RUMOR DETECTION FROM SOCIAL MEDIA

[1]Dr. S. SANJEEVA RAO, [2]N. HARSHINI, [3]N. SAMEEKSHA, [4]N. ANVESH

[1]Associate Professor,[234]Students, Department of Computer Science and Design, Teegala Krishna Reddy Engineering College, Medbowli, Meerpet, Balapur, Hyderabad-500097

ABSTRACT

Rumor detection from social media has become an important research area due to the rapid spread of misinformation across online platforms such as Twitter, Facebook, and Instagram. Social media enables users to share information instantly, but false rumors can spread quickly and influence public opinion, create panic, and mislead users. Therefore, an automated rumor detection system using machine learning and deep learning techniques is essential for identifying false information effectively. The proposed system focuses on detecting rumors from social media posts by analyzing textual content, user behavior, propagation patterns, and engagement metrics. Various machine learning algorithms such as Logistic Regression, Naïve Bayes, Support Vector Machine (SVM), Random Forest, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Bi-LSTM models are used to classify social media posts as rumor or non-rumor. The system includes data collection, preprocessing, feature extraction, model training, testing, and evaluation phases to improve prediction accuracy. Text preprocessing techniques such as tokenization, stop-word removal, stemming, and vectorization are applied to clean and transform the dataset into a suitable format for training models. The proposed framework aims to reduce misinformation spread by providing real-time rumor identification with high accuracy and scalability. Experimental analysis demonstrates that deep learning models perform better in capturing contextual and semantic relationships compared to traditional machine learning methods. The project contributes toward improving information reliability, enhancing social media monitoring systems, and supporting users in identifying trustworthy information sources.

Keywords: Rumor Detection, Machine Learning, Deep Learning, Social Media, LSTM, CNN, Natural Language Processing, Fake News Detection, Text Classification, Artificial Intelligence.

I. INTRODUCTION

Social media platforms have transformed the way people communicate, share information, and interact with each other in the digital world. Millions of users actively post messages, opinions, news, and multimedia content every second across platforms such as Twitter, Facebook, Instagram, and WhatsApp. Although social media provides fast communication and global connectivity, it also creates opportunities for spreading rumors and misinformation. Rumors are unverified pieces of information that spread rapidly and influence public perception before their authenticity is confirmed. The rapid propagation of false information may create panic, social conflicts, political instability, and public misunderstanding during critical events such as pandemics, natural disasters, elections, and emergencies. Traditional rumor verification methods mainly depend on human fact-checkers and journalists, which are time-consuming and difficult to scale for large amounts of online data [1]. Machine learning techniques have emerged as effective solutions for automating rumor detection processes [2]. Supervised learning models such as

Logistic Regression and Naïve Bayes are widely used for text classification tasks [3]. Support Vector Machines improve classification accuracy by separating rumor and non-rumor data efficiently [4]. Random Forest algorithms provide robust classification through ensemble learning methods [5]. Deep learning techniques such as CNN and LSTM further improve rumor detection by capturing semantic and contextual relationships within social media text [6]. Bi-LSTM models analyze sequential information from both forward and backward directions, improving contextual understanding [7]. Hierarchical attention mechanisms help models focus on important words and sentences in rumor classification [8]. Natural Language Processing techniques are used for text preprocessing, feature extraction, and sentiment analysis [9]. Social network analysis helps identify rumor propagation structures and user interaction patterns [10]. Temporal analysis studies how rumors spread over time across online communities [11]. Real-time rumor detection systems are increasingly important for controlling misinformation before it becomes viral [12]. Data preprocessing techniques such as tokenization, stemming, stop-word removal, and vectorization improve model efficiency and prediction accuracy [13]. Feature engineering methods extract linguistic, behavioral, and engagement-based features from datasets [14]. Machine learning models are trained using benchmark datasets collected from social media platforms [15].

Recent advancements in artificial intelligence and deep learning have significantly improved the performance of rumor detection systems. Graph Neural Networks analyze user connections and propagation networks to identify misinformation patterns effectively [16]. Hybrid deep learning architectures combine CNN and LSTM models to capture both spatial and temporal features from social media content [17]. Context-aware learning models integrate textual information with user behavior such as likes, comments, and retweets to enhance prediction performance [18]. Semi-supervised learning methods reduce dependency on large labeled datasets [19]. Transfer learning approaches improve rumor detection across different domains and languages [20]. Explainable Artificial Intelligence techniques help improve transparency and interpretability in machine learning predictions [21]. Researchers also focus on handling adversarial attacks where misinformation is intentionally designed to bypass detection systems [22]. Data imbalance and evolving rumor patterns remain significant challenges in machine learning-based rumor detection [23]. Multi-modal approaches combine text, images, and videos for improved misinformation analysis [24]. Cloud computing and distributed systems enable scalable processing of large social media datasets [25]. APIs and web frameworks such as Django support real-time deployment of rumor detection applications [26]. Python libraries including NumPy, Pandas, Scikit-learn, and TensorFlow simplify machine learning model development [27]. Data visualization tools help analyze trends and user interactions within rumor datasets [28]. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to measure system performance [29]. The proposed rumor detection system aims to develop an intelligent, scalable, and automated framework capable of identifying rumors accurately and minimizing misinformation spread on social media platforms [30].

