A novel approach for non-query fake news detection using K-SOM and Graph Neural Networks

Hoang-Danh Nguyen[1,2], Thanh-Phong To[1,2], Duc Quang Nguyen[1,2], Thanh-Trung Huynh[3]
Tham Tran[1,2], Cong-Tuan Bui[1,2], Tho Quan[1,2,*]
[1]*Ho Chi Minh City University of Technology, VNU-HCM*
[2]*Vietnam National University Ho Chi Minh City*
[3]*Ecole Polytechnique Fédérale de Lausanne*
[*]Corresponding author: `qttho@hcmut.edu.vn`

2023 15th International Conference on Knowledge and Systems Engineering (KSE) | 979-8-3503-2974-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/KSE59128.2023.10299416

***Abstract*—The widespread use of social networks has led to an increase in the number of users and posts on these platforms. However, the proliferation of fake news, particularly in the healthcare sector due to the COVID-19 pandemic, has become a significant concern. This has become a significant concern as accepting such fake news can have severe consequences on the health and lives of those who are exposed to it, leading to confusion, social disorder, and reputational damage for individuals, organizations, and businesses. Therefore, the automatic detection of fake news on social networks is of utmost importance. In this study, we propose the FANSOG model, which utilizes K-SOM to cluster articles and a graph-based model to automatically detect non-query fake news. Our findings demonstrate that the FANSOG model outperforms other state-of-the-art models in the same research direction.**

***Index Terms*—Fake News Detection, Graph Neural Network, K-SOM clustering**

I. Introduction

Detecting fake news on social media presents novel and complex research challenges. Despite the issue of fake news is not unprecedented, government and organizations have utilized various forms of media to conduct propaganda or influence activities for centuries. However, the exponential growth in the number of social media users and the posts has intensified the spread of fake news and made it more challenging to manage. Furthermore, fake news has become increasingly varied in structure and content. These characteristics have made the early detection of false information increasingly difficult, particularly for automated detection methods that leverage machine learning techniques.

The methods for detecting fake news can be broadly categorized into two approaches:

- Non-query approach: This method detects fake news solely based on the content and style of the article.
- Query approach: This method detects fake news by querying external information, such as user profiles, user interactions, engagement metrics, and news-related metadata. Additionally, search engines like Google can be leveraged to find additional evidence to distinguish between fake and real news.

The non-query method for detecting fake news through content and style features can aid in early detection. However, it is faced with difficulties due to the numerous types of fake news in terms of form and content. Depending solely on conventional text classification techniques can pose significant challenges, mainly due to the following two distinctions:

- Fake news with distinct content or subject matters may fall into the same category (Fake/Real).
- Fake and real news may contain comparable content (with only a few variations in writing style or character differences).

Consequently, the application of traditional text classification models results in a noisy learning model that fails to concentrate on the general themes or forms of fake news text groups, making it challenging to differentiate between the two classes when sentences with the same topic but different writing styles belong to separate categories.

To address the issue of using traditional text classification methods for fake news detection, we propose the **FANSOG** (**FA**ke **N**ews **D**etection using **SO**M and **G**raph Neural Network) model to automatically detect fake news based on their content, especially in the healthcare field. To achieve this, we use the K-SOM clustering technique to group similar text, creating a graph relationship between clusters. This graph relationship allows us to apply the Graph Neural Network-based method to classify fake and real news. Our contribution are stated as follows.

- We leveraged text clustering techniques to group similar content texts, thus augmenting the model's learning capacity to classify fake news based on links to other news in the same cluster.
- We utilized a graph neural network to extract post features, as it possesses the ability to assimilate information from neighboring news items in a cluster and to acquire knowledge regarding news representation through graph computations.

The remainder of the paper is organised as follows. Section II briefly reviews the literature of fake news detection. Section III describes the details of our proposed framework, including the five components. Section IV reports the experiments we conducted to study the performance of our technique compared to the state-of-the-arts. Section V concludes the paper.

