International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Rumor Detection in Social Media Using Word Embedding and LSTM-Based Networks

Aakash Kumar Yadav
Department of CSE with specialization in Internet of Things
School of Computing
Faculty of Engineering and Technology
SRM Institute of Science and Technology
Ramapuram-600089, India
ay1442@srmist.edu.in

S. Veena
Professor
Department of CSE - Internet of Things & CSBS
School of Computing
Faculty of Engineering and Technology
SRM Institute of Science and Technology
Ramapuram-600089, India
veenas1@srmist.edu.in

R Harisankar
Department of CSE with specialization in Internet of Things
School of Computing
Faculty of Engineering and Technology
SRM Institute of Science and Technology
Ramapuram-600089, India
rh2403@srmist.edu.in

P Dhanraj
Department of CSE with specialization in Internet of Things
School of Computing
Faculty of Engineering and Technology
SRM Institute of Science and Technology
Ramapuram-600089, India
dp8328@srmist.edu.in

Abstract— Rumor detection means classifying social media posts from Facebook, Twitter or Instagram into True Rumor, False rumor, Non-Rumor, Unverified categories. The traditional machine learning models uses basic handcrafted features for the classification and predictions; it fails to capture the contextual and sequential nature of the news data. Hence these limitations of traditional models are studied in this system and an efficient deep learning approach is proposed which can understand the contextual meaning of the data for efficient rumor classification. In this approach, textual data is first preprocessed through cleaning, tokenization into dense vector representations using Word2Vec word embedding technique. The LSTM network model processes these sequences to capture the contextual dependencies within the text. The final word embedding data is passed through dense layer for multi-class classification. Experimental evaluation showcase that the proposed deep learning LSTM model achieves reliable performance in identifying the fake rumor data, particularly in early stages. This approach provides a scalable and efficient solution for real-time rumor detection without relying on external metadata.

Keywords— ***Rumor Detection, Word Embeddings, Long Short-Term Memory (LSTM) model, Word2Vec word embedding, Deep Learning, Text Classification, Social Media Analysis, Sequence Modeling, Tokenization.***

I. INTRODUCTION

With the raise in social media, the true news helped in spreading the correct information across the world but also increase of fake rumor news became a challenge very quickly causing panic, confusion and causing serious social problems for the public.

The raise in various social media platforms like Twitter, Facebook and Instagram are sharing huge amounts of information is shared every second, so detecting fake information has become a very important and majorly concerning issue. Rumor detection is typically treated as a classification problem, where social media posts or data are categorized into True rumor, False rumor, non rumor and Unverified classes. This classification of news is hard to be done via traditional machine learning models with handcrafted features, which often fails to capture the contextual and sequential nature of the news data. Hence these limitations of traditional models are studied in this system and a suitable deep learning LSTM (Long Short-Term Memory) network model is used and tested for the sample real-world news data.

This system uses Word2Vec embedding technique for creating contextual meanings of the sentences and maintain the word order with their sequence of occurrence. After the data is embedded and transformed, it is used to train the deep learning LSTM model for the sequence learning of the sentence and to improve the accuracy of the prediction.

The advantage of the proposed approach is that it provides a better word meaning and sentence flow ultimately improving the classification accuracy. This works well for early detection of the fake rumor and since it doesn't depend upon the user data or any external metadata, which makes it highly scalable, simple and efficient for the real-life application.

II. LITERATURE SURVEY

Rumor detection is a critical issue in this modern social media world. The spread of misinformation is causing major mental issues, panic and untrust over the government. Various research has already been conducted to filter out the fake rumors using techniques like, Deep learning model, Graph-based models, transformers etc. These researches are divided into four parts [i] Early detection models, [ii] Semantic & stance-based model [iii] Structural model [iv] Graph-based semantic model. Different methods focus on different aspects of rumor detection.

In the early rumor detection models research of Shu et al. [1] has used transformers and reinforcement learning for early detection. Zhu et al. [3] has combined semantic, stance and temporal features for the early rumor classification. However, these approaches are generally complex and relies on extra user data making it less scalable.

