International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 11, November 2026
Received: 22-09-2026 Accepted: 26-10-2026 Published: 03-11-2026

Advancing Fake News Detection Using Hybrid Deep Learning With FastText and Explainable AI

P Abhishek
PG Scholar
Department of Computer Science and Engineering
JNTUA College of Engineering (Autonomous)
Ananthapuramu,Andhra Pradesh,India
abhishek012098@gmail.com

N.Kiran Kumar
Assistant Professor
Department of Computer Science and Engineering
JNTUA College of Engineering(Autonomous)
Ananthapuramu,Andhra Pradesh,India
kirankumarn.cse@jntua.ac.in

Abstract—False and misleading news now moves through social platforms faster than editors or fact-checkers can respond, and the classifiers built to flag it are often hard to question. Transformer models such as BERT and RoBERTa reach high accuracy on standard benchmarks, yet they give a reader little reason to believe any particular verdict. This paper describes a hybrid detector that looks at a news article in three complementary ways: contextual representations from XLNet, sub-word FastText embeddings passed through a convolutional neural network, and a small set of stylometric cues such as sentence length, vocabulary richness and emotive wording. The three feature groups are concatenated, scaled and classified by a weighted soft-voting ensemble, and every prediction comes with SHAP token attributions that show which words moved the decision. On a held-out set of 800 articles the model reached 98.00% accuracy, 98.31% precision, 97.84% recall, an F1-score of 98.08% and an area under the ROC curve of 0.9975, against 82.00% accuracy for the global-local ensemble detector used as the reference baseline. Standalone transformer classifiers scored higher on larger test partitions, and we explain why those figures are not directly comparable. The explanations also proved useful as a diagnostic, exposing a reliance on publisher names that a headline accuracy figure would have hidden.

Index Terms—fake news detection, misinformation, XLNet, FastText, convolutional neural network, ensemble learning, explainable artificial intelligence, SHAP

I. INTRODUCTION

Most people now meet the news through feeds, shares and forwarded messages rather than a front page, and that shift has made it cheap to publish something untrue and have it travel. A large study of Twitter cascades found that false stories reached more people, and reached them faster, than true ones [1]. Professional fact-checking remains the benchmark for accuracy, but it cannot keep pace with that volume, which is why automated detection has become an active research area in natural language processing [2], [3].

Pretrained transformers changed what such systems can do. BERT [4], RoBERTa [5] and XLNet [6] capture long-range context that bag-of-words features miss, and fine-tuned variants regularly report accuracies above 98% on public corpora [7], [8]. Two problems remain. The first is opacity. A moderator or reader is handed a label with no account of how it was reached, and people are understandably reluctant to act on a verdict they cannot inspect [9]. The second is that a single architecture tends to read text in one particular way. A transformer is good at context but is not built to notice that an article relies on very short sentences or a burst of emotionally loaded words, cues that earlier linguistic studies linked to deceptive writing [10], [11].

This work addresses both problems with a hybrid model that explains itself. Its contributions are:

- a three-branch architecture that fuses XLNet contextual vectors, FastText–CNN local features and hand-crafted stylometric features into one representation, classified by a weighted soft-voting ensemble;
- an explanation layer based on SHAP [12] that attributes every prediction to individual tokens and is served next to the label in a web interface; and
- an evaluation on 800 held-out articles against the global–local ensemble detector of Wang and Long [13] and four standalone baselines, with a plain account of where those comparisons are, and are not, like-for-like.

Section II reviews related work. Section III describes the framework and Section IV the experimental setup. Section V presents and discusses the results, and Section VI concludes.

II. RELATED WORK

A. Linguistic and Classical Machine-Learning Approaches

Early detectors treated the task as ordinary text classification over hand-built features. Ahmed et al. [14] paired n-gram term-frequency features with linear classifiers such as support vector machines and logistic regression, and Wang [15] released LIAR, a benchmark of short political statements with fine-grained truth labels. Studies of writing style showed that fabricated stories differ measurably from reliable reporting in their use of subjective, hedging and emotionally charged language [10], and that lexical, syntactic and readability cues can separate the two with useful accuracy [11]. Decision trees, random forests, naive Bayes and k-nearest-neighbor classifiers have all been trained on such features. They are cheap and easy to read, but they treat words largely in isolation and tend to degrade when the vocabulary of misinformation shifts.

