Received: April 3, 2024. Revised: May 9, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Hybrid Encoder and Deep Learning Model for Enhancing Rumor Detection from Social Media Texts

Siddheswaran Vanitha[1]* **Raju Prabahari[1]**

[1]*Department of Computer Science, Gobi Arts and Science College, Gobichettipalayam - 638453, Tamilnadu, India*
* Corresponding author's Email: vanithasiddheswaranphd@gmail.com

Abstract: Identifying rumors poses the greatest challenge when dealing with social media platforms. Recently, Deep Learning (DL) Models are widely utilized for the rumor identification using social media data. Amongst, Deep Feature Fusion for Rumor Detection (DFFRD) was used to capture the linguistic features from short-text source tweets for rumor detection. But, this model is not considered complicated Spatio-Temporal (ST) relationships that are sparse, incomplete or high-dimensional. To solve this, Bidirectional Encoder Representations from Transformers with Attention based Balanced Spatial-Temporal Graph Convolutional Networks (BERT-ABSTGCN) is developed in this paper to handle complex ST dependencies in tweet interactions for efficient rumor detection. BERT utilizes Transformer-Based Source Tweet Representation (TSTR) to retrieve context-dependent language features from the source tweet text data. This process alleviates data sparsity issues and performs well on large corpora. ASTGCN constitutes into two modules. A Spatial-Temporal Attention Model (STAM) learns the spatial correlations between various locations and detects dynamic temporal relations among different times in first module. In second module, the Spatial-Temporal Convolution Module (STCM) uses Graph Convolutions (GCs) to extract spatial information among neighbours from Twitter interactions and Temporal Convolutions (TCs) extracts temporal information from neighbourhood time slices. The Balanced ASTGCN (ABSTGCN) deals with irregular relationships in graphs. ABSTGCN adjusts spatial and temporal adjacency matrices dimensions to overcome difficulties in data adaptation and improve performance consistently. The features extracted from transformer representation and ABSTGCN are integrated and fed into softmax layer for the rumor detection. The experimental task imply that the proposed model obtained an accuracy of 94.48%, 93.28% and 93.49% on PHEME, Twitter15 and Twitter16 respectively which is higher than existing models like DFFRD, Convolutional Neural Network Information Gain-Ant Colony Optimization (CNN-IG-ACO), Knowledge Attention Graph Networks (KAGN), Topic and Structure Aware Neural Network (TSNN), Bi-Directional Multi-Level Graph Contrastive Learning (BiMGCL) and Multilayer Feature Fusion-Based GCN (MFF-GCN).

Keywords: Social media, Rumor identification, Deep learning, Transformer representation, Spatial-temporal convolution.

1. Introduction

Online Social Networking (OSN) is a critical inform information platform which have become integral to people's daily lives due to the rapid growth of mobile internet innovation [1]. Platforms like Facebook, Twitter, Weibo, and WeChat have revolutionized media interaction, but the prevalence of incorrect information has significantly altered culture and economy [2]. OSN users face difficulties accessing accurate content due to insufficient data on web pages [3].

Rumors are widely spread, incorrect information that can quickly and consistently cause harm to numerous individuals [4]. Rumors on media platforms can quickly spread, causing significant community damage and financial consequences. Unreliable falsehoods can lead to concerns and violence, as the general audience struggles to distinguish between established facts and rumors [5]. Rumors on OSN have become a significant societal issue, making human detection and categorization impractical. Reliable and automated rumor detection in OSN systems is crucial to overcome these challenges.

Various studies have been conducted to identify rumors on social media, focusing on grammatical, vocabulary and verbal features [6]. However, conventional models overlook global fundamental features in tweet data, hindering learning experiences. DL models have been introduced to solve this issue by extracting local and global structural features, textual attributes and social-temporal contexts of source tweets and related replies [7, 8]. DL models retrieves the latent contextual semantic relationships, learn hidden representations between terms in tweets and statistical properties of co-occurrence among them [9]. However, some models struggle to identify linguistic features, leading to high computational complexity.

To address this issue, DFFRD was developed [10] for Twitter rumor identificationl that extracts linguistic features from short-text tweets and temporal-structural data from dispersion trees. It uses a progressive embedding strategy to preserve propagation tree ST information and CNN to transform encrypted dispersion trees into temporal intrinsic parameters. Transformer methods like BERT and ROBERT reduces the data sparseness in low-lying text categorization. But, this model fails to represent the complex ST dependencies which are typically sparse, incomplete and high-dimensional.

In this paper, BERT-ABSTGCN is developed to address the complicated ST dependencies in tweet interactions for efficient rumor detection. This model utilizes BERT based source tweet representation to extract the context-dependent linguistic features from source tweet text of the collected dataset. This process effectively alleviates data sparsity issues and performs well on large corpora to benefits low-resource tasks. Then, ASTGCN is employed which constitutes two modules. (i) STAM is developed to understand the flexible ST relationships of Twitter exchanges by simulating complex geographical associations and dynamic temporal connections over specific time intervals. (ii) STCM module is developed to visualize the ST interactions of origin tweets, using GSs to capture geographic properties and temporal transformations. The integration of STAM and STCM balances the ASTGCN (ABSTGCN) when applied to the data with irregular connections in graphs. ABSTGCN adjusts spatial and temporal adjacency matrices dimensions to overcome difficulties in data adaptation and improve performance consistently. The features extracted from BERT and ABSTGCN are integrated and fed into softmax layer for the rumor detection. This model assists to handle the complicated ST modules for enhancing the performance of rumor detection using tweets.

The rest of the portions of the paper are prepared as follows: The recent work linked with the rumor detection models is presented in Section II. Section III describes the proposed BERT-ABSTGCN model, while Section IV demonstrates its efficacy. Section V summarizes this paper and suggests its possible improvement.

2. Literature survey

An automatic rumor detection was constructed [11] using CNN, Term Frequency–Inverse Document Frequency (TF-IDF) IG-ACO and NB for rumor classification. But, the accuracy was low as the model utilized only the textual attributes.

User-aspect Multi-View Learning with Attention Rumor Detection (UMLARD) was developed [12] which integrates Twitter content in a low-dimensional layer and uses stacked integration layer for user-level depiction. But, lower accuracy and F1-Score was obtained as the parameters were not adjusted properly.

A globally Distinct Attention Representative from Transformers (gDART) model was presented [13] using Branch-CoRR Attention Network (BCRR) and Feature Fusion Network Component (FFNC) to extract deep hidden features for rumor detection However, this model results in slower convergence rate which lead to lack accuracy.

A cost-sensitive cross-entropy (CSCE) was devised [14] using Deep Neural Networks (DNN) and AdCost function to solve data inconsistencies and change the cost array constants based on outcome percentages for rumor categorization. But, the precision and recall of this model was restricted due to the usage of unbalanced data.

