A Dual-Framework Approach for Fake News Detection Using Transformer-Based Embeddings and Explainable AI

Roshan M*[1], Monish K P[1], Adlin Layola J A[1], and Kiruba Wesley[1]

[1]Department of Artificial Intelligence and Machine Learning, St.Joseph's College of Engineering, Chennai 600119, India.

roshan162k@gmail.com

Abstract. The spread of incorrect information across digital media is a significant barrier to user confidence in the information and trustworthiness of digital content. In order to assist content users in spotting false information in online content, this article proposes a comprehensive dual framework that relies on social networks' community relational characteristics (graph-based propagation), as well as on where the content is shared (contextual embedding). The proposed architecture consists of multiple elements that include fine-tuning from the BERT model, transformer neural network structure to produce semantic representations of text-based content; weighted probability aggregators (WMA) for how information propagates within a community in the social network; attention visualizations; local interpretable model agnostic explanations (LIME); multi-layered approaches for evaluating the accuracy of information; and state-of-the-art pre-processing techniques like tokenization, lemmatization, and extracting contextual features, which provide the ability to build flexible annotation formats (binary, multi-class, and severity) for incorrectly identified media. We present experimental validation showing our proposed system outperforms other methods across multiple benchmark datasets regarding precision, accuracy, recall, and F1-score, quadrupling the baseline systems' performance. Lastly, this deployment-ready approach supports RESTful APIs for on-demand content verification.

Keywords: Misinformation Detection, Natural Language Processing , Deep Learning , Explainable AI · BERT , ss Content Verification

1 Introduction

One of the most radical changes has been brought into the limelight with the development of digital media. in history as it is concerned with the means through which false teaching is taught, forging fresh avenues of systematic attempts at misleading and deceiving the masses, at the same time as developing a completely new platform of the spread of facts and truth [1]. Most of the existing methods of detecting fake information heavily depend on. manually generated properties and classification methods, which are not flexible. with reference to various types of language and varying modes of deceptive behaviour [2].

S. P. Vijayaragavan et al. (eds.), *Proceedings of the Global Conference on Sustainable Energy Systems, Smart Electronics and Intelligent Computing (GCSESEIC 2025)*, Advances in Engineering Research 297,
https://doi.org/10.2991/978-94-6239-654-8_45

Traditional methods of verification of false information, succeed well in. they do not detect some of the most glaring types of misinformation as well. where confronted with the multiplicity of possible types of online misinformation Misinformed content can be generated through numerous methods, including total fabrication, selective omission, circumstantial manipulation, and exaggerated presentation; therefore, these diverse forms of misinformation will require more sophisticated analytical methods that can simultaneously analyse the text, contextual relationships, and the mechanisms by which information is disseminated as it pertains to misinformation [3].

Recent developments in deep learning methodology and architecture have resulted in an increase in the performance of transformer-based models and their potential for automated content verification [4]. Existing implementations do not take into account some of the most important elements that need to be considered during automated content moderation [5]; cross-platform generalisation, temporal consistency and interpretability, all of which are important for a successful implementation in current content moderation processes.

2 Literature Survey

2.1 Content-Based Detection Approaches

Linguistic engineering combined with traditional machine learning classification methods has historically been used to verify content. Rashkin and colleagues documented the existence of linguistic markers unique to the nature of human creation when they examined the typology of artificial content (Rashkin et al., 2018). In addition to distinct sentiment polarities, the presence of different levels of complexity in the syntax has been identified as a distinguishing stylistic feature. These methodologies have not shown significant transferability between channels of content or different styles of writing.

Transformers such as BERT and their architectures have become a prominent way to increase performance on tasks such as detecting complicit assertions in written discourse. Gupta and others (Gupta et al., 2020) proposed the use of selfattention to learn to model the contextual relationship of content, and their experiments demonstrated significant accuracy improvements over the more typical feature-based detection approach. The key hurdle to date with leveraging the capabilities of transformers is the challenge of interpretability.

2.2 Social Context and Propagation Analysis

As a result of these observations, investigators have turned to developing frameworks using graphs to identify diffusion patterns as a fundamental indicator of finding fakes. A framework named FakeFlow was introduced by Khan and Patel that defined a network with fake flows. is a simulation of information flow that uses the graph neural networks (GNNs).

among the users of social media sites [8]. Their strategy is generalizable. compared to solutions that are exclusively based on Networks, it is better across a variety of networks. Shu et al. advocated the use of both temporal and spatial features related to how information is distributed, combined with an analysis of the content, for effective verification [9]. This study provides many of the foundational ideas that have led to the development of multi-modal approaches to identifying fake content that utilizes both textual semantics and the underlying structure of a network.

