Impact of Transformer-Based Models and User Clustering in Early Fake News Detection in Social Media

Sakshi Kalra[1], Yashvardhan Sharma[1], Mehul Agrawal[1], Sai Ratna Kashyap Mantri[1] and Gajendra Singh Chauhan[2]

[1]*Department of CSIS, BITS Pilani, Pilani, 333031, Rajasthan, India*

[2]*Department of HSS, BITS Pilani, Pilani, 333031, Rajasthan, India*

Keywords: Early Fake News Detection, Neural Networks, Transformers, Attention Mechanism, User Clustering, Fuzzy C-Means Clustering, K-Means Clustering.

Abstract: People are now consuming news on social media platforms rather than through traditional sources as a result of easy access to the internet. This has allowed for the recent rise in the online dissemination of false information. The spread of false information seriously damages people's reputations and the public's trust in them. The research community has recently given fake news identification a great deal of attention, and prior studies have mainly concentrated on finding hints in news content or diffusion graphs. The older models, on the other hand, didn't have the key features needed to spot fake news quickly. We focus on finding fake news by using features that are available when it is just starting to spread. The current work suggests a new framework made up of content-based features taken from news articles and social-context features taken from user characteristics and responses at the sentence level. In addition, we extend our approach to Transformer-based models and leverage user clustering to demonstrate a considerable performance gain over the original model.

1 INTRODUCTION

Dealing with fake news has been part of our daily life in recent years. The spread of misinformation can heavily hamper a person's personal fame and public trust. Social media sites like Twitter and Facebook make it easier for people from all over the world to share information in real time. It has become the main way that people connect and share information online because it is easy to use, doesn't cost much, and moves quickly. With the popularity of social media growing so quickly, the internet has become a place where fake reviews, fake political statements, fake news, etc. are all over the place. For example, articles stating "COVID-19 vaccination causes autism and infertility among recipients[1]" can essentially impact the public trust and may prompt a drop in immunization drives. As per the research, fake news spreads much faster and deeper than factual news (Vosoughi et al., 2018). This has drawn significant attention among the industry leaders and research community as well. Even though the main motto of social media is to provide better communication, many users have started to confuse news from such platforms with main stream media. Traces of fake news started in 1439 itself (Klyuev, 2018) but the ease and scale at which it was disseminated changed drastically over time. In the modern day of the internet, fake news gained its importance only during the 2016 US presidential elections (Kshetri and Voas, 2017). There is no concrete definition of fake news. Based on existing literature, fake news can be loosely defined as "false stories that appear to be news and spread on the internet or other media, usually created to influence political views or as a joke, and depicting deliberate intentions". This depicts why fake news spread so rapidly, because most of the online news publishers have poor credentials and deny to identify themselves, which creates room to spread misinformation.

Early research on spotting fake news was mostly about finding better ways to spot fake news. This included, but wasn't limited to, understanding context, how news spreads, writing styles, syntactic analysis, etc. Getting such useful features is often hard and takes a lot of time, but users are smart enough to find ways around this. Recent research has focused on solving the above-mentioned problems. To learn how to represent a diffusion graph, for example, a lot of attention is paid to matrix factorization, graph neural networks, recurrent neural networks, and convo-

[1]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9359307/

Kalra, S., Sharma, Y., Agrawal, M., Mantri, S. and Chauhan, G.
Impact of Transformer-Based Models and User Clustering in Early Fake News Detection in Social Media.
DOI: 10.5220/0011684000003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 889-896
ISBN: 978-989-758-626-2; ISSN: 2184-4313

lutional neural networks. These methods only work well when they have enough information about how fake news is spreading. They are not good enough to find fake news in its early stages. As a human, when we are given a piece of information, we first use our intuition to judge its factual correctness. At times, we might also look for a reliable source to verify the information. This scenario motivates the importance of publisher and user credibility in detecting fake news in much earlier phases.

2 RELATED WORK

The detection of fake news on social media has drawn a lot of attention in recent years. One of the main goals of the studies that have already been conducted is to create machine and deep learning-based classifiers that can automatically tell if a news article spreading on social media is fake based on a number of news features. Early research focused on finding linguistic clues in news articles that could be used to spot fake news. This section gives an overview of the research that has been done on automatic extraction of the features for spotting fake news and closely related topics like spotting rumors or misinformation.