A KAGN model was created [15] using knowledge-aware attention and GCN to link entity references in entity posts and ideas in knowledge graphs for enabling rumor detection task. However, this strategy was restricted for obtaining higher accuracy due to veracity issue.

An Attention Mechanism (AM) method [16] was introduced using Gated Recurrent Unit (GRU) to extract long-distance features and CNN for rumor prediction and classification. However, the method was less precision as it does not consider other properties beyond text.

Received: April 3, 2024. Revised: May 9, 2024.

Diversified Contrary Evidence for Rumor Detection (DCE-RD) was presented [17] which utilizes GCN and subgraph formation techniques to increase diversity and interpretability in rumor detection. However, this model necessitated substantial number of training data to obtain high accuracy.

A TSNN model [18] was developed using granular concept signals and fine-grained topic signals to learn the subject and user credibility prediction to categorize rumors. But, the model was influenced by excessive noise which restricts to obtain higher accuracy.

A BiMGCL model [19] was developed by employing bidirectional network to describe rumor spread, creating optimistic occurrence pairs through three modifications and assessing their compatibility for rumor detection. But, poor accuracy was determined as the reported dataset was not balanced appropriately.

A MFF-GCN model [20] was designed using Heterogeneous Twitter Graph (HTG) to derive every sole-stage attributes to encode topic associations between tweets based on text content. But, the node label data was inadequate to obtain higher accuracy results.

3. Proposed methodology

In this part, the complete framework of suggested ABSTGCN is illustrated for the efficient rumor detection using tweet data. Fig. 1 portrays the pipeline of proposed model. Table 1 represents the notations used in this paper.

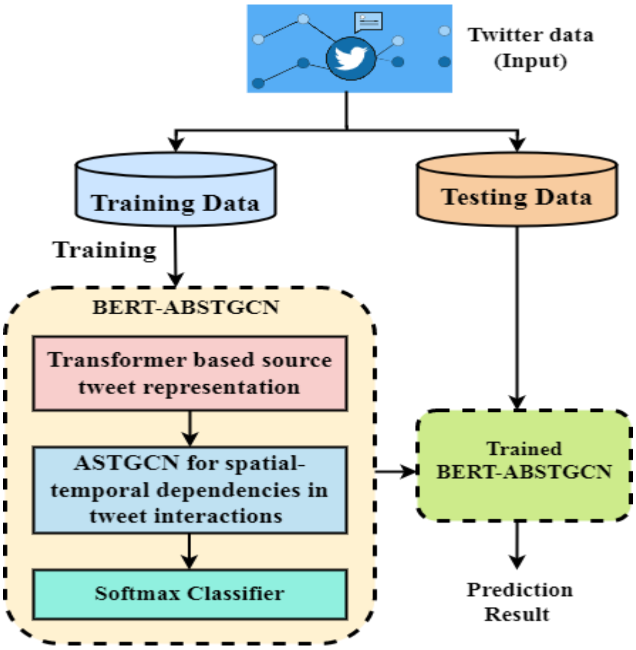

Figure. 1 Pipeline of the proposed model

Table 1. List of Notations

Notations	Description
$\hat{\jmath}_a$	Detected label probabilities with a^{th} class
$\mathcal{I}_n$	Unit matrix
$\mathbb{Q}^K$	Vertices feature with respect to K
$\mathcal{R}_a$	Set of retweets and comments
B_f	Bias of the FC layer
$\hat{F}$	Feature vector
G_θ	Signal graph filtered by kernel
T_D	Daily period
T_M	Monthly period
T_W	Weekly period
T_p	Present element
cl_{u-1}	Channels numbers in u^{th} layer
deg_{xx}	Node degree
$\hat{\jmath}$	Vector predicted probability
$\hat{\jmath}_T^x$	Data flow of node x at future time T
j_a	Actual label probabilities with a^{th} class
t_{u-1}	Temporal size in the u^{th}layer
w_F	Weight of the FC layer
$\hat{z}$	ABSTGCN based spatial-temporal relation
$\mathcal{A}^{(u-1)}$	Initial source of the u^{th} layer
$\mathcal{A}_D$	Every day attribute
$\mathcal{A}_M$	Monthly attribute
$\mathcal{A}_P$	Present attribute
$\mathcal{A}_T$	Attribute values of every node at time T
$\mathcal{A}_W$	Weekly attribute
$\mathcal{J}_a$	Integrated Label
$\mathcal{S}_a$	Source Tweet
$\mathcal{Z}_a$	Spatiotemporal correlations
$\hat{\mathcal{s}}$	Transformer based source tweet
θ_g	Parameter for the transformer model g
$\mathcal{E}$	Temporal relation matrix
$\mathcal{L}(\hat{\jmath})$	Loss function with respect to $\hat{\jmath}$
$\odot$	Hadamard product
C	Spatial attention matrix
G	Number of embedding tokens
H	Hidden layer
K	Tweet series distribution
$LapM$	Laplacian matrix
adj	Adjacent matrix
d	Rumour prediction dataset
deg	Diagonal degree
g	Graph convolution operation
n	Number of classes (rumor\non-rumor)
q	Embedding Vector
$\mathcal{G}$	Sampling task
$\boldsymbol{\mathcal{M}}$	Twitter Relation Network
Λ	Diagonal matrix
δ	Adjacency matrix
ψ	Fourier bias
ϕ	Parameters of temporal convolution kernel
$\boldsymbol{\delta}$	Dropout layer

3.1 Problem description

The collected rumor prediction dataset is described as the affirmation set $d = \{x_1, x_2, \dots\dots, x_{|d|}\ ]$ in which each affirmative $x_a = \{\mathcal{S}_a, \mathcal{Z}_a, \mathcal{J}_a\}$, $\mathcal{S}_a$ represent the source tweet, $\mathcal{Z}_a$ constitutes a ST correlations and integrated label $\mathcal{J}_a$.

The lightest text file is the origin tweet $\mathcal{S}_a$. The total response tweets is represented by t with each $\mathcal{S}_a$ constitutes a set of retweets and comments denoted as $\mathcal{R}_a = \{r_1^a, r_2^a, \dots, r_t^a\}$. A tree layout for $\mathcal{S}_a$ is formed by the files in the collection $\mathcal{R}_a$. The time and space relationships of an initial tweet $\mathcal{S}_a$ are shown as $\mathcal{Z}_a = (V, E)$ with $\mathcal{S}_a$ representing the base of the tree point v_0. In these relations, V and E are the array of nodes and edges accordingly. With respect to the primary tweet $\mathcal{S}_a$, the lapsed time of r_b^a is defined by node $v_b \in V$ and $\mathcal{S}_a$ will be set to 0. The presence of the unique connection is shown by edge $e_y \in E$. There is a class $\mathcal{J}_a$ indicating the beneficial phrases for each $\mathcal{S}_a$ which categorizes $\mathcal{J}_a$ as rumor and non-rumor. Classifiers may use the rumor detection issue to sort source tweets $\mathcal{S}_a$ and the related ST relations $\mathcal{Z}_a$ into appropriate categories. This model aims to enhance the reliability of rumor identification by utilizing social media sentiment analysis and linguistic elements from Twitter content to differentiate between rumors and non-rumors.

