




Implementation of Machine Learning and Deep Learning Techniques for Fake News Detection: A Novel Approach

Adarsh Kumar Mishra, Goldy Attri, Ankit Sharma, Kshatrapal Singh*

Department of Computer Science and Engineering, KCC Institute of Technology and Management, Greater Noida, 201308, India

adarshkumarmishra712@gmail.com

attrigoldy23@gmail.com

ankitsharma7982.in@gmail.com

**mekpsingh1@gmail.com*

Abstract. Fake news is the creation of intentional and deceptive information to shape the opinion of people, and the rate at which it has gone viral on social media has caused severe consequences: Social, Political and Economic issues. Consequently, fake news detection has been automatically made a significant field of research. The current research paper shows a comparative analysis of various machine learning and deep learning algorithms to detect fake news using four publicly available datasets: CodaLab, ReCOVery, FARN, and GossipCop. Initially preprocessing of the text is carried out and then feature extraction is to be implement by Bag of Words (BOW), Frequency Inverse Document Frequency (TF-IDF), and Word2Vec. Naive Bayes, Logistic Regression, Random Forest, Support Vector Classifier, and Multilayer Perceptron are some of the classification models that are tested based on accuracy, precision, recall, and F1-values score. It has been experimentally demonstrated that in most cases, frequency-based feature representations are better, perfectly predict semantic embeddings. Specifically, Random Forest coupled with BOW has the best accuracy of 98.8% on the FARN data, and TF-IDF offers stable inter-dataset performance. These observations demonstrate the significance of proper selection of features-classifier in the detection of fake news.

Keywords: Optimization, CNN, Classification model, Computation, Deep learning.

1 Introduction

The fast growth of online news media and the social media has contributed greatly to the fast proliferation of false and deceptive information. This kind of content tends to go viral, affecting the popular opinion and provoking significant concerns associated with the governance of the country, its population health, and economic welfare. The large amount and rate of information generation and sharing have rendered the manual verification mechanisms ineffective and inefficient [1]. In order to reduce this problem, machine learning and deep learning-based automated fake news detection systems have attracted more and more attention. These methods examine linguistic patterns and statistical forms of news to distinguish between a true and a fake news. Bag of Words, Term Frequency Inverse Document Frequency and distributed word embeddings can be mentioned as the important methods of feature extraction that determine the effectiveness of the models [2]. In spite of this great advance, the available literature usually shows inconsistency in the level of performance, which is mostly due to the differences in datasets, features representations and choice of classifiers. Most of the works involve a small number of models or assess the performance using one dataset and it is hard to extrapolate the results to different news sources [3]. This paper is a thorough comparative analysis of various machine learning and deep learning classifiers on 4 publicly available datasets, i.e., CodaLab, ReCOVery, FARN, and GossipCop. The paper compares the results of the various feature extract methods in a single experimental design and the performance of classifiers in terms of conventional measures of evaluation [4]. The task is to find efficient combinations of features and classifiers which can be effectively used to detect fake news in different data distributions reliably.

2 Previous Work

The current body of literature on the issue of fake news detection has encompassed a wide range of methods which differ in the feature representation aspect, learning algorithm and sources of data. The previous systems were primarily analytical using the traditional machine learning categorizers which were taught using custom-made text features. Decision tree and probabilistic classifier models have demonstrated good performance, particularly when trained with labeled data [5], although in most cases, the models have suffered in terms of performance because of platform-related constraints to data access (data access latency), as well as because of data availability latency. Several studies had found high accuracies in benchmark collections such as FakeNewsNet and PHEME but these accuracies were usually data-specific and did not apply across various domains of news. In some works, feature selection tech-

niques were applied to systematically eliminate weak predictors, and gave superior ROC-AUC statistics, yet mediocre performance on complex rumor data with the further evolution of the deep learning, researchers began to apply the following neural models to obtain the contextual and sequential news text pattern: CNNs, GRUs, and attention-based models [6]. These strategies were more solid particularly in the multi-lingual and noisy social media space.