2.1 Analysis Related to News Content Based Features

Many researchers use the simple method of just looking at the news content to spot fake news. They read the news article headlines, bodies of text, and in some cases, related images and videos (Jin et al., 2016). Some, such as (Gupta et al., 2014) counted the number of swear words and words that contained pronouns in order to create features to distinguish fake news from real news. (Castillo et al., 2011) adopted a list of content-based features, including emoticons, pronouns, sentiment of words, and punctuation marks, used to determine the veracity of news. Based on writing styles, (Afroz et al., 2012) found online fraud, deception, and hoaxes. They have used things like assertive verbs, factive verbs, and implicatives to figure out how likely web claims are to be true. These stylistic linguistic features can be easily manipulated and do not convey semantic meaning. These methods therefore have a lower likelihood of being successful in practical applications. Content-based detection methods (Sun et al., 2013) often have trouble finding fake news because it comes in many different forms, in many different ways, and on many different platforms. Additionally, news content features may be event-specific. As a result, features based on content that perform well on one dataset of fake news may not perform well on another.

2.2 Analysis Related to Social Context Based Features

Social interactions related to a news article are included in social context features. They might reveal information about whether a news article is accurate. Some research has already been done on the ways that social context is used to classify news. The most common types of social context features are based on the user, on the text, and on the structure. User profiles on social media, which show what kind of people use social media, can be used to get information about user-based features. (Castillo et al., 2011) used a list of fundamental user-based features supported by various social media platforms, such as the number of followers, friend count, and age of registration, to assess the accuracy of information posted by its source user. (Yang et al., 2012) added a few user features to Sina Weibo, a Chinese social media platform, in addition to the typical user characteristics, such as gender and registration area, to find rumors. Using only user-based features to decide if a news article is fake has a big drawback: people who make fake news often mix it with real news to make it more likely that people will believe it. So, even if the news article isn't true, just looking at how people use a resource doesn't give us a full picture. Information on the users who shared or retweeted a news article, however, may give us more insight into the authenticity of a news article. However, this type of feature is ignored by many existing studies. Text-based social context features can be accessed through the comments and discussions of social media users that show up under news articles. A number of temporal-based features extracted from the time series of user comments and time-stamped on user comments are proposed to detect false news. (Ma et al., 2015) used a time series of content and features based on social context, such as the percentage of microblogs with URLs and the percentage of verified users, to tell the difference between rumors and other types of content. But these "aggregated level" parts need a lot of statistical considerations in order to spot fake news as soon as it comes out. Many deep learning techniques, like RNN, are used by (Ma et al., 2016) to extract temporal-linguistic patterns from user comment sequences in order to identify rumors. At the beginning of the news propagation process, user responses may be very limited, which can have a significant negative impact on the performance of RNN models and lead to them becoming overfit. This is one of the main disadvantages of these methods. Social media users can connect with one another through directed or undirected links, such as friendship and following. When a news story is shared through these links, a propagation network can be formed. Existing studies have examined structural features extracted from propagation networks as a different method of identifying fake news. For rumor detection, (Yang et al., 2015) took advantage of the topology property of social networks. By comparing the diffusion patterns of rumors and non-rumors, (Liu et al., 2017) were able to identify rumors. The disadvantage of using structural features is that it typically requires a lot of time to observe a propagation network big enough to extract useful distinguishable features, so these approaches are not very effective in cases of early detection.

2.3 Research Objectives

Through this research work, we looked into the problems below and attempted to provide solutions using our suggested model and techniques.

1. How to minimize the generalization problem of content based models?
2. How to overcome the limited availability of structure based social context-features?
3. How to capture local and global variations in social context features?
4. How would an ensemble of content and social context features improve the performance?

3 DATASET USED

The dataset used for this study is the FakeNewsNet dataset described by (Shu et al., 2020). It combines information from Twitter about the users who tweeted the articles with a set of real and fake articles from PolitiFact and GossipCop that were manually labeled by humans.

4 PROPOSED METHODOLOGY

At the beginning of the spread, we have news content, user profiles of people who share news on social media, and their tweet responses, which we get within 5 minutes of the news being posted. We can use all of the available features in the early stages of propagation by combining the limited social context-based model with the content-based model.