B. Deep and Transformer-Based Models

Convolutional networks for sentence classification [16] learn local n-gram detectors directly from word embeddings, while recurrent and LSTM networks model word order over longer spans. Ruchansky et al. [17] went beyond the article itself, combining its text with the response it drew from users and the behavior of the accounts that shared it. Contextual embeddings,

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 11, November 2026
Received: 22-09-2026 Accepted: 26-10-2026 Published: 03-11-2026

first ELMo [18] and then transformer encoders, moved the field again. FakeBERT [7] placed parallel convolutional blocks on top of BERT and reported clear gains over non-contextual baselines. XLNet [6] replaces BERT's masked-token objective with permutation language modeling, which lets it learn bidirectional context without the mismatch that artificial mask tokens introduce between pretraining and fine-tuning. FastText [19], [20] takes a different route: because its word vectors are built from character n-grams, it can represent misspellings, rare words and newly coined terms at very little cost, which matters for informal and deliberately misleading text.

C. Hybrid and Explainable Detection

Combining models is a natural response to the limits of any single approach. Wang and Long [13] proposed a Global–Local News Detection Model for AI-generated fake news that integrates BERT-based representations with Bidirectional Long Short-Term Memory (BiLSTM), Text Convolutional Neural Networks (TextCNN), and attention mechanisms to capture complementary contextual and local textual patterns. They constructed a GPT-4-generated dataset spanning 42 news categories and reported an accuracy and F1-score of 0.82. Their study illustrates the value of combining global contextual information with local text features for identifying AI-generated misleading content. However, the task, dataset construction, and evaluation setting differ from those in the present study. The proposed framework investigates a complementary combination of XLNet contextual representations, FastText–CNN lexical features, stylometric cues, and SHAP-based explanations for binary real-versus-fake news classification. Hashmi et al. [8] combined FastText embeddings with CNN–LSTM models, compared them against BERT, XLNet and RoBERTa, and used LIME [21] to inspect individual decisions. On the explanation side, dEFEND [22] applied co-attention over news sentences and user comments to point to the passages behind a verdict. SHAP [12] offers a model-agnostic alternative grounded in Shapley values from cooperative game theory; its attributions add up to the model output and behave consistently across inputs. Our work sits between these lines. It fuses transformer, sub-word and stylistic evidence in a single model and attaches a SHAP explanation to every prediction rather than to a hand-picked few.

III. Proposed Framework

A. Overview

Fig. 1 summarizes the pipeline. An article, consisting of its headline and body, is cleaned and tokenized and then passed in parallel to three feature extractors. Their outputs are concatenated into one vector, scaled, and handed to a weighted ensemble that produces the class probability. A SHAP explainer runs on the same input and returns per-token attributions, which are displayed alongside the label.