In Semantic based approaches Mahbub et al. [4] has used psycho-linguistics features whereas other works [8] focused on understanding the word meaning (semantics) and based upon the understanding of data classifying them into categories. However, this method of classification focusses only on the features, ignoring the sequence of the text by not following the order of the word, making it un-reliable.

In Structural and Graph-based models Luo et al. [5] has used the combination of words with propagation structure for the mapping of the words in a graph-like structure for the easy propagation and word building. Other works like [9], [2] have also used graphs and spreading patterns. However, these methods become highly complex and ignores the sequential learning missing the text flow.

Despite all these advancements in the existing researches there still remains some limitations with the models or sometimes making it too complex, reducing its scalability. Most of these models focus only on one aspect (semantic / structure) ignoring the sequence of the texts and many other methods are either complex, need extra data or not suitable for real-time usage in applications. The existing methods are powerful but not simple or efficient.

After the observation of the existing researches, we discovered the need of balanced performance and simple model with good sequential learning and early detection of rumor data. The proposed solution used Word2Vec embedding technique for preserving the word meaning and LSTM (long short-term memory) model for the sequential learning of the data. This way it captures the both meaning and word order with less complexity and highly scalable in real life.

TABLE 1: Comparative Analysis of Existing Research and Identified Gaps

Study Category	Semantic Modelin	Structural Modeling	Sequential Modeling	Early Detection	Model Complexity
Early Detection Models [1]	Yes	Partial	Partial	Yes	High
Semantic + Stance Models [3]	Yes	Yes	Yes	Yes	High
Psycho-linguistic Models [4]	Yes	No	No	No	Medium
Graph-Based Semantic Models [9]	Yes	Yes	No	No	High
Proposed System	**Yes**	**No**	**Yes**	**Yes**	**Low**

Table 1 highlights the limitations of existing rumor detection researches which has clearly used the complex models and less scalable. The proposed system is less complex, provides better performance and easily scalable in real world.

III. PROPOSED SYSTEM

The proposed method uses a deep learning-based framework for the early rumor detection using words embedding and LSTM networks with dense layers. The overall workflow consists of four key stages: data preprocessing, word embedding, feature extraction, and classification.

A. Data Representation and Preprocessing

Each of the textual data from the social media post is treated as the input dataset. Let the dataset be represented as:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

where x_i denotes the i^{th} post, y_i denotes its corresponding class label. The textual raw data undergoes cleaning, tokenization, padding as part of the data preprocessing. Each sentence is converted into a sequence of tokens:

$$x_i = \{w_1, w_2, \ldots, w_m\}$$

these tokens are then mapped into dense vector layer using the Word2Vec embedding technique:

$$e_i = Embedding(x_i)$$

where e_i represents the embedded sequence.

B. Feature Extraction Using LSTM Network

The embedded data is passed to the LSTM for the sequential learning. The LSTM model maintains hidden states to understand the sequential learning of data.

$$h_t = LSTM(e_t, h_{t-1})$$

where h_t represents the hidden state for the time t.

The LSTM effectively captures the contextual relationship of the text data, enabling the model to understand the meaning of the text.

C. Classification Layer

The final sequential data is passed through the LSTM classification layer for the predication and classification:

$$y = Softmax(W \cdot h + b)$$

where W and b are varying learnable parameters.

To improve the classification separability, a Softmax activation function is used. The classification reduces overlap between rumor categories such as False Rumor, Non-Rumor, and Unverified.

The final prediction output classifies each event into one of four categories:

True Rumor, False Rumor, Non-Rumor, or Unverified.

D. System Integration and Operational Workflow

The system follows a step-by-step pipeline with the main stages as, preprocessing, embedding, sequence modeling, prediction and classification. The input text data is cleaned then converted into sequences before passing into the LSTM. The LSTM extracts feature by understanding the word order and context of the data. The system used only text data without the need of any user data keeping it safe and privacy-friendly. This system can be used across different platforms because of its scalability and flexible design.

