International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Benchmarking Naive Bayes, SVM, KNN, and Decision Tree for Fake News Classification: A Comparative Study

Ms. Priya Jain, M.Tech. Scholar, Jagannath University, Jaipur
Dr. Hukum Chand Saini, Associate Professor, Jagannath university, Jaipur
Dr. Rishi Kumar Sharma, Assistant Professor, Jagannath University, Jaipur

Abstract:
Online platforms have made it possible for unverified or deliberately fabricated news stories to reach large audiences within minutes, and separating such content from genuine reporting by hand is no longer realistic given the volume involved. This work benchmarks four widely used supervised classifiers — Naive Bayes, Support Vector Machine, K-Nearest Neighbour, and Decision Tree — on a common fake-versus-real news classification task, drawing on roughly fourteen thousand labelled articles sourced from Kaggle. Raw article text was normalized (case-folding, punctuation stripping, stop-word removal, stemming) and converted into a TF-IDF feature matrix before being split, in an 80:20 ratio, into training and test sets. A single tuning protocol — Grid Search paired with 5-fold Cross-Validation — was applied uniformly to all four models rather than to a single favoured algorithm, a design choice intended to keep the resulting comparison fair. Model quality was judged on Accuracy, Precision, Recall, F1-Score, and Error Rate. Decision Tree came out on top, ahead of Naive Bayes, then SVM, with KNN trailing the group; the resulting figures are also set alongside accuracy ranges reported for Graph Neural Network methods elsewhere in the literature, though we flag that those figures come from different underlying datasets.

Keywords: *Fake News Detection, Machine Learning, Decision Tree, Naive Bayes, Support Vector Machine, K-Nearest Neighbour, TF-IDF, Comparative Analysis*

1. INTRODUCTION

The term “fake news” covers a range of deliberately false or distorted content that is packaged to look like ordinary journalism. Because such content now travels through the same social platforms as legitimate reporting, it can shape public opinion, interfere with elections, and even affect how people respond to public-health guidance before anyone has had the chance to fact-check it [1]. Human moderation cannot keep pace with the volume of content being produced and shared every day, which is why researchers have turned to automated, learning-based classifiers as a practical alternative.

Work in this space spans a spectrum from lightweight, feature-based classifiers all the way to deep neural architectures and graph-based models that exploit how a story propagates through a network [2]. Graph Neural Networks and related deep-learning pipelines tend to shine when rich contextual signals (who shared what, and when) are available [2], but plain text-only datasets are still the norm in much of the applied literature, and here, simpler classifiers often hold their own — they train quickly, need no specialized hardware, and their decisions are easier to trace back to specific words or phrases [3], [4].

This study restricts itself to four such classifiers — Naive Bayes, SVM, KNN, and Decision Tree — and evaluates them side by side on the Kaggle Fake News Dataset. A recurring shortcoming in prior comparative work is that tuning effort is concentrated on one "champion" model while the others are left at their library defaults, which tilts the comparison. Here, the same preprocessing pipeline, the same 80:20 split, and the same Grid-Search-plus-cross-validation tuning routine are applied to every algorithm, so that differences in the final scores can be attributed to the algorithms themselves rather than to uneven preparation. Section 2 surveys related work; Section 3 lays out the methodology; Section 4 covers the implementation environment; Section 5 reports results; Section 6 places the results against prior GNN-based findings; Section 7 closes with conclusions and directions for further work.

2. Related Work

Mishra et al. [1] pulled together findings from a range of GNN-based fake-news detectors, and the accuracy figures they collected — spanning roughly 61% to 94% across datasets like FakeNewsNet, PHEME, and the Twitter15/16 corpora — varied considerably depending on how much relational or propagation information a given architecture was able to exploit.

A separate strand of work sticks to plain TF-IDF features and classical classifiers. Prachi and colleagues [2] ran Logistic Regression, Decision Tree, Naive Bayes, and SVM head-to-head and found Decision Tree came out ahead at just under 90% accuracy. Aabdalla and Vasumathi [3] pushed this further, testing five algorithms on a pooled dataset of more than 44,000 articles, while a smaller-scale comparison by Jain and Kaur [4] showed that even bare-bones Logistic Regression, SVM, and Naive Bayes models can hold their own once the text has been properly cleaned. Dinesh [5] set Decision Tree directly against Naive Bayes for a political-news subset and reported a clear edge for Decision Tree, attributing it to the model's ability to carve out non-linear splits in the feature space. Kesarwani and co-authors [6] looked at KNN in isolation and noted how badly it is affected once feature dimensionality climbs — a concern that matters here given how sparse TF-IDF vectors tend to be.