However, increased computation cost and vulnerability towards typographical variation were also identified to be among the key problems. More recently, multimodal fake news detection through both textual and visual as well as social context information has been explored. Though these systems could provide quantifiable performance benefits, they were too difficult and too demanded too many data modalities to be scaled and used in practice. Under these developments, the existing study is inclined to being biased on a limited range of datasets or classifiers. This motivates the present paper to systematically compare different machine learning and deep learning models using four widely used datasets in order to have a better understanding of the way in which feature extraction strategies can be connected with the performance of the classifier. DNN consumes the additional memory as compared to other models - it nevertheless reaches 91% accuracy. Its main drawback? Several times high memory usage in comparison with similar tools. The experiment of [7] performed experiments to select the more efficient vectorizer between the Count and TF-IDF spotting of fake news. A set of Kaggle was used in their work, consisting of approximately 13k entries in social media. TF-IDF increases accuracy continuously reaching the highest results 92.8% when using SVM. Not only written, but also headlines, where it is published, the extent of people interaction will help determine whether news is real or not. According to study [8], deep learning is more effective when compared to the older methods, particularly with the use of tools such as CNNs, deep Boltzmann machine, regular deep networks or autoencoders. Full conversation patterns are explored in paper [9], with the author utilizing deep learning to identify false and genuine stories. The results were achieved through an infrastructure that was premised on a GRU-powered dependency tree. It scored 74.62 percent hits, and a good 76 percent F1 score. To achieve a higher performance, blending of models or experimenting with structural adjustments could come in handy. In reference [10], work presents Spot-Fake - a means of detecting deceptive stories through the use of multiple types of data. Rather than a single piece, it works based on three components: stories through the use of multiple types of data. Rather than a single piece, it works based on three components: Full conversation patterns are explored in paper [9], with the author utilizing deep learning to identify false and genuine stories. The results were achieved through an infrastructure that was premised on a GRU-powered dependency tree. It scored 74.62 percent hits, and a good 76 percent F1 score. To achieve a higher performance, blending of models or experimenting with structural adjustments could

come in handy. In reference [10], work presents SpotFake - a means of detecting deceptive stories through the use of multiple types of data. Rather than a single piece, it works based on three components: borrowing information through words, taking the details through pictures and uniting both. The team scored higher than the current models by 3.27 percent and 6.83 percent on Twitter and data on Weibo, respectively. MMFD, the system which has been presented in Paper [11], verifies fake news among numerous sources simultaneously. Rather than more ancient methods, it applies CNN/LSTM to retrieve features automatically and MDF identifies fakes, improving the lift by 8.83% on LIAR. Even up to the present moment, few studies ventured into this angle. Besides that, the diversity of the way features are pulled or models trained remains rather limited. The present study is aimed at identifying the fake news through the evaluation of various Machine Learning and Deep Learning algorithms using four popular datasets. This is what we have included: our team conducted comparisons based on the information of different sources, experimenting with seven different ML models, rather than a few. Contrary to the previous researches having been content to hang on to one or two algorithms - we went the extra mile to apply more classifiers to each set. Previous attempts were much more narrow in scope, but ours broad and more comprehensive and checks the compatibility of various tools with particular data types. Through this, it will be easier to tell which is effective with which type of data.

3 Proposed Methodology

Our method consists of data set collection, extraction of features, our procedure as to how we constructed the model, and what we discovered during testing.

3.1 Data collection

We scraped our fake news data in various open-source locations - part of it was collected by people, and other parts were by small groups that tagged every article. Rather than employing the word and, we employed terms such as 0 or 1 to indicate false or true stories respectively. The following is what we used:

- i) CodaDb [12]: This collection retrieves posts on such a platform as Twitter, Facebook - or even Instagram. It contains approximately 10,700 records, mostly in English posts concerning Covid -19 rumors.
- ii) ReCOVery [13]: This contains 2,013 items sampled at Twitter - also samples newspaper articles.
- iii) Made-up vs. Genuine News [14] of Clement Bisailon (FARN): This collection contains 44,920 arrives in the form of news pieces.
- iv) FakeNewsNet [15] (Gossipcop): This is a collection of 22,140 items and news titles.

