
C	Number of classes contained in an episode
K	Number of instances in each class that are labeled
Q	Number of instances in each class that are unlabelled
x_i	Vectorized representation of the i -th character in a post
Q_m	Query matrix-vector
K_m	Key matrix-vector
V_m	Value matrix-vector
d_k	Dimension of a query and key vector
W_i	Weight matrices corresponding to the i-th header
W	Weight
b	Bias value
C_t	Storage cell state of BiLSTM
$\widetilde{C}_t$	Temporary cell state of BiLSTM
h_t	Hidden state of BiLSTM
f_t	Forgetting gate of BiLSTM
i_t	Memory gate of BiLSTM
o_t	Output gate of BiLSTM
σ	Logistic regression function
H^T	T hidden states of the whole BiLSTM
HD	Historical rumor dataset
ED	Epidemic rumor dataset
h_t	Hidden state
u_i	Support set vector
u_t	Query set vector
p	Number of neurons in the output capsule
a	Number of neurons (vector length) in each capsule
$\widehat{u}_{j\vert i}$	Prediction vector in the Capsnet
W_{ij}	Transformation matrix in the Capsnet
b_{ij}	Connection weight in the Capsnet
c_{ij}	Coupling coefficient in the Capsnet
s_j	Output of the j -th upper capsule in the Capsnet
v_j	General representation of the j -th historical rumor class
e_q	Epidemic rumor query sample
v_{pos}	A sample of the same category as e_q
v_{neg}	A sample of a different category from e_q

4.2. Embedding Module Based on BERT + BiLSTM. Several existing studies have shown that combining models yields better results for extracting and representing text features than a single network architecture [67, 68]. This paper embeds the CNFRD model with the BERT model for generating the character-level dynamic feature vector of rumor text, thereby relieving the lack of a lexicon for emerging epidemic rumors at the model input stage. Furthermore, the BiLSTM model incorporates both textual information and sequential features of sentences to extract semantic features of rumor texts. Our framework combines the BERT model with the BiLSTM model as shown in Figure 5, which can handle variable-length sequence data, allowing for more complex semantic features and accurate semantic representations. The embedding process is described as follows.

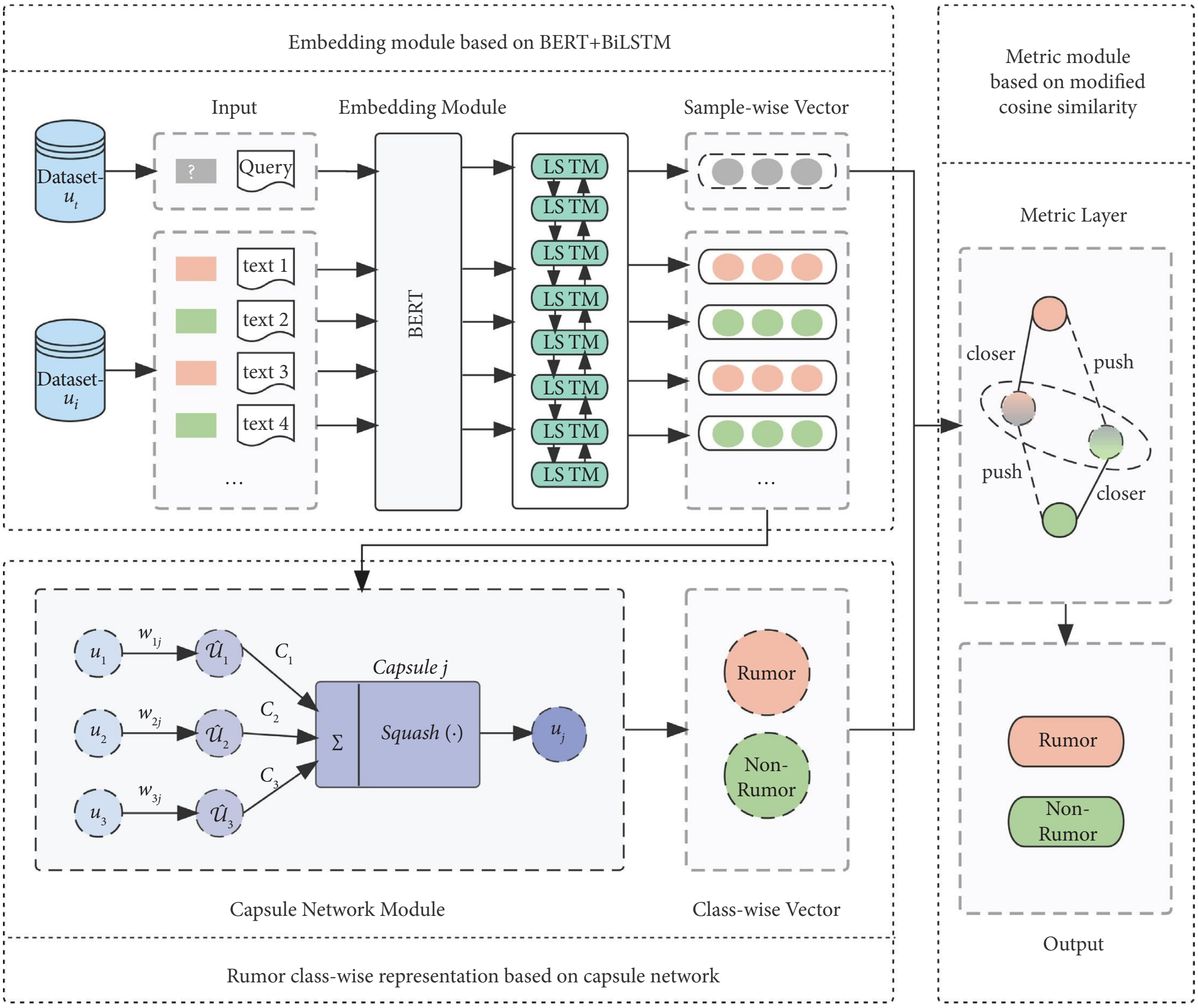


Figure 4: Model structure of the CNFRD.

BERT is used as the character vector embedding layer to achieve the process of creating a vectorized feature representation with respect to the pretrained text. First, the initialization of a text involves splitting it into individual characters, including the insertions of [*CLS*] and [*SEP*] identifiers at the beginning and end of the text, respectively. Second, each character in the text is encoded sequentially (starting at 0) to obtain position information. The character-level embedding vector and position vector are added as inputs to the model.

Given an input text $\mathscr{M} = (a_1, a_2, \ldots, a_N)$ represented by a sequence of words, our objective is to obtain a vectorized representation $e = (x_1, x_2, \ldots, x_N)$ of the text $\mathscr{M}$ based on the features extracted by the bidirectional transformer encoder, the core structure of BERT. The encoder will take the input sentence and pass it through the first sublayer i.e., the multihead self-attention layer. This layer will assist the encoder focus on the other words in the input sentence while it is encoding the sentence. Its output is then passed to the fully connected feed-forward layer. Normalization and residual connections are added to each sublayer. The total output of each sublayer can be calculated using the following equation:

$$\text{Output}_{\text{sublayer}} = \text{LayerNorm}\,(x + \text{sublayer}\,(x)). \tag{2}$$

Where sublayer$\,(x)$ is a function implemented by the sublayer itself. To facilitate these residual connections, all sublayers and embedding layers in the model produce output in $d_o = 512$ dimensions.

