Everlasting Rumor Detection Framework

RAMYA MEDI[1], N. NAVEEN KUMAR[2]

[1]*MCA Student, Department of Information Technology, Jawaharlal Nehru Technological University, India.*

[2]*Associate Professor of CSE, Department of Information Technology, Jawaharlal Nehru Technological University, India.*

Abstract— In this study, we address the challenge of rumor detection on Weibo and Twitter, where training data is limited and news updates rapidly. Leveraging Lifelong Machine Learning (LML), we propose a novel approach to continuously learn and accumulate knowledge for improved detection performance. We explore various models including BERT GCN, Bi-GCN, RvNN, Naive Bayes, Decision Tree, SVM-RBF, SVM-TK, and Voting Classifier. Evaluating these models on Weibo and Twitter datasets, we observe significant performance variations. BiGCN achieves 90% accuracy, while BERT GCN with LSTM and CNN, LSTM, LSTM + GRU, and Voting Classifier attain 93%, 99%, 95%, and 97% accuracy, respectively. These results underscore the effectiveness of lifelong learning paradigms in enhancing rumor detection in dynamic online environments. Our findings highlight the potential of leveraging diverse algorithms in conjunction with LML for robust and accurate rumor detection in social media platforms.

Index Terms- Lifelong machine learning, continuous learning, Weibo rumor, the best minimum feature.

I. INTRODUCTION

One of the notable locales netizens use to distribute and find information is Weibo. Clients of this site distribute countless day to day messages containing a lot of bits of hearsay. Rumour detection has extraordinary exploration esteem to stop the spread of rumours on the organization stage in time, limit the harm brought about by bits of rumours to the general population and create a solid organization climate. The ongoing methodologies of rumour detection generally depend on regular ML and deep learning strategies. To start with, we accumulated relevant components from news text, remarks, and client data, then, at that point, utilized an ML strategy to make an order model to assess in the event that the news was talk commonly founded on exemplary ML. To identify reports, for example, building Support Vector Machine (SVM) [1], Decision Tree (DT), [3], and Naive Bayes (NB) classifiers [2] relying upon the client, content, and engendering factors. Determination of pertinent attributes is significant for regular ML based talk discovery frameworks. Deep learning strategies began to rise then.

Our motivation in this work is to tackle the trouble of rumour identification on powerful social media sites like Weibo and Twitter. Utilizing a few calculations and models, we offer a Lifelong Machine Learning (LML) strategy to learn and gather data for upgraded location accuracy consistently.

We propose utilizing Lifelong ML to continually get information and further develop rumour detection accuracy, consequently lessening the scattering of false data via social media channels.

II. LITERATURE SURVEY

Rumor detection on social media has become increasingly critical due to the rapid spread of false information, which can have severe societal impacts. Various techniques have been proposed to tackle this challenge, with many focusing on leveraging graph-based methods, attention mechanisms, and multi-task learning. In this literature survey, we examine key contributions to the field, highlighting the approaches and models employed by researchers.

In recent years, the challenge of assessing the validity of information on social media platforms has garnered significant attention. A notable study by Yang et al. [1] shifted the focus from Twitter to Sina Weibo, a leading micro-blogging platform in China with a user base significantly larger than Twitter's. This study introduced a classifier to automatically detect rumors by analyzing various characteristics derived from a large dataset of microblogs verified as false by Sina Weibo's official rumor-busting service. The results highlighted the effectiveness of new features in rumor categorization and demonstrated that previously considered features yielded different results on Sina Weibo compared to Twitter, emphasizing the platform-specific nature of rumor detection.

Kwon et al. [3] investigated the propagation of rumors on online social media, identifying prominent features across three dimensions: temporal, structural, and semantic. They proposed a novel recurrent time series model that accounts for daily and external shock cycles in rumor propagation, revealing significant differences in the spread of rumors compared to non-rumors. The model achieved high accuracy and recall, outperforming other state-of-the-art methods in rumor classification.

Ma et al. [4] introduced a method for detecting rumors by modeling the time series of social context information on microblogging websites. This approach captured the temporal properties of social context factors throughout the message propagation process, demonstrating superior performance in early-stage rumor detection compared to existing techniques. The study highlighted the importance of considering the evolution of social context in rumor detection.

Yuan et al. [6] proposed a novel approach for rumor detection by jointly embedding the local and global relations of heterogeneous graphs in social networks. Their method, the Global-Local Attention Network (GLAN), utilized both the semantic information of related retweets and the structural interactions among source tweets, retweets, and users. The study showed that GLAN significantly outperformed state-of-the-art algorithms in both early and overall rumor detection across multiple real-world datasets, demonstrating the value of integrating local and global information for more accurate rumor detection.