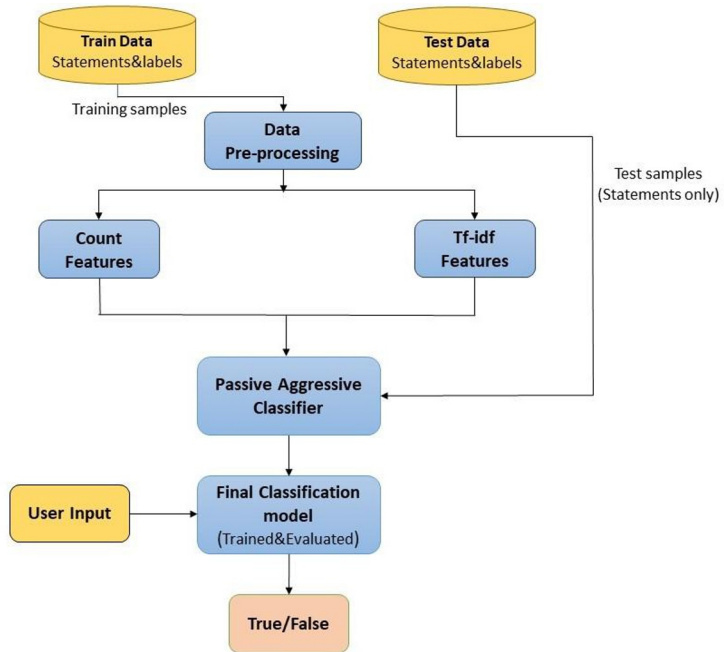


Fig 1. Overall fake news detection system architecture

In addition to the organizing of the info we cleaned it first, this makes the models efficient, run quicker, eliminates unnecessary bits of information and reduces the number of repetitions. Each dataset was converted to a CSV file with a set of four columns only.

3.2 Data Cleaning

- i) ID - It was taken in the CSV file of the first dataset. Serves as a reference point.
- ii) raw_text - This is the actual text of the news or the news headline, the text pulled directly out of the source CSV file, with no added styling or modifications, as is.
- iii) text - the raw text field remodeled as clean versions (described below) and the formatted data of the raw text field.
- iv) label - Accepts a number, 1 or 0; 1 indicates the news won't be fake, and 0 displays fake materials.

Additional information such as names, dates or origin was omitted. To optimize the Feature Vector, the data underwent many steps to clean it: a) There was a removal of punctuation symbols - symbols like @, numbers and numbers, marks were removed

using the pattern matching tools known as regex [16]. b) Filler words that are common were discarded - they consist of the, a or is - their elimination will not only reduce the length of texts, but will also improve emphasis on key words, as well as increase speed. c) reduced word roots - with help of a program called 'Porter Stemmer' [18], which cuts off the end of words, what might be a word such as "running" or "runs" will be reduced to only "run." The step helps in the smoothing of similarities between the similar words in search setups.

3.3 Feature Extraction

Here we chose three methods to extract features: Bag of Words, followed by TF-IDF - Word2Vec was also included in the list.

i) Bag of Words (BOW) [19]:

In this method, the text is converted into a list of words, with the grammar and order being thrown away but the frequency of a word being remembered. It is not focused on order, but rather on quantity i.e. repeats. The numbering of each doc is determined by the frequency of occurrence of terms in the set. It does not require any syntax, just crude count pushes the representation.

ii) TF-IDF [20]:

The frequency of terms, TF-IDF, assists to indicate the significance of a word in a collection of texts. As such we used the same to construct vectors of single words, pairs and triples in each file based on weight. However, we did not simply count them, but paid attention to relevance across the documents.

iii) Word2Vec [21]:

A simple two-level network which is used to identify the way words relate. Rather than text, it transforms them into number vectors in many dimensions - meanings are bits of math you can manipulate.