3.2 Transformer for tweet sources

The Transformer is a sequence transduction model that improves prediction quality by replacing recurrent layers with multi-headed self-attention. In this method, BERT is utilized which is a multi-staged language description model constitutes 12 transformer blocks and self-attention heads to acquire deep bidirectional representations from unclassed content. It converts phrases into word embeddings using WordPiece embeddings, with a 30K word vocabulary, and uses an autonomous learning strategy on large datasets like English Wikipedia and BooksCorpus.

An embedding vector $q = \{q_1, q_2, \dots, q_n\}$ is generated from a given source tweet $\mathcal{S}_a$ where n is embedded numerical characters derived by matching word integration algorithm employed by BERT. The transformer model takes the encoding vector q as an input variables. A matrix with dimensions $G * H$ is produced by the transformer's final hidden layer ∂. Here, G is the number of embedding tokens and H is the size of the last hidden layer (768 for BERT base models). The forwarding task is illustrated as follows in Eq. (1) and Eq. (2) :

$$\partial = Transformer_g(\theta_g, \boldsymbol{q}) \quad (1)$$

$$\hat{s} = \partial(1) \quad (2)$$

Where, $\partial(1)$ is the initial rank of ∂ and $\hat{s} = \{\hat{s}^1, \hat{s}^2, \dots, \hat{s}^H\}$ and θ_g represents the parameter for the transformer model g.

3.3 Attention based spatial-Temporal graph convolutional network

ASTGCN model is commonly used to predict data transmission from Twitter at every node in the relationship matrix by explicitly interpreting tweet content on its initial graph-based network. Assume $t-$ time series captured every nodes in the twitter relation network $\boldsymbol{\mathcal{M}}$ which will be the tweet series distribution, $k \in (1, \dots K)$, $a_T^{cl,x} \in \mathbb{Q}$ is employed to represent the c^{th} node attribute x with time T, $a_T^x \in \mathbb{Q}^K$ determines the vertices feature x at time T. The set $\mathcal{A}_T = (a_T^1,\ a_T^2, \dots, a_T^n) \in \mathbb{Q}^{n\times \not{f}}$ denotes the attribute values of every node at time T. Values of all characteristics of all nodes throughout τ time slices are described by $\mathcal{A} = (A_1,\ A_2, , . . , A_\tau)^t \in \mathbb{Q}^{n*\not{f}*\tau}$. Furthermore, the input $j_T^x = i_T^{F,x} \in \mathbb{Q}$ is modified to symbolize the data flow of node x at future time T.

An ASTGCN consists of four independent elements designed to model the ST correlation, including present, everyday-periodic, weekly, and monthly time dependencies of past information. The sampling task $\mathcal{G}$ is accessed with every day time instances samples as the current time being with t_0 and the forecasting window size is devised to T_c. The input for the every minute (present), day, week and month elements originates from the time node and four time sequence divisions with lengths T_P, T_D, T_W and T_M are observed. T_P, T_D, T_W and T_M are all integer multiples of T_c. The description of four-time series sections are as follows,

(i) Present Segment: The division of past time sequences are immediately preceding the forecasting time which constitutes to $\boldsymbol{\mathcal{A}_H} = \left(A_{t_0-T_P+1}, A_{t_0-T_P+2}, \dots.., A_{t_0}\right) \in \mathbb{Q}^{n\times \not{f}\times T_P}$. The creation and dispersal of tweet delays appear to be gradual. As a result, present tweet data flows will undoubtedly have an impact on future tweet interactions.

(ii) Everyday-Period Segment: The segments represent the preceding days within the same time interval as the prediction period, indicating that the segments consist of the same days within the same time interval which is shown in Eq. (3):

Received: April 3, 2024. Revised: May 9, 2024.