The self-attention layer contains three vectors for each word: the query matrix-vector (Q_m), the key matrix-vector (K_m), and the value matrix-vector (V_m). These vectors are derived by multiplying the embedding vector with three weight matrices. Each of them has a length of 64. The self-attention layer uses these vectors to determine the weight of each word in the input vector and maps the query (Q_m) and the key-value pairs $(K_m,\ V_m)$ to the output vector. The output is computed as a weighted sum of values. Each value is weighted by the compatibility function of the query with the corresponding key. The self-attention layer is calculated as shown in the following equation:

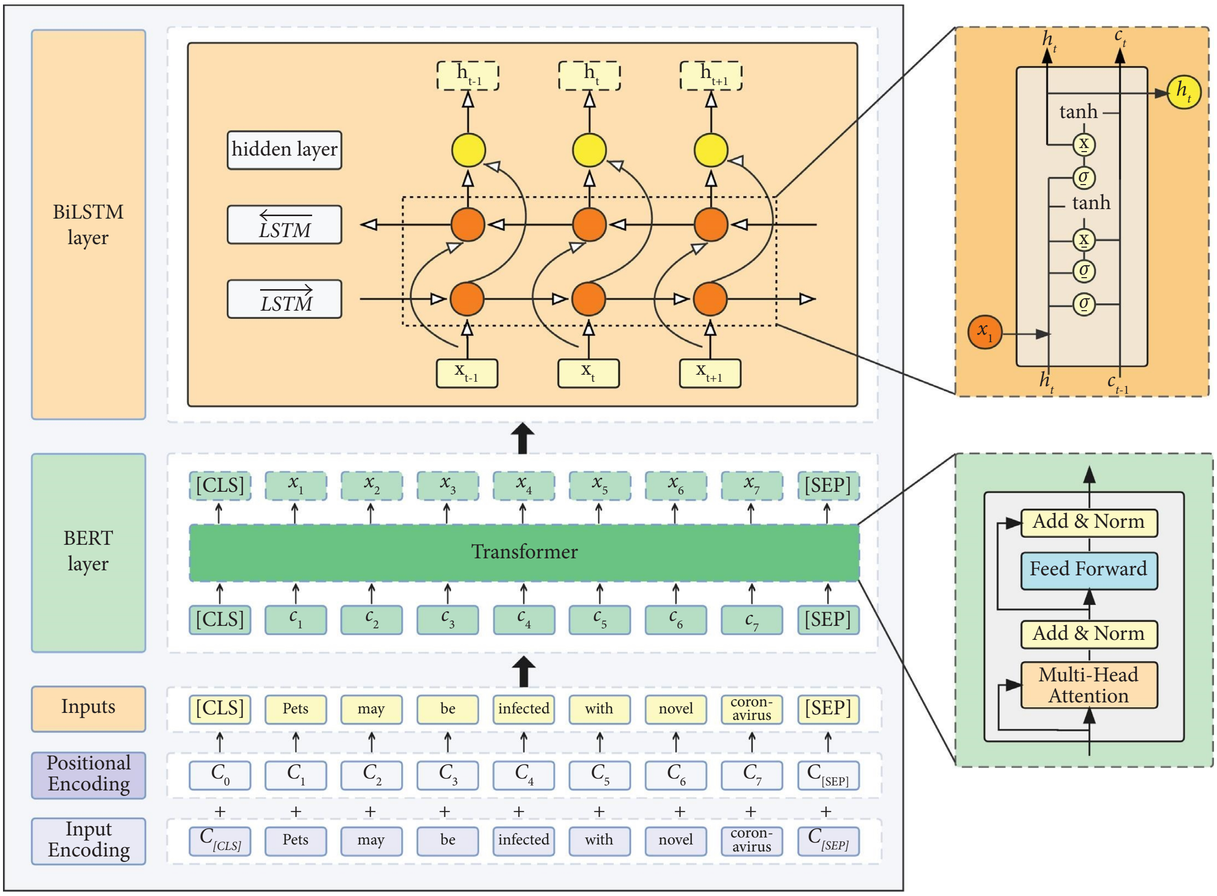


Figure 5: Framework diagram based on BERT + BiLSTM embedding module.

$$\text{Attention}\left(Q_m, K_m, V_m\right) = \text{softmax}\left(\frac{Q_m K_m^T}{\sqrt{d_k}} V_m\right), \tag{3}$$

where the dimension of a query and key vector is d_k. To make attention calculations more comprehensive, we employ a multihead attention mechanism. First, the input is linearly mapped several times to generate the query matrix, key matrix, and value matrix. Second, the matrices are used to calculate the scaled dot product attention for each input sentence. The result of each calculation is referred to as a head. Besides, the attention matrices obtained from multiple operations are stitched together horizontally, which are compressed into one matrix by multiplying a weight matrix. The specific calculation formulas are represented as follows:

$$\text{head}_i = \text{Attention}\left(Q_m W_i^{Q_m}, K_m W_i^{K_m}, V_m W_i^{V_m}\right), \tag{4}$$

$$\text{MultiHead}\left(Q_m, K_m, V_m\right) = \text{Concat}\left(head_1, head_2, \ldots, head_m\right) W^O, \tag{5}$$

Where $Q_m \subseteq R^{(m \times d_k)}$, $K_m \subseteq R^{(m \times d_k)}$, $V_m \subseteq R^{(m \times d_k)}$, and parameter m denotes the number of characters. $W_i^{Q_m}, W_i^{K_m}$, and $W_i^{V_m}$ represent the three weight matrices corresponding to the i-th header. The Concat function splices the computation results of multiple headers. W^O is the weight matrix used in the splicing.

Thereafter, the output of the multiheaded attention layer is fed into a fully connected feed-forward network layer, which contains multiple activation functions and can be summarized as follows

$$\text{Output}_{\text{FeedFN}} = \text{dropout}\left(\text{RELU}\left(W_A * \text{Attention}\left(Q_m, K_m, V_m + b_A\right)\right)\right) W_F + b_F, \tag{6}$$

where W_A and b_A are the weights and bias values of the multiheaded attention layer, respectively. W_F and b_F are the weights and bias values of the feed-forward network layer, respectively.

Character vectorized representations of the BERT layers are applied as the semantic representation of the rumor data, which is then input into BiLSTM for feature extraction and mining. The BiLSTM model is composed of the input characters X_t at moment t, the storage cell state C_t, the temporary cell state $\widetilde{C}_t$, the hidden state h_t, the forgetting gate f_t, the memory gate i_t, and output gate o_t. BiLSTM passes valid information to subsequent calculations by discarding useless information and remembering new information. The LSTM cell calculation equations (7)–(14) are represented as follows:

$$f_t = \sigma\left(W_f \cdot [x_t, h_{t-1}] + b_f\right), \tag{7}$$

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i), \tag{8}$$

$$\widetilde{C}_t = \tanh\,(W_C \cdot [x_t, h_{t-1}] + b_C), \tag{9}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \tag{10}$$

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o), \tag{11}$$

$$\overrightarrow{h_t} = o_{t-1} * \tanh(C_{t-1}), \tag{12}$$

$$\overleftarrow{h}_t = o_{t+1} * \tanh(C_{t+1}), \tag{13}$$

$$h_t = \text{Concat}\left(\overleftarrow{h_t}, \overrightarrow{h_t}\right), \tag{14}$$

where σ is the logistic regression function. W and b represent the weights and bias values of neurons, respectively. The operator $[a, b]$ denotes combining the vectors a and b into a single vector. The T hidden states $H = (h_1), (h_2, \ldots, h_T)$ of the BiLSTM model are used as input to a softmax classifier, which is a process to predict the label $y \in \{0, 1\}$. As a result, a vector with weight W can be obtained as shown in the following equation:

$$W = \text{softmax}\left(W_{s1} \tanh\left(W_{s2} H^T\right) + b_S\right), \tag{15}$$

where $W_{s1} \in R^{d_a \times 2u}$ and $W_{s2} \in R^{d_a}$ denote the weight matrices, d_a is a hyper-parameter, u is the hidden state size of each one-way LSTM, and b_S represents the bias value. The final vector representation e of a single text is a weighted sum of H, as shown in the following equation:

$$e = \sum_{t=1}^{T} W_t \cdot h_t. \tag{16}$$

4.3. Rumor Class-Wise Representation Based on Capsule Network. This section attempts to obtain an accurate and appropriate general representation of the historical rumor class. A learning approach based on the Capsnet model and dynamic routing algorithms is proposed for generalizing to unknown popular rumor classes. The vectorized capsules can encode spatial information more accurately than traditional scalarized neurons and can also be used to determine the probability of the existence of objects. Our purpose is to determine the semantics of classes that are invariant to sample-wise noise. The process is achieved by utilizing sample representations and class representations as input and output capsules, respectively. The direction of a capsule output vector reflects the semantic patterns or classifications of a text representation, while the vector length represents the probability of the text entity. In addition, capsules can express information about spatial entities, such as the angles and positions of a text.

The vector e obtained from the historical rumor dataset (HD) by performing (16) is used as the support set vector u_i. The support set data is generated by drawing K samples from each of the C classes, which is primarily to train the model parameters. A query set vector u_t is constructed using the rumor dataset (ED) from the epidemic to be detected. The query set data consists of several randomly selected data from the remaining parts of the C class. The data is applied to simulate real-world requests and calculate the loss.