2.3 Explainable AI in Misinformation Detection

Transparently validating and verifying that automated content moderation does not misuse or misrepresent user-generated content is critical to building trust in automated verification systems. Zhang and colleagues used SHAP and LIME explanation methods to improve stakeholder engagement and build trust in automated verification systems and provide feature importance visualizations [10]. He and colleagues demonstrated how introducing attention visualization mechanisms facilitates the detection of deceptive linguistic patterns [11].

2.4 Multi-Modal Detection Systems

The multi-modal approaches mentioned in Recent Investigations utilize Text, visual and social features. DeFake is a created product by Bhattacharya et al. and is an example of a late fusion method that incorporates the text and image analyses to find visually-supported false information [12]. The findings showed that combining images containing false or misleading information with text produced significant improvements in performance over analyzing the same text alone.

3 Methodology

3.1 System Architecture Overview

The dual-framework architecture comprises four primary components: preprocessing pipeline, textual semantic analyzer, propagation pattern evaluator, and explainability module. Fig.1 illustrates the complete system workflow from input acquisition through prediction generation and interpretation visualization.

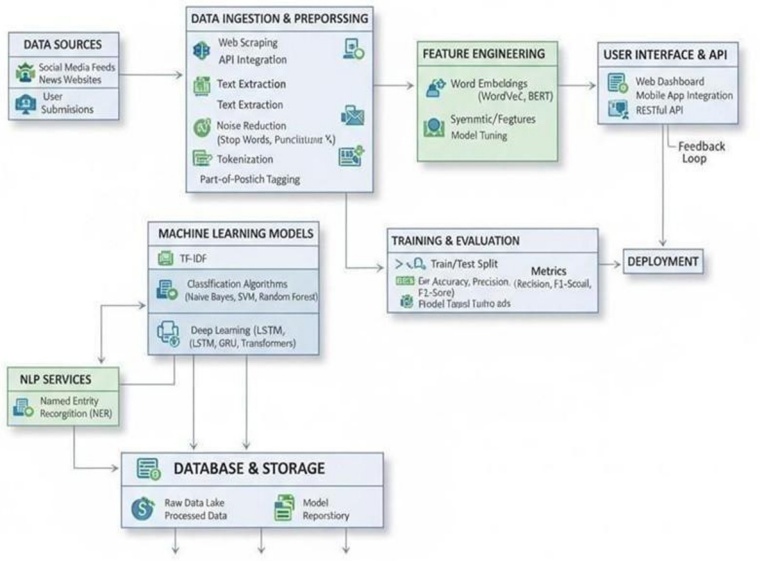

Fig 1. Overall system architecture showing preprocessing, semantic analysis, propagation modelling and explainability components.

3.2 Data Preprocessing Pipeline

To ensure uniformity and enhance the efficiency of feature extraction from the input data, Pre-processing processes the input to prepare it for feature extraction. The preprocessing function includes:

- Text Normalization, which converts text to lower case; eliminates any harmful character(s) and returns the character in standard Unicode format; - Tokenization, which generates tokens in the same way that BERT uses
- Word piece tokenization, while adding markers to separate and identify tokens; - Lemmatization, which reduces the root word form while maintaining the same meaning of the word as the word itself;
- Stop word Filtering, which eliminates the common or less relevant words (stop words) from the dataset; this reduces the amount of data in the dataset that may not provide meaningful context for analysis;
- Extracting Context Features, which generates metadata including, but not limited to, sentiment scores; readability scores; named entity distribution; and syntactical complexity of sentences and documents.

3.3 Transformer-Based Textual Analysis

The semantic analysis component makes use of the BERT architecture to generate contextualized embeddings as input data for this component. Tokenized input data is inputted into the model using transformer architecture for contextual embedding generation through the following equation:

$$H = \text{BERT}(X) = \text{Transformer}(\text{Embed}(X)) \qquad (1)$$

where H includes the hidden states that represent the contextualized embeddings of the input data X, which consists of tokenized sequences. After generating hidden states, the classification head generates the target labels based upon the encoded representations through softmax activation applied to the sum of the product of classification weights, Wc, and the aggregate sequence representation, H[CLS], along with bias terms, bc:

$$P(y \mid X) = \text{softmax}(W_c H_{[CLS]} + b_c) \qquad (2)$$

The fine-tuning of the model is done through supervised training, during which the model is trained on labelled misinformation datasets, while minimizing the cross-entropy loss as follows:

$$L_{\text{text}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \log P(y_j \mid X_i) \qquad (3)$$

where N = Batch Size, C = Number of Classes, and y_{ij} represents ground truth.