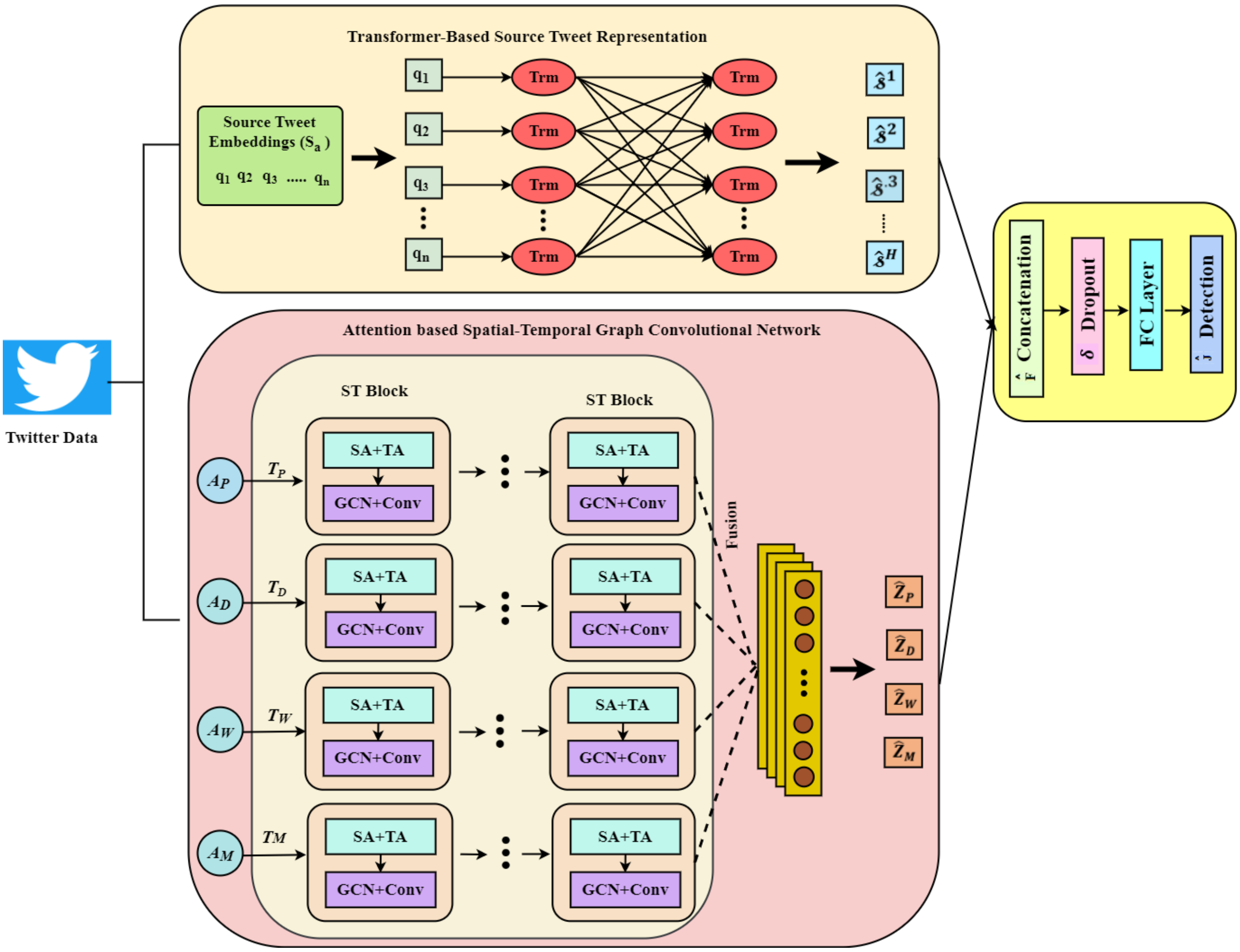

Figure. 2 Complete Framework of proposed BERT-ABSTGCN model

$$\boldsymbol{\mathcal{A}_D} = (A_{t_0-(T_D\backslash T_P)\times\mathcal{G}+1}, \cdots, A_{t_0-(T_D\backslash T_P)\times\mathcal{G}+T_c}, A_{t_0-(T_D\backslash T_P-1)\times\mathcal{G}+1}, \cdots, A_{t_0-\mathcal{G}+1}, \ldots, A_{t_0-\mathcal{G}+T_c}) \in \mathbb{Q}^{n\times\not{f}\times T_D} \quad (3)$$

The section is made up of current days that fall within the particular time range as the forecast period. Because of regular tweets posted by people, the twitter data might exhibit the recurring patterns i.e., number of people shared the posts. The main purpose of daily-period element to simulate the daily periodicity of tweet data.

(iii) Weekly Period Segment: This category comprises specific features where the prediction interval period is determined by comparing similar qualities in a week periods, it is shown in Eq. (4):

$$\boldsymbol{\mathcal{A}_W} = (A_{t_0-7\times(T_D\backslash T_P)\times\mathcal{G}+1}, \cdots, A_{t_0-7\times(T_D\backslash T_P)\times\mathcal{G}+T_c}, A_{t_0-7\times(T_D\backslash T_P-1)\times\mathcal{G}+1}, \cdots, A_{t_0-7\times(T_D\backslash T_P-1)\times\mathcal{G}+T_c}, \cdots, A_{t_0-7\times\mathcal{G}+1}, \ldots, A_{t_0-7\times\mathcal{G}+T_c}) \in \mathbb{Q}^{n\times\not{f}\times T_W} \quad (4)$$

In basis of weekly divisions, current Monday tweet posting patterns are usually comparable to past Monday patterns, however they might differ substantially from those on weekends. As a result, the weekly-recurring element intends to identify weekly interval features in the tweet post data.

(iv) Monthly Period Segment: This part constitutes to a month wise computation by considering the tweet post of total days and weeks accumulated in complete month and it is compared with preceding month as the detection period is shown in Eq. (5):

$$\boldsymbol{\mathcal{A}_M} = (A_{t_0-n\times(T_D\backslash T_P)\times\mathcal{G}+1}, \cdots, A_{t_0-n\times(T_D\backslash T_P)\times\mathcal{G}+T_c}, A_{t_0-n\times(T_D\backslash T_P-1)\times\mathcal{G}+1}, \cdots, A_{t_0-n\times(T_D\backslash T_P-1)\times\mathcal{G}+T_c}, \cdots, A_{t_0-n\times\mathcal{G}+1}, \ldots, A_{t_0-7\times\mathcal{G}+T_c}) \in \mathbb{Q}^{n\times\not{f}\times T_M} \quad (5)$$

The current monthly tweet patterns, including all days and weekly posts, are compared to past month patterns, but may differ significantly on a daily and weekly basis. The monthly period component of a Twitter data model identifies patterns over n days and weeks using a recurrent learning architecture. The model incorporates convolution units and STAM, and the final forecast uses a parameter vector to describe variable ST interactions in the tweet data relation. Fig. 2 provides the complete framework of proposed model for rumor detection.

3.3.1. Spatial-Temporal Attention Module (STAM)

In order to capture the dynamic geographical and temporal correlations on the twitter interactions network, this model incorporates a unique STAM. It incorporates Spatial Attention (SA) and Temporal Attention (TA) which is briefly illustrated below.

Spatial Attention: This SA assist to identify the tweet shared locations which have directed among each other and reciprocal influence is greatly effective. The current SA element is considered as an example in which the attention mechanism adapts to record the constant interactions among elements in spatial variables are represented Eq. (6) and Eq. (7):

$$\mathcal{C} = v_p . \sigma(\begin{matrix} \left(\mathcal{A}_p^{(u-1)} w_1\right) \times \\ w_{2\times} \left(w_3 \mathcal{A}_p^{(u-1)}\right) \end{matrix}^{t} + j_p \tag{6}$$

$$\mathcal{C}'_{x,y} = \frac{\exp(\mathcal{C}_{x,y})}{\sum_{y=1}^{n} \exp\,(\mathcal{C}_{x,y})} \tag{7}$$

Where, $\mathcal{A}_p^{(u-1)} = (\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{t_{u-1}}) \in \mathbb{Q}^{n \times cl_{u-1} \times t_{u-1}}$ defines the initial source of the u^{th} ST block. cl_{u-1} denotes the channels numbers of the initial source in the u^{th} ramge. When $u = 1$, $cl_0 = K . t_{u-1}$ is the length of the temporal size in the u^{th}layer. Whe n $u = 1$, $t_0 = T_p$ for the present element, $t_0 = T_D$ (daily period), $t_0 = T_W$ (weekly period) and $t_0 = T_M$ (monthly period). $v_p, j_p \in \mathbb{Q}^{n\times n}, w_1 \in \mathbb{Q}^{t_{u-1}}, w_2 \in \mathbb{Q}^{cl_{u-1} t_{u-1}}, w_3 \in \mathbb{Q}^{cl_{u-1}}$ are trainable variables and sigmoid σ is employed for the activation function. The weights that are affected by a graph may be dynamically adjusted using the adjacency matrix δ and the SA matrix $\mathcal{C}' = \mathbb{Q}^{n*n}$.