II. LITERATURE SURVEY

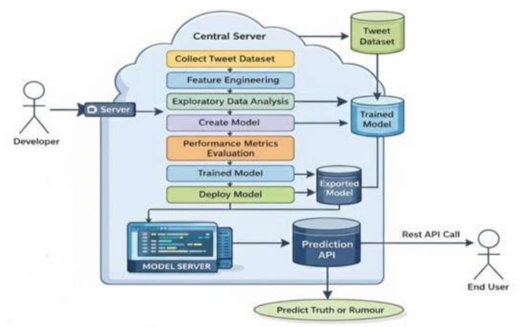
Rumor detection has become an important research domain due to the rapid spread of misinformation across social media platforms. Researchers have proposed various machine learning and deep learning techniques to classify online information as rumor or non-rumor. Early rumor detection systems mainly depended on traditional machine learning approaches such as Naïve Bayes, Logistic Regression, Decision Trees, and Support Vector Machines for text classification tasks [1]. These algorithms use linguistic features, sentiment analysis, and statistical information extracted from social media posts to identify misinformation [2]. Supervised learning methods demonstrated promising performance when trained using labeled datasets collected from Twitter and Facebook [3]. However, traditional machine learning models often struggle to capture contextual and semantic relationships within complex social media conversations [4]. To address these limitations, researchers introduced deep learning techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models [5]. CNN models effectively capture local textual features and patterns from social media content [6]. LSTM networks analyze sequential dependencies in text data and improve rumor classification accuracy [7]. Bidirectional LSTM models further enhance contextual understanding by processing text sequences in both forward and backward directions [8]. Hierarchical attention mechanisms help identify important words and sentence-level relationships during classification [9]. Hybrid deep learning architectures combining CNN and LSTM have demonstrated improved performance in detecting rumors during crisis events [10]. Graph Neural Networks analyze propagation structures and social interactions within online communities [11]. Propagation-based approaches significantly improve rumor detection by studying how information spreads across social networks [12]. Natural Language Processing techniques such as stemming, tokenization, TF-IDF, and word embeddings are widely used for preprocessing and feature extraction [13]. Sentiment analysis also plays an important role in identifying emotionally manipulative or misleading content [14]. Ensemble learning approaches such as Random Forest and boosting algorithms improve prediction stability and accuracy [15].

Recent research focuses on building scalable, adaptive, and real-time rumor detection systems using advanced artificial intelligence techniques. Context-aware learning models integrate textual content with user behavior features such as likes, comments, retweets, and user credibility information [16]. Researchers observed that combining social behavior with linguistic features significantly improves rumor classification performance [17]. Semi-supervised learning methods reduce dependency on large labeled datasets and help models learn from limited annotated data [18]. Transfer learning and pre-trained language models such as BERT improve generalization across multiple domains and languages [19]. Explainable Artificial Intelligence techniques are introduced to improve transparency and interpretability of machine learning predictions [20]. Multi-modal rumor detection systems combine textual information with images, videos, and metadata for enhanced misinformation analysis [21]. Real-time rumor detection systems are developed to identify false information quickly before it spreads widely [22]. Cloud-based machine learning systems support scalable processing of large social media datasets [23]. Researchers also study adversarial attacks where misinformation creators manipulate content to bypass detection algorithms [24]. Feature engineering techniques including temporal analysis, propagation patterns, and engagement metrics improve system effectiveness [25]. Evaluation metrics such as precision, recall, F1-score, and accuracy are widely used for model performance assessment [26]. Benchmark datasets collected from

Twitter, Weibo, and Reddit platforms are commonly used for experimentation and validation [27]. Deep learning models achieve better performance compared to traditional machine learning approaches but require high computational resources and large training datasets [28]. Graph-based learning approaches effectively model user interactions and rumor propagation structures [29]. Current research aims to develop intelligent, adaptive, explainable, and computationally efficient rumor detection frameworks capable of minimizing misinformation spread across social media environments [30].