E. Computational Complexity and Runtime Performance

The preprocessing of the data works in a linear time complexity depending upon the length of the data since each word is processed only once. Let n denote the maximum sequence length and d denote the embedding dimension. The embedding layer performs a lookup operation with a linear time complexity. The LSTM network model gives the complexity of:

$$O(n \cdot d \cdot h)$$

where h represents the number of hidden units in the LSTM layer. Since LSTM processes input data sequentially, maintaining the state memory of the previous words, making it efficient for capturing the contextual relationship. During training, the overall complexity depends on the number of epochs and training dataset size. The overall model works efficiently for short texts suitable for real-world application.

Algorithm 1: LSTM Model Rumor Detection Algorithm
Input: Dataset D = $\{(x_i, y_i)\}$
Output: Trained model

1: Load the dataset *D*.
2: Text preprocessing (cleaning, tokenization)
3: Convert texts into sequences & apply padding
4: Convert sequences into embeddings
5: Pass embeddings into LSTM model
6: Extract important features
7: Pass through dense layer & apply Softmax activation
8: Train the model using labeled data
9: Save the trained model

During testing phase, the new text is preprocessed and converted into embeddings. The LSTM model predicts the rumor category in real-time.

F. System Architecture

The proposed LSTM based model consist of four major components in its architecture, preprocessing, words embeddings, sequential modelling of words in LSTM, and finally the classification of the data into different classes.
The preprocessing steps consist of normalizing the sentences, removing null or unnecessary words, removing noise, Tokenization (splitting words), padding (same lengths for all inputs), lowercasing etc. all these process helps in the standardizing the text before using it from the LSTM model. These processed data are then converted into numerical vector data via Word2Vec embedding techniques, for the better understanding of the word data and also for understanding the similarities between the words. The LSTM model maintain the state internally for the building of contextual data understanding, relationship by maintain the memory from the previous states. The outputs from the LSTM layer are forwarded to fully connected (dense layers) where it is further processed. At the end, a Softmax activation function is used for the calculation of the probability distribution across the target classes and the class with the highest probability is selected as the predicted label.

The system is designed to operate without relying on external metadata such as user profile data making it highly scalable, privacy-friendly and simple to use in the real-life application over large datasets.