Let p be the number of neurons in the output capsule and a be the number of neurons (vector length) in each capsule. Then, the input $u_i \in \mathbb{R}^a$ can be transformed into the prediction vector $\widehat{u}_j i$ in the following equation:

$$\widehat{u}_{ji} = W_{ij} u_i \in \mathbb{R}^p, \tag{17}$$

where $W_{ij} \in \mathbb{R}^{p \times a}$ is the transformation matrix indicating the important connections between encoding low-level capsule vectors (features) and high-level capsules vectors (features), which can be learned and updated by the back-propagation algorithm of the neural network.

Subsequently, the dynamic routing mechanism reported in [23] is applied in our approach. The mechanism has focused on replacing the pooling operation of the CNN. Researchers have been able to adjust the association strength of the upper and lower capsules by iteratively updating the coupling coefficient c_{ij}. Consequently, the model can capture the part-whole relationship and predict the similarity of upper and lower capsules without sacrificing certain essential features. As shown in Algorithm 1, the basic dynamic routing scheme is described as follows:

This algorithm initially establishes the connection weight $b_{ij} = 0$ of the i-th capsule to the j-th upper-level capsule. Coupling coefficients c_{ij}, quantifying the connection between a low-level capsule and its parent capsule, are defined as probability distributions of the i-th capsule that activates the j-th capsule. The coefficients can be formalized as follows, and we have $\sum_j c_{ij} = 1$:

$$c_{ij} = \text{softmax}\left(b_{ij}\right) = \frac{\exp\left(b_{ij}\right)}{\sum_k \exp\left(b_{ij}\right)}. \tag{18}$$

The output s_j of the j-th upper capsule is calculated by summing the coupling coefficients c_{ij} as weights with the corresponding prediction vectors $\widehat{u}_{ji}$, as shown in the following equation:

```
Procedure ROUTING $(\widehat{u}_{ji}, r, l)$
  for all capsule (HTML translation failed) in layer $l$ and capsule $j$ in layer $(l+1)$: $b_{ij} \leftarrow 0$.
  **for**$r$ iterations **do**
    for all capsule $i$ in layer $l$: $c_i \leftarrow$ softmax $(b_i)$
    for all capsule $j$ in layer $(l+1)$: $s_j \leftarrow \sum_i c_{ij} \widehat{u}_{ji}$
    for all capsule $j$ in layer $(l+1)$: $v_j \leftarrow \text{squash}(s_j)$
    for all capsule $j$ in layer $l$ and capsule $j$ in layer $(l+1)$:
      $b_{ij} \leftarrow b_{ij} + \widehat{u}_{ji} \cdot v_j$
**return** $v_j$
```

Algorithm 1: Dynamic routing algorithm.

$$s_j = \sum_i c_{ij} \widehat{u}_{ji}. \tag{19}$$

Upper and lower capsule weights are updated by the inner product of $\widehat{u}_{ji}$ and v_j. This process increases and decreases the coupling coefficient between similar capsules and between unrelated capsules, respectively. As a result, capsules of similar characteristics are more likely to be coupled. However, it is necessary to express the existence probability of the corresponding entity by the vector length of the capsule output. We construct a nonlinear function "Squashing" that compresses the length of the output vector to [0, 1] without modifying the direction. Long and short vector lengths are compressed as much as possible to 0 and 1, respectively. The final vector output of the capsule after reaching the set routing times is formalized by the following equation:

$$v_j = \text{squash}\left(s_j\right) = \frac{\left\| s_j \right\|^2}{1 + \left\| s_j \right\|^2} \frac{s_j}{\left\| s_j \right\|}. \tag{20}$$

4.4. Metric Module Based on Modified Cosine Similarity. The BiLSTM + BERT embedding module is used to encode each query text in the epidemic rumor dataset into a query vector e_q. Once the general representation of the historical rumor class v_j has been obtained, this query vector can be used to compare with the class-wise representations in order to determine the particular class of the query text. Two vectors will be generated and sent to the metric module for further processing. The metric module calculates the correlation between the epidemic rumor query sample and the historical rumor class, i.e., calculate the distance between each pair (v_j, e_q). Finally, the module utilizes the nearest neighbor principle to complete the classification. A modified cosine similarity function (21) is constructed in the metrics module for normalizing the data by subtracting a mean value from the cosine similarity in all dimensions, thus correcting for the error caused by insensitivity to values.

$$\text{Sim}\left(v_j, e_q\right) = \frac{\sum \left(v_j - \overline{CR}\right)\left(e_q - \overline{CR}\right)}{\sqrt{\sum \left(v_j - \overline{CR}\right)^2} \sqrt{\sum \left(e_q - \overline{CR}\right)^2}}. \tag{21}$$

Where $\overline{CR}$ is the mean value. The metric module returns a similarity score between 0 and 1. Cosine values closer to 1 indicate greater similarity between the two vectors.

4.5. Objective Function. Cross-entropy loss can explore the relationship between a single sample and all classes. Besides, triplet losses can significantly simplify and streamline the training process. A neuron network model can benefit from the simplicity of metric learning and excellent nonlinear modeling abilities by using triplet loss. Hence, a combination of cross-entropy and a triplet loss function is used to train our model. Model performance can be improved by combining both of them. Hence, the similarity scores $\text{Sim}(v_j, e_q)$ should be regressed to the true label y_q. The similarity between class and query sample matching pairs is 1, while the similarity between unmatched pairs is 0.

Given the number of query sets NQ and the number of class N, the cross-entropy loss function is defined as follows:

$$\text{Loss}_{\text{cp}} = -\frac{1}{N} \sum_q^{NQ} \sum_i^N \log \frac{\exp\left(Sim\left(e_q, v_j\right)\right)}{\sum_{k=1}^N \exp\left(\text{Sim}\left(e_q, v_k\right)\right)}. \tag{22}$$

The triplet loss function can be formalized as follows:

$$Loss_{tl} = \sum_q^{NQ} \max\left(Sim\left(e_q, v_{\text{pos}}\right) - \text{Sim}\left(e_q, v_{\text{neg}}\right) + \text{margin}, 0\right). \tag{23}$$

Where v_{pos} is a sample of the same category as e_q, v_{neg} is a sample of a different category from e_q, and margin is a hyper-parameter that increases (res. decreases) the distance between samples belonging to different (res. same) labels. Two loss functions are combined during training and the global loss function can be expressed as follows:

$$\text{Loss}_{\text{Global}} = \alpha \times \text{loss}_{\text{cp}} + \beta \times \text{loss}_{tl}. \tag{24}$$

5. Experimental Evaluations

This section introduces the datasets, baseline methods, and experimental settings used in our experiments on CNFRD. We then report experimental results and analysis to identify rumors. Ablation experiments are performed to analyze the effectiveness of each module in the model. Finally, the complexity and overhead analysis are presented for the proposed method.

5.1. Datasets. Chinese and English datasets are used to test the CNFRD model's performance in detecting rumor patterns. The relevant statistics are shown in Table 3.

(i) Chinese dataset ($Dataset_{\mathrm{CN}}$): (i) The historical rumor dataset was adopted from the Chinese rumor dataset (https://github.com/thunlp/Chinese_Rumor_Dataset) provided by Song et al. [69]. We also collect a large amount of historical rumor data in various fields and aspects for the supplement. All data are from the platform of false information reporting on Sina Weibo. (ii) A CHECKED dataset provided by Yang et al. [70] is used to build COVID-19 rumor data. On this basis, the officially recognized rumor samples are collected from the Sina Community Management Center to form the COVID-19 rumor corpus required by the experiment. The data were obtained from the Chinese social networking platform Weibo.

(ii) English dataset ($Dataset_{EN}$): (i) Historical rumor data is drawn from FakeNewsNet dataset (https://github.com/KaiDMML/FakeNewsNet) proposed by Shu et al. [71], which is mainly data samples collected from PolitiFact and GossipCop. In order to complete the experiment, we select the rumor events summarized on Snope to complement the historical rumor dataset. (ii) The COVID-19 rumor epidemic rumor data were collected using a dataset provided by Cheng et al. [72], which consist of rumors from two sources: the news from various news websites and the data from Twitter. The veracity status of the rumors has been determined via fact-checking websites.