Temporal Attention: An AM is employed to automatically apply different standards to information, and the temporal factor exposes relationships among traffic instances over several time sections, which change in Eq. (8) and Eq. (9): by event.

$$\mathcal{E} = v_D . \sigma\left(\left(\mathcal{A}_p^{(u-1)}\right)^t R_1\right) \times$$

$$R_2 \left(R_3 \mathcal{A}_p^{(u-1)}\right) + j_D \tag{8}$$

$$\mathcal{E}'_{x,y} = \frac{\exp(\mathcal{E}_{x,y})}{\sum_{y=1}^{t_{u-1}} \exp\,(\mathcal{E}_{x,y})} \tag{9}$$

Where, v_D , $j_D \in \mathbb{Q}^{t_{u-1} \times t_{u-1}}, R_1 \in \mathbb{Q}^{n}, R_2 \in \mathbb{Q}^{cl_{u-1} \times n}$ and $R_3 \in \mathbb{Q}^{cl_{u-1}}$ represents the learnable variables. The temporal relation matrix E is obtained by distinctive inputs. The component integer $\mathcal{E}_{x,y}$ in $\mathcal{E}$ semantically indicates the correlation among x and y. Finally, E is the output of the softmax operation with the TA matrix, which quickly processes the input and finds to merge relevant data and make flexible modifications to the input.

3.3.2. Spatial-Temporal convolution module (STCM)

By modifying data and sending it to the STCM, the STAM ranks network information in priority. Included in this package is spatial graph convolution, which captures local relationships and temporal graph convolution which exploits adjacent times.

Graph convolution in spatial dimension: Spectral graph theory is used to analyze Twitter interaction networks by translating node attributes into graph indications, while direct signal analysis using spatial dimension signal correlations enhances traffic network design by converting the graph into algebraic form. The Laplacian matrix ($LapM$) and its eigenvalues are the objects evaluation. $LapM$ of the graph is determined as $lap = deg - adj$ and its normalized terms in Eq. (10)

$$\begin{aligned} lap = \mathcal{I}_n - deg^{-1/2} \times \\ adj \times deg^{-1/2} \in \mathbb{Q}^{n\times n} \end{aligned} \tag{10}$$

Where adj, $\mathcal{I}_n$, $deg \in \mathbb{Q}^{n\times n}$ and $deg_{xx} \in \mathbb{Q}^{n*n}$ represents the matrix modules of adjoining unit, degree and diagonal with node degrees accordingly. Eigen value decomposition in $LapM$ is $lap = \psi \wedge \psi^t$ where $\wedge = dl([\mu_0, \dots, \mu_{n-1}]) \in \mathbb{Q}^{n\times n}$ represents the crosswise matrix and ψ is the fourier bias. Bu using tweet data at time T, the input for the complete graph will be $a = a_T^k \in \mathbb{Q}^n$ and the Graph Fourier Transform (GFT) of the signal is illustrated as $\hat{a} = \psi^t a$. In respective to the criteria of $LapM$, Inverse Fourier Transform (IFT) will be $a = \psi\,\hat{a}$ with respect to orthogonal matrix ψ. The graph convolution involves applying a diagonalized linear function to the Fourier domain. Considering this, the input a with graph g is excluded by a kernel G_θ in Eq. (11) as:

Received: April 3, 2024. Revised: May 9, 2024.

$$G_\theta \times g^a = g(lap) \times a = G_\theta\left(\psi \wedge \psi^t\right) \times a = \psi_{G_\theta}(\wedge)\, \psi^t \tag{11}$$

In Eq. (8), g defines the graph convolution function. GFT is the multiplication of G_θ and a and IFT are all part of the process of convolution for a graph signal. However, Chebyshev Polynomials (CSP) are being utilized to effectively address the costly issue of eigenvalue reduction in large graph sizes in Eq. (12)

$$G_\theta \times g^a = g(lap) \times a = \sum_{m=0}^{m-1} \theta_m\, t_\theta\left(\widehat{lap}\right) \times a \tag{12}$$

Where, the variable $\theta \in \mathbb{Q}^m$ is a binomial co-integer matrix. $\widehat{lap} = \frac{2}{\mu_{max}}\, lap - \mathcal{I}_n$, μ_{max} defines the greatest eigen integer of the $LapM$. The iterative illustration of the CSP is $t_m\,(a) = 2at_{m-1}(a) - t_{m-2}(a)$ in which $t_0(a) = 1$; $t_1(a) = a$. By applying the approximation to CSP, data from the adjacent formulation is 0 to $(m-1)^{th}$ arranged with adjoining positions on every node in the graph using the convolution kernel G_θ . The graph convolution method operates the Rectified Linear Unit (ReLU) as the last activation function i.e., ReLU $(G_\theta \times g^a)$. To modify the connection among the nodes for each label of CSP, $t_m\left(\widehat{lap}\right)$ with the SA vector $\mathcal{C}' \in \mathbb{Q}^{n\times n}$ is determines as $t_m\left(\widehat{lap}\right) \odot \mathcal{C}'$ where $\odot$ is the Hadamard operation. Moreover, the graph convolution significantly changes to $G_\theta \times g^a = g(lap)a = \sum_{m=0}^{m-1} \theta_m \left(t_m\left(\widehat{lap}\right) \odot \mathcal{C}'\right) \times a$.

Normalized Convolution in Temporal Dimension: Graph convolution operations record adjacent data with each node in the spatial dimension, with a typical temporal convolution layer upgrading node input by integrating data with the adjoining time slice. Assume the u^{th} layer operations of the as the current element instances as in Eq. (13),

$$\boldsymbol{\mathcal{A}}_p^{(u)} = ReLU\left(\phi \times ReLU\left(G_\theta \times g^{\hat{A}_P^{(u-1)}}\right)\right) \in \mathbb{Q}^{n\times cl_u \times t_u} \tag{13}$$

In above Eq. (10), ϕ represents the temporal convolution kernel variables. The STCM effectively collects temporal and geographical information from Twitter data using a block layout, STAM and convolution module. Multiple blocks are layered for dynamic correlations, and a FC layer ensures outputs fit forecasting objectives for rumor prediction. The impacting weights for all nodes are calculated based on the earlier post data when merged. Therefore, final prediction of ASTGCN after the integration is provided in Eq. (14)

$$\hat{z} = \mathcal{W}_P \odot \hat{Z}_P + \mathcal{W}_D \odot \hat{Z}_D + \mathcal{W}_W \odot \hat{Z}_W + \mathcal{W}_M \odot \hat{Z}_M \tag{14}$$

Where, $\mathcal{W}_P, \mathcal{W}_D\,, \mathcal{W}_W\ and\ \mathcal{W}_M$ are the learning variables indicating the influences degrees of the four ST elements. An extracted features from the STAM and STCM, $\hat{Z}_P$, $\hat{Z}_D$,$\hat{Z}_W$ and $\hat{Z}_M$ is applied for rumor detection. The model balances STAM features with STCM, utilizing ABSTGCN for training, optimizing spatial and temporal adjacency matrices to eliminate irregular connections and enhance rumor detection.