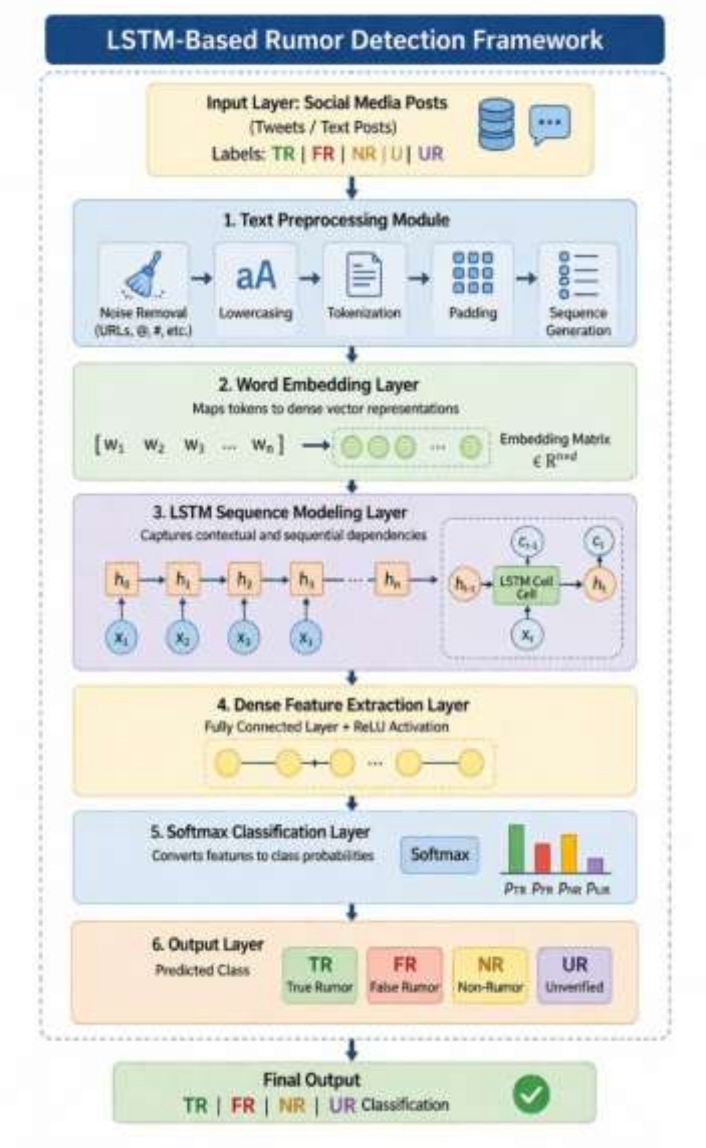

Figure 1. Architecture diagram

Figure 1 shows the complete architecture diagram of the LSTM based Rumor detection system working. The diagram includes the complete data flow pipeline of the system including the preprocessing stage, embedding layer, LSTM model training and classification of data based upon the prediction.

IV. Results And Discussion

The implementation results of the proposed Rumor Detection in Social Media using Word Embeddings and LSTM Networks are presented in this section. The evaluation focuses on textual pattern analysis, comparative model performance, and the effectiveness of deep learning LSTM model for rumor classification. The result is derived from an actual real-life dataset, showcasing how well the model works using real data.

A. Textual Pattern Analysis using WordCloud

For the clear understanding of the textual data, WordCloud representation is used to visualize the frequent words from the dataset. Separate WordCloud are shown for the True rumor and Fake rumor dataset.

Figure 2. WordCloud Representation of True Rumor Data

Figure 2. highlights the WordCloud representation of the Truth factual posts data. The WordCloud showed the most frequently repeated words from the factual text data, which mostly contained the real-world location, events, and factual descriptions. The distribution of words appears more structured and less exaggerated data which is likely the case in Fake Rumor dataset, the true rumor data tends to use contextually consistent and meaningful vocabulary, which is easily captured using word embedding techniques.

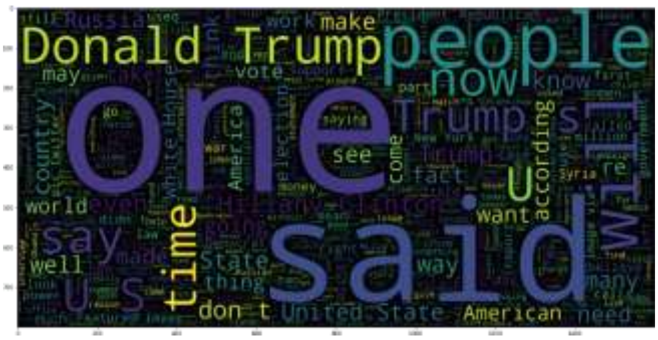

Figure 3. WordCloud Representation of Fake Rumor Data

Figure 3. highlights the WordCloud representation of Fake rumor data which reveals a different linguistic pattern compared to the true rumors. The frequently occurred words from fake rumor data category includes sensational, exaggerated, or emotionally driven terms, which are generally used in fake news for attracting the attention of the readers ultimately misleading them. The vocabulary distribution is more diverse and less structured, inconsistent in content.

This observation validates the use of word embeddings, as it helps in understanding the meaning of the data, making the classification easier for the model. Word2Vec embedding technique is used for converting the English words into numerical numbers, clearing out the un-necessary stop words and helps in capturing the context and relationships of the textual data. This allows the model to better understand the consistency or inconsistency in the vocabulary of the sentence ultimately improving the model prediction and classification performance. Additionally, Word2Vec embedding helps in improving the normalization of the data by preserving contextual information across different sample records of data, enhancing the overall performance of classification models.

B. Machine Learning Model Performance

We used different machine learning models to evaluate the effectiveness of different approaches for rumor detection; multiple machine learning models were trained and tested on the same pre-processed textual dataset which we obtained after the Word2Vec embedding and cleaning of the data. The models we tested for are AdaBoost Classifier, Gradient Boosting Classifier, Random Forest Classifiers and other baseline models. The performance of these models was analysed using the confusion matrix heatmaps for the classification accuracy and error distribution. The results indicated that the ensemble-based models performed better than the traditional simple classifier models due to their ability to capture non-linear relationship in textual data. However, variations in performance are observed depending on how well each model handles semantic complexity.