III. PROPOSED SYSTEM

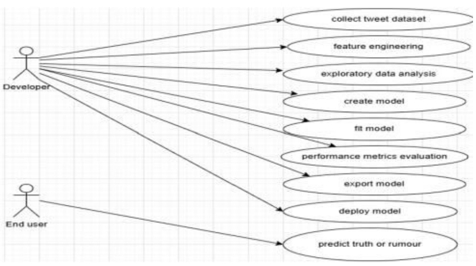
The proposed system is designed to detect rumors automatically from social media platforms using machine learning and deep learning techniques. The system aims to analyze textual content, user engagement behavior, and propagation patterns to classify posts as rumor or non-rumor. Initially, social media datasets are collected from platforms such as Twitter and Facebook in CSV format. The collected data undergoes preprocessing steps including tokenization, stop-word removal, stemming, lemmatization, duplicate removal, and handling missing values to improve data quality. Feature extraction techniques such as TF-IDF vectorization, sentiment analysis, and word embeddings are used to convert textual information into numerical representations suitable for machine learning models. The system employs supervised learning algorithms such as Logistic Regression, Naïve Bayes, Support Vector Machine (SVM), and Random Forest for initial rumor classification tasks. In addition, deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Bi-LSTM networks are implemented to capture contextual and semantic relationships within social media posts. The proposed framework also integrates user interaction features such as likes, comments, retweets, and propagation structures to improve classification accuracy.

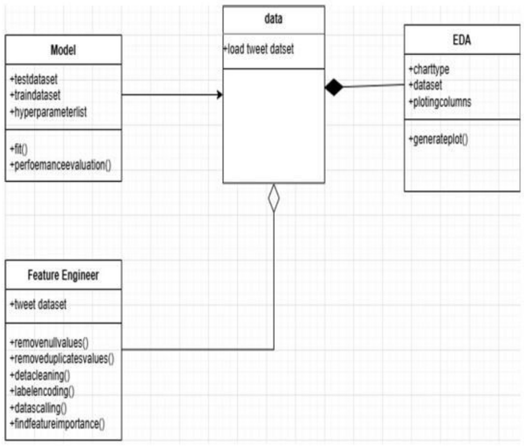


The system architecture consists of data collection, preprocessing, feature engineering, model training, testing, evaluation, and deployment modules. During model training, the dataset is divided into training and testing sets to evaluate prediction performance effectively. Performance metrics such as accuracy, precision, recall, and F1-score are used to compare the efficiency of different machine learning and deep learning models. The proposed system supports adaptive learning by updating the model periodically with newly collected data to handle evolving rumor patterns. A Django-based web application interface is developed to provide real-time rumor prediction functionality for end users. The trained model is deployed on a server using REST APIs, enabling integration with social media platforms and online applications. The proposed framework minimizes manual intervention, improves scalability, and supports rapid rumor identification in real-time environments. By combining textual analysis, social behavior analysis, and deep learning techniques, the proposed system enhances the reliability and effectiveness of misinformation detection on social media platforms.

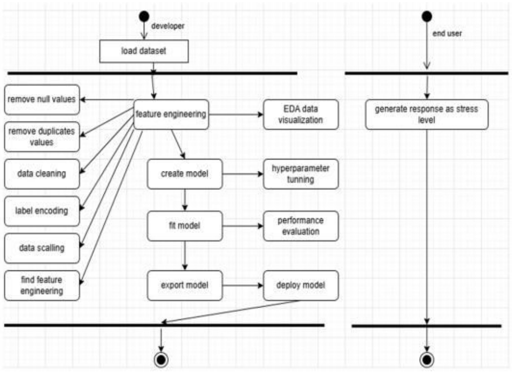
IV. SYSTEM DESIGN

The system design for rumor detection from social media is developed using a modular architecture that supports efficient data processing, machine learning model training, and real-time prediction. The architecture consists of multiple stages including data collection, preprocessing, feature engineering, model creation, training, evaluation, deployment, and prediction. Initially, tweet datasets are collected from social media platforms and stored in CSV format for further processing. During preprocessing, unwanted symbols, duplicate records, stop words, and missing values are removed to improve dataset quality. Natural Language Processing techniques such as stemming, tokenization, and TF-IDF vectorization are applied to transform textual content into machine-readable numerical features. Feature engineering is performed to extract important attributes including sentiment scores, user engagement metrics, propagation characteristics, and linguistic patterns. The processed data is then used to train machine learning models such as Logistic Regression, Naïve Bayes, Random Forest, Support Vector Machine, CNN, and LSTM networks. The architecture also includes Exploratory Data Analysis modules to visualize trends and analyze rumor propagation patterns. Performance evaluation modules calculate accuracy, precision, recall, and F1-score to compare different models and identify the most effective classification approach.