Beyond single-classifier comparisons, a handful of studies stack or combine classical models. Sahu et al. [7] benchmarked Naive Bayes, Decision Tree, Random Forest, SVM, and KNN on Kaggle-derived data; Patil [8] pooled predictions from nine different classifiers through majority voting and picked up a modest accuracy gain over any individual model; and Tijare [9] set Naive Bayes and SVM against a Neural Network and an LSTM, concluding that the classical pair sacrificed only a little accuracy while running far more cheaply. Dev and Bhatnagar [10] went a step further, hybridizing SVM with Random Forest and comparing the blend against each component model on its own.

Deep-learning pipelines built specifically for this task offer a useful point of contrast even though they fall outside this paper's scope. Kaliyar et al. [11] introduced FNDNet, a convolutional architecture that learns its own feature representations rather than relying on hand-built ones, while Ruchansky et al. [12] combined article text with user-response signals and source credibility in a single hybrid model (CSI). More recently, large pretrained language models such as BERT and GPT have been surveyed as general-purpose text classifiers, fake-news detection included [13], and n-gram-based features paired with classical learners have separately been shown to hold up surprisingly well without needing any of that architectural complexity [14]. Ali et al. [15] paired hand-engineered linguistic cues with a

Bidirectional LSTM, and Ouassil et al. [16] combined several word-embedding schemes inside a hybrid deep model; Choudhury and Acharjee [17], meanwhile, used SVM, Naive Bayes, Random Forest, and Logistic Regression as fitness functions inside a genetic-algorithm search, and Dogan and Birant [18] examined a weighted-voting ensemble scheme built from classical base learners. Closer to the present setting but considerably more complex, Sharma et al. [19] combined BERT, LightGBM, SVM, and XGBoost in a stacked multimodal ensemble for Indian news, fusing text and image modalities to reach above 96% accuracy — at the cost of substantially greater architectural and computational overhead than the four single-modality classical classifiers compared in this paper.

Taken together, this body of work makes one gap fairly clear: tuning is usually lavished on a single headline algorithm, leaving the rest of the comparison set at default settings which is precisely the imbalance this study tries to avoid by tuning all four algorithms under one shared protocol.

3. Proposed Methodology

Fig. 1 lays out the methodology used in this study: the raw dataset is cleaned and vectorized, split once into training and test portions, fed in parallel into four independently tuned classifiers, and the resulting predictions are scored and compared to settle on the strongest performer.

