RESEARCH ARTICLE OPEN ACCESS

Context Aware Deep Ensemble Learning Model for Rumor Detection

D. Rajasekar*, Dr. L. Robert
Department of Computer Science, Government Arts College, Coimbatore-641018

ABSTRACT
Rumor recognition is the most challenging task on social media platforms. The local and global structural features existing between the original tweet and the responses to it have been the focus of numerous rumor recognition models. Twitter word-embedded Ensemble Graph Convolutional neural network (T-EGCN) model trained to distinguish rumors from a huge number of tweets by creating word embedding and combining them with other tweet-related properties. But its robustness was lessened since it ignored the correlations and differences between various contexts (called responses) over time. The social-temporal contexts of source tweets are critical for resolving this issue. Hence, this article proposes a Twitter word-embedded and Time-series EGCN (TT-EGCN) model for rumor recognition. The fundamental goal of this model is to incorporate a Long Short-Term Memory (LSTM) network into the T-EGCN in order to better handle the time-varying nature of replies. Together, the Text-Convolutional Neural Network (T-CNN), stacked LSTM networks, and GCN make up the TT-EGCN model, which is used to represent tweets' textual contents, social-temporal contexts, and global structural aspects. This model is able to pick up on how rumors spread in their formative stages. In addition, a hierarchical attention network is used to teach various context inputs how to form a single attentive context embedding. The tweets are then categorized as rumor or non-rumor using a softmax algorithm. At last, the test results show that the TT-EGCN attains 90.02% accuracy, which is higher than all other existing rumor recognition models.

Keywords-Rumor recognition, T-EGCN, Time-series data, LSTM network, Context data, Hierarchical attention strategy

---

Date of Submission: 24-06-2023 Date of acceptance: 05-07-2023

I. INTRODUCTION

Microblogging platforms are important information repositories that have been successfully used to study sociopragmatic factors such as beliefs, opinions, and sentiments in a technological sense. Before it makes it into traditional media outlets, breaking news is typically announced on microblogs. When it comes to studying Natural Language Processing (NPL), Twitter is one of the best places to look [1-2]. The availability of such vast amounts of data has both positive and negative consequences. Unreliable information is readily available from certain sources, and its proliferation is difficult to curb. When information is presented as having been efficiently collected, it might be difficult to tell what is true and what is just a rumor [3-4]. For example, a piece of data can be rapidly widespread by simply tapping the Re-tweet option on Twitter. The term rumor can be interpreted in a variety of ways. Rumor is a term for information that generates and circulates among people without being validated by empirical evidence [5]. It is untested information that could turn out to be true or false, positive or negative, or it could go unanswered. Though its political or commercial sources and motives are visible, it is a remark of uncertain credibility with no discernible basis [6-7].

Due to the possible rapid growth of global social media platforms like Facebook, Twitter, and Sina Weibo [8], it appears to be quite a major challenge. They can proliferate swiftly before being remedied or discovered, owing to the ease with which data can be obtained on numerous digital channels. On the web, the propagation of erroneous data can affect people's behavior and cause a social crisis [9]. Breaking news about recent issues occurs frequently on social media, and the content focuses on different subjects. Some rumors are made to look like news in order to serve several agendas, some of which are immoral. People have a hard time discerning between true and false rumors.

Understanding how to identify rumors in online communities has been the subject of numerous studies on rumor identification models. These models utilize a wide variety of machine learning techniques, various methods such as Random Forest (RF) and Support Vector Machine (SVM), have been developed for mining rumor-related material and context features [10, 11]. Syntactic, lexical, and semantic details are all included in the content attributes, while structural details are part of the context attributes. In many of the most popular models, the global structural qualities shared by the first tweet and the replies are disregarded. Learning is further influenced by the content and local structural features. The Source-Replies (SR) relation graph, developed by Bai et al. [12], is an attempt to solve these problems by exclusively using global structural traits and content data. The rumor's origin was tracked down by constructing an EGCN loaded with NPAM from the SR-graphs. This methodology was taught to identify rumors by analyzing specific textual, regional, and global structural features. In addition, the dimensions of the word vectors were tuned to achieve acceptable efficiency. Standard word-embedding was used to train the word vectors; however, this method struggles to make sense of big Twitter datasets. For massive Twitter datasets, an unsupervised word-embedding method needs to be created. The T-EGCN framework [13] was proposed for this function; it uses word embedding, a technique based on unsupervised learning, to detect rumors in large Twitter datasets. The model incorporated the latent contextual semantic relationship between terms in tweets and the statistical properties of co-occurrence among them. Tweet rumor attribute vectors were generated by combining word embeddings with GloVe model word attribute vectors, Twitter-specific attributes, and n-gram attributes. The EGCN also utilized this characteristic vector to classify rumors within a massive Twitter data set. However, the recognition performance may be affected by the fact that responses to each tweet change over time. This problem can be solved by handling the time-series variation of responses and learning the social-temporal contexts of source tweets.