II. Related Work

A. Fake News Detection

As mentioned, there are two main approaches for fake news classification, namely non-query and query approaches [1], [2]. In this paper, we focus on query approaches. In this direction, many studies focus on analyzing the content of the post, the title, and even the image and video posted, such as [3]. [4] uses a list of content-based attributes such as punctuation marks, emoticons, positive or negative words. [5] leverages the occurrence of rude words or self-pronouns used in the post as signs of fake news. [6], [7] indicate that the writing style of the article plays an important role in expressing the credibility of the article. These approaches do not include semantic significance and are easily overcome. [8] studied news content at four levels: vocabulary (lexicon), syntax, semantics, and discourse, and used a machine learning model to discover common patterns of fake news. [9], [10] leverage techniques such as *n-gram* information retrieval [11] and word representation learning using *Word2Vec* [12].

In recent years, there has been more focus on research on natural language processing for pre-trained models. BERT is the most popular language model that has been pre-trained. These models have also been used in fake news detection research. BERT is used in some fake news detection models [13]–[15] to classify real or fake news. These studies leverage semantic, content, and writing style features to build fake news prediction models to make predictions as soon as a news article is posted. However, for completely new content with breaking news or fake writing styles, these models may be easily overlooked.

B. Kohonen Self Organizing Map

Self-Organizing Maps (SOM), also known as K-SOM [16], are a variant of competitive learning models that consist of one- or two-dimensional arrays of nodes or neurons adjusted according to different input data patterns. In two dimensions, K-SOM typically employs two commonly used grid topologies, namely rectangular and hexagonal. The primary objective of K-SOM is to transform arbitrary-sized input data patterns into a discrete map with fewer dimensions, while preserving the essential interrelationships among the original data elements. K-SOM is the popular choice for applications such as image classification, text clustering, and so on.

Suppose that the input data set consists of n samples represented as a vector set $\mathbf{X} = \{x_1, x_2, x_3, ..., x_n\}$, where each vector has p dimensions. The chosen K-SOM has a size of m, and the weight vector set of the nodes in the network is denoted as $\mathbf{W} = \{w_1, w_2, w_3, ..., w_m\}$. Corresponding to each vector x_i, the weight vector w_j of node j in the K-SOM also has p dimensions. The K-SOM algorithm consists of the following steps.

Step 1: Initialize weights for nodes in the K-SOM.

Step 2: Randomly select a point from the input data set, denoted as x_i.

Step 3: Find the winning node w_j for the point x_i.

The winning node is the node whose weight vector has the smallest Euclidean distance or the smallest dot product with the vector of the input data point.

Step 4: From the winning node w_j, update the weight vectors of all nodes in the K-SOM matrix or only a set of nodes within a predetermined radius from the node w_j as the center (1), which is known as the neighborhood function.

$$w_k^{t+1} = w_k^t + \alpha(t) \times \theta(r) \times (x_i - w_k^t) \tag{1}$$

In (1), t is the t-th iteration step, $\alpha(t)$ is the learning rate function, which decreases after each iteration step, $\theta(r)$ is the function used to decrease the amount of weight update based on the distance r from node w_k to the winning node w_j. The farther away from the winning node, the smaller the value of the function, and $\theta(r)$ have the value of 0 when reaching a certain distance.

Step 5: Repeat Step 2 until the weight vectors converge or until a specific number of iterations is reached.

III. Proposed Method

The general architecture of the FANSOG model is depicted in Fig. 1. We acquire information of labeled posts and perform feature extraction to extract relevant features. We then use K-SOM to cluster the posts and form a relation graph from the resulting clusters. Graph Neural Network is utilized to extract features from the graph, and finally, an Logistic Regression Classifier is trained on the graph features, serving as the ultimate fake news classifier.

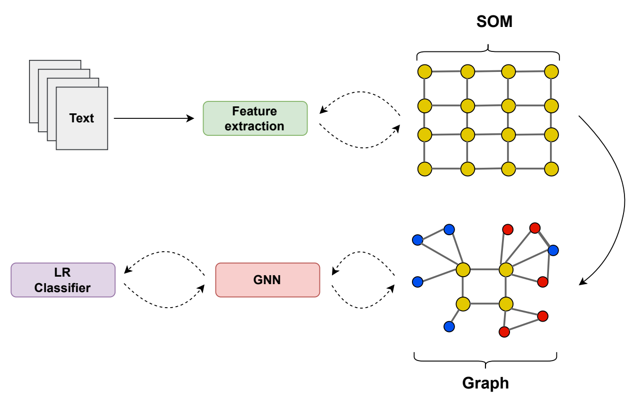

Fig. 1. The general architecture of FANSOG.