4.1 Content Based Models

Two content-based models are used. One is based on glove embeddings and convolution neural networks to extract latent features, and the other is based on transformers, which are state-of-the-art in many NLP tasks.

4.1.1 Glove and CNN Based Feature Extraction

Firstly, we prepare a 3D input for a convolutional-based model. The idea isn't to get an embedding for the whole article, but to break it up into sentences and add the number of sentences as a third dimension to the input. The main advantages of breaking down an article into sentences are:

1. As each sentence in an article is represented by a separate feature vector in the input tensor, we can encode the positional information of the sentences.
2. Other extra features at sentence level are also combined using this methodology.

So, we transform the news headline and body into a 3-dimensional tensor, where the headline and sentences represent the first dimension, words represent the second dimension, and glove word vectors represent the third dimension. Then, we feed this input into a convolutional network to get a single feature vector for the whole article. Figure 1 shows the 3D input tensor. This 3D input tensor's size must be fixed, so two

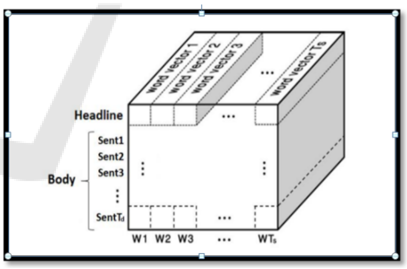

Figure 1: 3D Input Tensor.

thresholds are set: one to limit the number of sentences in an article ((T_d) and the other to limit the number of words in a sentence ((T_s). Any article having sentences longer than (T_s) is truncated, and lesser ones are padded. Similarly, for sentences extending the word limit, truncation is done; otherwise, they are padded to the word limit. Based on how the model was built and some statistical analysis of the dataset, we chose (T_d) = 100 (About 5% of sentences have more than 100 sentences). For calculating (T_d), we obtained the mean no. of sentences in an article (μ) and its standard deviation from the mean (σ) added them to obtain the value of (T_d). (T_d) = ($\mu + \sigma$). This

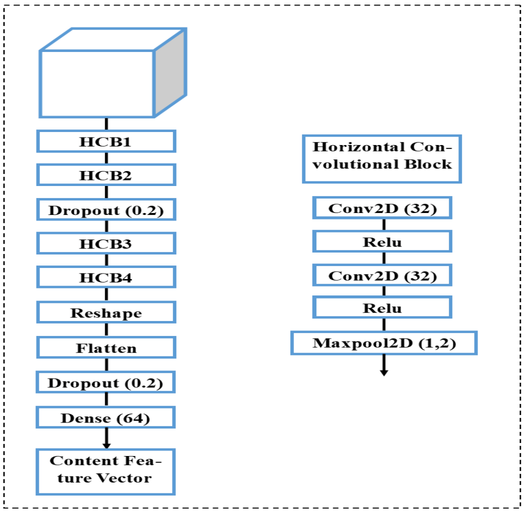

Figure 2: Glove and CNN Based Content Feature Vector Extraction.

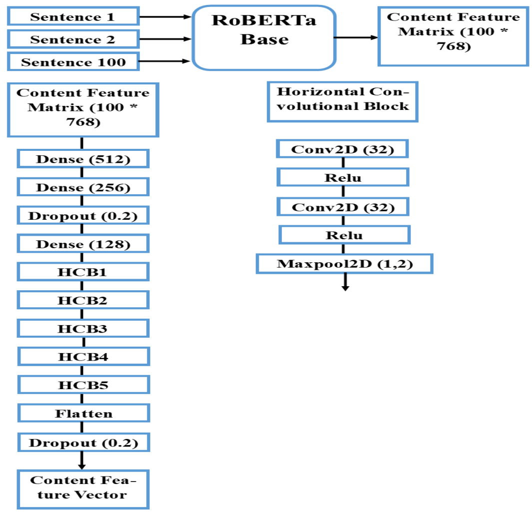

Figure 3: Transformer Based Content Feature Vector Extraction.

prevented the construction of very large and sparse tensors as it ignored the outlier sizes of articles in the dataset. By doing similar statistical analysis, we obtained $(T_w) = 46$. The architecture of the convolution-based network can be observed in Figure 2. In the input layer, the whole news article is represented as a 3D tensor. Then there are four horizontal convolutional blocks (HCB), using which we extract one feature vector for each sentence, thereby obtaining a matrix of size (100, 32) and then flattening and passing it through a dense layer (64) to obtain a single feature vector for the entire article. HCB is made up of two convolution layers that come one after the other, followed by a ReLU layer and then a max pooling layer. This content feature vector is concatenated with the social context-based feature vector to obtain the final feature vector, which can be used to classify news as fake or real.