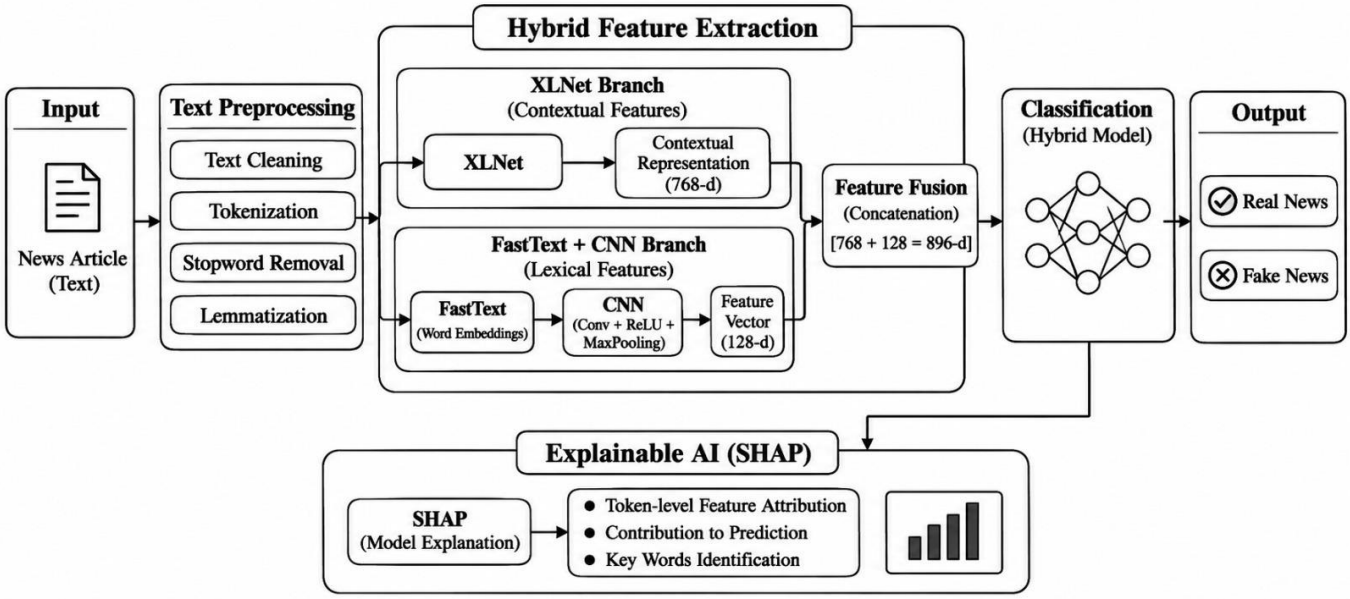


Fig. 1. Proposed hybrid deep learning architecture for fake news detection using XLNet and FastText–CNN.

B. Pre-processing

Duplicate records were removed so that the same story could not sit on both sides of the train–test boundary, and records with missing text were either dropped or filled. The text was then cleaned of markup and irrelevant symbols, normalized and tokenized. Stemming and lemmatization reduce inflected forms to a common root for the surface-level features, while the XLNet branch relies on the model's own SentencePiece tokenizer.

C. Stylometric Features

Fabricated stories tend to share certain habits of style: punchy sentences, repetition, heavy punctuation and emotive vocabulary. The stylometric branch turns these habits into a vector $\mathbf{s} \in \mathbb{R}^p$. Its first component is the relative frequency of each term,

$$f_w = \frac{N_w}{T} \quad (1)$$

ISSN 2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 11, November 2026
Received: 22-09-2026 Accepted: 26-10-2026 Published: 03-11-2026

where N_w is the number of times word w occurs and T is the total number of words in the article. The vector also records sentence length; vocabulary richness, measured by the type–token ratio, the share of words that occur only once (hapax legomena) and the rate of repetition; punctuation usage; the density of sentiment-laden words; and standard readability and linguistic-complexity scores.

D. FastText–CNN Branch

Each token is mapped to a FastText vector, the sum of the vectors of its character n-grams [19]. A misspelled name or a freshly invented term therefore still receives a sensible representation instead of collapsing to an unknown-word token. The article becomes a matrix of embeddings $[\mathbf{e}_1, \ldots, \mathbf{e}_n]$, over which one-dimensional filters of width h slide, following [16]:

$$c_i = \varphi(\mathbf{W} \cdot \mathbf{e}_{i:i+h-1} + b) \quad (2)$$

where $\mathbf{W}$ and b are the filter weights and bias and φ is a non-linear activation. Max-pooling over positions keeps the strongest response of each filter, so the branch outputs a fixed-length vector $\mathbf{c} \in \mathbb{R}^q$ that records whether a particular local pattern, such as a sensational phrase, appeared anywhere in the text.