This research proposes the TT-EGCN model for rumor recognition, which accounts for the time-varying nature of answers and uses an LSTM network to learn the social-temporal contexts of tweets from their sources. In this model, the T-CNN, stacked LSTM, and GCN are combined to independently learn the textual attributes, social-temporal contexts, and global structural attributes of the given source tweets. This learnt representation can be used to simulate how rumors spread in their formative phases. After these representations have been learned, a hierarchical attention network is employed to train an attentive context embedding from a variety of context inputs simultaneously. Finally, the softmax function is used to classify rumor and non-rumor tweets. Thus, this model can improve recognition accuracy by considering both textual and contextual attributes.

The rest of the manuscript is laid out as follows: Section II provides the works that are linked to the rumor recognition frameworks. Section III provides background information on the TT-EGCN model, whereas Section IV presents the obtained results. The study's potential ramifications are discussed in the final section (V).

II. LITERATURE SURVEY

Yu et al. [14] created the GCN model that represents the pattern of rumor dissemination using the graph convolution operator to adjust the node vectors and recognize the rumors. Both stable and changing characteristics of the real-world social media corpus were incorporated into this model. The attribute merging and pooling units also underwent adjustments to improve precision. But if there isn't enough data for training, it will be even less effective.

Wu et al. [15] designed a Propagation Graph Neural Network (PGNN) model to detect the rumor and generate reliable interpretations for each PG node. In this paradigm, updates to node interpretations were made in real time by sharing information through surrounding nodes' association pathways. Also accepted were PGNN-based methods for global embedding (GLO-PGNN) and ensemble learning (ENS-PGNN). The weight of the node was also dynamically adjusted using an attention policy. On the other hand, it has a high level of complexity to adjust the parameters because there are more learning variables.

An autonomous creation methodology was created by Wang et al. [16] to build a sentiment dictionary that discovers the nuanced human emotional responses to events. The time-span distribution data from microblog events was then kept using a 2-stage dynamic time series method. A new 2-layer Cascaded Gated Recurrent Unit (CGRU) framework for identifying rumor events was developed on the basis of these principles. However, the kinds of emotions expressed by rumors and by occurrences that were not rumors varied significantly.

Asghar et al. [17] studied the rumor identification problem by looking into several deep learners, with a focus on models that account for context while forwarding and inverting passage instructions. Initially, they utilized Bidirectional LSTM (BiLSTM) to train the persistent dependence in a tweet by factoring in the preceding context as well as potential background information. The tweet was then mined for attributes using CNN to determine whether it was a rumor or not. However, it is less effective because it does not take into account other properties than text.

In order to detect rumors with fewer false positives, Sicilia et al. [18] suggested a new feature selection method that uses a representation of the feature space to narrow down sample configurations. Initially, the topology scheme was developed to provide knowledge about sample scattering. Then, the feature selection scheme was adopted to use these privileged data about samples in the feature space to recognize the descriptors, which creates better classifier decision limits. But it needs to analyze the sensitivity by optimizing the weights in the analysis criteria. Using deep representation training, Asghar et al. [17] studied the rumor identification problem by looking into several deep learners, with a focus on models that account for context while forwarding and inverting passage instructions. Initially, they utilized Bidirectional LSTM (BiLSTM) to train the persistent dependence in a tweet by factoring in the preceding context as well as potential background information. However, there wasn't enough data, and some user profiles had been deleted.

Nguyen et al. [20] developed a Just-in-time rumor recognition model called JUDO, which was constructed on top of the continuous scoring of rumor-related signals. In this model, the social graphs were treated as a data stream, and the anomaly score of possible rumors was determined at both the element and subgraph levels. But, the drawback was addressed when a user was only interested in rumors regarding a particular topic so the efficiency was limited.

Tu et al. [21] designed the Rumor2vec model with combined text and propagation structure representation training. Initially, the principle of the union graph was presented to integrate the propagation structures of each tweet, which solves the sparsity problem. After that, the network embedding was leveraged to train interpretations of nodes in the union graph. Also, a model was adopted to learn and recognize the rumor interpretations. But, it needs to analyze the temporal patterns of rumors' propagation structures and so mine highly useful attributes.

In order to process the post-content of rumor events in their nascent stages of dissemination, Luo et al. [22] created a new post-based enrichment interpretation approach called the Backward Compression Mapping approach (BCMM). The BCMM was integrated with the GRU to define post content, topological network of posts, and metadata mined from post corpora. As well, a 3-layer GRU was used to enhance the database's interpretation one hour after a social media event occurred. But, it has high computational complexity.