In light of the unique characteristics of epidemic rumors, we select the rumors from Xinhua (http://www.xinhuanet.com), People's Daily Online (http://www.people.com.cn), CCTV (https://www.cctv.com), and other official websites in the fields of medicine, influenza, and pneumonia. In addition, we resample the data during preprocessing to control for the uncontrollable quality of user responses. Specifically, we removed the user responses that contained fewer words (less than three words), included more emojis (over 80%), and possessed a single hyperlink and no other information. Some posts that contain too much useless information also require removal.

5.2. Baseline Approaches. This study specifies six approaches to the CNFRD model as comparison baselines from two perspectives. Experimental results demonstrate that a few-shot learning framework is effective for detecting rumors. The benefits of introducing the Capsnet as a class generator are also verified within the learning framework. The baseline deep learning architecture employed by previous rumor detection models is used for the rumor detection task. Experimental comparison groups divide the baseline model dataset according to a ratio of 8 : 2, where 80% for training and 20% for final model validation.

(i) SVM-TS [73] proposes a linear SVM classifier that uses time-series structures to model the variation of social context features

(ii) GRU [27] means multilayer generalized recurrent neural network with gates, which trains a rumor classifier with microblogs modeled as variable time series.

(iii) CNN [74] is the abbreviation of a convolutional neural network that is used to categorize and represent rumor representation by framing related posts into fixed-length convolutional sequences.

To assess the performance of the few-shot learning framework used in this paper, we select three representative few-shot learning methods:

(i) Matching network [75] is a few-shot learning model using a metric-based attention approach.

(ii) Prototype network [21] is a depth metric-based approach based on sample averaging as a class prototype.

(iii) Relation network [22] is a few-shot learning model in which the distance measure is a neural network and the class vector is a sample vector from the support set.

Additionally, we develop several variants of CNFRD to demonstrate the effectiveness of the model's components, as detailed in Section 5.6.

5.3. Experimental Setup. This study implements the CNFRD model experiment using Python 3.6 and Pytorch 1.9.0 deep learning frameworks. The bert-base-uncased version released by Hugging Face [76] is applied as the BERT encoder. We develop a 2-way 15-shot model following the approach of Yu et al. [77]. The specific experimental parameters are detailed in Table 4.

5.4. Evaluation Metrics. The proposed CNFRD model will be evaluated using four standard evaluation metrics, i.e., Accuracy, Precision, Recall, and Macro F1. Macro F1 is utilized to balance the distribution of rumor posts. The metric allows us to analyze the classifier from a wider range and solve the defects of unbalanced labels. According to the confusion matrix shown in Table 5. Precision, Recall, Macro F1, and Accuracy are defined sequentially. The confusion matrix [78] is reproduced from Alejandro Morales-Hernández et al.

Table 3: The statistics of the real-world datasets.

$Dataset_{CN}$	*HD*	*ED*	*All*
# of rumors	5597	2344	7941
# of non-rumors	7968	3760	11728
Min.Length # of posts	40	31	31
Max.Length # of posts	219	117	219
Ave.Length # of posts	172	146	159
$Dataset_{EN}$	*HD*	*ED*	*All*
RumorNum	3851	1981	5832
NonRumorNum	5712	3278	8990
Min.Length	36	17	17
Max.Length	166	89	166
Ave.Length	157	122	139

Table 4: The specific experimental parameters setting.

Parameter	Parameter value	Parameter explanation
Lstm_dim	128	Hidden layer dimension of LSTM
Iterations	3	Iterations are used by the routing algorithm
Seed	42	Random seed
Margin_loss_scaling	10^{-7}	Scale margin loss
Dropout	0.1	Drop rate
Adam_epsilon	$1e-8$	Epsilon for adam optimizer
Train_batch_size	1000	Total batch size for training
Eval_batch_size	84	Total batch size for eval
Test_batch_size	84	Total batch size for test
Num_train_epochs	50	Total number of training epochs to perform
Learning_rate	$3e-6$	The initial learning rate for adam
Max_seq_length	164	The maximum length of the rumor text sequence

$$\begin{aligned}
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\
\text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\
F1 &= 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}},\\
\text{Macro}\, F1 &= \sum_{c=1}^{n} F\frac{1}{n},\\
\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.
\end{aligned} \tag{25}$$

5.5. Results and Analysis. This section demonstrates the effectiveness of the proposed model CNFRD by evaluating it using two datasets $Dataset_{CN}$ and $Dataset_{EN}$ and performing extensive comparisons with other deep learning and few-shot models. Their data details are shown in Table 3. The experimental results of the quantitative evaluation are presented in Tables 6 and 7. The scores in bold text style indicate the best results in each evaluation metric. The crucially relevant findings from the results are summarized as follows.

Overall, the proposed method, CNFRD, outperforms other baseline algorithms on both real-world benchmark datasets in terms of Accuracy, Precision, Recall, and Macro F1. More comprehensive performance improvements show that our model integrating the few-shot learning method and capsule network provides effective information that supports improving the performance of rumor detection.

Table 6 and Figure 6 enumerate the Accuracy, Precision, Recall, and Macro F1 scores for the proposed model CNFRD and three deep learning models (SVM-TS, CNN, and GRU) in $Dataset_{CN}$ and $Dataset_{EN}$, respectively. The current experiment found that the CNFRD model with only 15 support set samples per episode achieves 88.92% accuracy in the Chinese dataset and 87.07% accuracy in the English dataset. The scores have a significant 7.1% to 23.7% improvement over the three existing classical deep learning models with fewer epidemic rumor samples.

The improvement can be attributed to the fact that the CNFRD model generalizes the class-wise vectors well by incorporating multiple historical rumor sample vectors through the prototype network framework. The vector integration removes the noise between different expressions in the same category and improves the efficiency of the subsequent metric module. These results corroborate the effectiveness of the metric learning framework based on few-shot learning for rumor detection. Another important observation is that SVM-TS performs the worst among all models, indicating that the hand-crafted features are weak and insufficient to identify rumors. CNN performs better than SVM-TS since it is supervised and captures the local features between different words. GRU performs significantly better than SVM-TS and CNN on both datasets since

Table 5: Confusion matrix.

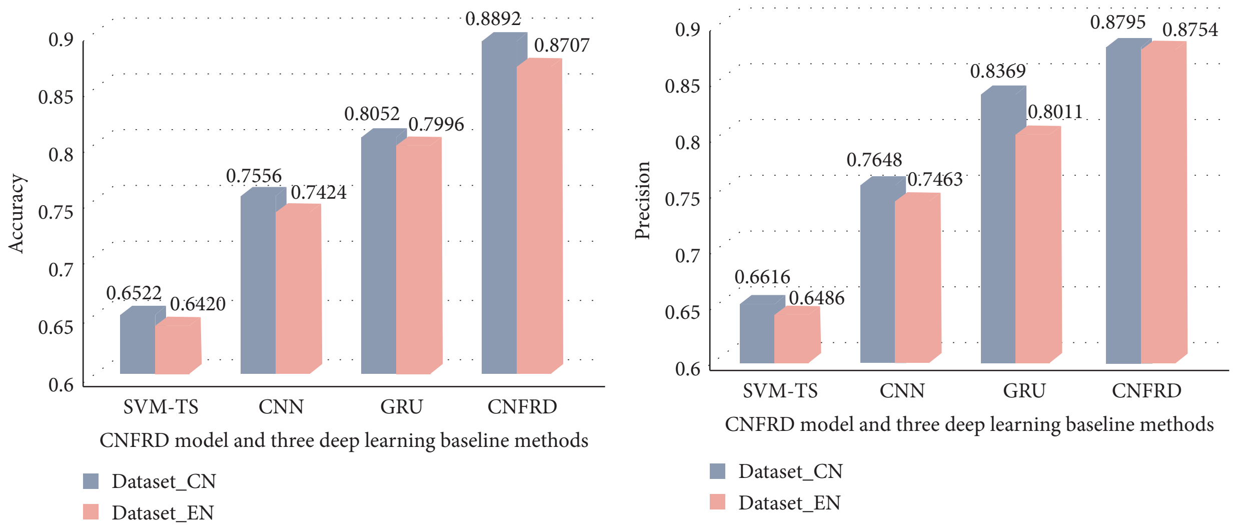
		p	n	
Actual value	p'	True positive (TP)	False negative (FN)	P'
	n'	False positive (FP)	True negative (TN)	N'
		P	N	

Table 6: The performance of CNFRD with compared deep learning models on two datasets.