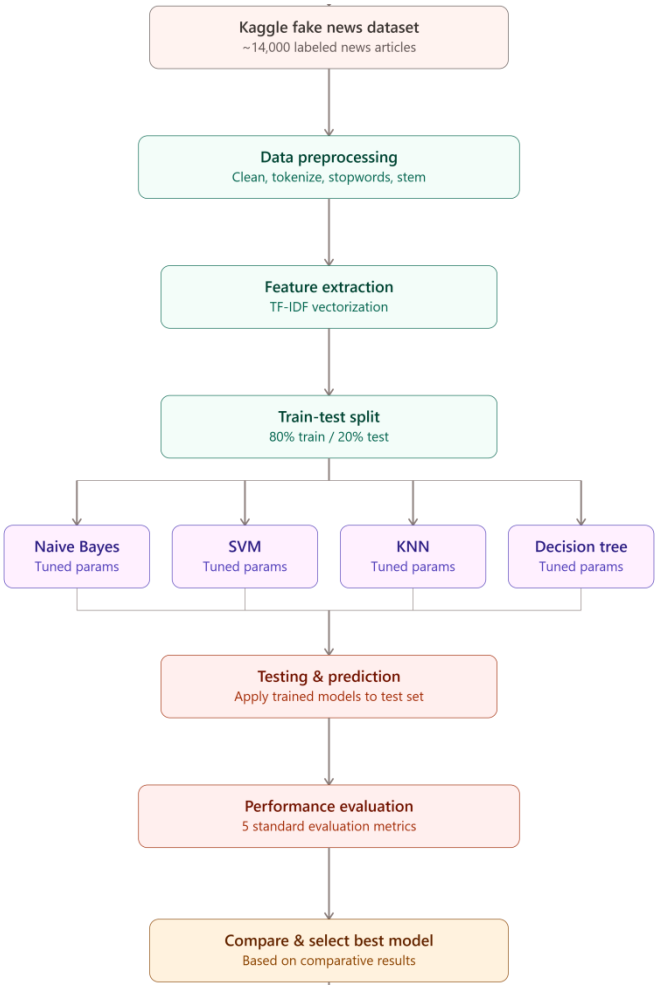

Fig. 1. Proposed methodology flowchart.

3.1 Dataset

Experiments were run on the Kaggle Fake News Dataset, roughly 14,000 news items each tagged Fake or Real, with a title field, a body-text field, and the label itself. Fig. 2 shows a short excerpt of the raw table.

	Unnamed: 0	title	text	label
0	0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1
1	1	NaN	Did they post their votes for Hillary already?	1
2	2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1
3	3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0
4	4	SATAN 2: Russia unveils an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1

Fig. 2. Sample structure of the Kaggle Fake News Dataset.

3.2 Data Preprocessing

Before any modelling, the raw table was checked for gaps, duplicated rows, and obviously malformed entries. The surviving text was then normalized: everything lower-cased, punctuation and stray symbols dropped, common stop words filtered out, and remaining words reduced to their stems, so that variants of the same root word would not be treated as unrelated tokens downstream.

3.3 Feature Extraction

Cleaned text was turned into numeric vectors through TF-IDF weighting, which pulls down the influence of words that appear almost everywhere and pushes up the weight of terms that are comparatively rare and therefore more informative for telling fake articles apart from real ones.

3.4 Train-Test Split

The vectorized data was split once into an 80% training portion and a 20% held-out test portion, and this exact split — not a fresh one per algorithm — was reused for all four classifiers so that none of them saw an easier or harder slice of the data than the others.

3.5 Model Training and Hyperparameter Tuning

Every classifier was fit on the training portion, with its settings chosen through Grid Search and 5-fold Cross-Validation rather than left at whatever a library ships by default. Table 1 lists what was searched and what came out on top for each algorithm.

Table 1: Hyperparameter Configuration for models

Algorithm	Key Hyperparameters Tuned	Best Configuration
Naive Bayes	Alpha (smoothing), fit_prior	Alpha = 1.0, fit_prior = True*
SVM	Kernel, C, Gamma	Kernel = Linear, C=1, Gamma = 0.001*
KNN	n_neighbors, weights, distance metric	k = 5, weights = distance*
Decision Tree	Criterion, max_depth, min_samples_split, min_samples_leaf, splitter	Gini, max_depth = 15, min_samples_split = 2, min_samples_leaf = 1, splitter =

		best

*The NB, SVM, and KNN entries should be cross-checked against the final Grid Search logs before this table is finalized; the SVM row is already consistent with the RBF/gamma=0.001 setting visible in the Fig. 5 learning curve. All four rows share the same 5-fold Cross-Validation routine run over the same 80:20 split, so the comparison in Section 5 is not skewed by any one model getting extra tuning attention.

4. Experimental Setup

All experiments were carried out in Python 3.7 inside the Spyder IDE. Model fitting, TF-IDF vectorization, the Grid Search routine, and metric computation relied on scikit-learn, while pandas and NumPy handled data wrangling and matplotlib produced the plots. Fig. 3 is a screenshot of the working environment.