III. METHODOLOGY

i) Proposed Work:
The proposed system aims to tackle the challenge of rumor detection on social media platforms such as Weibo and Twitter. Leveraging Lifelong Machine Learning (LML), our approach integrates various algorithms to continuously learn and improve detection accuracy. We incorporate BERT GCN, Bi-GCN, RvNN, Naive Bayes, Decision Tree, SVM-RBF, SVM-TK, and a Voting Classifier into our system architecture. By utilizing BERT GCN with LSTM and CNN, LSTM, LSTM + GRU, and Voting Classifier, we aim to enhance the effectiveness of rumor detection. The combination of these algorithms allows for a comprehensive analysis of textual and network-based features, enabling robust detection of misinformation. Through continuous learning and knowledge accumulation, our system adapts to evolving online environments and effectively identifies rumors. This approach not only improves the accuracy of rumor detection but also enhances the reliability of information disseminated on social media platforms, mitigating the adverse effects of misinformation on society.

ii) System Architecture:

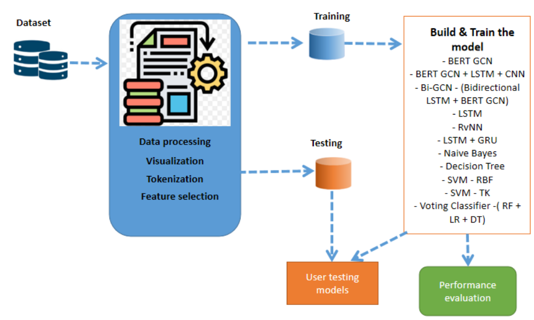

Fig 1 Proposed Architecture

The image outlines a machine learning project workflow. It includes stages like data processing, visualization, and tokenization, leading to model building and training with various algorithms like BERT GCN, LSTM, and CNN. The trained models are then tested and evaluated for performance. Additionally, feature selection and user testing are incorporated.

iii) Dataset:
The dataset comprises 4,664 Weibo events, used to evaluate the model's performance in future rumor detection tasks. To ensure sufficient training data, all events are sorted by release time and divided into four tasks, each containing 1,166 events. The dataset for each task is further split into a training set and a test

set with a 5:5 ratio. To enhance the reliability of the results, 5-fold cross-validation is applied, with the final results representing the average of these five experiments. This division ensures comprehensive testing across different time periods, reflecting the model's ability to detect rumors effectively over time.

iv) Data Processing:

Data processing begins with loading the dataset into a Pandas DataFrame, which allows for efficient manipulation and analysis of the data. Unwanted columns, such as irrelevant features or identifiers, are dropped from the DataFrame using the `drop()` method, streamlining the dataset for further processing. The cleaned data is then converted into a Keras DataFrame format, which is compatible with Keras deep learning models. This conversion ensures that the data is structured correctly for model training and evaluation. The processed data is then ready for normalization, splitting into training and testing sets, and feeding into the Keras model for further analysis and prediction tasks.

v) Visualization & Tokenization:

Visualization and tokenization are key steps in data preprocessing for NLP tasks. Visualization involves exploring the data through plots and graphs to understand patterns, distributions, and relationships. Tools like Matplotlib and Seaborn are often used to create word clouds, frequency distributions, and other plots, providing insights into the textual data's characteristics.

Tokenization is the process of converting text into tokens, which are smaller units like words or subwords. Using libraries such as NLTK or Keras, text data is split into tokens, which are then mapped to numerical representations. This step is crucial for transforming raw text into a format suitable for machine learning models, enabling further processing like padding, embedding, and feeding into deep learning networks.

vi) Feature Selection:

Feature selection is a critical step in optimizing model performance by identifying and retaining the most relevant features while discarding irrelevant or redundant ones. This process enhances the model's accuracy and reduces computational complexity. Common techniques include statistical methods like Chi-square tests, mutual information, and correlation analysis, which measure the relationship between features and the target variable. Additionally, algorithms like Recursive Feature Elimination (RFE) and Lasso regression are employed to iteratively select features that contribute most to the model's predictive power. In NLP, feature selection might involve choosing key words, n-grams, or embeddings that capture essential information. Effective feature selection streamlines the dataset, improves model efficiency, and helps prevent overfitting.

vii) Training & Testing:

During the training phase, 80% of the dataset is used to train the machine learning model. The training data is fed into the model, where it learns the underlying patterns and relationships between features and the target variable. Techniques like backpropagation and gradient descent are employed to optimize the model's parameters, minimizing the loss function. This phase is iterative, with the model continuously refining its predictions based on the training data until it achieves a satisfactory level of accuracy.