A lightweight Propagation Path Aggregating (PPA) neural network model was created by Zhang et al. [23] to aid in the embedding and classification of rumors. All rumor architectures were described in this way, each as a distinct set of propagation channels that define the original post in different conversational settings. Then, the information from all the different paths was combined to derive an explanation for the overall structure of the propagation. To locate stance patterns that are not affected by external events, a neural topic model with a Wasserstein Autoencoder (WAE) structure was also employed. However, it requires additional attributes such as user attributes to improve accuracy.

A powerful model for detecting rumors in tweets was developed by Ali and Malik [24]. In the first place, features were extracted using word2vec embedding and the Bidirectional Encoder Representations from the Transformers (BERT) technique. Then, the most relevant features were chosen and given to the different machine learning algorithms to classify tweets as rumor or non-rumor. But it needs additional social and semantic features to increase the model's accuracy. Kumar et al. [25] presented a hybrid model for rumor classification using CNN and an Information Gain-Ant Colony Optimized (IG-ACO) Naive Bayes (NB) classifier. First, the textual attributes were learned by the CNN, which were then merged with the optimized attribute vector created by the IG-ACO. After that, the resultant optimized vector was utilized to train the NB classifier to classify the rumor events. But it utilized only the textual attributes, whereas the context attributes of tweets can be learned independently to construct a robust model.

III. PROPOSED METHODOLOGY

This section provides a quick summary of the TT-EGCN paradigm for rumor recognition. Initially, a large-scale Twitter corpus involving more tweets and their reactions is acquired. In contrast to the GCN, which uses the SR-graph between a source tweet and its reply to learn high-level features [13], the T-CNN extracts and learns features such as n-gram attributes, structural attributes, and twitter-specific attributes. In addition, a stacked LSTM network is utilized to learn the social-temporal settings of the source tweets, modeling the dynamics of rumor transmission during the formative stages of an epidemic. Then, a hierarchical attention network with the NPAM is employed to cooperatively learn attentive context embedding across many context inputs and create an ensemble network model for different topics. Moreover, a softmax function is used to classify the learned feature representations as rumor or non-rumor tweets.

3.1 Preliminaries

Early on in the development of an event, a rumor statement is typically derived from a candidate tweet x_i at time t_i, which can be viewed as a possible rumor event's source. Each of the i candidate tweets in the set $X = \{x_1, \dots, x_n\}$, where $x_i = \{[CC_i, CM_i], t_i\}, x_i \in X$ has two connected responses CC_i and CM_ithroughout the time series t_i, is a candidate tweet. Take j to be the total number of replies to all tweets used as input sources. Here, $CC_i = \{cc_{i,0}, cc_{i,1}, \dots, cc_{i,j}\}$is a collection of time-ordered responses to the content of the current context, and $CM_i = \{cm_{i,0}, cm_{i,1}, \dots, cm_{i,j}\}$ is a collection of time-ordered responses to the metadata of the current context (an SR-graph). Let's pretend that the set of binary labels is represented by $y = \{0,1\}$. The goal is to find the most likely label, based on source tweet content and all context sub-events CC_i and CM_i, for all candidate source tweets x_i at a given time $t_i \subseteq [0, j]$. If rumor x_i is true, $y_i = 1$, and if not, it is 0.

3.2 Overview of Model Structure

Fig 1 depicts the whole architecture of the proposed TT-EGCN model. In this research, we construct a neural network model that takes as input and output predictions $\hat{y}_i$ the source tweets x_i and their associated contexts (CC_i and CM_i). This TT-EGCN model encompasses five major modules: (i) T-CNN, (ii) GCN, (iii) stacked LSTM network, (iv) hierarchical attention models, and (v) softmax classifier.

The rumor recognition using the TT-EGCN model follows different processes. In this step, they preprocess tweets from potential sources X along with their associated context inputs (CC_i and CM_i), raw tweet source content is used to extract the tweet-related attributes, whereas the SR-graph (i.e., CM_i) is used to learn the high-level structural characteristics using the GCN. Also, the raw source context content is fed to the stacked LSTM network and hierarchical attention models for contextural modeling. A stacked LSTM is built from many LSTM networks and takes into account input representations (i.e., CC_i) in a specific order.

Consider the number of layers is L, so L-layer LSTMs are used to process the raw context data. The recurrent structure learns characteristics of sequential data and utilizes soft hierarchical attention models (initial attention layer) to generate an optimal representation. The contextual embedding from the LSTM and GCN layer outputs (i.e., H_{CC}^i and H_{CM}^i) are temporally fused to create a joint representation (H_C^i).

After applying the second attention layer and layer normalization (masked), the joint sequential embedding H_C^i yields a compact representation of context sequences V_C^i. Finally, in the classification layer (softmax function), these various embeddings of source content and context utilizing NPAM are used to produce the final rumor source representation. This is the final layer of output that gives the result of rumor recognition. The entire network is fine-tuned based on the calculated cross-entropy loss.