4.1.2 Transformers Based Feature Extraction

Transformers-based models are the state-of-the-art in various Natural Language Processing tasks. They have a deeper understanding of the language and have been pre-trained in both directions on large datasets. We developed contextual embedding representations for each sentence in an article and hence obtained a 2D content feature matrix having a dimension of (number of sentences(100) * embedding vector dim(768)) as shown in Figure 3.

Later, this content feature matrix was passed through a stack of dense layers and a few horizontal convolution blocks to obtain a single feature vector for the entire article. Since it also extracts feature representation for each sentence in the article, like SLCNN , it also encodes the positional information of the sentences, hence including features at the sentence level. The RoBERTa Base was used for finding embeddings of the sentences since it yielded the best accuracy and an F1 score as shown in Table 1.

Table 1: Comparative Analysis based on Various Transformer-based Architectures.

Evaluation Parameter	bert-base	distilbert-base	XLM RoBERTa	RoBERTa-base
Accuracy	78.7%	77.3%	75.5%	81.2%
F1-Score	76.3%	76.2%	73.5%	80.5%

4.2 User and Social Context Based Model

User profiles of news spreaders on social media and their tweet responses when they posted their tweet are considered. For each article only K tweets are considered by the assumption that we will get those K tweets in the first 10-15 minutes of the tweet being posted. In that case we were able to get an average of 10 tweets within 10-15 minutes of posting, hence we used K=10. We have used this constraint of using only limited number of tweets to ensure early detection of fake news.

4.2.1 Extracting Tweet Feature Matrix Using Glove-Based Architecture

For each news article we consider 10 tweets and hence 10 tweet responses, each tweet response has an average of 15 words and each word will be represented by glove vector, hence dimension of our input vector would be : (K * max no of words in a tweet response * glove word vector size). We have used a

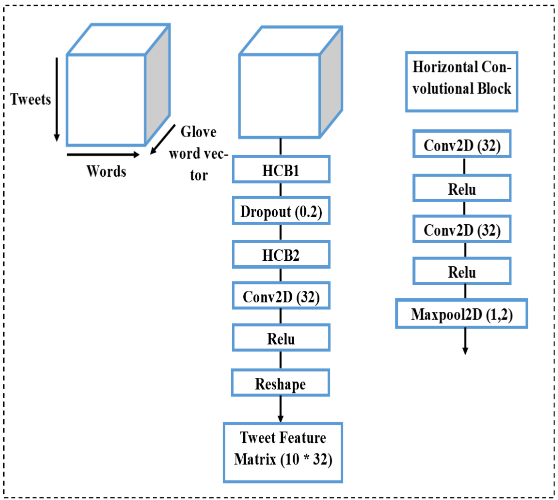

Figure 4: Tweet Feature Matrix Extraction using Glove-based Architecture.

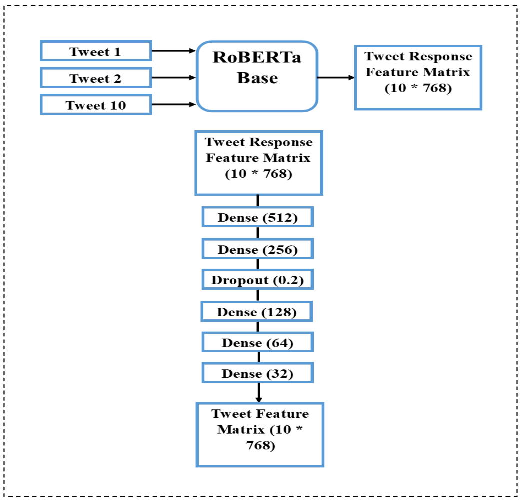

Figure 5: Tweet Feature Matrix using Transformer-based Architecture.