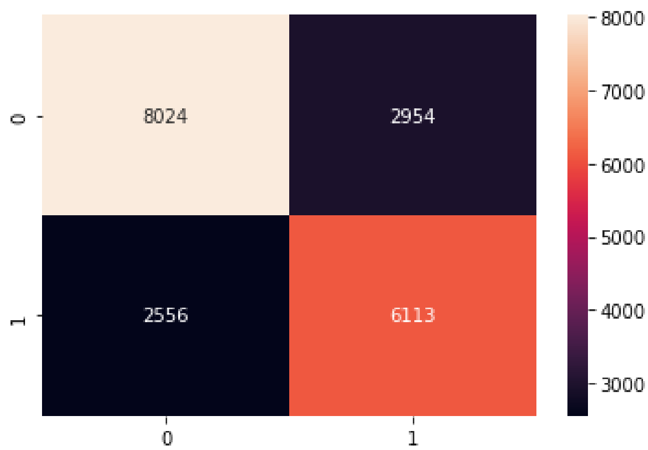

Figure 4. Confusion Matrix of AdaBoost Classifier

The AdaBoost Classifier model worked averagely and showed moderate classification performance. While a reasonable number of data were incorrectly classified (diagonal values), there are noticeable misclassifications between similar category classes. This suggests that AdaBoost has limited capability in capturing complex contextual relationships in textual data.

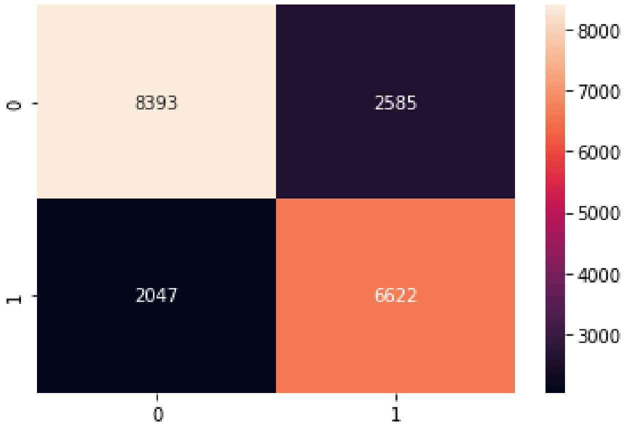

Figure 5. Confusion Matrix of Gradient Boosting Classifier

The Gradient Boosting classifier model performed better than the AdaBoost classifier. It gave more accurate and correct predictions (stronger diagonal in heatmap). However, still some errors happened when classifying the similar classes.

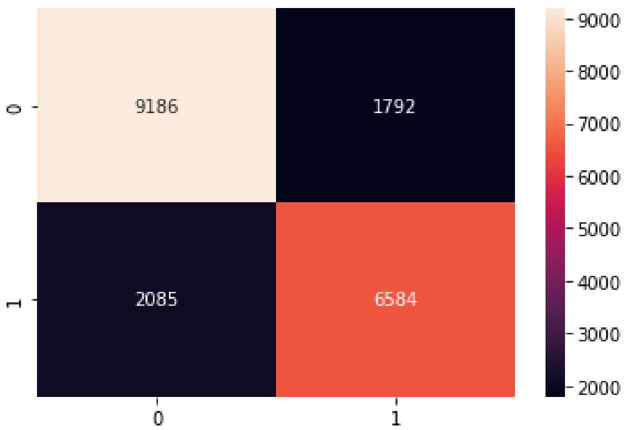

Figure 6. Confusion Matrix of Random Forest Classifier

Among all the model performances Random Forest classifier performed the best with highest number of correct predictions, showing a higher concentration of correctly classified textual data records in the confusion matrix. This is compared based upon the stronger diagonal dominance observed in the heatmap compared to AdaBoost and Gradient Boosting models. Random Forest classifier was better at handling complex and non-linear relationships but this model also had limitations. The rumor classes are very similar in meaning hence it was hard to separate them using only features.