A. Feature Extraction

Initially, an article undergoes preprocessing and tokenization techniques. Once the content has been cleaned and tokenized, pre-trained BERT-based model are utilized. For this study, the DistilRoBERTa-base model is employed, which has an enhanced ability to extract sentence features. This is due to its training on sentence pairs and utilization of a dataset with over one billion pairs. Following the feature extraction process for each token within the sentence, we compute the average of all token feature vectors to obtain the vector v_i^d associated with the i^{th} news article.

B. K-SOM Construction

Once the dataset of news articles has been processed through the feature extractor, we obtain a set of n vectors $\mathbf{V} = \{v_1^d, v_2^d, v_3^d, ..., v_n^d\}$, with each vector representing a news article. The K-SOM, which has a size of k, is chosen, and the set of weight vectors for each node in the network is represented as $\mathbf{W} = \{w_1^d, w_2^d, w_3^d, ..., w_k^d\}$. Vector w_j^d, for the j^{th} node in the K-SOM is initialized having the equal number of dimensions. The distance between v_i and w_j is computed by using Euclidean distance.

The SOM algorithm is implemented in its original form, as discussed in section II-B. The function $\alpha(t)$ in (1) represents the model's learning rate, while the function $\theta(r,t)$ influenced by distance and time, which is computed using (2).

$$\begin{cases} \alpha(t) = a_0 \times \exp\left(\dfrac{-t}{\lambda_a}\right) & \text{(2a)} \\ \sigma(t) = r_0 \times \exp\left(\dfrac{-t}{\lambda_r}\right) & \text{(2b)} \\ \theta(r,t) = a_0 \times \exp\left(\dfrac{-r^2}{2 \times \sigma(t)^2}\right) & \text{(2c)} \end{cases}$$

In (2), a_0, r_0, λ_a, λ_r are hyperparameters, and r is the Euclidean distance from the winning node to the node whose weight needs to be updated on the K-SOM, calculated using (3).

$$r = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \tag{3}$$

In (3), (x_j, y_j) and (x_k, y_k) represent the respective coordinates of the winning node and the node requiring a weight update on the two-dimensional K-SOM.

C. Graph Construction

Once the K-SOM model has completed training to group similar texts together, a text classifier is required for the two categories of news: fake and real. To classify the article, we proposed a *heterogeneous graph* employing the similarity of text nodes and the K-SOM's node connections, as illustrated in Fig. 2.

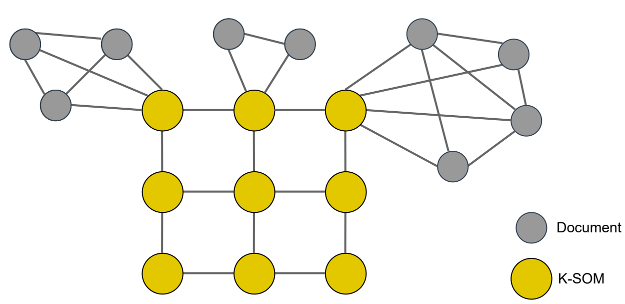

Fig. 2. The graph constructed from document nodes and K-SOM nodes.

Each feature vector of the i^{th} document, denoted as v_i^d, is represented by a corresponding *Document node*. Similarly, the K-SOM weight vector, as mentioned in III-B, is assigned a *K-SOM node* for graph construction process. To construct the graph, we map each document node to its corresponding K-SOM node. Document nodes belonging to the same point in the K-SOM are linked to the mapped K-SOM node and connected to each other to form a sub-graph in two ways:

- (i) Pairwise connect for forming a complete graph
- (ii) Compute the similarity between each pair of feature vectors for each document node to generate a similarity matrix with symmetric properties, consisting of values representing the similarity between each pair of nodes in the sub-graph. For the threshold parameter β, we select nodes that are connected if their similarity value falls within the highest group with a ratio greater than α.