CNN based model to extract35 tweet feature matrix from this 3D input tensor. It uses few horizontal convolutional blocks along with reshape layer in the end to convert it into a 2D matrix of size (10 * 32) i.e., each tweet response being represented by a feature vector of size 32. Figure 4 shows the tweet feature matrix extraction using glove-based architecture.

4.2.2 Extracting Tweet Feature Matrix Using Transformer-Based Architecture

We developed contextual embedding representations for each tweet. Hence it generates a 2D input having dimension: (no of tweets * embedding vector dim). This matrix is further passed through a series of dense layers to reduce the dimension of embedding vector and obtain the tweet feature matrix of size (10 * 32). Figure 5 shows the tweet feature matrix extraction using transformer-based architecture.

4.3 User Feature Matrix Extraction for Measuring User Credibility

There are various features of the user that are readily available on the user profile and can be used to measure the credibility of the user posting the news. We have used the following features to describe each user: Follower's count, Friends count, Statuses count, Account verified or not and Location mentioned or not. A user feature vector can be obtained by using the normalized values of above measures. We then stack the set of user feature vectors of all tweets corresponding to a news article to obtain a user feature matrix for that article. So, if a user feature vector is of size 32, then a user feature matrix will be of size

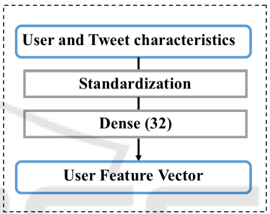

Figure 6: Extracting User Feature Vector.

(10, 32) because we have considered 10 tweets corresponding to a news article. Figure 6 shows the user feature extraction model and Figure 7 shows the complete architectural view.

4.4 Social Context Feature Matrix

We have obtained both tweet response (10, 32) and user feature matrix (10, 32). Now to obtain social context feature matrix we will just concatenate the above matrices and the size of the matrix obtained will be (10, 64).

4.4.1 Analysis of Social Context Feature Matrix

We need to capture both local and global variations in the social context data. Global variations can be captured using self-attention like mechanisms as they analyse the entire set of tweet responses and user characteristics for a particular news and select globally which are the prominent ones for classifying the news correctly. To capture local features, time series analysis of the social context data can be done using RNNs. They analyse the variation in the social context data as time progresses. We have used GRUs for this purpose. Later both global and local variations can be concatenated to obtain the final social context vector for the news article.

4.4.2 Self-Attention Mechanism (Capturing Global Variations)

Given a sequence of K tweets (tweet response + user characteristics) which is represented by social context feature matrix (K * 64), not all of them have the same ability to discriminate true and fake news. Some special text response generated by some special type of user may reflect the truthfulness of a concerned news article more significantly, thus should be somehow highlighted in the entire propagation path. Thus, our detection model should learn how much attention should be given to each tweet. The self-attention module will multiply each tweet vector with an attention score between 0 and 1. Hence all the relevant tweet vectors will be multiplied with an attention score close to 1 and all irrelevant tweets will be multiplied by scores close to 0. These attention scores will be calculated using weight matrix which will be trained along with the model. The weighted sum of all these tweet vectors will form global social context vector.

4.4.3 Time Series Analysis Using Gated Recurrent Units (Capturing Local Variations)

We have a sequence of K tweets $<(x_1,t_1),\ldots,(x_k,t_k)>$ where x_j is the vector representing concatenation of user characteristics and tweet response and t_j is the time of posting of tweet. Now we will feed this sequence of tweets to GRU for obtaining hidden state at each time step which will later be used to obtain local social context vector.

4.4.4 GRU Based Local Feature Extraction

For the $t^t h$ social context vector in the sequence i.e., x_t , a GRU takes in input as x_t , $h_(t-1)$ and produces h_t as output according to the following formulas:

$$z_t = \sigma(U_z x_t + W_z h_{t-1}) \tag{1}$$

$$r_t = \sigma(U_r x_t + W_r h_{t-1}) \tag{2}$$

$$\widetilde{h_t} = \tanh(U_h x_t + h_{t-1} \odot w_h r_t) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{4}$$

We then apply mean pooling to reduce the sequence of output vectors $<h_1,\ldots,h_k>$ produced by GRU units into a single vector which is the average of the above vectors produced at each time step. This vector obtained is the local social context vector.