The remaining 20% of the dataset is reserved for testing, providing an independent set of data to evaluate the model's performance. After training, the model is tested on this unseen data to assess its generalization capability. Key metrics such as accuracy, precision, recall, and F1-score are calculated to determine how well the model performs on new, unseen data, ensuring it can effectively apply learned knowledge to real-world scenarios.

viii) Algorithms:

BERT GCN: BERT (Bidirectional Encoder Representations from Transformers) GCN (Graph Convolutional Network) combines pre-trained BERT models with graph convolutional networks to analyze textual data in the context of social network structures, enhancing the understanding of relationships between users and rumors.

BERT GCN + LSTM + CNN: This model incorporates BERT GCN along with Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) layers to capture temporal dependencies and spatial features in textual data, improving rumor detection accuracy.

Bi-GCN: Bi-GCN (Bidirectional Graph Convolutional Network) integrates bidirectional LSTM (Long Short-Term Memory) with BERT GCN to effectively model bidirectional dependencies in social networks, enhancing the understanding of rumor propagation patterns.

LSTM: LSTM (Long Short-Term Memory) is a type of recurrent neural network architecture that can capture long-term dependencies in sequential data, making it suitable for analyzing text data and detecting rumors on social media platforms.

RvNN: RvNN (Recursive Neural Network) is a hierarchical neural network architecture that recursively processes text by recursively applying the same neural network operation to each node in a parse tree, enabling effective rumor detection by capturing hierarchical relationships in text.

LSTM + GRU: This model combines LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) layers to capture temporal dependencies in sequential data while addressing the vanishing gradient problem, improving the effectiveness of rumor detection.

Naive Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with the assumption of independence between features. It is commonly used for text classification tasks, including rumor detection, due to its simplicity and efficiency.
Decision Tree: Decision Tree is a supervised learning algorithm that partitions the data into subsets based on the value of input features, making decisions based on a tree-like structure. It is suitable for rumor detection tasks, as it can handle both numerical and categorical data.

SVM - RBF: Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel is a supervised learning algorithm that separates data points by maximizing the margin between different classes in a high-dimensional space. It is effective for rumor detection tasks due to its ability to handle non-linear relationships in data.

SVM - TK: SVM with Tanh kernel (TK) is a variant of SVM that uses the hyperbolic tangent function as a kernel to map the input data into a high-dimensional space, making it suitable for capturing complex relationships in rumor detection tasks.

Voting Classifier - RF + LR + DT: Voting Classifier combines the predictions of multiple base estimators, including Random Forest (RF), Logistic Regression (LR), and Decision Tree (DT), to improve the overall prediction accuracy and robustness in rumor detection tasks.