As a result, sub-graphs are constructed corresponding to each K-SOM node. Based on the properties of the K-SOM, document nodes belonging to neighboring K-SOM nodes share similar features. Therefore, we suggest connecting K-SOM nodes according to their original network structure to leverage this property of K-SOM while aiming to construct a graph based on text similarity. The similarity between any two document nodes is computed using cosine similarity. The graph $\mathcal{G}$ construction process is summarized in pseudo-code in Algorithm 1.

Algorithm 1: Graph Construction Algorithm

```
Input  : f_KSOM(·), The list of feature of node post,
         The list of feature of node KSOM, Threshold
         β
Output: Graph
G ← Empty_Graph
for node_KSOM in all KSOM node do
    G.Add_Node(node_KSOM)
    for each neighbor_node in N(node_KSOM) do
        G.Create_Link(node_KSOM, neighbor_node)
    end
end
for post in all post do
    G.Add_Node(post)
    winning_node ← f_KSOM(post)
    Add the post in the sub-graph of node KSOM
      winning_node
end
for node_KSOM in all KSOM node do
    list_post_in_KSOM ←
      get_list_post(node_KSOM)
    pairwise_sim_matrix ←
      pdist(list_post_in_KSOM)
    top_alpha_sim ←
      top_k(pairwise_sim_matrix, β)
    for (node_x, node_y in top_alpha_sim do
        G.Create_Link(node_x, node_y)
    end
end
return G
```

D. Graph Encoder

After constructing the graph, we employ the GNNs to address the node-level classification problem on the graph. Therfore, we model the feature matrix $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$. The adjacency matrix $\mathbf{A}$ is readily derived from the graph constructed in section III-C. On the other hand, the feature matrix $\mathbf{X}$ is obtained from the vertex features introduced in section III-C, specifically in (4).

$$\mathbf{X} = \{x_i \in \mathbb{R}^h, i = \overline{1, n+k}\} \tag{4}$$

In (4), h, n, and k represent the dimensions of the feature vectors v_i^d, the number of documents, and the number of K-SOM nodes, respectively. To be more specific, the first k rows of the matrix $\mathbf{X}$ correspond to the k weight vectors of the K-SOM nodes, and the remaining n rows of the matrix $\mathbf{X}$ correspond to the n feature vectors of the n text samples.

1) Neighbor Sampling: For large-scale data problems, sampling is a necessary step in the training process. Due to the unique sampling process in graphs, we clarify the graph sampling process using the neighborhood sampling method introduced in [17]. As the graph derived from section III-C is a heterogeneous graph, we consider the various relationships between neighbors based on the characteristics of each vertex. Specifically, the graph contains three types of relationships:

- ***"d-d"*** relationship: The relationship between documents.
- ***"k-d"*** relationship: The symmetric relationship between individual K-SOM nodes and documents.
- ***"k-k"*** relationship: The relationship between two nodes in K-SOM.

Excessive layer stacking or excessive influence hops from a node in graph neural networks can result in over-smoothing. Therefore, we must carefully consider the selection of neighboring samples to mitigate this phenomenon. By explicitly specifying the nature of relationships, the sampling process can be customized to determine the number of neighboring samples required for each relationship at each hop.

To leverage the structural connectivity of the K-SOM model, a target node requires a 3-hop computational graph (corresponding to 3 GNN layers) to learn the feature vector of text samples located in neighboring K-SOM clusters.

In Fig. 3, it is evident that the sampling process excludes relationships related to K-SOM nodes (*"k-k"*, *"k-d"*) at the third hop to prevent excessive information aggregation, which can result in the loss of original information of the target node.

2) GNNs Layer: The feature vectors of each vertex are updated at each GNN layer. For the FANSOG model, the GraphSAGE model introduced in [17] was selected to aggregate information from neighboring vertices due to its suitability for recursive problems and robustness with respect to unseen data. The resulting GNN layers produce representative vectors for each vertex within the graph. Subsequently, these representative vectors are fed through a fully connected layer to classify whether the news article is fake or real.