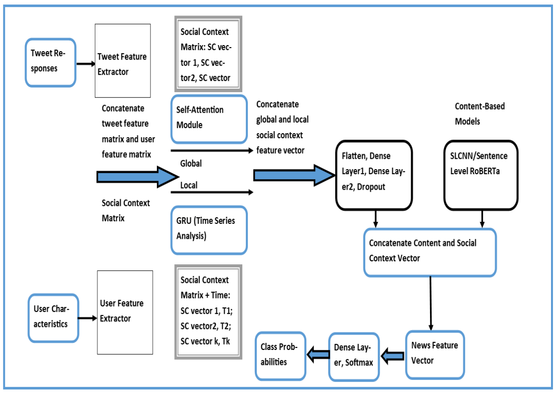

Figure 7: Complete Model Architecture View.

4.4.5 Concatenation of both Representations (Local and Global) of Social Context Vector

Both the representations can be concatenated into a single vector that represents the final social context vector. It can then be concatenated with content feature vector to obtain final feature vector which can be fed into a multi-layer feedforward neural network that predicts the class label for the news.

4.5 User Clustering

Social context features usually includes user-based features and text-based features. Apart from these two, it also captures structure-based features which involves the relationships among the users that are involved. In the practical scenario, with the limited availability of open dataset that includes the user relationships, even from popular microblogging websites like twitter, finding an alternative way to capture the structure-based features is necessary. In this regard, we would like to extend the base architecture with user clustering. Usually, there are two major categories of clustering:

- **Hard Clustering.** Where each data point has a fixed cluster label.
- **Soft Clustering.** A data point can co-exist in multiple clusters with certain probability.

Extended the framework to implement K-means which is hard clustering method and Fuzzy C-means clustering algorithms which falls under soft clustering. User behaviour is not always binary, few of the users tend to spread both fake news as well as real news (knowingly or unknowingly). Having the fuzziness gives us the flexibility to have a control over the cluster membership thresholds, which is more practical in the real world. To understand the clustering

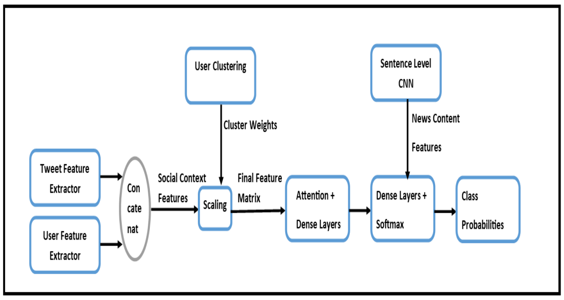

Figure 8: Model Architecture with User Clustering.

methods that are in discussion in brief, K-means clustering initialize the centers, assign memberships to the datapoints and recompute the centers and repeat until it converges. Whereas, Fuzzy C-Means Clustering initialise the memberships randomly first, updates the class centers using these memberships, computes Euclidean distance of samples from centers, finally re-update the memberships and repeat until it converges. Once we obtain the cluster memberships, cluster weights are given to each of the cluster and its members based on the relative size of the cluster they are in. Now this cluster weights are used to scale the feature vectors of corresponding users. As users with huge friendship/follower network can be an outlier and could possibly impact the model via its feature vectors. To reduce the impact of outliers, the cluster weights are user to scale the feature vectors of the respective users. Figure 8 shows the model architecture with User Clustering.

5 EXPERIMENTS AND RESULTS

The FakenewsNet repository was used for the experimental analysis. The labeled data has been fact-checked by PolitiFact and GossipCop. Table 2 shows the accuracy and F1 score of the PolitiFact data. Table 3 shows the accuracy and F1 score of the GossipCop data. Table 4 shows the Accuracy and F1 score related to various content based and content-social context ensemble based models.

Table 2: Accuracy and F1-score on PolitiFact Data.

PolitiFact	Content (Glove based)+ Social context(global)	Content (Glove based) + social context (local +global)	Content (transformers) + social context (global)	Content (transformers) + social context (local + global)
Validation Accuracy	85.5%	86.18%	83.22%	84.12%
F1-Score	84.33%	85.25%	83.56%	82.22%

Considering local and global social context-based features with the content-based features improved the performance by significant margin. And since we have used only a limited no. of tweets, our model will be able to detect news at an early stage of news propagation. The results of our model are comparable to the results of models that uses entire propagation network which takes months to build and hence violates the constraint of early detection of fake news.