Methods		$Dataset_{CN}$		
	Accuracy	Precision	Recall	Macro F1
SVM-TS [73]	0.6522	0.6616	0.6654	0.6488
CNN [74]	0.7556	0.7648	0.7691	0.7706
GRU [27]	0.8052	0.8369	0.8144	0.7960
CNFRD	0.8892	0.8795	0.8836	0.8862
Methods		$Dataset_{EN}$		
	Accuracy	Precision	Recall	Macro F1
SVM-TS [73]	0.6420	0.6486	0.6419	0.6481
CNN [74]	0.7424	0.7463	0.7357	0.7495
GRU [27]	0.7996	0.8011	0.8078	0.7952
CNFRD	0.8707	0.8754	0.8620	0.8781

Table 7: The performance of CNFRD with compared few-shot models on two datasets.

Methods		$Dataset_{CN}$		
	Accuracy	Precision	Recall	Macro F1
Matching network [75]	0.7956	0.7894	0.7921	0.7996
Prototype network [21]	0.7823	0.7866	0.7873	0.7895
Relation network [22]	0.8169	0.8143	0.8105	0.8096
CNFRD	0.8892	0.8795	0.8836	0.8862
Methods		$Dataset_{EN}$		
	Accuracy	Precision	Recall	Macro F1
Matching network [75]	0.7905	0.7981	0.7954	0.7968
Prototype network [21]	0.7801	0.7796	0.7756	0.7822
Relation network [22]	0.8002	0.8037	0.8049	0.7946
CNFRD	0.8707	0.8754	0.8620	0.8781



(a)

(b)

Figure 6: Continued.

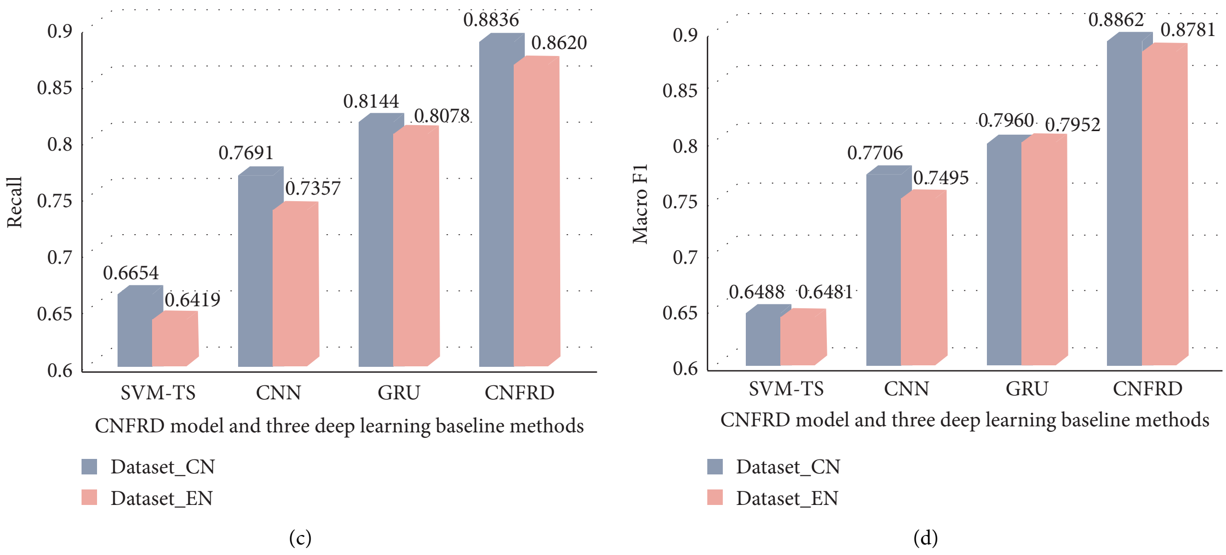


Figure 6: Comparison results with three deep learning baseline methods on two datasets. (a) Accuracy. (b) precision. (c) Recall. (d) Macro F1.

RNN can implicitly handle variable-length sequences of posts, whereas CNN requires more data to make decisions.

Table 7 and Figure 7 describe the comparison results among our established CNFRD model and three few-shot models: the matching network, the prototype network, and the relation network. The three baselines belong to the distance metric learning model in FSL. Compared with the existing classical FSL models, the CNFRD improves significantly by 7.0%–10.6% with the same 15 samples from the support set. The reason is that the learning and updating of the baseline models occur in the process of representing features and measuring distances at the sample level, while the CNFRD model constructs a class-level induction module based on the capsule network, focusing on the class-level representation, and uses capsule and dynamic routing algorithms to ensure that the CNFRD is robust to changes in the support set samples. This improvement validates the effectiveness of introducing the capsule network as a class generator in the few-shot rumor detection model.

5.6. Ablation Study. Deep learning experiments commonly involve ablation experiments as a form of module performance analysis. Module combination or removal is used to verify the influence of a particular module on the whole CNFRD model. This section implements the ablation experiments on both the Chinese and the English datasets. The methods in our experiments can be divided into the following three groups.

(i) CNFRD W/O BERT: A variant of the CNFRD, which removes the component of BERT that generates character-level dynamic feature vectors for rumor texts in the embedding module. The variant only applies BiLSTM to model variable-length sequence texts.

(ii) CNFRD W/O Capsnet: A variant of the CNFRD without the capsule network layer that generates historical rumor class vectors. The average of all sample vectors of the same category is used following the idea of the prototype network [21].

(iii) CNFRD W/EuD: A variation of the CNFRD that uses Euclidean distance to calculate similarity in the metric module. Euclidean distance is the most commonly used formula for calculating distance and measures each dimension's absolute value.

The results of Accuracy, Precision, Recall, and Macro F1 for the compared CNFRD variants are shown in Table 8 and Figure 8.

First, we evaluate the performance of CNFRD and CNFRD W/O BERT on two datasets. The variant CNFRD W/O BERT achieves Accuracy scores of 85.5% and 84.9% on $Dataset_{CN}$ and $Dataset_{EN}$, respectively, which suffers lower accuracy compared to the CNFRD model. Experimental results demonstrate that the BERT module is conducive to mining the semantic features of rumor texts and improving the accuracy of rumor detection. Using BERT to generate character-level dynamic feature vectors of rumor texts can effectively solve the problem of the missing lexicon in emerging epidemic rumors.

Second, we assess the performance of CNFRD and CNFRD W/O Capsnet. The complete model CNFRD beats the variant CNFRD W/O Capsnet by 4.8% on $Dataset_{CN}$ and 4.0% on $Dataset_{EN}$, demonstrating the necessity and superiority of the capsule network layer. Using the average of all sample vectors of the same class as the rumor class-wise vectors results in a significant reduction in model performance on both datasets. The capsule network layer is designed to model the relationship between local and global information, which allows it to automatically generalize knowledge learned from historical data to various new epidemic scenarios.