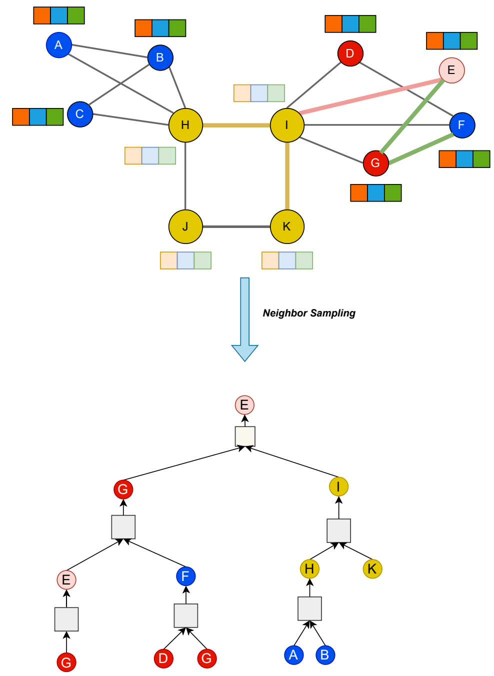

Fig. 3. Neighbor sampling for the target node E.

E. Inference

When a news article necessitates inference, the FANSOG model assigns the article to a K-SOM cluster. The linking process between the article and other articles within the cluster is predicated on the minimum similarity threshold in the . The process of connecting with K-SOM nodes remains the same as described in III-C.

IV. Experiments

A. Dataset

In this experiment, we employed five standard datasets to evaluate the FANSOG model's performance in fake news detection without using evidence. These datasets consist of LIAR [18], Fake or Real [19], KDD2020 [20], COVID-19 [21].

In practical applications, the volume and diversity of news data sources are continuously increasing. The ability of the FANSOG model to cluster related news articles together more closely by utilizing the clustering capability of the K-SOM map can enhance the efficiency of classifying new articles by focusing on news articles that are "close" in proximity, without considering unrelated news articles with entirely different content and topics. To closely simulate real-world scenarios and demonstrate the effectiveness of the FANSOG model with a large and diverse corpus of news articles, we created Combination datasets through the convergence of training datasets from various sources into a comprehensive training dataset and the integration of testing datasets from these sources into a unified testing dataset. We developed two Combination datasets containing different quantities and

TABLE I
EXPERIMENTAL RESULT ON SYNTHETIC AND STANDARD DATASET

	Combination 1		Combination 2		Liar		Fake or Real		KDD2020		COVID-19	
	Acc. ↑	F1 ↑	Acc. ↑	F1 ↑	Acc. ↑	F1 ↑	Acc. ↑	F1 ↑	Acc. ↑	F1 ↑	Acc. ↑	F1 ↑
BERT	0.748	0.742	0.747	0.743	0.605	0.599	0.980	0.980	0.802	0.788	0.838	0.802
BERT-LSTM	0.742	0.741	0.738	0.732	0.612	0.607	0.983	0.983	0.795	0.784	0.822	0.792
FakeBERT	0.747	0.743	0.756	0.739	0.614	0.608	0.978	0.978	0.776	0.769	0.824	0.797
BERT-2	0.750	0.738	0.744	0.731	0.592	0.591	0.997	0.997	0.803	0.786	0.814	0.775
BERT-LSTM-2	0.750	0.746	0.743	0.741	0.603	0.584	**0.998**	**0.998**	0.806	0.788	0.810	0.793
FakeBERT-2	0.751	0.749	0.742	0.748	0.607	0.606	0.997	0.997	0.818	0.802	0.811	0.776
FANSOG-Unconnected	0.759	0.754	0.753	0.752	0.611	0.613	0.979	0.978	0.806	0.803	0.836	0.800
FANSOG-1	0.761	0.751	0.769	0.768	0.625	0.619	0.971	0.969	0.810	0.806	0.839	0.814
FANSOG	**0.785**	**0.760**	**0.778**	**0.777**	**0.636**	**0.628**	0.983	0.981	**0.823**	**0.819**	**0.849**	**0.824**

diverse news articles as follows (note: the dataset names were assigned by the authors).