C. Model Optimization & Training Strategy

We have used various effective strategies for the performance improvement of the rumor detection system, such as preprocessing, embedding, and model design strategies. Text data was cleaned using Tokenization strategy, removing the unnecessary words, stop-words, and normalization of the data to reduce noise and improve feature quality of the text data. These processed texts were converted into padded sequence for maintaining the uniform input size. Word2Vec embedding technique was used for capturing the meaning and relationship between the words. These pre-trained embeddings were added to the LSTM model for the better context awareness and understanding of the text data.

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         24084400

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 24201777 (92.32 MB)
Trainable params: 117377 (458.50 KB)
Non-trainable params: 24084400 (91.87 MB)
_________________________________________________________________
```

Figure 7. Architecture of LSTM-based Rumor Detection Model

The Deep learning LSTM model includes embedding layers, followed by a 128-unit LSTM layer also a dense output layer for classification. The LSTM model was later trained using the Adam optimizer for efficient training and binary cross-entropy loss function for the classification.

During training we observed steady improvement in accuracy with minimal overfitting also the validation accuracy remained closer to the training accuracy, proving less overfitting. Additionally, ensemble models such as Random Forest model, AdaBoost model, and Gradient Boosting model were used to provide a baseline comparison between the traditional machine learning models and LSTM model.

D. Deep Learning Model Training & Evaluation

We trained the Long Short-Term Memory (LSTM) model for multiple epochs, with each epoch, the model kept improving. The training accuracy reached ~ 99% and the validation accuracy reached ~ 95-96%. This shows that the model performs well on both training and the new testing data, with gradual increasing its accuracy. With each epoch the loss values decreased steadily, this indicated that the learning was done properly and the model is converging effectively. We used pre-trained Word2Vec embedding technique for the faster training and better understanding of word meaning for the model.

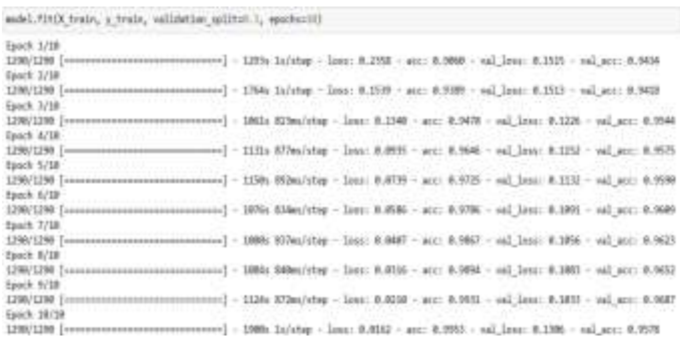

Figure 8. Epoch Training & Performance of LSTM Model

For assurance and validation, we performed a similarity check with the Word2Vec data, this confirms the semantic learning of the model data.

```
w2v_model.wv.most_similar("innovation")
```

```
[('innovative', 0.8034091591835022),
 ('technological', 0.7662263512611389),
 ('sustainable', 0.7536510825157166),
 ('technology', 0.7362650036811829),
 ('entrepreneurship', 0.7077240347862244),
 ('development', 0.6952726244926453),
 ('creativity', 0.6924847364425659),
 ('ecosystem', 0.6850614547729492),
 ('competitiveness', 0.6803798675537109),
 ('robotics', 0.6770955324172974)]
```

Figure 9. Similarity checks using Word2Vec Embeddings

E. Comparative Analysis of Models

We performed the comparison analysis on the traditional machine learning models and out of them the best performance was observed in the Random Forest model, but these ML models have a limitation of judging based upon the manual feature (feature-based learning), they don't understand the word order or sentences. Due to this we switched to the Deep learning model (LSTM) long short-term memory model. LSTM uses word embedding and sequence learning for the better context understanding of the text data, which helped in the improvement of the performance and overall predication.