3.4 Feature integration and classification

n order to eradicate overfitting problems in the training dataset, the features retrieved from the transformer-based initial tweets $\hat{s}$ Eq. (15) and the ABSTGCN-based spatial temporal relation $\hat{z}$ are combined into a feature vector $\hat{F}$ in Eq. (16). The FC layer uses the soft max operation to determine the label of the tweet data for rumor detection after receiving the result of the dropout layer $\boldsymbol{\delta}$ in Eq. (17):

$$\hat{F} = [\hat{s}\,, \hat{z}] \tag{15}$$

$$\delta = \ Dropout\ (\hat{F}) \tag{16}$$

$$\hat{j} = softmax\ (\delta \times\ w_F + B_f) \tag{17}$$

Where, w_F and B_f are the weight and bias related with the FC layer, $\hat{j}$ defines the vector predicted probabilities of the class labels.

3.5 Training model generation

During training, the three trainable modules constitutes to develop the suggested BERT-ABSTGCN collectively. In model training, the loss function is characterized as the Cross-Entropy (CE) of the prediction outcomes ($\hat{j}$).

$$\mathcal{L}(\hat{j}) = -\sum_{a=1}^{n} j_a \log(\hat{j}_a) + (1 - j_a)\log\,(1 - \hat{j}_a) \tag{18}$$

In Eq. (18), n represents the number of classes (rumor\non-rumor), j_a and $\hat{j}_a$ represents the real and predicted class probability in the a^{th} label

Table 2. Parameter values for proposed and existing model

Models	Parameters	Range	Models	Parameters	Range
DFFRD [10]	BERT – Transformer Blocks	12	BiMGCL [19]	Learning rate	0.001
	Hidden Size	768		Weight decay	0.05
	Self-Attention Heads	12		Epochs	150
	Learning rate	0.001		Batch size	64
	Epochs	200		Activation Function	ReLU
	Batch size	32		Optimizer	Adam
	Dropout rate	0.5		Loss Function	CE
	Activation Function	ReLU	MFF-GCN [20]	Learning rate	0.02
	Optimizer	AdamW		Window Size	20
IG-ACO [11]	Learning rate	0.1		Embedding Size	200
	Epochs	100		Batch size	24
	Batch size	32		Activation Function	ReLU
	Dropout rate	0.9		Optimizer	Adam
	Activation Function	RMSprop		Loss Function	CE0
	Optimizer	Sigmoid	Proposed BERT-ABSTGCN	GC layer	64
	Loss Function	Binary CE		Word Embedding	300
KAGN [15]	Filters	4		Temporal Convolutional layer	64
	Word Embedding	300		Stride; Padding	2; 3
	Learning rate	0.003		Learning rate	0.0001
	Epochs	50		Weight decay	1e-3
	Batch size	16		Epochs	250
	Dropout rate	0.5		Batch size	64
	Activation Function	ReLU		Dropout rate	0.7
	Optimizer	AdamW		Activation Function	ReLU
	Loss Function	CE		Optimizer	Adam
TSNN [18]	Learning rate	0.001		Loss Operation	CE
	Weight decay	0.004			
	Epochs	80			
	Batch size	64			
	Dropout rate	0.7			
	Activation Function	Tanh			
	Optimizer	AdaGrad			
	Loss Function	MSE			

correspondingly. The three parts of a model's trainable parameters θ are varied in the direction of gradient descent and back-propagation technique.

Algorithm: BERT-ABSTGCN

Input: Twitter Dataset – Tweets

Output: Target label of test data **(**Rumor or not)

1. Start
2. Split the gathered twitter data into training and testing set.
3. Employ the BERT model to retrieve the context-dependent attributes from the source tweet data;
4. Construct the ASTGCN model to find complex ST dependencies in tweet interactions;
5. Split the ASTGCN into STAM and STCM;

6. Utilize the STAM to learn the complex spatial and temporal interactions;
7. Identify adjacent time dependences form tweets using TCs;
8. Devise the STCM for ST dependencies of source tweets representations;
9. Arrange the GCs to analyze the closest time portions of tweets for spatial and temporal features estimations;
10. Adjust the spatial and temporal adjacency matrices dimensions for ABSTGCN;
11. Concatenate the features extracted from BERT and ABSTGCN and fed into the softmax classifier;
12. Train the LSTM by the ReLU activation to get the trained model;
13. Test the data target is predicted based on trained LSTM model Evaluate the efficiency of prediction;
14. End

4. Result and discussion

4.1 Dataset description

PHEME dataset [21]: The dataset presents a variety of Twitter rumors for and non-rumors, disrupting current or interesting events. It contains rumors about nine incidents and each rumor is marked with its credibility level: true, false or unverified. This dataset is focused on two events the german wing crash and charlie hebdo. The dataset contains over 60,000+ rows in which 62446 tweets have been considered.

Twitter15 and Twitter 16 [22]: This dataset has been widely adopted as standard data in the field of rumor detection. Twitter15 and Twitter16 contains 1490 and 818 tweets propagations respectively. Each tweet propagation is labeled with one of four types, including non-rumor, false rumor, true rumor and unverified rumor.

4.2 Performance analysis

This section compares the efficiency of the BERT-ABSTGCN model to existing rumor detection techniques like DFFRD [10], CNN-IGACO [11] KAGN [15], TSNN [18] BiMGCL [19] and MFF-GCN [20] on the considered dataset (described in Section 4.1). Both the proposed and existing models are implemented in Python 3.11 and executed on a system with an Intel® CoreTM i5-4210 CPU @ 3GHz, 4GB RAM, and a 1TB HDD running on Windows 10 64-bit with the stimulation parameter listed in Table 2.

For the experimental purposes, the collected datasets are individually divided into 70% for training and remaining 30% for testing. The performance metrics used to evaluate the proposed and existing algorithms are described below:

Accuracy: It indicates the proportion of correct predictions among the entire number of tweets studied. Eq. (19),

$$Accuracy = \frac{True\ Positive\ (TP)+True\ Negative\ (TN)}{TP+TN+False\ Positive\ (FP)+False\ Negative\ (FN)} \quad (19)$$

Here, TP and FN represent the amount of tweets correctly detected. Also, FP and TN represent the amount of tweets that are incorrectly recognized.