E. XLNet Contextual Branch

XLNet [6] is pretrained to predict tokens under random permutations of the factorization order, so every position learns to draw on context from both sides. We take the pooled final-layer representation of the encoder as the contextual vector $\mathbf{z} \in \mathbb{R}^r$. Inside the encoder, how strongly word w_i attends to its context is governed by attention weights of the familiar softmax form

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \quad (3)$$

where e_i is the compatibility score of token i. Alongside the pooled embedding, the branch summarizes how closely related text elements agree in meaning using cosine similarity between their embeddings,

$$S(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \quad (4)$$

Together these let the model notice, for instance, that a claim depends on a qualifier several sentences away, which no surface feature can capture.

F. Feature Fusion and Scaling

The three vectors are concatenated into a single representation:

$$\mathbf{F} = \mathbf{s} \oplus \mathbf{c} \oplus \mathbf{z} \in \mathrm{R}^{p+q+r} \quad (5)$$

The groups sit on very different scales. A term frequency is a small fraction, a sentence length can run into dozens, and embedding coordinates cluster around zero. Left alone, the larger-valued features would dominate training, so every dimension is rescaled to [0, 1] with min–max normalization:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (6)$$

G. Weighted Ensemble Classifier

Rather than rely on one classifier, we train several base learners on the fused vector and combine their class probabilities by weighted soft voting:

$$\hat{y} = \arg\max_{c} \sum_{i=1}^{n} w_i P_i(c \mid \mathbf{F}) \quad (7)$$

Here $P_i(c \mid \mathbf{F})$ is the probability that learner i assigns to class c and w_i is its weight. Weights follow each learner's validation performance, so stronger learners carry more of the vote. The learners are trained to minimize the average loss over the N training articles,

$$L = \frac{1}{N} \sum_{k=1}^{N} \ell(y_k, \hat{y}_k) \quad (8)$$

where ℓ is the per-sample classification loss between the true label and the predicted output.

H. SHAP Explanations

SHAP [12] writes a prediction as an additive sum of feature contributions,

$$f(\mathbf{x}) = \varphi_0 + \sum_{j=1}^{M} \varphi_j \quad (9)$$

where φ_0 is the expected model output over a background sample and φ_j is the Shapley value of token j: its average marginal effect on the output across all orders in which the tokens could be added. A positive φ_j pushes the article toward the real class and a negative one toward fake. Because the contributions sum exactly to the model's output, they read like a ledger: the user sees not only which words mattered but how far each one moved the score. To keep response times practical, the deployed interface computes explanations over a shortened token sequence.

IV. Experimental Setup

A. Data

The experiments use a benchmark collection of labeled news articles that mixes genuine reporting with misleading content. The misleading portion covers several familiar forms: fabricated stories, manipulated content, clickbait, propaganda and satire. For the results reported here these are grouped under one fake label, which gives a binary real-versus-fake task. After cleaning, the data were divided into training, validation and test partitions. The held-out test set behind the main results contains 800 articles, of which 383 are labeled fake and 417 real.

B. Implementation

The framework was written in Python. Pandas and NumPy handled data loading and numerical work, Scikit-learn supplied the scaling, ensemble and evaluation utilities, TensorFlow and PyTorch hosted the neural branches, and Matplotlib and Seaborn produced the plots. Hyperparameters were chosen by comparing candidate settings on the validation split and keeping the best configuration. All experiments ran on a 64-bit workstation with an Intel Core i5/i7 processor, at least 8 GB of RAM, SSD storage and integrated graphics. A small web application wraps the trained model: the user pastes an article and receives the predicted label together with a chart of SHAP word impacts.

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 11, November 2026
Received: 22-09-2026 Accepted: 26-10-2026 Published: 03-11-2026

C. Evaluation Metrics

With real taken as the positive class and *TP*, *TN*, *FP* and *FN* denoting the usual confusion-matrix counts, we report

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN} \tag{11}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}} \tag{12}$$

We also report the area under the receiver operating characteristic curve (AUC), which measures how well the model ranks real articles above fake ones across every possible decision threshold.