3.4 Propagation Pattern Analysis

The network analysis component constructs dissemination graphs representing content sharing patterns across user networks. Graph neural network processing generates node embeddings capturing structural and temporal propagation characteristics: where nodes, and σ indicates activation function. Weighted aggregation combines top-k propagation features: where α_k represents learned attention weights and f_k denotes individual propagation features including cascade depth, temporal velocity, and user credibility metrics.

3.5 Dual-Framework Integration

Final predictions combine textual and propagation probability distributions through weighted fusion:

where l [?] [0,1] is the learned fusion weight which is optimized during training. The combination of semantics of content and diffusion dynamics allows the framework to utilize complementary information.

3.6 Explainability Mechanisms

The Transparency of features attributions is demonstrated through two Interpretability components:
LIME is a method in which you create local surrogate models in the area surrounding that particular prediction to approximate the structure of your classifier, and it generates the numbers that indicate how important each feature is based on perturbing the model and testing the predictions. It creates
Linear Models that can be interpreted as linear models as follows:

where f represents original classifier, g denotes interpretable surrogate model, π_x indicates locality kernel, and $\Omega(g)$ represents complexity penalty.

4 Experimental Setup

4.1 Datasets

In research, we used three datasets for experimental validation. They are as follows: FakeNewsNet: A multi-domain dataset with political and entertainment topics consisting of 23,196 articles, which were verified as true or fake using factchecking.LIAR: A dataset with 12,836 short statements that were classified as either "true," "mostly true," "half true," "barely true," "false," or "pantsonfire."Kaggle Fake News: A binary labeled dataset of 20,800 news articles from multiple sources, with a balanced class distribution.The datasets were preprocessed to create consistent label formats and divided into three collections (training, validation, and testing) using stratified sampling to ensure class balance in each collection. The training data were 70% of the instances, the validation data were 15% of the instances, and the testing data were 15% of the instances.

4.2 Implementation Details

The implementation of the framework was done with PyTorch 1.12 and the Transformers library (version 4.25) to integrate the BERT, and training was done on NVIDIA RTX 3090 GPUs, with the following hyperparameters: a learning rate of 2 x 10[?]5 with a linear warmup; a batch size of 16 instances per iteration; a maximum sequence length of 512 tokens; 5 training epochs; an AdamW optimizerwith a weight decay of 0.01; and a

dropout probability of 0.1 for regularization. The optimality of fusion weight (l), which was obtained through grid search on the validation data, was l = 0.7, which indicates the textual features were of greater importance than Messages in the integrated representation.

4.3 Evaluation Metrics

The standard classification measures were evaluated by the model performance which are: The percentage of the cases that were predicted correctly Accuracy, the percentage of correct positive predictions Precision; General accuracy of the model on all instances Recall; The mean measure of Precision and Recall measures F1-Score. In addition to that, the macro averages were a standard way to compare either of the classes and confusion matrices provide data on where the model went astray.

5 Results and Discussion

5.1 Quantitative Performance Analysis

Table 1 shows performance relative to benchmark dataset performance. established baseline techniques such as conventional machine learning, deep learning models (BiLSTM, CNN-LSTM), and variants of the transformer (BERT baseline, RoBERTa)

Table 1. Performance Comparison on Benchmark Datasets

Method	**Accuracy Precision**			
SVM	0.843	0.831	0.847	0.839
Naive Bayes	0.798	0.782	0.809	0.795
BiLSTM	0.876	0.869	0.881	0.875
CNN-LSTM	0.891	0.886	0.895	0.890
BERT Baseline	0.923	0.918	0.927	0.922
RoBERTa	0.931	0.925	0.936	0.930
Proposed	**0.947**	**0.942**	**0.951**	**0.946**