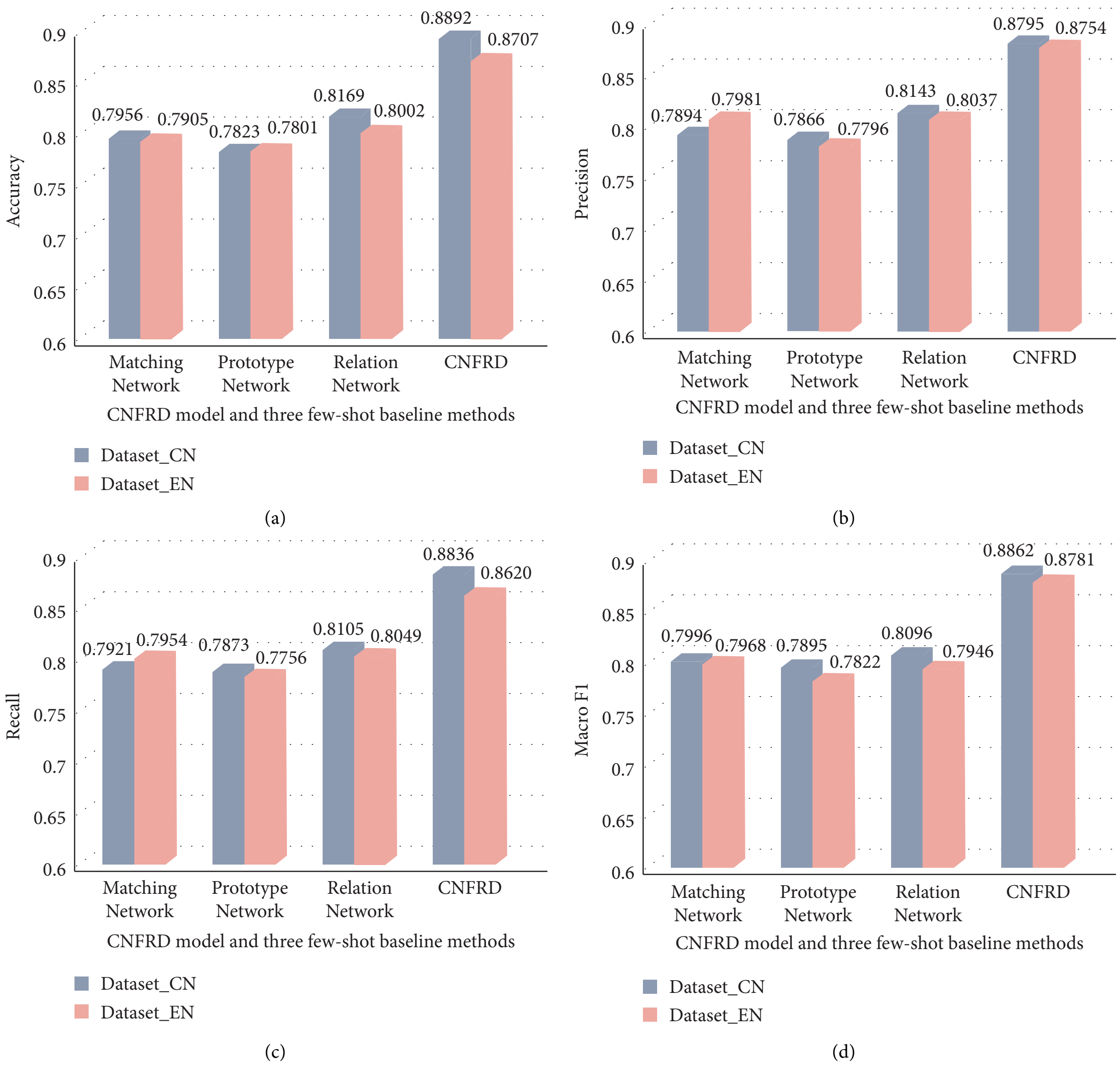


Figure 7: Comparison results with three few-shot baseline methods on two datasets. (a) Accuracy. (b) Precision. (c) Recall. (d) Macro-F1.

Table 8: The performance of CNFRD and its variants on two datasets.

Methods	$Dataset_{CN}$			
	Accuracy	Precision	Recall	Macro F1
CNFRD W/O BERT	0.8550	0.8596	0.8501	0.8527
CNFRD W/O Capsnet	0.8388	0.8367	0.8264	0.8325
CNFRD W/EuD	0.8628	0.8659	0.8616	0.8642
CNFRD	0.8892	0.8795	0.8836	0.8862
Methods	$Dataset_{EN}$			
	Accuracy	Precision	Recall	Macro F1
CNFRD W/O BERT	0.8499	0.8482	0.8459	0.8506
CNFRD W/O Capsnet	0.8301	0.8298	0.8322	0.8287
CNFRD W/EuD	0.8652	0.8549	0.8589	0.8604
CNFRD	0.8707	0.8754	0.8620	0.8781

Third, we compare the performance of CNFRD and CNFRD W/EuD. CNFRD performs better than CNFRD W/EuD, e.g., 2.4% and 0.4% higher than CNFRD W/EuD on $Dataset_{CN}$ and $Dataset_{EN}$, respectively. The modified cosine similarity is used in the CNFRD model, which yields better results than the Euclidean distance for the variant model.

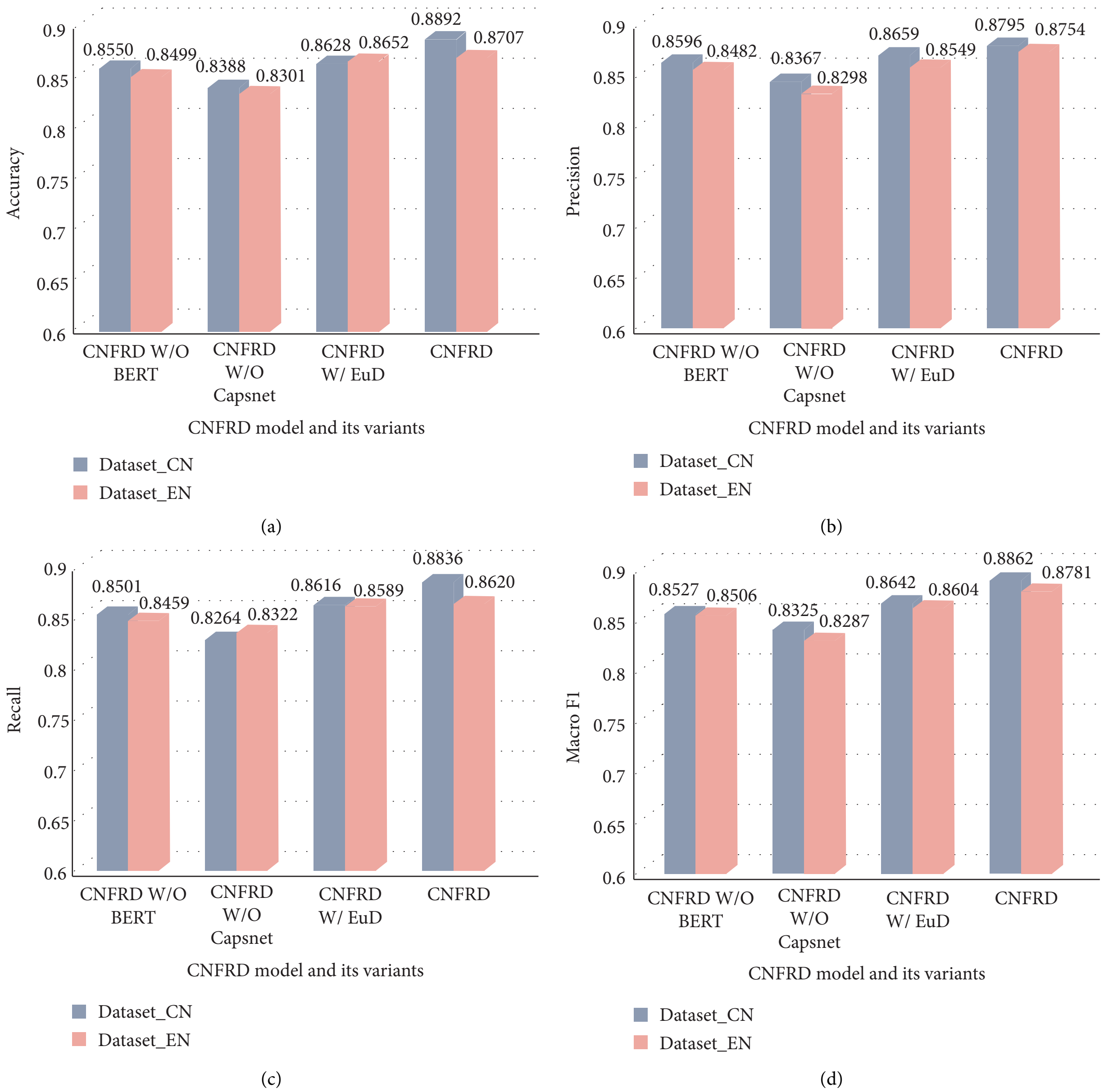


Figure 8: The comparison results of CNFRD and its variants on two datasets. (a) Accuracy. (b) Precision. (c) Recall. (d) Macro-F1.

Cosine similarity measures the relative difference between dimensions, while Euclidean measures the absolute value of the numerical difference. Cosine similarity is often used because it tends to give better results.

5.7. Complexity and Overhead Analysis. Based on the analysis of the dynamic routing algorithm in the capsule network in Section 4.3, we can conclude that the algorithm automatically updates the coupling coefficient through several iterations, according to which the part-whole relationship can be obtained. In this process, we assume that the number of iterations is r and the number of capsule network layers is l, indicating the time complexity of this algorithm is $O(rl)$. The training time required for each round is 790 s.