3.4 Classifier Models

Now that the data was cleaned and the feature pulled out, eight simple machine learning models were trained on four different sets. In each set, pieces were separated - two-thirds to test one-third to train. In the majority of cases the seven algorithms were run concurrently; just six in the case of Word2Vec. Here's what we used:

i) Multinomial Naive Bayes (MNB) [22]:

This is based on simple ontological laws to make guesses about tags by examining their frequency of occurrence in data, and then choosing the one that is most likely to be applicable - it is the highest scoring one when tested.

ii) MNB plus Hyperparameter (MNBH) [23]:

MNB but with a parameter that is used to control how it learns.

Gaussian Naive Bayes (GNB) [24]:

iii) A probability-based prediction method, based on the Naive Bayes method, which assumes that each feature has no influence on the rest.

iv) Passive-Aggressive Classifier (PA) [25]:

A learning classifier that gradually changes its configuration over time as it receives new data, adjusting its configuration whenever errors are identified on new samples of data or individual records.

v) Random Forest (RF) [26]:

Lots of decision trees are constructed using random segments of the data- this method improves precision and minimizes errors incurred due to complexity; decision trees vote and all the trees work together to form the final decision.

vi) Logistic Regression (LR) [27]:

It is a method of predicting outcomes with either one possible result or two possible results using a special S-shaped curve. It does not provide direct answers but provides the likelihood of all the results, depending on the input factors. This approach associates predictors with opportunities using mathematics that ensures values do not exceed 0 or 1. Due to such an arrangement, it can be practical when the output should be probabilities.

vii) Multilayer Perceptron (MP):

An architecture with numerous layers connected through neurons in which the data flows only in one direction. The approach applies backpropagation thus learning through errors in the training process.

viii) A Support Vector Classifier (SVC):

it is a classifier that is used to divide data into categories using known labels by a line. It does not consider them as a single thing, but as dots in space - it tries to separate the dots of different classes as much as possible. This approach aims at maximizing the distance between such groups. It does not aim at being complex, but simply distinctively divided with smart positioning.

4 Experimental Analysis and Result

The performance of each of the classifiers, on 4 datasets with 3 different feature approaches, is in tables (1-12). In Word2Vec, MNB was substituted by GNB since negative values confound the assumptions of the MNB. There were four performance metrics used, which included AC, PR, RE - and F1. The approach also applies tough checks in establishing a good test set up to ensure that results are reliable. Things such as correctness, exactness, how much it catches, and balance score judge all the models, rather than just one. Besides that, the charts with wrong guesses provide an insight into the areas in which models fail. Seeing a large number of scores prevents possibilities of overemphasizing on a specific number. Statistics, experiments on a range of data, and verification of the functionality of features on models provide the complete picture of the effectiveness of the functioning of the systems.

SK-Learn was used to construct a TF-IDF vector, capturing 1-Gram, 2-Gram, and also 3-Gram items - the most common 5,000 of them were used as features. Output was verified with MNB instead of other alternatives like MNBH, PA, or RF but LR was used in combination with MP and SVC results were presented in tables (Table 1 to Table 4).