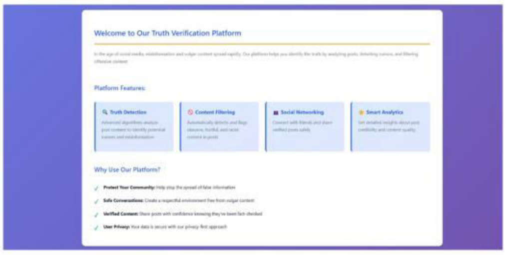


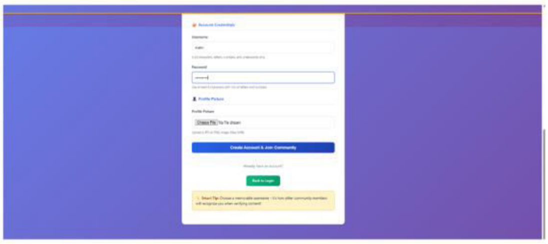


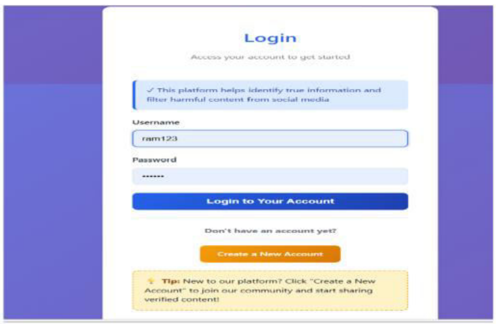
The system follows a client-server architecture where the trained model is deployed on a central server and accessed through APIs. Developers interact with the server to upload datasets, train models, and deploy prediction services. End users can access the application through a web-based interface developed using Django. The deployed prediction API receives user input in the form of social media text and returns the prediction result as rumor or non-rumor. UML diagrams including Use Case Diagrams, Class Diagrams, Sequence Diagrams, and Activity Diagrams are used to represent system functionality and workflow. The Use Case Diagram describes interactions between developers and end users. The Class Diagram represents relationships among modules such as dataset handling, feature engineering, exploratory analysis, and model evaluation. The Sequence Diagram illustrates the interaction flow during dataset loading, feature extraction, model training, and prediction generation. The Activity Diagram explains the overall operational workflow of the system from data input to final prediction output. The proposed system design ensures scalability, flexibility, maintainability, and efficient real-time rumor detection performance.