Precision: It is the proportion of correctly recognized rumor posts to overall detected tweets in a rumor category. It is shown in Eq. (20),

$$Precision = \frac{TP}{TP+FP} \quad (20)$$

Recall: Eq. (21) calculates the proportion of rumor tweets that are correctly recognized.

$$Recall = \frac{TP}{TP+FN} \quad (21)$$

F1-score: It represents the partial mean of precision and recall. It is represented in Eq. (22)

$$F - measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (22)$$

Fig. 3 displays the efficacy of different various models on PHEME dataset for rumour recognition. Accordingly, it realizes that the accuracy of the BERT-ABSTGCN is 20.34%, 15.61%, 12.22% 7.92%, 5.61% and 2.48%; precision of BERT-ABSTGCN is 20.65%, 13.72%, 11.3%, 7.85%, 6.31% and 2.87%; recall of the BERT-ABSTGCN model is 17.05%, 12.86%, 9.56%,6.42%, 4.85% and 2.13; F-measure of BERT-ABSTGCN model is 20.59%, 15.48%, 13.27%,9.04%, 6.92% and 2.82% higher than the other existing models like DFFRD, IG-ACO, KAGN, TSNN, BiMGCL and MFF-GCN models respectively.

Fig. 4 displays the efficacy of various models on Twitter15 dataset for rumour recognition. Accordingly, it is understood that the accuracy of BERT-ABSTGCN is 20.19%, 15.83%, 12.68%, 9.39%, 6.94% and 1.95%, precision of BERT-ABSTGCN is 20.63%, 16.52%, 12.16%, 10.42%, 6.14% and 2.41%; recall of the BERT-ABSTGCN

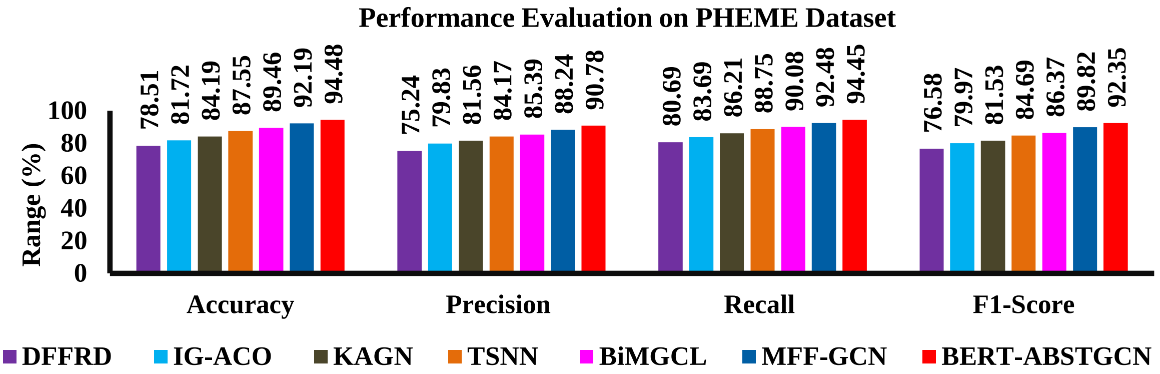

Figure. 3 Performance analysis of proposed and existing models using PHEME dataset

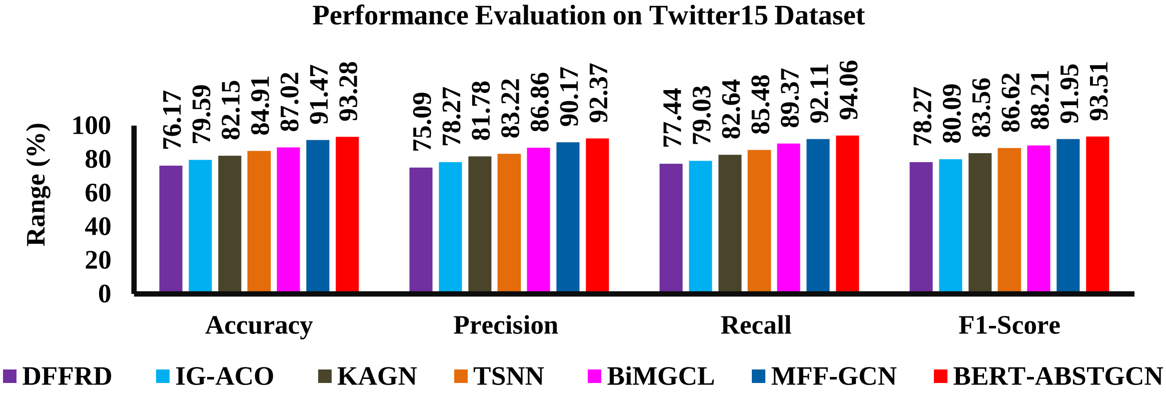

Figure. 4 Performance Evaluation of Existing and Proposed Models on Twitter15 dataset

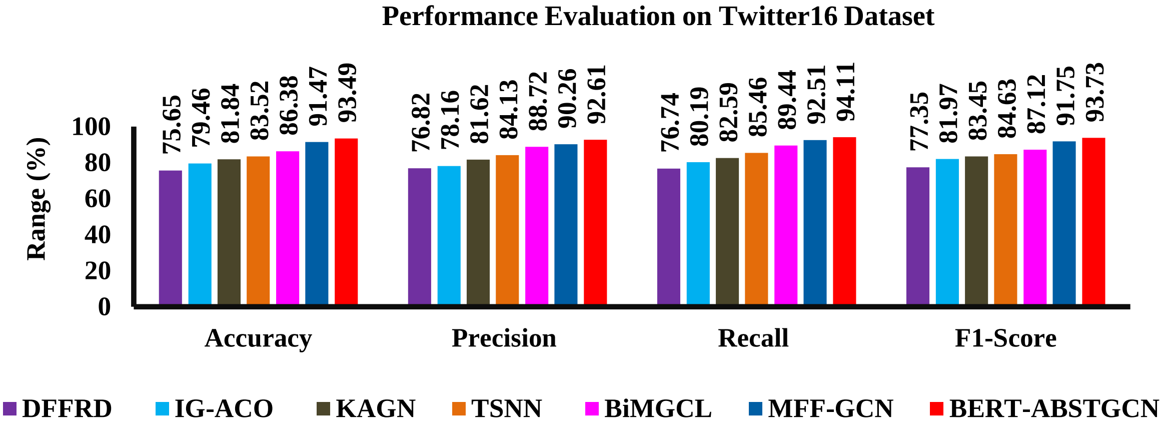

Figure. 5 Performance Evaluation of Existing and Proposed Models on Twitter16 dataset

model is 19.38%, 17.36%, 12.92%, 9.55%, 5.11% and 2.09%; F-measure of BERT-ABSTGCN is 17.74%, 15.46%, 11.23%, 7.65%, 5.83% and 1.68% higher than the other prediction models like DFFRD, IG-ACO, KAGN, TSNN, BiMGCL and MFF-GCN models respectively.