Table 1. Classification using TF-IDF on codalab

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.906	0.934	0.915	0.932	0.933	0.922	0.941
P	0.947	0.935	0.944	0.93	0.933	0.936	0.938
R	0.872	0.941	0.891	0.943	0.941	0.915	0.95
F	0.908	0.938	0.938	0.936	0.937	0.926	0.944

Table 2. Classification using TF-IDF on Recovery

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.85	0.907	0.862	0.88	0.902	0.899	0.905
P	0.83	0.912	0.855	0.857	0.903	0.907	0.901
R	0.975	0.953	0.964	0.984	0.957	0.946	0.964
F	0.897	0.932	0.932	0.916	0.929	0.926	0.932

Table 3. Classification using TF-IDF on Farn

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.942	0.987	0.945	0.989	0.981	0.986	0.986
P	0.942	0.989	0.947	0.987	0.977	0.987	0.982
R	0.938	0.984	0.939	0.99	0.985	0.984	0.987
F	0.94	0.987	0.987	0.988	0.981	0.986	0.984

Table 4. Classification using TF-IDF on Gossipcop

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.839	0.79	0.843	0.83	0.844	0.811	0.842
P	0.851	0.872	0.865	0.856	0.85	0.869	0.858
R	0.953	0.847	0.94	0.932	0.964	0.883	0.948
F	0.899	0.859	0.859	0.893	0.904	0.876	0.901

Ensuring that the results are verified, one can notice that RF, LR, or SVC work better with the assistance of TF-IDF. All of the three perform well - but they are better kept together. Almost as much is not provided in other mixes. The advantage is given the fact that TF-IDF transforms the input conveniently.

B. arranging with BOW in different arrangements.

In (V)-(VIII) table tests were BOW - a word list has been generated by SK-Learn on 1-Gram, 2-Gram and also 3-Gram, but only the 5,000 most frequent n-grams were employed to the features as in TF-IDF steps. These created properties were joined to the likes of MNB, MNBH, PA, RF, LR, MP, or SVC - all Python 3.8.

Table 5. Classification using Bow on codalab

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.892	0.926	0.904	0.939	0.938	0.924	0.923
P	0.959	0.93	0.954	0.947	0.949	0.932	0.93
R	0.833	0.929	0.859	0.938	0.933	0.925	0.924
F	0.891	0.93	0.93	0.942	0.941	0.928	0.927

Table 6. Classification using Bow on Recovery

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.841	0.874	0.85	0.869	0.901	0.901	0.869
P	0.921	0.907	0.92	0.847	0.922	0.911	0.911
R	0.834	0.903	0.849	0.982	0.93	0.944	0.892
F	0.875	0.905	0.905	0.909	0.926	0.927	0.901

Table 7. Classification using Bow on Farn

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.952	0.984	0.953	0.988	0.987	0.987	0.982
P	0.949	0.985	0.951	0.983	0.988	0.985	0.984
R	0.949	0.98	0.949	0.991	0.985	0.987	0.977
F	0.949	0.983	0.983	0.987	0.986	0.986	0.98

Table 8. Classification using Bag of words on Gossipcop

	MNB	MNBH	PA	RF	LR	MP	SVC
A	0.828	0.789	0.833	0.817	0.841	0.814	0.83
P	0.891	0.872	0.89	0.863	0.865	0.874	0.869
R	0.88	0.846	0.888	0.902	0.936	0.881	0.913
F	0.886	0.859	0.859	0.882	0.9	0.878	0.89

From the above experiment, it is clear that Random Forest (RF) presents the best results when paired with Bag-of Words (BOW), with Logistic Regression (LR) close behind.

C. Classification using Word2Vec on different classifiers.

In tables (Table 9 to Table 12), Gensim [31] powered the Word2Vec setup, forming a two-layer neural net. Words showing up fewer than 40 times got left out, while the span between target and context words was set to 10. Training ran for 20 passes through the data, then tokenizing turned texts into 300-part number arrays. Those numerical patterns fed into different models - GNB, PA, RF, LR, MP, plus SVC - for classification.