We compared the CNN method in terms of the number of trainable parameters. The CNN method involves 35 M parameters since it relies on a large amount of data and layers containing numerous feature maps to learn and update. In this case, it is not very efficient. Consequently, large datasets require many redundant feature detectors. Conversely, the capsule network encloses 8 M parameters, only a quarter of the CNN baseline. The reason for this is that the Capsnet is more generalizable and theoretically is able to use fewer parameters and obtain better results. Although capsule network parameters have been significantly reduced compared with the baseline standard CNN, the number of parameters of the capsule layer is still relatively large. This fact will lead to the problem of a slow training process. In the future, more novel and efficient routing algorithms should be designed for optimization.

6. Conclusion and Future Work

COVID-19 exhibited many rumors combined with numerous facts, disrupting social order. It is crucial to mine and analyses these texts in order to prevent and control epidemics. Traditional rumor detection approaches are unsuitable for current severe epidemic scenarios owing to their defects of being time-consuming and inefficient. Therefore, we propose a few-shot learning rumor detection model based on capsule networks (CNFRDs), which utilizes a metric learning framework and the capsule network to detect rumors of posts during unexpected epidemic events.

The experimental analysis demonstrates that: (1) the model CNFRD proposed in this paper can effectively improve the rumor detection effect in the case of less epidemic rumor data. The experimental scores are much better than other models in several indexes of Accuracy, Precision, Recall, and Macro F1, achieving an accuracy rate of 88.92% in the Chinese dataset and 87.07% in the English dataset. The CNFRD rumor detection model is more accurate than the three existing deep learning classical models by 7.1% to 23.7%, proving its effectiveness; (2) leveraging capsule network neural layer can significantly improve the detection accuracy of rumors; (3) modified cosine similarity can produce better results than Euclidean distance in rumor detection of few-shot learning.

During the COVID-19 epidemic, numerous rumors have been incorporated with countless pieces of information. Public anxiety is continuously increasing as social networks spread, adversely impacting the management of epidemics, the implementation of national policies, and society's stability. The insights into model construction gained from this study may be of assistance to rumor detection on OSN with a small amount of epidemic data. The focused technologies can be applied to build rumor detection on Facebook in a chrome environment or an automatic rumor analytical system on the Sina Weibo platform, which we are committed to doing. From the model's perspective, future work is dedicated to introducing different modal information for rumor detection, such as adding image information based on text content to make the received feature expression more comprehensive. These research programs will improve the accuracy of the rumor detection algorithm.

Data Availability

The data used to support the findings of this study are publicly available at https://github.com/lexieechen/CNFRD.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation (Grant nos. 61902324, 11426179, and 61872298), the Science and Technology Program of Sichuan Province (Grant nos. 2023YFD0424, 2021YFQ0008), the Foundation of Cyberspace Security Key Laboratory of Sichuan Higher Education Institutions (Grant no. sjzz2016-73) and Innovation Fund of Postgraduate, Xihua University (Grant no. YCJJ2021030).

References

[1] K. Alieyan, A. Almomani, M. Anbar, M. Alauthman, R. Abdullah, and B. B. Gupta, "Dns rule-based schema to botnet detection," *Enterprise Information Systems*, vol. 15, no. 4, pp. 545–564, 2021.

[2] Y. Du, Y. Wang, J. Hu, X. Li, and X. Chen, "An emotion role mining approach based on multiview ensemble learning in social networks," *Information Fusion*, vol. 88, pp. 100–114, 2022.

[3] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang, "*K*-Clique Community Detection in Social Networks Based on Formal Concept Analysis," *IEEE Systems Journal*, vol. 11, no. 1, pp. 250–259, 2017.

[4] S. Srinivasan and L. D. D. Babu, "A social immunity based approach to suppress rumors in online social networks," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 5, pp. 1281–1296, 2021.

[5] D. E. Salhi, A. Tari, and M. T. Kechadi, "Email classification for forensic analysis by information gain technique," *International Journal of Software Science and Computational Intelligence*, vol. 13, no. 4, pp. 40–53, 2021.

[6] J. Gao, F. Hao, Z. Pei, and G. Min, "Learning concept interestingness for identifying key structures from social networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3220–3232, 2021.

[7] F. Hao, S. Li, G. Min, H.-C. Kim, S. S. Yau, and L. T. Yang, "An efficient approach to generating location-sensitive recommendations in ad-hoc social network environments," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 520–533, 2015.

[8] A. K. Jain, S. R. Sahoo, and J. Kaubiyal, "Online social networks security and privacy: comprehensive review and analysis," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2157–2177, 2021.

[9] S. R. Sahoo and B. B. Gupta, "Fake profile detection in multimedia big data on online social networks," *International Journal of Information and Computer Security*, vol. 12, no. 2/3, pp. 303–331, 2020.

[10] S. Li, D. Qin, X. Wu, J. Li, B. Li, and W. Han, "False alert detection based on deep learning and machine learning," *International Journal on Semantic Web and Information Systems*, vol. 18, pp. 1–21, 2022.

[11] B. Chen and X. Chen, "MAUIL: Multilevel Attribute Embedding for Semisupervised User Identity Linkage," *Information Sciences*, pp. 527–545, 2022.

[12] F. Hao, G. Min, M. Lin, C. Luo, and L. T. Yang, "Mobifuzzytrust: an efficient fuzzy trust inference mechanism in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2944–2955, 2014.

[13] W. Jing and H. Kang, "An effective isdpr rumor propagation model on complex networks," *International Journal of Intelligent Systems*, vol. 34, 2022.

[14] L. Hu, T. Yang, L. Zhang et al., "Compare to the knowledge: graph neural fake news detection with external knowledge," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing, pp. 754–763, ACL/IJCNLP, Bangkok, Thailand, August 2021.

[15] R. A. Fiorini, "Computational intelligence from autonomous system to super-smart society and beyond," *International Journal of Software Science and Computational Intelligence*, vol. 12, no. 3, pp. 1–13, 2020.

[16] J. Brennen, F. Simon, P. Howard, and R. Nielsen, *Types, Sources, and Claims of Covid-19 Misinformation*, Reuters Institute, Norham Gardens, Oxford OX2 6PS, United Kingdom, 2006.

[17] C. Song, K. Shu, and B. Wu, "Temporally Evolving Graph Neural Network for Fake News Detection," *Information Processing & Management*, vol. 22, Article ID 102712, 2021.

[18] Y. Zhao, J. Da, and J. Yan, "Detecting Health Misinformation in Online Health Communities: Incorporating Behavioral Features into Machine Learning Based Approaches," *Information Processing & Management*, vol. 58, Article ID 102390, 2021.

[19] T. Takahashi and N. Igata, "Rumor detection on twitter," in *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems*, pp. 452–457, SCIS), India, December 2012.

[20] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *Proceedings of the IEEE 13th International Conference on Data Mining*, pp. 1103–1108, Dallas, TX, USA, December 2013.

[21] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 4077–4087, Long Beach, California, USA, December 2017.

[22] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: relation network for few-shot learning," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, Salt Lake City, UT, USA, June 2018.

[23] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 3856–3866, Long Beach, California, USA, December 2017.

[24] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics - MDS 12*, New YorkNYUnited States, Auguest 2012.

[25] C. Song, N. Ning, Y. Zhang, and B. Wu, "A Multimodal Fake News Detection Model Based on Crossmodal Attention Residual and Multichannel Convolutional Neural Networks," *Information Processing & Management*, vol. 12, Article ID 102437, 2021.

[26] X. Chen, H. Peng, J. Wang, and F. Hao, "Supervisory Control of Discrete Event Systems under Asynchronous Spiking Neuron P Systems," *Information Sciences*, pp. 253–273, 2022.

[27] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 3818–3824, IJCAI, New York City, July 2016.

[28] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3901–3907, IJCAI, Melbourne Australia, Augest 2017.

[29] T. Chen, X. Li, H. Yin, and J. Zhang, "Call attention to rumors: deep attention based recurrent neural networks for early rumor detection," in *Proceedings of the Trends and Applications in Knowledge Discovery and Data Mining - PAKDD 2018 Workshops*, pp. 40–52, Melbourne, June 2018.

[30] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang, "Rumor detection on social media with graph structured adversarial learning," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pp. 1417–1423, IJCAI, Yokohama, Japan, January 2020.