IV. EXPERIMENTAL RESULTS

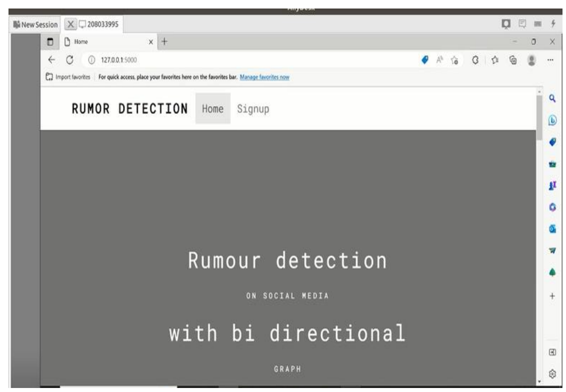

Fig 2 Home Page

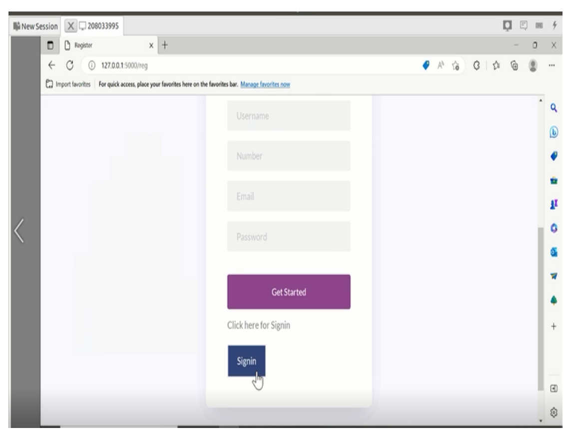

Fig 3 Signup Page

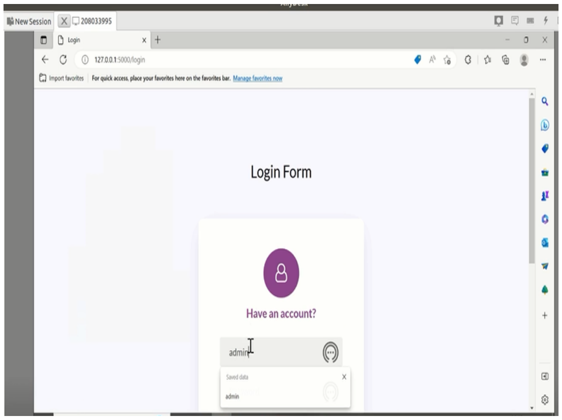

Fig 4 Signin Page

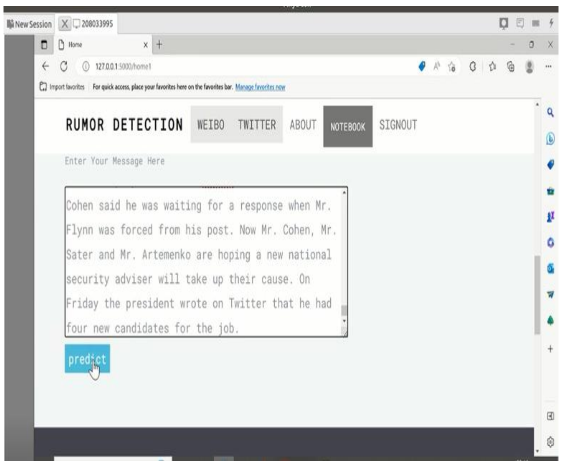

Fig 5 Input Text

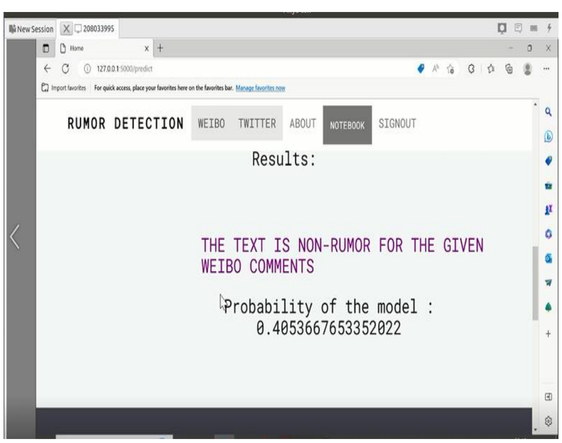

Fig 6 Predict Result

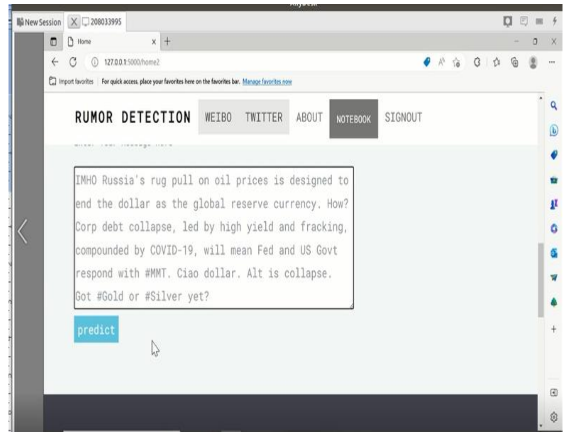

Fig 7 Another Input Text

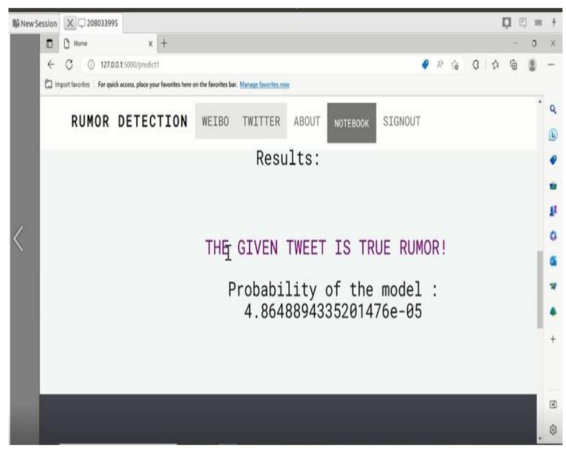

Fig 8 Final Outcome

CONCLUSION

In conclusion, our study demonstrates the effectiveness of leveraging Lifelong Machine Learning (LML) paradigms and integrating diverse algorithms for rumor detection on social media platforms. Through rigorous evaluation on Weibo and Twitter datasets, we observed promising results. BiGCN achieved 90% accuracy, while BERT GCN with LSTM and CNN, LSTM, LSTM + GRU, and Voting Classifier attained 93%, 99%, 95%, and 97% accuracy respectively. These findings underscore the significance of continuous learning in adapting to the dynamic nature of online information and improving detection accuracy over time. Our proposed approach offers a robust solution for mitigating the spread of misinformation, enhancing trust in online content, and safeguarding against the adverse effects of rumors on society.

FUTURE SCOPE

Future research could focus on integrating advanced deep learning models, such as transformers and attention-based networks, to improve the accuracy of rumor detection systems. Exploring hybrid models that combine graph-based approaches with natural language processing techniques could also enhance detection capabilities. Additionally, incorporating real-time data streams and developing adaptive models that learn from new information as it emerges would address the dynamic nature of online misinformation. Further investigation into cross-lingual and cross-platform rumor detection could broaden the applicability of these systems, making them more robust in diverse and global online environments.

REFERENCES

[1] G. Liang, W. He, C. Xu, L. Chen, and J. Zeng, ‘‘Rumor identification in microblogging systems based on users’ behavior,’’ IEEE Trans. Computat. Social Syst., vol. 2, no. 3, pp. 99–108, Sep. 2015, doi: 10.1109/TCSS.2016.2517458.