- **Combination 1**: This dataset was created by merging three datasets, LIAR, Fake or Real, and KDD2020. Although this dataset contains a range of news articles, it primarily focuses on political and social news since the three datasets mainly comprise news articles related to politics and society.
- **Combination 2**: This dataset was created by merging four datasets, LIAR, Fake or Real, KDD2020, and COVID-19. Compared to the Combination 1 dataset, this dataset has a larger number of news articles and a more diverse range of topics, including news articles related to the Covid-19 pandemic in addition to political and social news.

B. Model Configuration

The hyper-parameters our FANSOG model are set as follows: Primarily, the threshold parameter β is assigned a value of 0.5, indicating the selection of edges with similarity scores falling within the top 50% percentile of the similarity matrix. To leverage information aggregation from text nodes across neighboring K-SOM clusters, we adopt a 3-hop aggregation for this particular problem.

Concerning the neighborhood sampling methodology discussed in section III-D1, distinct sampling schemes are employed for each type of relationship:

- ***"d-d"*** relationship: sample 10, 5, and 10 neighbors in each respective hop.
- ***"d-k"*** relationship: sample 1, 0, and 0 neighbors in each respective hop.
- ***"k-k"*** relationship: sample 0, -1, and 0 neighbors in each respective hop.

  Here, the value -1 denotes the inclusion of all available neighbors associated with the given node.

Moreover, the number of K-SOM clusters is determined as $5\sqrt{n}$, where n represents the total number of data points present in the dataset.

C. Baselines

To demonstrate the effectiveness of our proposed model FANSOG, we compare it with several existing method:

- BERT [13]: This model utilizes the pre-trained BERT model to extract features, followed by passing the [CLS] token through fully connected layers to classify news.
- BERT-LSTM [15]: This model employs the pre-trained BERT model to extract features for each token, followed by using a single BiLSTM layer along with a multi-layer fully connected classification module.
- FakeBERT [14]: This model utilizes the pre-trained BERT model to extract features for each token, then applies multiple one-dimensional convolutional layers with varying filter sizes and kernel dimensions that slide over the tokens in the sentence. The author then concatenates the resulting vectors together, passes them through additional convolutional layers, flattens them, and feeds them into a classification module comprised of fully connected layers.

The present study also aimed to evaluate the efficacy of the two pre-trained models mentioned in III-A for feature extraction. In order to achieve this, we tested baselines with the DistilRoBERTa-base model, which was denoted by appending a "2" suffix to the model name. For instance, the model FakeBERT-2 was utilized to indicate that the FakeBERT architecture was employed, but the content sentence feature extraction layer was replaced with the pre-trained DistilRoBERTa-base model. The same naming convention was applied to the BERT-2 and BERT-LSTM-2 models.

We conducted experiments on variants of FANSOG models, including FANSOG-1 that used bert-base-cased for feature extraction, and our FANSOG model that employed DistilRoBERTa-base. We also investigated the FANSOG-Unconnected model, which excluded connections between K-SOM clusters to examine the impact of neighboring clusters on the current cluster.

D. Experimental Result

The present study reports experimental results on a synthetic dataset and standard dataset, as summarized in Table I. The FANSOG model is observed to outperform other models by leveraging local clustering and learning techniques. For standard dataset, the FANSOG model demonstrated enhancements in both accuracy and F1 Macro when compared to contemporary text classification models, indicating the efficacy of utilizing the similarity between news content within the same cluster and neighboring clusters to augment the local learning capability of the model. FANSOG model demonstrated superior performance by 1% to 3% when utilizing the same pre-trained model for sentence feature extraction, relative to the baseline models. However, our FANSOG model exhibited suboptimal performance in comparison to the baseline models on the Fake or Real dataset. This observation may potentially be attributed to the substantial degree of sentence similarity within the dataset. Therefore, the application of clustering to reinforce the FANSOG model's local learning capability did not result in significant improvements in this specific scenario.

With the synthetic dataset, our evaluation on the Combination 1 dataset yielded accuracy and F1 Macro scores of 78.47%, and 76.01%, respectively. These metrics exhibit an improvement of 2% to 3% compared to the state-of-the-art baseline model, FakeBERT-2. Similarly, on the Combination 2 dataset, the FANSOG model demonstrated superior performance by 2% to 4% compared to other contemporary models.