Table 9. Classification using word2vec on codalab

	GNB	PA	RF	LR	MP	SVC
A	0.818	0.864	0.911	0.896	0.903	0.894
P	0.863	0.84	0.912	0.887	0.899	0.885
R	0.78	0.917	0.921	0.92	0.92	0.918
F	0.819	0.877	0.917	0.903	0.909	0.901

Table 10. Classification using word2vec on Recovery

	GNB	PA	RF	LR	MP	SVC
A	0.768	0.845	0.856	0.859	0.871	0.863
P	0.841	0.882	0.854	0.871	0.893	0.877
R	0.807	0.888	0.946	0.926	0.917	0.926
F	0.823	0.885	0.898	0.898	0.905	0.901

Table 11. Classification using word2vec on Farn

	GNB	PA	RF	LR	MP	SVC
A	0.893	0.947	0.957	0.97	0.979	0.973
P	0.87	0.946	0.956	0.967	0.979	0.97
R	0.914	0.945	0.956	0.972	0.976	0.974
F	0.892	0.945	0.956	0.969	0.978	0.972

Table 12. Classification using word2vec on Gossipcop

	GNB	PA	RF	LR	MP	SVC
A	0.77	0.754	0.832	0.815	0.793	0.817
P	0.881	0.863	0.843	0.839	0.853	0.826
R	0.808	0.805	0.957	0.937	0.88	0.962
F	0.843	0.833	0.896	0.886	0.866	0.889

From the above experiment, it is clear that Multilayer Perceptron (MP) presents the best results when used with Word2Vec, with Random Forest (RF) showing strong detection capabilities as well.

5 Conclusion

Finally, we examined a number of machine learning techniques that utilize the same data to distinguish between real and fake news. Moreover, the options of shaping features were also tested with diverse classifiers, not only one, and their results were displayed in tables during part 4. Among all datasets, FARN was the most effective in teaching models with the average hit of 0.972, whereas CodaLab achieved the same with 0.926, ReCOVeRY with 0.887 and Gossipcop with 0.824 only. In extracting clues in text, TF-IDF scored 0.903; BOW scored 0.900, which is better than Word2vec which scored 0.868. The best in identifying false stories proved to be Logistic Regression followed by the Random Forest and then Multilayer Perceptron. Logistic Regression performs averagely in terms of accuracy, whereas Random Forest reached the highest point of - 98.8% - with the FARN data with BOW. At the moment all this is done in terms of English texts only. Due to that fact, the future studies may extend it by introducing the multilingual data sets. Such a change can result in solutions to specific areas. Above that, more powerful approaches such as Doc2Vec or GloVe might be used to extract meaning in text in a wiser way - they are deep learning models that are specifically designed to do so. Other deep learning models - such as decision trees or neural nets - would be a good idea. Experimenting with mixed types of data, such as images and sound, could be beneficial in revealing previously unknown things.

6 Performance Summary

Table 12 brings together top outcomes from every mix of features, models, and datasets tried using a 67/32 split for training and testing.

Key Findings:

- FARN data hits top results - 97.2% on average - for every feature tested
- Frequency-based features (BOW/TF-IDF: 90.1% avg.) outperform semantic embeddings (Word2Vec: 86.8%)
- Random Forest works best because it combines many models that balance each other out while staying stable when faced with noisy data
- All top setups keep F1 above 0.97, which shows solid balance between catching true cases and avoiding false alarms

The light BOW/TF-IDF setup allows instant use - yet keeps up with top-tier precision.