Fig. 5 displays the efficacy of various models on Twitter16 dataset for rumour recognition. Accordingly, it is devised that the accuracy of BERT-ABSTGCN is 21.09%, 16.22%, 13.28%, 11.26%, 7.90% and 2.18%; precision of BERT-ABSTGCN is 18.63%, 16.92%, 12.61%, 9.59%, 4.29% and 2.57%; recall of the BERT-ABSTGCN model is 20.33%, 15.97%, 13.03%, 9.63%, 5.08% and 1.71%; F-measure of BERT-ABSTGCN is 19.14%, 13.38%, 11.60%, 10.20%, 7.30% and

2.13% higher than the other models like DFFRD, IG-ACO, KAGN, TSNN, BiMGCL and MFF-GCN models respectively.

In the literature, IG-ACO [11], KAGN [15] and MFF-GCN [20] dataset have utilized PHEME dataset for the evaluation. Similarly, in this model, DFFRD [10], KAGN [15], TSNN [18] and BiMGCL [19] models have considered Twitter15 and Twitter16 for the performance task. Hence, this work evaluates proposed and existing models on PHEME, Twitter15 and Twitter16 datasets using the parameters in Table 2. From the above comparison, it is proved that the proposed BERT-ABSTGCN model determines efficient results on collected datasets than other existing models for the rumour detection and classification. This is because the suggested model effectively handles irregular connections in graphs by modifying spatial and temporal adjacency matrix dimensions and eliminating intricate ST dependencies in tweet interactions using feature representations derived from graphs for effective rumour prediction.

5. Conclusion

In this paper, BERT-ABSTGCN is proposed for rumor prediction using twitter dataset. This method employs TSTR to retrieve the context-dependent vocabular features from tweet text, reducing data sparsity and excelling on large corpora. The ASTGCN learns complicated spatial and dynamic temporal relationships by combining GCs for spatial features and TCs for local time sections. The ABSTGCN is employed to data with irregular graph relationships, adjusting spatial and temporal adjacency matrices dimensions to improve performance. The extracted features are integrated into a softmax layer for rumor detection and categorization. Finally, the suggested method obtains 94.48%, 93.28% and 93.49% accuracy on PHEME, Twitter 15 and Twitter 16 respectively which is greater than other models like DFFRD, IG-ACO, KAGN, TSNN, BiMGCL and MFF-GCN.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, Vanitha; formal analysis, investigation, Prabahari; resources, data curation, writing—original draft preparation, Vanitha; writing—review and editing, Vanitha; visualization, supervision, Prabahari;

References

[1] M. R. Zheltukhina, G. G. Slyshkin, E. B. Ponomarenko, M. V. Busygina, and A. V. Omelchenko, "Role of Media Rumors in the Modern Society", *International Journal of Environmental and Science Education,* Vol. 11, No. 17, pp. 10581-10589, 2016.

[2] A. Dang, M. Smit, A. Moh'd, R. Minghim, and E. Milios, "Toward understanding how users respond to rumours in social media," In: *Proc. of 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM),* pp. 777-784, 2016.

[3] N. DiFonzo and P. Bordia, "Rumors influence: Toward a dynamic social impact theory of rumor", *The science of social influence, Psychology Press,* pp. 271-295, 2011.

[4] C. Wang, G. Wang, X. Luo, and H. Li, "Modeling rumor propagation and mitigation across multiple social networks", *Physica A: Statistical Mechanics and its Applications,* Vol. 535, pp. 122240, 2019.

[5] S. M. Alzanin and A. M. Azmi, "Detecting rumors in social media: A survey", *Procedia Computer Science,* Vol. 142, pp. 294-300, 2018.

[6] H. Bingol and B. Alatas, "Rumor Detection in Social Media using machine learning methods", In: *Proc. of 2019 1st International Informatics and Software Engineering Conference (UBMYK),* pp. 1-4, 2019.

[7] X. He, G. Tuerhong, M. Wushouer, and D. Xin, "Rumors detection based on lifelong machine learning", *IEEE Access*, Vol. 10, pp. 25605-25620, 2022.

[8] B. Pattanaik, S. Mandal, and R. M. Tripathy, "A survey on rumor detection and prevention in social media using deep learning", *Knowledge and Information Systems,* pp. 1-42, 2023.

[9] L. Tan, G. Wang, F. Jia, and X. Lian, "Research status of deep learning methods for rumor detection", *Multimedia Tools and Applications*, Vol. 82, No. 2, pp. 2941-2982, 2023.

[10] Z. Luo, Q. Li, and J. Zheng, "Deep feature fusion for rumor detection on twitter", *IEEE Access*, Vol. 9, pp. 126065-126074, 2021.

[11] A. Kumar, M. P. S. Bhatia, and S. R. Sangwan, "Rumour detection using deep learning and filter-wrapper feature selection in benchmark twitter dataset", *Multimedia Tools and Applications*, Vol. 81, No. 24, pp. 34615-34632, 2022.

[12] X. Chen, F. Zhou, G. Trajcevski, and M. Bonsangue, "Multi-view learning with

distinguishable feature fusion for rumor detection", *Knowledge-Based Systems*, Vol. 240, pp. 108085, 2022.

[13] S. Roy, M. Bhanu, S. Saxena, S. Dandapat,and J. Chandra, "gDART: Improving rumor verification in social media with Discrete Attention Representations", *Information Processing & Management* , Vol. 59, No. 3, pp. 102927, 2022.

[14] Z. Zojaji and B. Tork Ladani, "Adaptive cost-sensitive stance classification model for rumor detection in social networks", *Social Network Analysis and Mining*, Vol. 12, No. 1, pp. 134, 2022.

[15] W. Cui and M. Shang, "KAGN: knowledge-powered attention and graph convolutional networks for social media rumor detection", *Journal of big Data*, Vol. 10, No. 1, pp. 45, 2023.

[16] P. Wan, X. Wang, G. Pang, L. Wang, and G. Min, "A novel rumor detection with multi-objective loss functions in online social networks", *Expert Systems with Applications*, Vol. 213, No. 119239, 2023.

[17] K. Zhang, J. Yu, H. Shi, J. Liang, and X. Y. Zhang, "Rumor Detection with Diverse Counterfactual Evidence", In: *Proc. of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3321-3331, 2023.

[18] Z. Chen, L. Wang, X. Zhu, and S. Dietze, "TSNN: A Topic and Structure Aware Neural Network for Rumor Detection", *Neurocomputing*, Vol. 531, pp. 114-124, 2023.

[19] W. Feng, Y. Li, B. Li, Z. Jia, and Z. Chu, "BiMGCL: rumor detection via bi-directional multi-level graph contrastive learning", *PeerJ Computer Science*, Vol. 9, pp. e1659, 2023.

[20] S. Chen, M. Li, and W. Yang, "Multilevel Feature Fusion-Based GCN for Rumor Detection with Topic Relevance Mining", *Advances in Multimedia*, 2023.

[21] https://www.kaggle.com/datasets/nicolemichelle/pheme-dataset-for-rumour-detection.

[22] J. Ma, W. Gao, and K. F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning", In: *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1, pp. 708-717, 2017.