[31] T. Bian, X. Xiao, T. Xu et al., "Rumor detection on social media with bi-directional graph convolutional networks," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 549–556, AAAI, New York, USA, Feburary 2020.

[32] D. Rao, X. Miao, Z. Jiang, and R. Li, "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, EMNLP, Honolulu Hawaii, October 2021.

[33] S. Srinivasan, "A parallel neural network approach for faster rumor identification in online social networks," *International Journal on Semantic Web and Information Systems*, vol. 15, no. 4, pp. 69–89, 2019.

[34] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing*, vol. 100, Article ID 106983, 2021.

[35] J. V. Tembhurne, M. M. Almin, and T. Diwan, "Mc-dnn: fake news detection using multi-channel deep neural networks," *International Journal on Semantic Web and Information Systems*, vol. 18, pp. 1–20, 2022.

[36] Y. Dun, K. Tu, C. Chen, C. Hou, and X. Yuan, "Kan: knowledge-aware attention network for fake news detection," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 81–89, AAAI, Delhi, India, Feburary 2021.

[37] Z. Guo, K. Yu, A. Jolfaei, A. K. Bashir, A. O. Almagrabi, and N. Kumar, "Fuzzy detection system for rumors through explainable adaptive learning," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3650–3664, 2021.

[38] L. Wei, D. Hu, W. Zhou, X. Wang, and S. Hu, "Modeling the uncertainty of information propagation for rumor detection: a neuro-fuzzy approach," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[39] H. Zhou, T. Ma, H. Rong, Y. Qian, Y. Tian, and N. Al-Nabhan, "Mdmn: multi-task and domain adaptation based multimodal network for early rumor detection," *Expert Systems with Applications*, vol. 195, Article ID 116517, 2022.

[40] S. Santhoshkumar and L. D. Dhinesh Babu, "Earlier detection of rumors in online social networks using certainty-factor-based convolutional neural networks," *Social Network Analysis and Mining*, vol. 10, no. 1, p. 20, 2020.

[41] E. C. Marín, C. B. González-Tejero, M. G. García, and F. J. S. García, "Catholic impact evolution through public twitter data during covid-19," *International Journal of Cloud Applications and Computing*, vol. 12, pp. 1–17, 2022.

[42] F. Li, R. Fergus, and P. Perona, "One-shot Learning of Object Categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 56, pp. 594–611, 2006.

[43] J. Ma, H. Xie, G. Han, S. Chang, A. Galstyan, and W. AbdAlmageed, "Partner-assisted learning for few-shot image classification," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*, pp. 10553–10562, ICCV), Venice, Italy, June 2021.

[44] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12450–12459, Vancouver, Canada, June 2021.

[45] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, "Temporal-relational crosstransformers for few-shot action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 475–484, CVPR, San Juan, PR, USA, June 2021.

[46] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 328–339, ACL, Melbourne, July 2018.

[47] H. Zhang, Y. Zhang, L. Zhan et al., "Effectiveness of pre-training for few-shot intent classification," in *Findings of the Association for Computational Linguistics*, pp. 1114–1120, EMNLP, 2021.

[48] J. Zhang, T. Bui, S. Yoon et al., "Few-shot intent detection via contrastive pre-training and fine-tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1906–1912, EMNLP, Honolulu Hawaii, October 2021.

[49] C. Wang, J. Wang, M. Qiu, J. Huang, and M. Gao, "Trans-prompt: towards an automatic transferable prompting framework for few-shot text classification," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2792–2802, EMNLP, Punta Cana, November 2021.

[50] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-Vaegan-D2: A feature generating framework for any-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10275–10284, CVPR, Honolulu, HI, USA, July 2019.

[51] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the 2nd International Conference on Learning Representations*, ICLR, Banff, AB, Canada, April 2014.

[52] I. J. Goodfellow, J. PougetAbadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, vol. 27, pp. 2672–2680, Vancouver, Canada, December 2014.

[53] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proceedings of the ICML Deep Learning Workshop*, New York City, NY, USA, June 2015.

[54] R. Geng, B. Li, Y. Li, X. Zhu, P. Jian, and J. Sun, "Induction networks for few-shot text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3902–3911, EMNLP-IJCNLP, Hong Kong, China, November 2019.

[55] W. Li, J. Xu, J. Huo, L. Wang, Y. Gao, and J. Luo, "Distribution consistency based covariance metric networks for few-shot learning," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 8642–8649, AAAI, North Miami Beach, Florida, March 2019.

[56] Z. Zhao, Q. Liu, W. Cao, D. Lian, and Z. He, "Self-guided information for few-shot classification," *Pattern Recognition*, vol. 23, Article ID 108880, 2022.

[57] S. Xu and Y. Xiang, "Frog-gnn: multi-perspective aggregation based graph neural network for few-shot text classification," *Expert Systems with Applications*, vol. 176, Article ID 114795, 2021.

[58] N. Pang, X. Zhao, W. Wang, W. Xiao, and D. Guo, "Few-shot text classification by leveraging bi-directional attention and cross-class knowledge," *Science China Information Sciences*, vol. 64, no. 3, Article ID 130103, 2021.

[59] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 1165–1174, WWW, Lyon, France, April 2018.

[60] J. Kim, S. Jang, E. L. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.

[61] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3110–3119, Brussels, Belgium, Auguest 2018.

[62] C. Xia, C. Zhang, X. Yan, Y. Chang, and P. S. Yu, "Zero-shot user intent detection via capsule neural networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3090–3099, Brussels, Belgium, October 2018.

[63] H. Ren and H. Lu, "Compositional coding capsule network with k-means routing for text classification," *Pattern Recognition Letters*, vol. 160, pp. 1–8, 2022.

[64] Z. Chen, S. Li, L. Ye, and H. Zhang, "Multi-label classification of legal text based on label embedding and capsule network," *Applied Intelligence*, 2022.

[65] A. Kenarang, M. Farahani, and M. Manthouri, "Bigru attention capsule neural network for Persian text classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 8, pp. 3923–3933, 2022.

[66] Y. Wang, S. Qian, J. Hu, Q. Fang, and C. Xu, "Fake news detection via knowledge-driven multimodal graph convolutional networks," in *Proceedings of the 2020 on International Conference on Multimedia Retrieval*, pp. 540–547, ICMR, Dublin, Ireland, June 2020.

[67] M. Y. Cheng, D. Kusoemo, and R. A. Gosno, "Text mining-based construction site accident classification using hybrid supervised machine learning," *Automation in Construction*, vol. 118, Article ID 103265, 2020.

[68] R. Guha, M. Ghosh, P. K. Singh, R. Sarkar, and M. Nasipuri, "A hybrid swarm and gravitation based feature selection algorithm for handwritten indic script classification problem," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 823–839, 2021.

[69] C. Song, C. Yang, H. Chen, C. Tu, Z. Liu, and M. Sun, "Ced: credible early detection of social media rumors," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3035–3047, 2021.

[70] C. Yang, X. Zhou, and R. Zafarani, "Checked: Chinese Covid-19 Fake News Dataset," *Social Network Analysis and Mining*, vol. 58, 2021.

[71] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social media: A Data Mining Perspective," *SIGKDD Explorations*, vol. 3, pp. 22–36, 2017.

[72] M. Cheng, S. Wang, X. Yan et al., "A covid-19 rumor dataset," *Frontiers in Psychology*, vol. 12, p. 644801, 2021.

[73] J. Ma, W. Gao, Z. Wei, Y. Lu, and K. Wong, "Detect rumors using time series of social context information on microblogging websites," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pp. 1751–1754, CIKM, Melbourne, October 2015.

[74] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *Proceedings of*

the 26th International Joint Conference on Artificial Intelligence, pp. 3901–3907, IJCAI, Melbourne, May 2017.

[75] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3630–3638, 2016.

[76] T. Wolf, L. Debut, V. Sanh, J. Chaumond, and C. Delangue, "Transformers: state-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, EMNLP, Abu Dhabi, November 2020.

[77] M. Yu, X. Guo, J. Yi et al., "Diverse few-shot text classification with multiple metrics," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1206–1215, NAACL-HLT, New Orleans, Louisiana, May 2018.

[78] A. M. Hernández, I. V. Nieuwenhuyse, and S. Rojas-Gonzalez, "A Survey on Multi-Objective Hyperparameter Optimization Algorithms for Machine Learning," *CoRR*, vol. 25, 2021.