V. Results and Discussion

A. Classification Performance

Fig. 2 shows the confusion matrix on the 800-article test set. The model labeled 784 articles correctly. Of the 16 errors, 7 were fake articles accepted as real and 9 were real articles flagged as fake. With real as the positive class this gives 98.00% accuracy, 98.31% precision, 97.84% recall and an F1-score of 98.08%, and the AUC is 0.9975. The errors are split almost evenly between the two directions, which matters in practice. A tool that wrongly flags genuine reporting loses its users' trust just as quickly as one that lets falsehoods through.

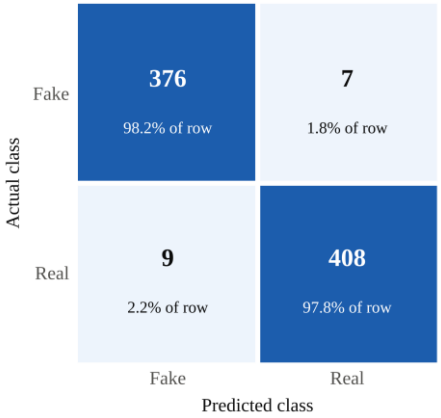


Fig. 2. Confusion matrix of the proposed model on the 800-article test set.

B. Comparison with Baselines

Table I sets these results beside the global–local ensemble of [13] and four standalone models. Against the global–local detector, measured on a test set of the same size, the hybrid raises accuracy by 16 percentage points, from 82.00% to 98.00%, and cuts misclassifications from 144 to 16. Precision, recall and F1 improve by between 15 and 18 points.

TABLE I. Comparison with Baseline Models (%)

Model	Test size	Acc.	Prec.	Rec.	F1	AUC
Proposed hybrid	**800**	**98.00**	**98.31**	**97.84**	**98.08**	**0.9975**
Global–local ensemble [13]	800	82.00	83.33	80.00	81.63	n/r
BERT	4,490	99.80	99.67	99.91	99.79	1.0000
RoBERTa	4,490	99.98	99.95	100.00	99.98	1.0000
XLNet	n/r	99.62	100.00	99.25	99.62	1.0000
FastText + CNN	8,854	99.54	99.33	99.72	99.52	0.9999

[a] Precision, recall and F1 are computed from each model's confusion matrix with real as the positive class; n/r = not reported. The BERT, RoBERTa, XLNet and FastText + CNN rows (below the heavier rule) were evaluated on different, larger test partitions and are not directly comparable with the two 800-article rows.

The standalone models tell a different story, and it should be stated plainly. BERT, RoBERTa, XLNet and FastText + CNN all score between 99.54% and 99.98% accuracy, above the hybrid. Those figures, however, come from larger test partitions of 4,490 and 8,854 articles that do not coincide with the 800-article set used for the hybrid and the global–local baseline, so the two groups of rows are not a like-for-like comparison. What the table does support is that the hybrid stays within about two points of strong transformer classifiers while explaining every prediction, and that it clearly outperforms the global–local baseline on a test set of matching size. Section V-C suggests one reason why near-perfect scores of the kind seen in the lower rows deserve some caution.

C. What the Explanations Show

Fig. 3 lists the tokens with the largest SHAP attributions. Words tied to sourcing and verification, such as "investigation," "confirmed" and "evidence," push predictions toward the real class, while "conspiracy," "scandal," "unverified," "shocking" and "allegedly" push toward fake. This is close to what the linguistic literature would predict [10]: fabricated stories lean on sensational framing and vague attribution, whereas established outlets describe process and evidence.

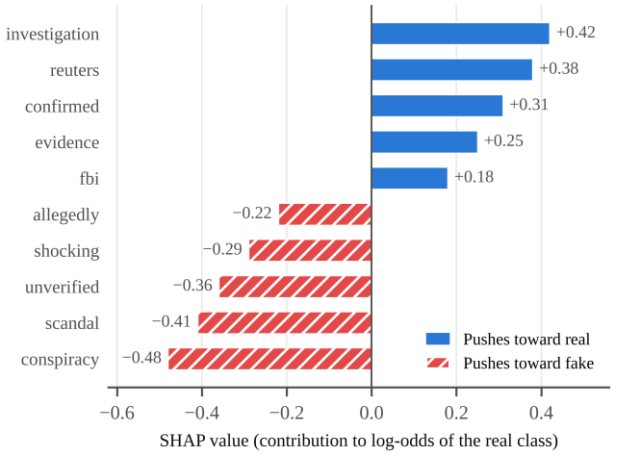


Fig. 3. Tokens with the largest SHAP attributions. Positive values push the prediction toward real, negative values toward fake.