Table 3: Accuracy and F1-score on GossipCop Data.

GossipCop	Content (glove) + social context (global)	Content (transformers) + social context (global)	Content (transformers) + social context (global)	Content (transformers) + social context (local + global)
Validation Accuracy	87.2%	88.4%	83.22%	84.12%
F1-Score	86.7%	87.3%	83.56%	82.22%

We can observe that sentence level models, sentence level CNN and sentence level Roberta, Sentence Level CNN with Fuzzy clustering performed better than their counterparts which considered whole article as a single sentence/entity as shown in Table 4. Also, adding social context-based features to the content-based features improved the performance by significant margin. And since we have used only a limited no. of tweets, our model will be able to detect news at an early stage of news propagation. The results of our model are comparable to the results of models that uses entire propagation network which takes months to build and hence violates the constraint of early detection of fake news.

6 CONCLUSION AND FUTURE WORK

Even using only about 20 % social context data, we were able to achieve accuracy comparable to models that uses entire social context data hence violate the constraint of early detection. Ensembling the content-based model with social context based is the way to deal with generalizability issue of content-based model. Using both CNN and RNN (Time Series Analysis) we can capture both local and global variations in the social context data. Cluster credibility and self-attention helps to identify which tweet and user should be taken into consideration to classify news as fake or real. For the future work retweets can be considered with the tweets which were obtained in the same time window. Comments on tweets and retweets can also be considered. After including retweets, we can build a social context graph for the propagation of news and instead of using text CNN or sentence level CNN we can use graph CNN for feature extraction and news classification. Introducing fuzzy nature to the clustering certainly improved the performance of

Table 4: Accuracy and F1 score related to various content based and content-social context ensemble based models.

	Base Model	Content-Based			Content-Social Context Ensemble			
Evaluation Metrics	Text-CNN	Sentence Level CNN	RoBERTa	Sentence - Level RoBERTa	Sentence-CNN+ Social Context Based Model	Sentence RoBERTa + Social Context Based Model	Clustering Methods	
							Sentence CNN+K-Means	Sentence CNN+ Fuzzy C-Means
Accuracy	73.3%	80.5%	75.4%	78.6%	86.4%	83.3%	80.2%	83.6%
F1-Score	0.67%	0.77%	0.73%	0.76%	0.85%	0.82%	0.69%	0.84%

the model. As Fuzzy C-means clustering only considers Euclidian distance to compute, using a metric like Mahalanobis distance which also captures the spatial metrics can improve the model.

REFERENCES

Afroz, S., Brennan, M., and Greenstadt, R. (2012). Detecting hoaxes, frauds, and deception in writing style online. In *2012 IEEE Symposium on Security and Privacy*, pages 461–475. IEEE.

Castillo, C., Mendoza, M., and Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

Gupta, A., Kumaraguru, P., Castillo, C., and Meier, P. (2014). Tweetcred: Real-time credibility assessment of content on twitter. In *International conference on social informatics*, pages 228–243. Springer.

Jin, Z., Cao, J., Zhang, Y., Zhou, J., and Tian, Q. (2016). Novel visual and statistical image features for microblogs news verification. *IEEE transactions on multimedia*, 19(3):598–608.

Klyuev, V. (2018). Fake news filtering: Semantic approaches. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 9–15. IEEE.

Kshetri, N. and Voas, J. (2017). The economics of "fake news". *IT Professional*, 19(6):8–12.

Liu, Y., Jin, X., Shen, H., and Cheng, X. (2017). Do rumors diffuse differently from non-rumors? a systematically empirical analysis in sina weibo for rumor identification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 407–420. Springer.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., and Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks.

Ma, J., Gao, W., Wei, Z., Lu, Y., and Wong, K.-F. (2015). Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1751–1754.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

Sun, S., Liu, H., He, J., and Du, X. (2013). Detecting event rumors on sina weibo automatically. In *Asia-Pacific web conference*, pages 120–131. Springer.

Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *science*, 359(6380):1146–1151.

Yang, F., Liu, Y., Yu, X., and Yang, M. (2012). Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*, pages 1–7.

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