```
y_pred = (model.predict(X_test) >= 0.5).astype("int")
614/614 [==============================] - 163s 264ms/step
```

```
accuracy_score(y_test, y_pred)
0.9581106530259073
```

```
print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.96      0.97      0.96     10978
           1       0.96      0.94      0.95      8669

    accuracy                           0.96     19647
   macro avg       0.96      0.96      0.96     19647
weighted avg       0.96      0.96      0.96     19647
```

Figure 10. Model Prediction Accuracy, F1 Report

Overall, the machine learning models provide a great baseline performance but have a limited understanding of the context while the LSTM model provides better contextual understanding and performance due to sequence learning.

V. Conclusion and Future Work

We have built a rumor detection system using LSTM (Long-short Term Memory) model. It mainly focuses on analyzing the social media fake new texts and classifying them into different categories using only the text data format. This system basically includes three major workings, the first part includes the data preprocessing, here the raw text data gets converted into proper dataset, by passing them through various null checks, transformations, duplication detection checks etc. Later this data gets converted into numbers (word embeddings) which could be fed into the model directly. Finally, the embedded words are passed through the model to get trained and tested. The LSTM model understands the word orders and context better and helps in creating a meaning from the text data. The system predicts the news text data and classifies them into True Rumor, False Rumor, Non-Rumor, and Unverified Rumor. The major strength of this system is that it keeps the system simple and efficient and tries to make a meaning out of the input texts. It uses text data only to keep less load on the model and maintains the privacy. Finally, the system is working well and accurately for the real-time rumor detection.

Future Work

Advance deep learning models can be used for better and even more accurate understanding of context sequence. Training deep learning model for detecting not just text news, but various other type of news data, such as: images, audio, behaviors etc. Training the model for supporting multiple languages can help in the spreading of the application world-wide. Making the model explain it's decision can become a very trustworthy source of truth in the sensitive places like fake new detection, online social media news.

.

References

[1] C. Shu, X. Tang, J. Xiao and Y. Chen, "Unverified Rumor Detection," in IEEE Access, vol.13, pp.75635-75647, 2025, doi:10.1109/ACCESS.2025.3564638.

[2] Y. FengYing, A. Din, L. HuiChao, M. Babar and S. Ahmad, "Decentralized Consensus in Robotic Swarm for Collective Collision and Avoidance," in IEEE Access, vol. 12, pp. 72143-72154, 2024, doi: 10.1109/ACCESS.2024.3402564.

[3] Y. Zhu, G. Wang, S. Li and X. Huang, "A Novel Rumor Detection Method Based on Non-Consecutive Semantic Features and Comment Stance," in IEEE Access, vol. 11, pp. 58016-58024, 2023, doi: 10.1109/ACCESS.2023.3284308.

[4] S. Mahbub, E. Pardede and A. S. M. Kayes, "COVID-19 Rumor Detection Using Psycho Linguistic Features," in IEEE Access, vol. 10, pp. 117530-117543, 2022, doi: 10.1109/ACCESS.2022.3220369.

[5] Z. Luo, Q. Li and J. Zheng, "Deep Feature Fusion for Rumor Detection on Twitter," in IEEE Access, vol. 9, pp. 126065-126074, 2021, doi: 10.1109/ACCESS.2021.3111790.

[6] M. Chen and W. Jiang, "SCDVit: Semantic Change Detection Based on Sam-Vit and Semantic Consistency," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 18, pp. 2269-2282, 2025, doi: 10.1109/JSTARS.2024.3514926.

[7] X. Ning, Y. He, H. Zhang, R. Zhang, D. Chang and M. Hao, "Semantic Information Collaboration Network for Semantic Change Detection in Remote Sensing Images," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 12893-12909, 2024, doi: 10.1109/JSTARS.2024.3418632.

[8] D. Wheeler and B. Natarajan, "Engineering Semantic Communication: A Survey," in IEEE Access, vol. 11, pp. 13965-13995, 2023, doi: 10.1109/ACCESS.2023.3243065.