The presence of inter-cluster connectivity in K-SOM is essential for effectively synthesizing information from news articles with similar content, as such articles may not necessarily belong to the same K-SOM cluster. The results have demonstrated the crucial role of inter-cluster connectivity, as evidenced by the markedly inferior performance of the FANSOG model without connectivity in comparison to the FANSOG model with connections.

V. Conclusion

In this study, we introduced the FANSOG model which is designed to improve local learning and reduce noise encountered by independent text classification models through analyzing connections between news articles. Our experimental results demonstrate that the FANSOG model achieves promising performance on synthetic datasets, indicating its potential effectiveness in real-world applications. However, this model has some limitations. Firstly, it is not an end-to-end model as it requires training a K-SOM layer before the graph neural network layers. We resolved the issue by substituting K-SOM with an end-to-end module, capable of establishing relationships among articles. This can be achieved through a link prediction model based on graph neural networks, as well as the utilization of knowledge graphs not reliant on feature extraction. Secondly, in some cases, the subgraph of each K-SOM node may exhibit low correlation, which could introduce noise in the learned information and lead to erroneous predictions. Hence, further investigations to the FANSOG model may be necessary to mitigate these limitations effectively.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number IZVSZ2.203310.

References

[1] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "Declare: Debunking fake news and false claims using evidence-aware deep learning," 2018.

[2] L. Wu, Y. Rao, L. Sun, and W. He, "Evidence inference networks for interpretable claim verification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 14058–14066, May 2021.

[3] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, (New York, NY, USA), p. 795–816, Association for Computing Machinery, 2017.

[4] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, (New York, NY, USA), p. 675–684, Association for Computing Machinery, 2011.

[5] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: A real-time web-based system for assessing credibility of content on twitter," *CoRR*, vol. abs/1405.5490, 2014.

[6] K. Popat, "Assessing the credibility of claims on the web," in *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, (Republic and Canton of Geneva, CHE), p. 735–739, International World Wide Web Conferences Steering Committee, 2017.

[7] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Proceedings - 2012 IEEE Symposium on Security and Privacy, S and P 2012*, Proceedings - IEEE Symposium on Security and Privacy, pp. 461–475, Institute of Electrical and Electronics Engineers Inc., 2012. 33rd IEEE Symposium on Security and Privacy, S and P 2012 ; Conference date: 21-05-2012 Through 23-05-2012.

[8] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani, "Fake news early detection: A theory-driven model," *Digital Threats*, vol. 1, jun 2020.

[9] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, (Edinburgh, Scotland, UK.), pp. 1589–1599, Association for Computational Linguistics, July 2011.

[10] A. Zubiaga, M. Liakata, and R. Procter, "Exploiting context for rumour detection in social media," in *Social Informatics* (G. L. Ciampaglia, A. Mashhadi, and T. Yasseri, eds.), (Cham), pp. 109–123, Springer International Publishing, 2017.

[11] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51–89, 2003.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[13] J. Y. Khan, M. T. I. Khondaker, S. Afroz, G. Uddin, and A. Iqbal, "A benchmark study of machine learning models for online fake news detection," *Machine Learning with Applications*, vol. 4, p. 100032, 2021.

[14] R. K. Kaliyar, A. Goswami, and P. Narang, "Fakebert: Fake news detection in social media with a bert-based deep learning approach," *Multimedia Tools Appl.*, vol. 80, p. 11765–11788, mar 2021.

[15] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake news classification using transformer based enhanced lstm and bert," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, 2022.

[16] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[17] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[18] W. Y. Wang, ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Vancouver, Canada), pp. 422–426, Association for Computational Linguistics, July 2017.

[19] J. Y. Khan, M. T. I. Khondaker, S. Afroz, G. Uddin, and A. Iqbal, "A benchmark study of machine learning models for online fake news detection," *Machine Learning with Applications*, vol. 4, p. 100032, 2021.

[20] Kaggle, "Fake news detection challenge kdd 2020," 2020. Accessed on April, 2023.

[21] A. Koirala, "Covid-19 fake news dataset," Feb 2021.