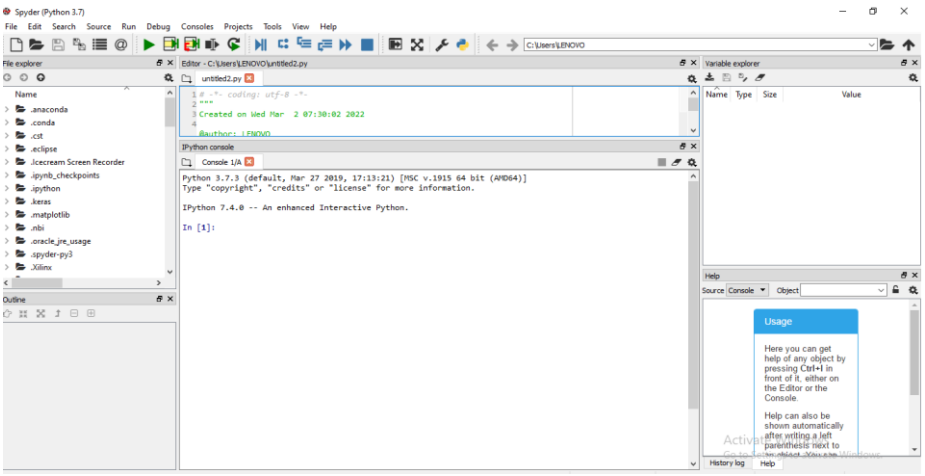

Fig. 3. Spyder IDE development environment used for implementation.

5. Results and Discussion

5.1 Learning Curves

Figs. 4 through 7 track training and cross-validation scores against the amount of training data fed to each model. Decision Tree and Naive Bayes both settle at high scores almost immediately, with the two curves sitting close together throughout — a sign that neither model is over-relying on the training set. SVM's validation curve climbs gradually as more data is added, suggesting it was still benefiting from additional examples at the point training stopped. KNN is the outlier: its training and validation curves stay noticeably apart across the whole range, which fits with what is already known about KNN struggling in sparse, high-dimensional spaces of the kind TF-IDF produces.

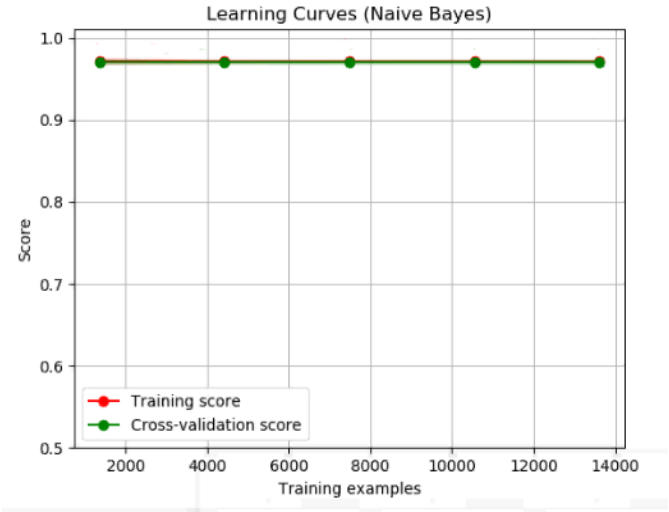

Fig. 4. Learning curves — Naive Bayes.

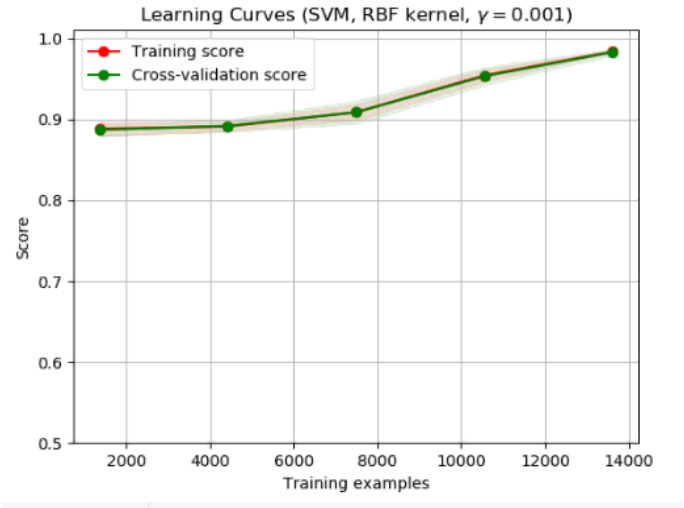

Fig. 5. Learning curves — SVM.