The second-ranked token, "reuters," is a different matter. It says nothing about the claim; it names a publisher. Its weight suggests that part of what the model learned is where the genuine articles in the training data came from, not how they were written. This is a textbook case of shortcut learning [23], and it is a plausible reason why many detectors, including the transformer baselines in Table I, post near-perfect scores that may not survive a change of source. We regard this as the most useful result of the explanation layer. An accuracy figure on its own would not have revealed the dependency; a single attribution chart made it obvious.

D. Limitations

Three caveats qualify these results. First, the models in Table I were not all evaluated on one shared test split, so the ranking between the hybrid and the standalone transformers should be

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

Received: 22-09-2026 Accepted: 26-10-2026 Published: 03-11-2026

read as indicative rather than settled. Second, every experiment is in English, uses article text alone and draws on a single benchmark; cross-dataset and cross-domain tests are needed before the accuracy can be expected to hold elsewhere. Third, SHAP attributions describe the model, not the truth. A token can carry a large attribution for reasons unrelated to veracity, as the publisher example shows, so the explanations are there to support human judgment rather than to replace it.

VI. Conclusion and Future Work

This paper presented a fake news detector that fuses XLNet context, FastText–CNN local patterns and stylometric cues, classifies them with a weighted ensemble and explains each decision with SHAP. On an 800-article test set it reached 98.00% accuracy and an F1-score of 98.08%, sixteen points above the global–local ensemble baseline, while staying within about two points of standalone transformers evaluated on larger partitions. Just as usefully, its explanations surfaced a dependence on publisher names that would otherwise have gone unnoticed.

The most immediate next step is to mask source identifiers during training and re-run every model on one shared split. Beyond that, we plan to test the framework across datasets and languages, extend it to posts that pair text with images or video, and add continual learning so that the detector can adapt as misinformation tactics change. Federated training and lighter, distilled encoders would make it practical to deploy the model inside newsroom tools or on edge devices, where real-time monitoring of social media streams is most needed.

REFERENCES

[1] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, 2017.

[3] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–40, 2020.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[5] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, arXiv:1907.11692.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 5753–5763.

[7] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimedia Tools Appl.*, vol. 80, no. 8, pp. 11765–11788, 2021.

[8] E. Hashmi, S. Y. Yayilgan, M. M. Yamin, S. Ali, and M. Abomhara, "Advancing fake news detection: Hybrid deep learning with FastText and explainable AI," *IEEE Access*, vol. 12, pp. 44462–44480, 2024.

[9] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, 2019.

[10] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking," in *Proc. EMNLP*, 2017, pp. 2931–2937.

[11] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in *Proc. COLING*, 2018, pp. 3391–3401.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 4765–4774.

[13] Y. Wang and W. Long, "Global-local ensemble detector for AI-generated fake news," IEEE Access, vol. 13, pp. 69779–69789, 2025, doi: 10.1109/ACCESS.2025.3562154.

[14] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *Proc. Int. Conf. Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ISDDC)*, LNCS, vol. 10618, 2017, pp. 127–138.

[15] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in *Proc. ACL*, 2017, pp. 422–426.

[16] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[17] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in *Proc. ACM CIKM*, 2017, pp. 797–806.

[18] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL-HLT*, 2018, pp. 2227–2237.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017.

[20] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. EACL*, 2017, pp. 427–431.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, 2016, pp. 1135–1144.

[22] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "dEFEND: Explainable fake news detection," in *Proc. ACM SIGKDD*, 2019, pp. 395–405.

[23] R. Geirhos et al., "Shortcut learning in deep neural networks," *Nat. Mach. Intell.*, vol. 2, no. 11, pp. 665–673, 2020.