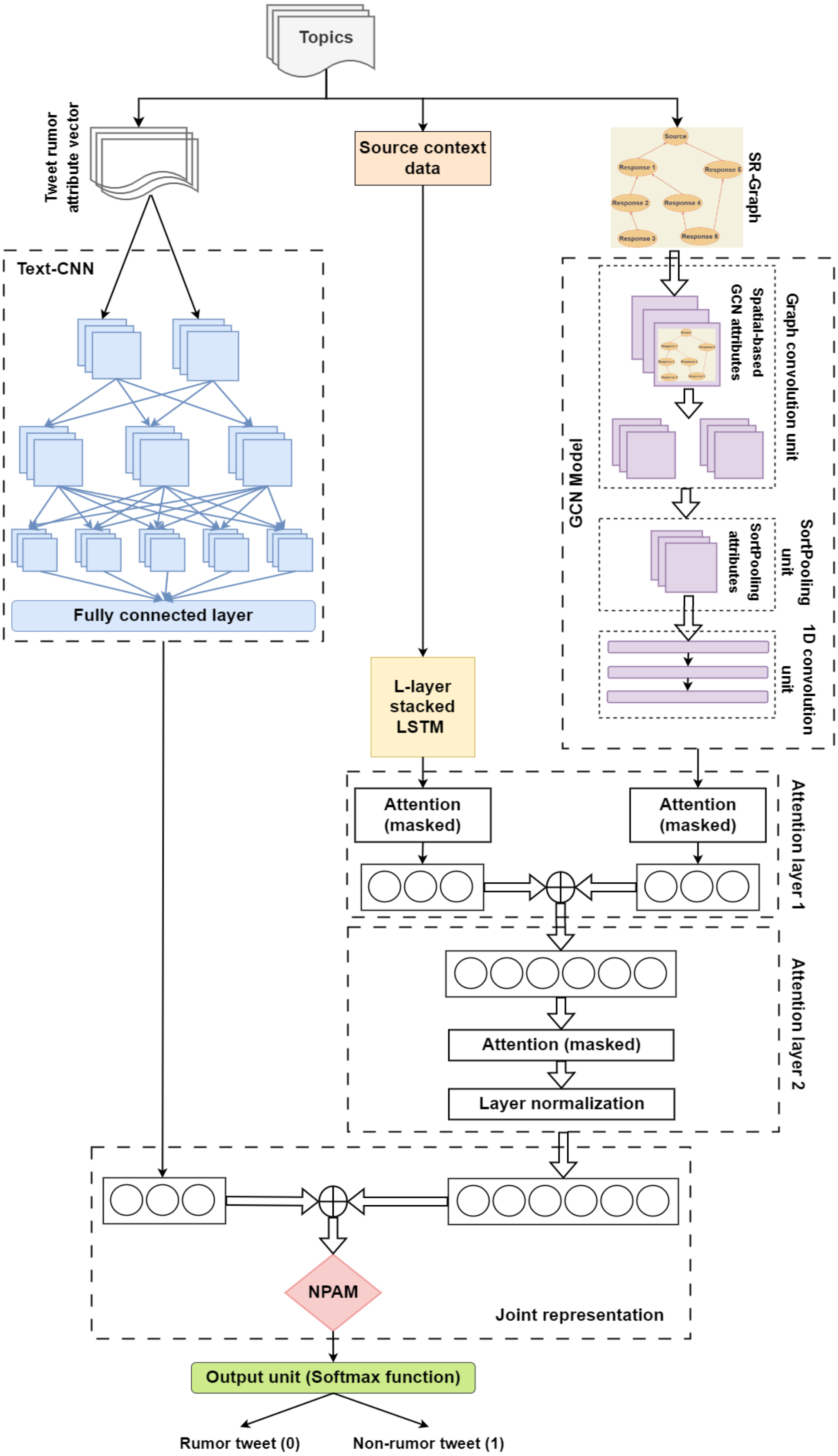

Figure 1. Structure of TT-EGCN Model for Rumor Recognition

3.3 Stacked LSTM Network Module with Hierarchical Attention

The main aim is to adopt LSTM for modeling rumor context. The LSTM processes a sequential input effectively. Where h_t denotes the hidden state at t and W denotes the weights of the LSTM network, it applies the operation $h_t = f_W(x_t, h_{t-1})$ to each potential tweet context (x_t) in a series. Each time t passes, the current concealed state is affected by the previous hidden state. Therefore, the timing of the context input for a reaction based on a time series is crucial. This method enables LSTMs to predict the dissemination patterns of public responses to all source claims and their associated metadata. Furthermore, it accepts inputs of varying durations.

A simultaneous context embedding is adapted to process two linked context inputs, and a 2-layer of forward LSTM network is used to learn more abstract features across multiple context data. The embedded context content, denoted by the notation F_{CC}^{i}, is fed into two forward LSTM layers. The following formula can be used to determine the output state of the context H_{CC}^{i} at time t: $(\overrightarrow{h_{CC,t}^{i}} = \overrightarrow{LSTM_l}\left(\overrightarrow{h_{CC,t-1}^{i}}, v_{CC,t}^{i}\right), \forall t \in [0, j]$.