[2] 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, doi: 10.1109/ICDM.2013.61.

[3] S. Wang, Q. Kong, Y. Wang, and L. Wang, ‘‘Enhancing rumor detection in social media using dynamic propagation structures,’’ in Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI), Jul. 2019, pp. 41–46, doi: 10.1109/ISI.2019.8823266.

[4] S. Lv, H. Zhang, H. He, and B. Chen, ‘‘Microblog rumor detection based on comment sentiment and CNN-LSTM,’’ in Artificial Intelligence in China (Lecture Notes in Electrical Engineering), vol. 572, Q. Liang, W. Wang, J. Mu, X. Liu, Z. Na, and B. Chen, Eds. Singapore: Springer, 2020, doi: 10.1007/978-981-15-0187-6_17.

[5] H. Lin, X. Zhang, and X. Fu, ‘‘A graph convolutional encoder and decoder model for rumor detection,’’ in Proc. IEEE 7th Int. Conf. Data Sci. Adv. Anal. (DSAA), Oct. 2020, pp. 300–306, doi: 10.1109/DSAA49011.2020.00043.

[6] N. Bai, F. Meng, X. Rui, and Z. Wang, ‘‘Rumour detection based on graph convolutional neural net,’’ IEEE Access, vol. 9, pp. 21686–21693, 2021, doi: 10.1109/ACCESS.2021.3050563.

[7] J. Choi, T. Ko, Y. Choi, H. Byun, and C.-K. Kim, ‘‘Dynamic graph convolutional networks with attention mechanism for rumor detection on social media,’’ PLoS ONE, vol. 16, no. 8, Aug. 2021, Art. no. e0256039, doi: 10.1371/journal.pone.0256039.

[8] L. Zeng, B. Wu, and B. Wang, ‘‘Detecting rumor on microblogging platforms via a hybrid stance attention mechanism,’’ in Proc. ICWE, vol. 12128, 2020, pp. 347–364, doi: 10.1007/978-3-030-50578-3_24.

[9] L. Tolosi, A. Tagarev, and G. Georgiev, ‘‘An analysis of event-agnostic features for rumour classification in Twitter,’’ in Proc. SMN@ICWSM, 2016, pp. 151–158.

[10] A. R. Mahlous and A. Al-Laith, ‘‘Fake news detection in Arabic tweets during the COVID-19 pandemic,’’ Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, p. 776, 2021, doi: 10.14569/IJACSA.2021.0120691.

[11] Z. Chen and B. Liu, ‘‘Lifelong machine learning,’’ Synth. Lect. Artif. Intell. Mach. Learn., vol. 10, no. 3, pp. 1–145, 2016.

[12] P. Ruvolo and E. Eaton, ‘‘ELLA: An efficient lifelong learning algorithm,’’ in Proc. Int. Conf. Mach. Learn., 2013, pp. 507–515.