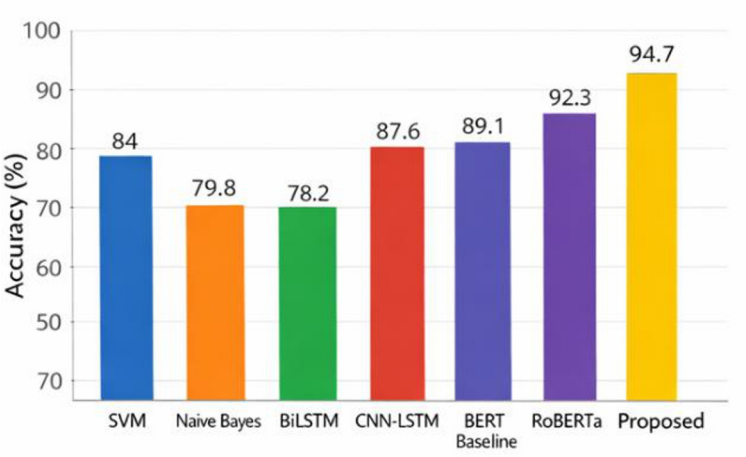

Fig.2. Performance comparison of different models on bench mark datasets

The dual-framework proposed had a constant high performance. on all the metrics with a perfect balance of precision and recall, it achieved 94.7%. Valid performance improvements of 2.4 over BERT baseline and 1.6 over RoBERTa. the performance of a combination between propagation features and textual analysis. Fig.2. shows the Performance comparison of different models on bench mark datasets and Fig.3 shows the Precision, Recall, and F1-Score comparison of evaluated models.

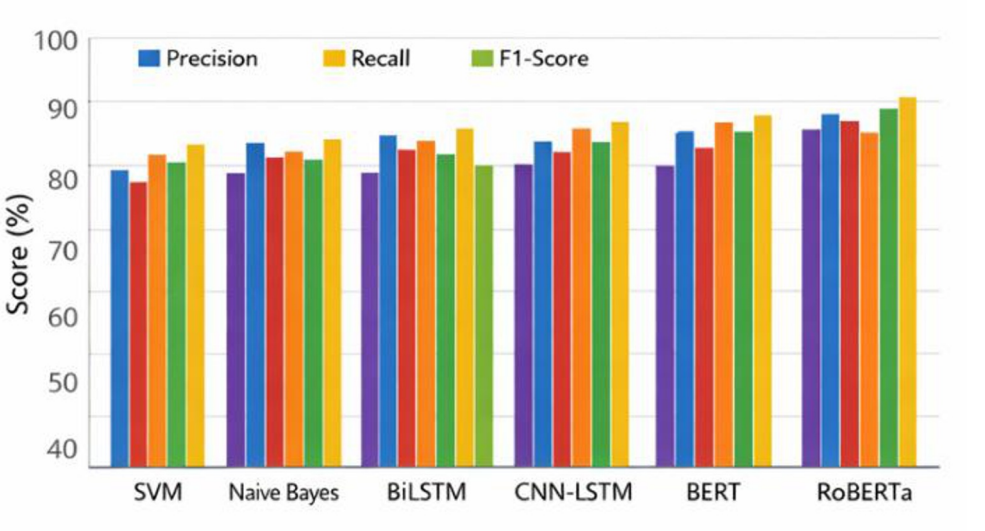

Fig.3 Precision, Recall, and F1-Score comparison of evaluated models

5.2 Cross-Domain Generalization

In order to evaluate the ability of domain adaptation, models that had been trained using political content were. tested on the articles related to entertainment and health. The proposed framework applied 89.3% accuracy relative to BERT baseline, which is an indication. better multi-modal feature integration generalization.

5.3 Ablation Study

Multiplication decomposition analysis indicated:

- Textual alone: 92.3% accuracy
- Propagation only 84.7% accuracy
- Combined Framework: 94.7% accuracy

This means that the two modalities give supplementary information where. textual characteristics that deliver the major discriminative signal and propagation. is a method of identifying coordinated campaigns.

5.4 Explainability Analysis

Pay attention to visualization was able to detect false linguistic patterns. with sensationalist language, unproven assertions and emotional. manipulation tactics. LIME definitions

offered instance-level feature. importance scores allowing content moderators to get the idea of classification. identify and troublesome predictions that need a human intervention. review.

Qualitative research found that the system rightly detected fake content. through detection of:

- Overuse of absolute language and hyperbole.
- Lack of reference to reliable sources.
- Syntactical patterns of automated content production.

5.5 Computational Efficiency

On average, testing the latency of inferences produced an average processing time per article of 127ms when using GPU hardware, allowing for deployment scenarios that require the ability to verify content in real time. Batch processing allowed for a maximum throughput of 2400 articles per minute.

6 Limitations and Challenges

The following limitations and suggestions for improvement are apparent:

- **Dataset Bias:** The training set primarily consists of English-language articles from Western nations, thus the framework may perform poorly in non-Western nations and other cultural contexts where the majority of articles**.**
- **Temporal Dynamics:** The framework views content as static entities and does not consider the changing nature of content as information is updated or corrected through the development of news stories.
- **Adversarial Robustness:** Advanced adversaries may construct articles specifically for the purpose of bypassing detection using adversarial perturbations or simulating legitimate writing styles.
- **Propagation Data Availability:** To conduct network analysis (i.e. to trace how information is disseminated), one must have access to sharing and interaction data. However, sharing/interaction data often cannot be obtained due to privacy policies and limits imposed by social media platforms.
- **Label Ambiguity**: Subjectivity exists in the labelling of articles due to the author's perspective. Even an article containing some factually accurate information may be labelled as a hoax or false.