An LSTM hierarchy is then built using the shallow characteristics collected from the overt data in social responses. This model is able to learn the complicated hierarchical socio-temporal structure's latent behavioral and social dynamics. State H_{CM}^{i} of the context output is defined as follows: $\overrightarrow{h_{CC,t}^{i}} = \overrightarrow{GCN}\left(\overrightarrow{h_{CC,t-1}^{i}}, v_{CC,t}^{i}\right), \forall t \in [0, j]$. Further, to strengthen the contribution of significant context elements and remove unwanted details in the final representation, a hierarchical attention strategy is introduced in this model. This approach can concentrate on the most important details since it uses attention throughout many stages. In order to improve recognition performance, the attention method is used in conjunction with multi-level context embedding to filter out extraneous data and collect more detailed information about the context in which the target object is being used.

By specifying each t for the context embedding layers, attention weights may be computed. Input is a context sequence of length j, and it uses attention methods to learn a mapping between the input and the output sequence. Using a probabilistic distribution over inputs representing the conversation's context as a function of time, the end-to-end rumor recognition model may estimate where the focus should be placed. Normalized probability distribution of significance throughout the complete context is approximated using the usual softmax function. H_C is the context of tweets that is often kept secret.

$$\alpha_C^t = softmax(\tanh(W_h h_C^t + b_h)), \forall t \in [0, j] \tag{1}$$

$$h_{C\_new}^t = \alpha_C^t h_C^t \tag{2}$$

The weights of the attention layer, denoted by W_h and b_h in Eqns. (1) and (2), are initially randomized before being trained to optimal values. Variable lengths can also be handled with zero padding. The padded values are hidden behind a negative infinity float, just like the stacked LSTM layer does. The new context embedding, denoted by h_{C_new}, has been reweighted.

In the TT-EGCN architecture, the stacked LSTM layers and the joint representation unit receive the attention approach. In the first attention layer, H_{CC}^t and H_{CM}^t are the outputs of the LSTM and the GCN context, respectively (see Eqns. (3) and (4)). The trained model consists of two attention layers and two recurrent layers with independently trained hidden states.

$H_{CC\_new}^t$ and $H_{CM\_new}^t$ are the outputs of the two attention models, respectively. The combined representation from the two context networks is then input to the second attention layer, together with the weighted hidden state vectors for each t.

$$H_{CC\_new}^t = attention_1(H_{CC}^t) \tag{3}$$

$$H_{CM\_new}^t = attention_1(H_{CM}^t) \tag{4}$$

To determine the inference correlation between two related context embedding's, the attention model is used as a composition layer to integrate the two varieties of sub-event inference data. The second attention layer is distinct from the first in that it uses a weighted sum to integrate hidden states based on their relative significance, with the goal of learning shared semantics between content and responses. This unifies the representation of replies and distribution patterns, which enhances performance on the rumor identification problem.

$$h_C^t = attention_2\left(h_{CC\_new}^t \oplus H_{CM\_new}^t\right) \quad (5)$$

$$v_c = \sum_t h_C^t \quad (6)$$

The context hidden states, h_C^t, are denoted in Eqns. (5) & (6), and the final context vector, v_c, is defined as the sum of h_C^t for all times t. The softmax algorithm is then used to distinguish between rumor and non-rumor tweets based on this aggregated feature representation.

Algorithm for TT-EGCN-based Rumor Recognition

Input: PHEME dataset

Output: Rumor and non-rumor events

1.**Begin**
2.Divide the PHEME dataset into training and test sets;
3.$\boldsymbol{for}(each\ instances\ in\ the\ training\ set)$
4. Various textual features, including n-grams, Twitter-specific features, grammar, and the semantic content of tweets, can be extracted.
5. GloVe and Word2Vec may be trained to produce accurate word vectors.
6. Train the T-CNN model by fusing word vectors to obtain the unified tweet rumor attribute vector;
7. Create the SR-graph for use in training a GCN model capable of extracting global-level structural features.
8. Train the stacked LSTM network using the source context data;
9. Apply a hierarchical attention strategy to learn social-temporal contexts of the source tweets;
10. Merge all text-based and context-based attributes by the NPAM to get a joint representation;
11. Train the softmax function;
12.$\boldsymbol{end\ for}$
13.$\boldsymbol{for}(each\ instances\ in\ the\ test\ set)$
14. Apply the trained model;
15. Classify each tweet instance as either rumor or non-rumor event;
16. Validate the model performance;
17.$\boldsymbol{end\ for}$
18.**End**

IV. EXPERIMENTAL RESULT

Here, the TT-EGCN model is run in Java to evaluate how well it performs. Its effectiveness is measured against that of the EGCN [12], the T-EGCN [13], the GLO-PGNN [15], and the CNN-IG-ACO-NB [25] models already in use. The Text-CNN and GCN hyperparameters are set up as described in [13]. The analysis compares results using accuracy, precision, recall, and the f-measure.