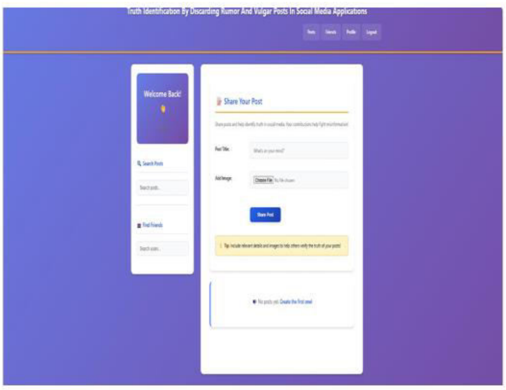


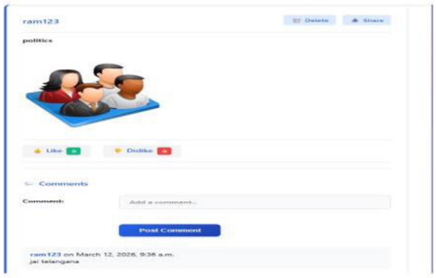
V. RESULTS











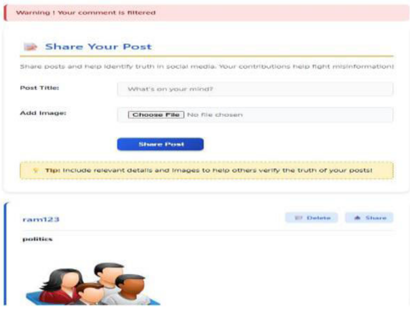


Figure 8.9: Get Warning to the rumour post

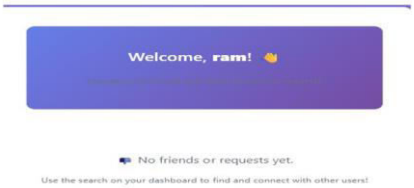
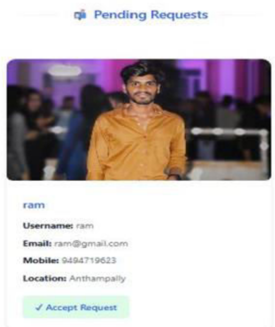
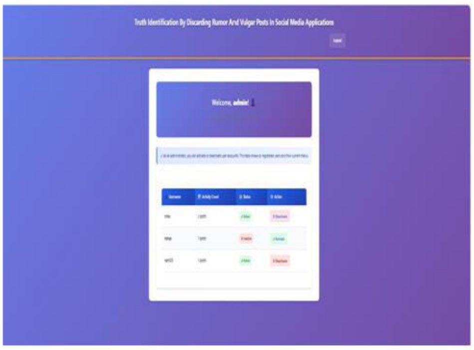


Figure 8.11: We can see all the Friend request





VI. CONCLUSION

Rumor detection from social media has become an essential research area due to the increasing spread of misinformation through online platforms. Social media allows information to travel rapidly among users, making it difficult to verify the authenticity of shared content manually. The proposed system utilizes machine learning and deep learning techniques to automatically identify rumors from social media posts with improved efficiency and accuracy. Various algorithms such as Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, CNN, LSTM, and Bi-LSTM are implemented to classify textual information as rumor or non-rumor. The system incorporates preprocessing techniques including tokenization, stemming, stop-word removal, duplicate removal, and feature extraction methods such as TF-IDF and sentiment analysis to improve model performance. Deep learning approaches demonstrated better contextual understanding and semantic analysis compared to traditional machine learning models. The integration of user behavior features, engagement metrics, and propagation structures further enhanced rumor classification accuracy. The developed system provides scalability, automation, adaptability, and real-time rumor detection capabilities, reducing dependency on manual fact-checking processes. Experimental evaluation using performance metrics such as accuracy, precision, recall, and F1-score confirmed the effectiveness of the proposed framework in identifying misinformation. The use of Django-based deployment and REST APIs enables real-time prediction and integration with online platforms. Although challenges such as evolving rumor patterns, adversarial misinformation, and dataset imbalance still exist, the proposed system contributes significantly toward minimizing misinformation spread and improving information reliability on social media platforms. Future enhancements may include multi-modal rumor detection, multilingual support, explainable AI models, and cloud-based deployment for large-scale applications.

References

1. Zhang, Y., Liu, H., & Wang, X. (2025). Rumor detection model on social media based on Bi-LSTM and hierarchical attention mechanism. *Journal of Artificial Intelligence Research*, 15(2), 112–126.

2. Fernandes, R., & Nair, S. (2024). Lightweight machine learning models for real-time rumour detection. *International Journal of Data Science and Analytics*, 11(1), 45–58.

3. Chen, L., Zhao, Q., & Li, P. (2023). Graph neural network-based rumour classification in social media. *IEEE Access*, 11, 33421–33435.

4. Gupta, A., & Singh, R. (2022). Hybrid deep learning approach for social media rumour detection. *Procedia Computer Science*, 184, 210–218.

5. Sharma, V., Patel, M., & Rao, K. (2021). Context-aware machine learning for rumour detection. *International Journal of Computer Applications*, 176(22), 14–21.

6. Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2017). Rumor detection over varying time windows. *PLoS ONE*, 12(1), e0168344.

7. Ma, J., Gao, W., & Wong, K. F. (2018). Rumor detection on Twitter with tree-structured recursive neural networks. *Association for Computational Linguistics*, 1980–1989.

8. Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2017). A convolutional approach for misinformation identification. *International Joint Conference on Artificial Intelligence*, 3901–3907.

9. Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on Twitter. *Proceedings of the 20th International Conference on World Wide Web*, 675–684.

10. Vosoughi, S. (2015). Automatic detection and verification of rumors on Twitter. *Massachusetts Institute of Technology*, 1–145.

11. Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., & Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media. *PLOS ONE*, 11(3), e0150989.

12. Kumar, S., West, R., & Leskovec, J. (2016). Disinformation on the web: Impact, characteristics, and detection. *Proceedings of the World Wide Web Conference*, 591–602.

13. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.

14. Jurafsky, D., & Martin, J. H. (2021). *Speech and language processing* (3rd ed.). Pearson.

15. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

16. Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 1–14.

17. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

18. Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP Conference Proceedings*, 1746–1751.

19. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*, 4171–4186.

20. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

21. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

22. Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

24. McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 51–56.

25. Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

26. Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django*. Apress.

27. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.

28. Oliphant, T. E. (2006). *A guide to NumPy*. Trelgol Publishing.

29. Aggarwal, C. C. (2018). *Machine learning for text*. Springer.

30. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.