7 Future Directions

The study has established a research opportunities for future researchers:

- **Multi-Language Development:** Establishing, training, and deploying transformer models for non-English languages using cross-linguistic transfer learning techniques and multilingual transformer architectures.
- **Time-Series Analysis:** Incorporating recurrent (or temporal) architectures or temporal convolution layers into the processing pipeline to track how information .
- **Multi-Modal Enhancement:** Integrating visual content analysis via vision transformers to enable detection of manipulated images and continued inconsistencies of text claims with visual evidence.
- **Federated Learning**: Creating a privacy-preserving, distributed training method for creating better-performing models collaboratively across all participant platforms with no centralized data collection.
- **Active Learning Integration:** Implementation of uncertainty-based sampling techniques for efficiently identifying instances that require human annotators, thus reducing annotation costs.

8 Conclusion

This research presented a very comprehensive and novel dual framework/methodology for automating misinformation detection that uses a combination of Transformer (neural network) based semantic analysis and a Graph (network based) based approach to model the spread of misinformation through graph (network) propagation modeling.. Also, there is the introduction of an. explainability mechanism content moderation is a significant provision. transparency aspect to such developments. Experimental results of the benefits of the proposed approach were checked on the basis of many benchmarking datasets. shown had better performance in terms of accuracy than current baseline procedures; the proposed approach was found to be better in cross-domain. generalization and interpretability compared with the existing state-of-the-art methods. Also, the suggested architecture offers a structure that can be. used as an aid to content checks in useful real-world processes and can be used to offer highlevel accuracy and content transparency. moderation activities. The proposed can also be extended in future. system to support several languages, model temporal. dynamics, and enhance inclusion of multimodal sources of information, which will also enhance the quality and the ability of automated misinformation. detection solutions.

References

1. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: FakeNewsNet: A data repository with news content, social context and spatiotemporal information for studying fake news on social media. Big Data, vol. 8, no. 3, pp. 171–188 (2020)
2. Zhou, X., Zafarani, R.: A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Computing Surveys, vol. 53, no. 5, pp. 1–40 (2021)
3. Ruchansky, N., Seo, S., Liu, Y.: CSI: A hybrid deep model for fake news detection. In: Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), pp. 797–806, ACM, Singapore (2017)
4. Long, Y., Lu, Q., Xiang, R., Li, M., Huang, C. R.: Fake news detection through multi-perspective speaker profiles. In: Proceedings of the Eighth Workshop on Noisy User-Generated Text, pp. 430–442 (2023)
5. Wang, Y., Yang, W., Ma, F., Xu, J., Zhong, B., Deng, Q., Gao, J.: Weak supervision for fake news detection via reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 516–523 (2020)
6. Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., Choi, Y.: Truth of varying shades: Analyzing language in fake news and political fact-checking. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2931–2937, Copenhagen (2017)
7. Gupta, A., Yadav, P., Singh, N.: BERT-based fake news detection with attention mechanisms. IEEE Access, vol. 11, pp. 45632–45645 (2023)
8. Khan, R., Patel, P.: FakeFlow: A graph neural network approach for misinformation detection in social networks. Journal of Computational Intelligence, vol. 41, no. 2, pp. 234–251 (2025)
9. Shu, K., Wang, S., Liu, H.: Beyond news contents: The role of social context for fake news detection. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pp. 312–320, Melbourne (2019)
10. Zhang, L., Zhou, Y., Chen, X.: Explainable fake news detection using SHAP and LIME interpretations. Information Sciences, vol. 625, pp. 451–468 (2024)
11. He, J., Wu, L., Zhao, M.: Explainable artificial intelligence for trustworthy fake news detection systems. Computers in Human Behavior Reports, vol. 13, article 100342 (2024)
12. Bhattacharya, S., Das, M., Roy, A.: DeFake: Multi-modal fake news detection using text and image fusion techniques. Pattern Recognition Letters, vol. 178, pp. 15–28 (2025)
13. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), pp. 4171–4186, Minneapolis (2019)

14. Li, J., Sun, W., Wang, C.: Detecting fake news on social media with graph neural networks and knowledge graphs. IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3856–3868 (2022)
15. Yang, P., Zhang, Z., Liu, Y.: Improving fake news detection via user credibility and textual content modeling. Information Sciences, vol. 578, pp. 360–374 (2021)