Two LSTM forward layers are used in a stacked network, and the hidden unit size is configured to be twice as large as the input size. A 1e^{-4} learning rate and a 1e^{-5} weight decay are employed. Each training instance and its associated context inputs are cycled through in each and every epoch, with a batch size of 125. Overfitting can be prevented by limiting the number of iterations to 10.

4.1 Dataset Description

The enlarged PHEME corpus [26] is taken into account for this experiment, which includes both rumor and non-rumor tweets sent out during the broadcast of breaking news. Each rumor has a label indicating whether it is true, false, or unverified in relation to one of the nine instances. Numbers and descriptions of incidents are listed in Table 1.

Table 1. Details about PHEME Corpus

Incidents	#Rumors	#Non-rumors	#Threads	#Tweets
Ferguson	284	859	1143	24175
Sydney siege	522	699	1221	23996
Charlie Hebdo	458	1621	2079	38268
Ottawa shooting	470	420	890	12284
Germanwings-crash	238	231	469	4489
Putin missing	126	112	238	832
Prince Toronto	229	4	233	902
Gurlitt	61	77	138	179
Ebola Essien	14	0	14	226
Sum	2402	4023	6425	105354

This TT-EGCN model is tested by the five foremost incidents, which produce a data-balanced corpus. The five most notable incidents are Charlie Hebdo, Ferguson, Gencrash, the Ottawa shooting, and the Sydney siege. This means that there are a total of 5802 annotated tweets in the final corpus, with 1972 rumor tweets and 3830 non-rumor tweets. 75% of these are used for instructional purposes, while the remaining 25% are put to the test.

4.2 Performance Metrics

- **Accuracy:** It is the ratio of perfect recognitions over the sum number of tweets tested.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{TP + TN + False\ Positive\ (FP) + False\ Negative\ (FN)} \quad (7)$$

In this situation, the sum of tweets that were accurately recognized (TP) and false negatives (FN) are the same thing. Similarly, the fractional and total numbers of misrecognized tweets (FP and TN).

- **Precision:** It's the ratio of fully recognized tweets within a rumor label to all tweets within the label that have been acknowledged.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

- **Recall:** It's the fraction of tweets containing perfectly identifiable rumors.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

- **F-measure:** It represents an equilibrium point between precision and recall.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

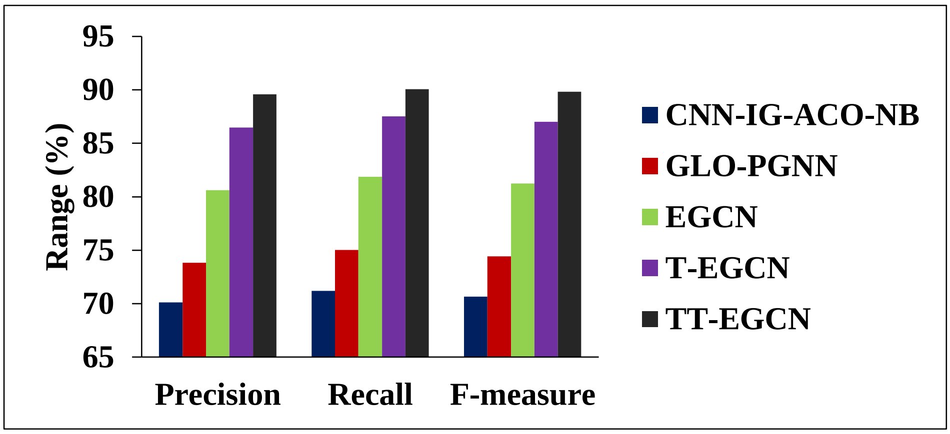

Figure 2. Comparison of TT-EGCN with Existing Rumor Recognition Models in terms of Precision, Recall, and F-measure

Fig 2 is a scatter plot showing the accuracy, recall, and f-measure of the models on the PHEME dataset. Based on the results, it is clear that the TT-EGCN model, which learns both textual and context characteristics from the source tweets, achieves better recognition performance than the other existing models. The TT-EGCN increases the precision by 27.8%, 21.3%, 11.1%, and 3.6% compared to the CNN-IG-ACO-NB, GLO-PGNN, EGCN, and T-EGCN models, respectively. The recall of the TT-EGCN model is maximized by 26.5%, 20%, 10%, and 2.9% in contrast with the CNN-IG-ACO-NB, GLO-PGNN, EGCN, and T-EGCN, correspondingly. Similarly, the TT-EGCN enhances the f-measure values by 27.1%, 20.7%, 10.5%, and 3.2% compared to the CNN-IG-ACO-NB, GLO-PGNN, EGCN, and T-EGCN, respectively.