[9] S. Sun, S. Dustdar, R. Ranjan, G. Morgan, Y. Dong and L. Wang, "Remote Sensing Image Interpretation with Semantic Graph-Based Methods: A Survey," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 4544-4558, 2022, doi: 10.1109/JSTARS.2022.3176612.

[10] Z. Lin et al., "Image Style Transfer Algorithm Based on Semantic Segmentation," in IEEE Access, vol. 9, pp. 54518-54529, 2021, doi: 10.1109/ACCESS.2021.3054969.

[11] R. Mihalcea and C. Strapparava, “The lie detector: Explorations in the automatic recognition of deceptive language,” in Proc. Conf. Short Papers (ACL-IJCNLP), 2009, pp. 309–312.

[12] F. Bovolo and L. Bruzzone, “The time variable in data fusion: A change detection perspective,” IEEE Geosci. Remote Sens. Mag., vol. 3, no. 3, pp. 8–26, Sep. 2015.

[13] Ericsson. (2021). Ericsson Mobility Report. [Online]. Available: https://www.ericsson.com/4ad7e9/assets/local/reports-papers/mobility-report/documents/2021/ericsson-mobility-report-november-2021.pdf.

[14] L. Wang, J. Yan, L. Mu, and L. Huang, “Knowledge discovery from remote sensing images: A review,” WIREs Data Mining Knowl. Discov., vol. 10, no. 5, pp. 1–31, May 2020.

[15] A. Fernandez, P. Illera, and J. L. Casanova, “Automatic mapping of surfaces affected by forest fires in Spain using AVHRR NDVI composite image data,” Remote Sens. Environ., vol. 60, no. 2, pp. 153–162, May 1997.

[16] L. Bruzzone, R. Cossu, and G. Vernazza, “Detection of land-cover transitions by combining multidate classifiers,” Pattern Recognit. Lett., vol. 25, no. 13, pp. 1491–1500,2004.

[17] P. Yuan, Q. Zhao, X. Zhao, X. Wang, X. Long, and Y. Zheng, “A transformer-based Siamese network and an open optical dataset for semantic change detection of remote sensing images,” Int. J. Digit. Earth, vol. 15, no. 1, pp. 1506–1525, 2022.

[18] L. Zhao, Q. Mo, S. Lin, Z. Wang, Z. Zuo, H. Chen, W. Xing, and D. Lu, “UCTGAN: Diverse image inpainting based on unsupervised cross-space translation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5741–5750.

[19] K. Zhou, C. Shu, B. Li, and J. H. Lau, “Early rumour detection,” in Proc. Conf. North, 2019, pp. 1614–1623.

[20] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, “Prominent features of rumor propagation in online social media,” in Proc. IEEE 13th Int. Conf. Data Mining, Dec. 2013, pp. 1103–1108.

[21] J. Ma, W. Gao, S. Joty, and K.-F. Wong, “An attention-based rumor detection model with tree-structured recursive neural networks,” ACM Trans. Intell. Syst. Technol., vol. 11, no.4, pp. 1–28, Aug. 2020.

[22] C. Li and M. Wand, “Combining Markov random fields and convolutional neural networks for image synthesis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun.2016, pp. 2479–2486.

[23] X. Chen, F. Zhou, G. Trajcevski, and M. Bonsangue, “Multi-view learning with distinguishable feature fusion for rumor detection,” Knowledge-Based Syst., vol. 240, Mar. 2022, Art. no. 108085.

[24] Q. Li, Q. Zhang, and L. Si, “Rumor detection by exploiting user credibility information, attention and multi-task learning,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 1173–1179.

[25] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 694–711.

[26] W. Zou and L. Tang, “What do we believe in? Rumors and processing strategies during the COVID-19 outbreak in China,” Public Understand. Sci., vol. 30, no. 2, pp. 153–168, Feb. 2021.

[27] K. Yang, “Asymmetric Siamese networks for semantic change detection in aerial images,” IEEE Trans. Geosci. Remote Sens., vol. 60, Oct. 2021, Art. no. 5609818, doi:10.1109/TGRS.2021.3113912.

[28] L. Zhao, A. Li, S. Lin, W. Xing, and D. Lu, “SpatialGAN: Progressive image generation based on spatial recursive adversarial expansion,” in Proc. ACM Multimedia Conf., 2020, pp. 2336–2344.

[29] J. Ma, W. Gao, and K. F. Wong, “Rumor detection on Twitter with tree-structured recursive neural networks,” in Proc. ACL, Melbourne, VIC, Australia, 2018, pp. 1980–1989.

[30] D. Lu, P. Mausel, E. Brondizio, and E. Moran, “Change detection techniques,” Int. J. Remote Sens., vol. 25, no. 12, pp. 2365–2407, Jun. 2004.