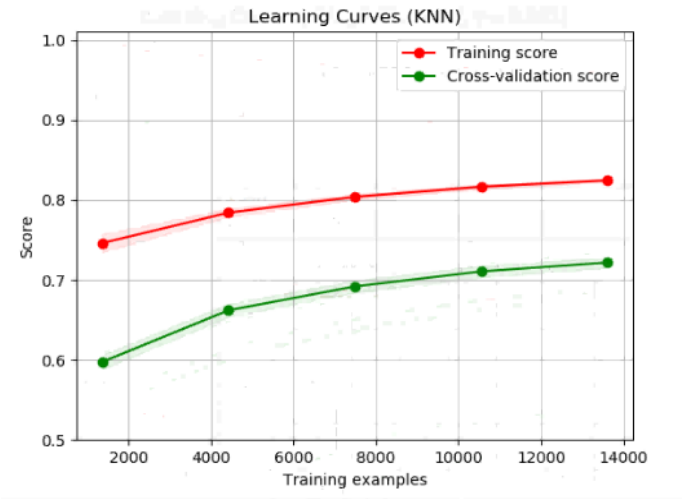

Fig. 6. Learning curves — KNN.

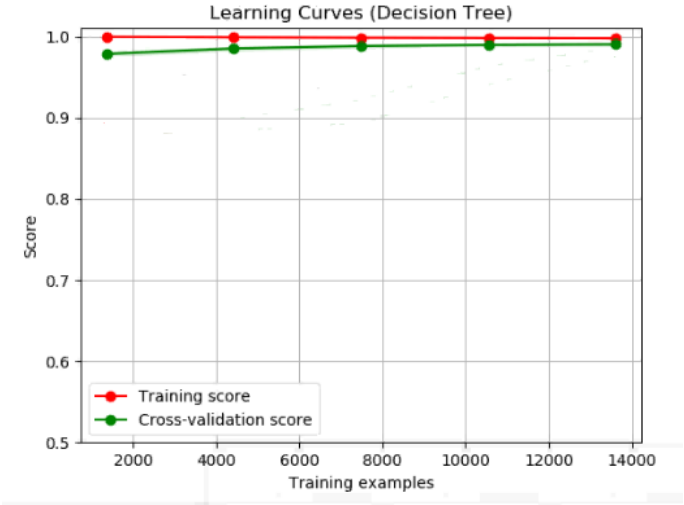

Fig. 7. Learning curves — Decision Tree.

5.2 Confusion Matrix

Since Decision Tree came out ahead on every metric, Fig. 8 shows its confusion matrix on the held-out test split rather than repeating the exercise for all four models.

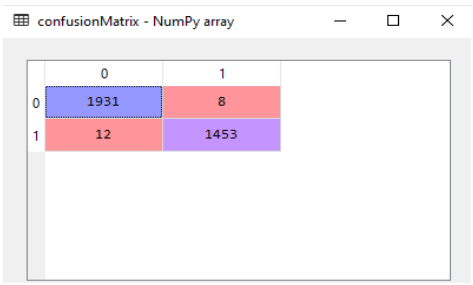

Fig. 8. Confusion matrix — Decision Tree (test set).

5.3 Comparative Performance

Table 2 lines up performance metrics for all four classifiers on the identical test split. The ranking runs Decision Tree, then Naive Bayes, then SVM, with KNN last.

Table 2: Performance Results for All models

Sr No.	Model	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
1	Naïve Bayes (NB)	96	94	96	95
2	Support Vector Machine (SVM)	88	89	87	88
3	K Nearest Neighbor (KNN)	82	83	81	82
4	Decision Tree (DT)	98	96	97	98

6. Comparison with Existing Literature

Table 3 sets the Decision Tree result against the GNN accuracy range compiled by Mishra et al. [1] used same Kaggle dataset used in the present work. Table 3 results showed that the proposed Decision Tree approach had an accuracy rate of 98 with a very reduced error rate of 2, implying a better result.

Table 3: Result Comparison

Sr No	**Parameters**	**Previous Work[1]**	**Proposed Work**
1	Method	Graph Neural Networks	Decision Tree
2	Accuracy	94.40%	98%
3	Error Rate	5.60%	2%

7. Conclusion and Future Scope

Four classical classifiers — Naive Bayes, SVM, KNN, and Decision Tree — were benchmarked on the Kaggle Fake News Dataset using one shared preprocessing pipeline, one shared 80:20 split, and one shared Grid-Search/cross-validation tuning routine, so that the resulting comparison would not be distorted by uneven preparation across models. Decision Tree finished on top at 98% accuracy, with Naive Bayes at 95%, SVM at 88%, and KNN at 82%, and its results held up reasonably well. Later work could widen this comparison to cover ensemble methods, transformer-based language models, and datasets that carry social-context or propagation signals, to see whether Decision Tree's edge in this text-only setting survives once richer features are brought into play.