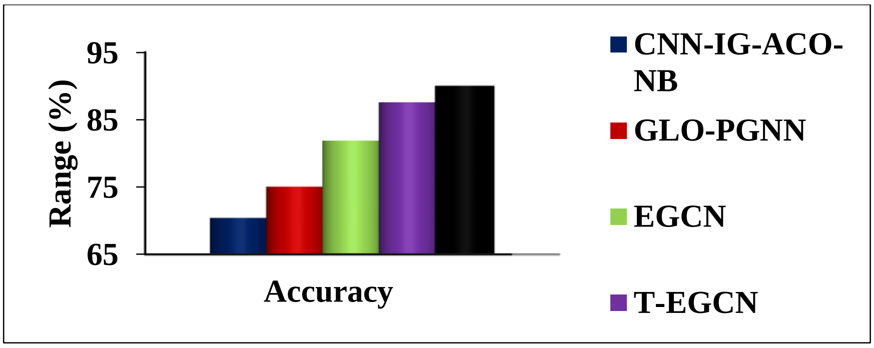

Figure 3. Comparison of Accuracy for TT-EGCN and Existing Models

Fig 3 plots the accuracy of the TT-EGCN and other existing rumor recognition models on the PHEME dataset. It is observed that the TT-EGCN model achieves better accuracy than all other models for rumor recognition by efficiently learning the text-based attributes and social-temporal contexts of the source tweets. The TT-EGCN model maximizes the accuracy by 27.9%, 20%, 10%, and 2.8% compared to the CNN-IG-ACO-NB, GLO-PGNN, EGCN, and T-EGCN models, respectively.

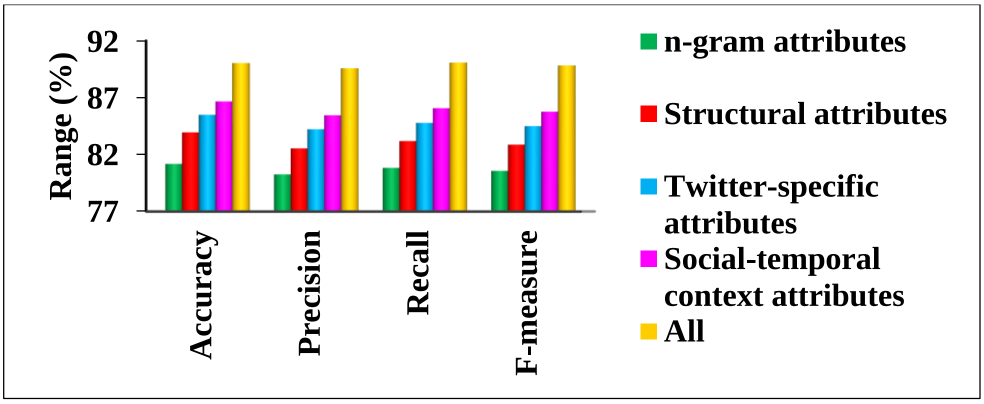

Figure 4. Comparison of TT-EGCN Model for Different Tweet Characteristics Vectors

Fig 4 shows how the TT-EGCN model for rumor detection learns many source tweet attributes simultaneously. It is noted that the TT-EGCN increases the recognition performance by capturing and learning both text- and context-based characteristics from the given tweet data compared to learning independent characteristics. Since text and context information surrounding the tweets and their answers are so important, the TT-EGCN model is able to achieve higher levels of accuracy, precision, recall, and f-measure than the competition.

V. CONCLUSION

For rumor recognition, the TT-EGCN model was introduced here, which can learn both the social-temporal and linguistic aspects of the source tweets. The social-temporal contexts of the given source tweets were captured by the stacked LSTM networks, which can model the rumor propagation patterns for better recognition. Also, the hierarchical attention strategy layer was applied to merge the textual and contextual features from the T-CNN, stacked LSTM layers, and GCN to create a joint feature representation. This final representation was classified by the softmax function to recognize rumor or non-rumor events. The TT-EGCN model using the PHEME dataset has an accuracy of 90.02%, which is 14.4% better than all other known models for efficient rumor recognition, as demonstrated by the experimentation described in the conclusion.

REFERENCES

[1]. L. Wu, N. A. Dodoo, T. J. Wen, and L. Ke, Understanding Twitter conversations about artificial intelligence in advertising based on natural language processing. International Journal of Advertising, 41(4), 685-702.

[2]. M. A. Al-Garadi, S. Kim, Y. Guo, E. Warren, Y. C. Yang, S. Lakamana, and A. Sarker, Natural language model for automatic identification of intimate partner violence reports from twitter. Array, 15, 2022, 1-8.