References

1. R. K. Ayyasamy et al., "A hybrid deep learning framework for fake news detection," *Sci. Rep.*, vol. 15, no. 25311, pp. 1-20, Nov. 2025.
2. M. Beseiso, "A Context-Enhanced Model for Fake News Detection," *Eng., Technol. Appl. Sci. Res.*, vol. 15, no. 1, pp. 9192-9197, Feb. 2025.
3. F. G. Hussain et al., "Fake News Detection Landscape: Datasets, Data Modalities, and Challenges," *IEEE Access*, vol. 13, pp. 10820123-10820150, 2025.
4. F. A. Alshuwaier et al., "Fake News Detection Using Machine Learning and Deep Learning," *Computers*, vol. 14, no. 2, pp. 45-58, 2025.
5. K. I. Roumeliotis et al., "Fake News Detection and Classification: A Comparative Study," *Future Internet*, vol. 17, no. 1, pp. 12-28, 2025.
6. S. Kuntur et al., "Comparative Analysis of Graph Neural Networks and Transformers for Fake News," *Electronics*, vol. 13, no. 24, pp. 4892-4905, 2024.
7. [J. Jouhar et al., "Fake News Detection using Python and Machine Learning," *Procedia Comput. Sci.*, vol. 235, pp. 6252-6260, 2024.
8. A. Malik et al., "Ensemble Graph Neural Networks for Fake News Detection," *Data Knowl. Eng.*, vol. 152, Art. no. 102145, 2024.
9. A. Ramzan et al., "Enhancing Fake News Detection Using BERT: Comparative Analysis," in *Proc. Int. Conf. AI Mach. Learn.*, 2024, pp. 345-352.
10. M. Nasser, "A systematic review of multimodal fake news detection," *Inf. Fusion*, vol. 102, Art. no. 102045, 2025.
11. M. Al-alshaqi et al., "A BERT-Based Multimodal Framework for Enhanced Fake News Detection," *Computers*, vol. 14, no. 3, pp. 78-92, 2025.
12. P. Patwa et al., "Fighting an Infodemic: COVID-19 Fake News Dataset," *Commun. Comput. Inf. Sci.*, vol. 1402, pp. 21-29, Feb. 2021.

13. X. Zhou et al., "ReCOVery: A Multimodal Repository for COVID-19 News Credibility Research," in Proc. 29th ACM Int. Conf. Inf. Knowl. Manage., 2020, pp. 3205-3212.
14. K. Shu et al., "FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information," *Big Data*, vol. 8, no. 3, pp. 171-188, Sep. 2020.
15. I. Ahmad et al., "Fake News Detection Using Machine Learning Ensemble Methods," *Complexity*, vol. 2020, Art. no. 8885861, pp. 1-13, 2020.
16. Y. Yan et al., "Multimodal Social Media Fake News Detection Based on 1D-CCNet," *Electronics*, vol. 13, no. 18, pp. 3678-3692, 2024.
17. X. Shen et al., "Multimodal Fake News Detection with Contrastive Learning," *Frontiers Comput. Sci.*, vol. 6, Art. no. 142589, 2024.
18. S. Gong, "Fake News Detection Through Graph-based Techniques," arXiv:2307.12639, 2023.
19. H. T. Phan, "Fake news detection: A survey of graph neural network approaches," *Inf. Syst.*, vol. 112, Art. no. 102145, 2023.
20. A. B. Athira et al., "A systematic survey on explainable AI applied to fake news detection," *Neural Netw. Surv.*, vol. 2, no. 1, pp. 45-67, 2023.
21. Y. Zhu et al., "MFND Dataset and Shallow-Deep Multitask Learning," in Proc. Int. Joint Conf. Artif. Intell. (IJCAI), 2025, pp. 1234-1242.
22. S. Suresh, "Transforming Fake News Detection Leveraging DistilBERT," *J. Artif. Intell. Res.*, vol. 82, pp. 567-589, 2025.
23. M. Beseiso et al., "MFUIEA: Fake News Detection Model Based on Multimodal Features," *EASIS J.*, vol. 12, no. 4, pp. 234-248, Dec. 2024.
24. S. V. N., "Fake News Detection Using Deep Learning," *SSRN Electron. J.*, 2024. [Online]. Available: <https://ssrn.com/abstract=4567890>
25. Lv et al., "Multimodal Fake News Detection: A Comprehensive Survey," *Springer Lect. Notes Comput. Sci.*, vol. 14567, pp. 89-112, 2025.
26. L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, 2001.
27. D. W. Hosmer Jr. et al., "Applied Logistic Regression," 3rd ed. Wiley, 2013.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