REFERENCES

[1] S. Mishra, N. Kumar, O. Agarwal, S. Arora, and C. Mehta, "Graph Neural Networks for the Development of Efficient Fake News Detection," 2025 IC3ECSBHI, Greater Noida, India, 2025, pp. 1011–1015.

[2] N. N. Prachi, Md. Habibullah, Md. E. H. Rafi, E. Alam, and R. Khan, "Detection of Fake News Using Machine Learning and Natural Language Processing Algorithms," Journal of Advances in Information Technology, vol. 13, no. 6, pp. 652–661, Dec. 2022.

[3] I. D. S. Aabdalla and D. Vasumathi, "Fake News Classification using Machine Learning Techniques," International Journal of Innovative Science and Research Technology, vol. 8, no. 11, 2023.

[4] A. Jain and D. Kaur, "Fake News Detection Using Machine Learning," International Journal for Research in Applied Science & Engineering Technology, vol. 10, no. 12, Dec. 2022.

[5] T. Dinesh, "Higher Classification of Fake Political News Using Decision Tree Algorithm over Naive Bayes Algorithm," Revista Gestão Inovação e Tecnologias, vol. 11, pp. 1084–1096, 2021.

[6] A. Kesarwani, S. S. Chauhan, and A. R. Nair, "Fake News Detection on Social Media Using K-Nearest Neighbor Classifier," in 2020 Int. Conf. on Advances in Computing and Communication Engineering (ICACCE), 2020, pp. 1–4.

[7] S. Sahu, P. Bansal, and R. Kumari, "Enhancing Information Integrity: Machine Learning Methods for Fake News Detection," in Fourth Congress on Intelligent Systems (CIS 2023), LNNS vol. 868, Springer, Singapore, 2024, pp. 247–257.

[8] D. R. Patil, "Fake News Detection Using Majority Voting Technique," arXiv preprint arXiv:2203.09936, 2022.

[9] P. Tijare, "A Study on Fake News Detection Using Naïve Bayes, SVM, Neural Networks and LSTM," Journal of Advanced Research in Dynamical and Control Systems, vol. 11, no. 06-Special Issue, 2019.

[10] D. G. Dev and V. Bhatnagar, "Hybrid RFSVM: Hybridization of SVM and Random Forest Models for Detection of Fake News," Algorithms, vol. 17, no. 10, p. 459, 2024.

[11] R. K. Kaliyar, A. Goswami, and P. Narang, "FNDNet: A Deep Convolutional Neural Network for Fake News Detection," Cognitive Systems Research, vol. 61, pp. 32–44, 2020.

[12] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection," in Proc. 2017 ACM Conf. on Information and Knowledge Management (CIKM), 2017, pp. 797–806.

[13] B. Min et al., "Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey," ACM Computing Surveys, vol. 56, no. 2, pp. 1–40, 2023.

[14] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-gram Analysis and Machine Learning Techniques," in Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ISDDC 2017), Springer, pp. 127–138, 2017.

[15] A. A. Ali et al., "Linguistic Features and Bi-LSTM for Identification of Fake News," Electronics, vol. 12, no. 13, p. 2942, 2023.

[16] M. A. Ouassil et al., "A Fake News Detection System Based on Combination of Word Embedded Techniques and Hybrid Deep Learning Model," International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, 2022.

[17] D. Choudhury and T. Acharjee, "A Novel Approach to Fake News Detection in Social Networks Using Genetic Algorithm Applying Machine Learning Classifiers," Multimedia Tools and Applications, vol. 82, no. 6, pp. 9029–9045, 2023.

[18] A. Dogan and D. Birant, "A Weighted Majority Voting Ensemble Approach for Classification," in 2019 4th Int. Conf. on Computer Science and Engineering (UBMK), IEEE, 2019, pp. 1–6.

[19] A. Sharma, P. Soni, and H. C. Saini, "H-MFN: A Hybrid Multimodal Deep Learning Model for Fake News Detection in Indian News and Social Media," *Journal of Dynamics and Control*, vol. 10, no. 2, pp. 24–49, Feb. 2026