[3]. N. Rani, P. Das andA. K. Bhardwaj, A. K, Rumor, misinformation among web: a contemporary review of rumor detection techniques during different web waves. Concurrency and Computation: Practice and Experience, 34(1), 2022, 1-21.

[4]. L. C. Passaro, A. Bondielli, P. Dell'Oglio, A. Lenciand F. Marcelloni, In-context annotation of topic-oriented datasets of fake news: a case study on the notre-dame fire event. Information Sciences, 615, 2022, 657-677.

[5]. A. Galli, E. Masciari, V. Moscato and G. Sperlí, A comprehensive benchmark for fake news detection. Journal of Intelligent Information Systems, 59(1), 2022, 237-261.

[6]. F. Giardini, D. Balliet, E. A. Power, S. Számadóand K. Takács, Four puzzles of reputation-based cooperation: content, process, honesty, and structure. Human Nature, 33(1), 2022, 43-61.

[7]. K. Eismann, Diffusion and persistence of false rumors in social media networks: implications of searchability on rumor self-correction on Twitter. Journal of Business Economics, 91(9), 2021, 1299-1329.

[8]. C. Melchior and M. Oliveira, Health-related fake news on social media platforms: a systematic literature review. New Media & Society, 24(6), 2022, 1500-1522.

[9]. T. Buchanan, Why do people spread false information online? The effects of message and viewer characteristics on self-reported likelihood of sharing social media disinformation. Plos One, 15(10), 2020, 1-33.

[10]. P. Meeland D. K. Vishwakarma, Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. Expert Systems with Applications, 153, 2020, 1-52.

[11]. A. R. Pathak, A. Mahajan, K. Singh, A. Patiland A. Nair, Analysis of techniques for rumor detection in social media. Procedia Computer Science, 167, 2020, 2286-2296.

[12]. N. Bai, F. Meng, X. Ruiand Z. Wang, Rumour Detection Based on Graph Convolutional Neural Net. IEEE Access, 9, 2021, 21686-21693.

[13]. D. Rajasekar and L. Robert, Unsupervised word embedding with ensemble deep learning for twitter rumor identification. Revue d'Intelligence Artificielle, 36(5), 2022, 769-776.

[14]. K. Yu, H. Jiang, T. Li, S. Han and X. Wu, Data fusion oriented graph convolution network model for rumor detection. IEEE Transactions on Network and Service Management, 17(4), 2020, 2171-2181.

[15]. Z. Wu, D. Pi, J. Chen, M. Xie and J. Cao, Rumor detection based on propagation graph neural network with attention mechanism. Expert Systems with Applications, 158, 2020, 1-31.

[16]. Z. Wangand Y. Guo, Rumor events detection enhanced by encoding sentimental information into time series division and word representations. Neurocomputing, 397, 2020, 224-243.

[17]. M. Z. Asghar, A. Habib, A. Habib, A. Khan, R. Aliand A. Khattak, Exploring deep neural networks for rumor detection. Journal of Ambient Intelligence and Humanized Computing, 12(4), 2021, 4315-4333.

[18]. R. Sicilia, M. Merone, R. Valenti and P. Soda, Rule-based space characterization for rumour detection in health. Engineering Applications of Artificial Intelligence, 105, 2021, 1-14.

[19]. X. Chen, F. Zhou, F. Zhang, and M. Bonsangue, Catch me if you can: A participant-level rumor detection framework via fine-grained user representation learning. Information Processing & Management, 58(5), 2021, 1-19.

[20]. T. T. Nguyen, T. T. Nguyen, T. T. Nguyen, B. Vo, J. Joand Q. V. H. Nguyen, Judo: Just-in-time rumour detection in streaming social platforms. Information Sciences, 570, 2021, 70-93.

[21]. K. Tu, C. Chen, C. Hou, J. Yuan, J. Liand X. Yuan, Rumor2vec: A rumor detection framework with joint text and propagation structure representation learning. Information Sciences, 560, 2021, 137-151.

[22]. Y. Luo, J. Maand C. K. Yeo, BCMM: A novel post-based augmentation representation for early rumour detection on social media. Pattern Recognition, 113, 2021, 1-12.

[23]. P. Zhang, H. Ran, C. Jia, X. Liand X. Han, A lightweight propagation path aggregating network with neural topic model for rumor detection. Neurocomputing, 458, 2021, 468-477.

[24]. G. Ali and M. S. I. Malik, Rumour identification on Twitter as a function of novel textual and language-context features. Multimedia Tools and Applications, 82(5), 2023, 7017-7038.
[25]. A. Kumar, M. P. S. Bhatiaand S. R. Sangwan, Rumour detection using deep learning and filter-wrapper feature selection in benchmark twitter dataset. Multimedia Tools and Applications, 81(24), 2022, 34615-34632.
[26]. https://figshare.com/articles/dataset/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078.