Sakshi Santosh Kasar, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Fake News Detection and Classification Using Machine Learning

Sakshi Santosh Kasar
Under The Guidance of Dr. Anita Mahajan
M.E. Student, Department of Computer Engineering Ajeenkya D. Y. Patil School of Engineering, Pune, India

Abstract- The rapid growth of online news, social networking platforms, and instant messaging has made information easier to access, but it has also increased the circulation of misleading claims. Manual verification alone cannot keep pace with the volume and speed of online content. This paper presents a machine-learning-based framework for detecting and classifying potentially fake news from textual content. The proposed workflow uses a labelled dataset, text cleaning, feature extraction, supervised classification, and evaluation through standard performance measures. CountVectorizer and Term Frequency–Inverse Document Frequency (TF-IDF) are considered for converting text into numerical features. Candidate classifiers include Support Vector Machine (SVM), Logistic Regression, Decision Tree, Random Forest, Naive Bayes, K-Nearest Neighbours (KNN), and XGBoost; an LSTM model may be explored where sufficient data and computing resources are available. The LIAR/LIAR-PLUS datasets are relevant options for experiments involving fact-checked political statements. Model quality should be assessed using accuracy, precision, recall, F1-score, and a confusion matrix rather than accuracy alone. As no verified experimental scores were supplied in the source documents, this paper describes a proposed framework and evaluation plan rather than claiming a measured winning model. The system is intended to assist users and fact-checkers, not replace human judgement or source verification.

Keywords—Fake news, misinformation, machine learning, natural language processing, TF-IDF, CountVectorizer, text classification.

I. INTRODUCTION

Online platforms allow news to reach large audiences within seconds. The same speed can also help unverified stories, misleading headlines, manipulated claims, and fabricated reports spread before they are checked. Such content may influence public opinion, create confusion, damage reputations, or affect decisions about health, society, and public affairs. The term "fake news" is used broadly, so a detection system must be designed carefully: a statement can be inaccurate, partly true, satirical, outdated, or presented without context, and these cases are not always equivalent.

Human fact-checking remains important, but reviewing every post manually is difficult at internet scale. Natural language processing (NLP) and machine learning can help by identifying textual patterns learned from previously labelled examples. A model can analyse a headline, statement, or article and estimate which class it resembles. Its prediction is a screening signal, not proof that a claim is true or false.

This paper proposes a reproducible text-classification pipeline. It focuses on preprocessing, feature representation, comparison of conventional classifiers, and careful evaluation. It also highlights limitations such as dataset bias, changing language, context dependence, and false predictions.

II. PROBLEM STATEMENT

The problem addressed is to develop a machine-learning-based system that analyses news text and assigns a class indicating whether the content is likely to be genuine or misleading, based on labelled training examples. The system should support consistent preprocessing, feature extraction, model training, and testing on unseen samples. It should be evaluated with measures that reveal both correct predictions and errors, especially cases where genuine news is incorrectly flagged or misleading content is missed.

© 2026 Sakshi Santosh Kasar, This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

A classifier cannot independently establish factual truth unless its prediction is supported by reliable evidence and context. Therefore, the proposed system is framed as an assistance tool for preliminary screening and prioritisation.

Objectives and Scope

- Collect and prepare a suitable labelled dataset of news statements or articles.
- Clean text consistently and convert it into machine-readable features.
- Train and compare supervised classifiers using the same data split and evaluation procedure.
- Measure performance using accuracy, precision, recall, F1-score, and confusion matrices.
- Inspect false positives and false negatives and consider the ethical impact of incorrect labels.
- Provide a foundation for testing unseen text through a simple user interface in future work.

The source seminar material identifies the LIAR and LIAR-PLUS datasets, particularly for political claims and fact-checked statements. The approach can be adapted to other domains, but a model trained on one dataset may not generalise to breaking news, other languages, or different platforms without further validation.

III. RELATED WORK

Prior work on automatic deception detection has examined linguistic patterns, source information, social signals, and machine-learning classifiers. Conroy, Rubin, and Chen survey approaches to automatic deception detection and describe the challenge of identifying misleading information from content alone [1]. Tandoc, Lim, and Ling discuss the varied meanings of "fake news," underscoring the importance of defining the target label and context before building a classifier [2].

Research has also explored models that combine content with online activity. Vedova et al. study automatic online fake-news detection using content and social signals [3]. Seddari et al. present a hybrid linguistic and knowledge-based approach for social-media content [4]. Galli et al. provide a benchmark-oriented perspective on fake-news detection [5]. These lines of work suggest that text features can be useful, while additional evidence and carefully designed evaluation may be necessary for robust real-world use. The seminar report motivating this paper describes conventional supervised methods, including Logistic Regression, Support Vector Machine, decision trees, Random Forest, boosting methods, and neural approaches. It also outlines CountVectorizer, TF-IDF, and word embeddings as alternative text representations. This paper organises these elements into a common experimental workflow rather than assuming one algorithm will perform best in every dataset.

IV. PROPOSED METHODOLOGY

1. Dataset Collection

The first step is to obtain a labelled dataset whose source, labelling procedure, and class definitions are documented. LIAR or LIAR-PLUS may be used for experiments with fact-checked political statements. The dataset should be inspected for missing values, duplicate entries, class imbalance, and possible overlap between training and testing records. The label scheme must be preserved or explicitly mapped to the intended classification task. If labels contain multiple truthfulness levels, collapsing them into "real" and "fake" should be justified and documented.

2. Text Preprocessing

Text preprocessing makes the input more consistent. The proposed steps are: (1) remove empty or unusable records; (2) normalise text, including lowercasing where appropriate; (3) handle punctuation, extra whitespace, and selected numeric or special tokens; (4) tokenise text; and (5) remove stop words only when doing so is beneficial. Preprocessing choices should be tested rather than applied blindly, because punctuation, numbers, names, and short words may carry useful signals. The same preprocessing pipeline must be used during training and prediction.

3. Feature Extraction

Machine-learning classifiers require numerical input. CountVectorizer represents a document using token counts, creating a vocabulary from the training data. TF-IDF weights terms according to their frequency in a document and their

distribution across the corpus, reducing the influence of terms that occur everywhere. Unigrams and bigrams can capture individual words and short phrases. Dense word embeddings,
such as those produced with an NLP library, may be evaluated as an alternative, although they can require more memory and computation.
To avoid data leakage, the vocabulary and TF-IDF weights must be fitted only on the training partition. The fitted transformation is then applied to validation and test data. Optional metadata or linguistic features should be added only if they are available consistently and do not reveal the target label indirectly.

4. Model Training
The proposed comparison includes the following supervised models:

- Support Vector Machine (SVM): identifies a separating boundary between feature vectors and can work well with sparse text features.
- Logistic Regression: estimates class probabilities and provides a strong, interpretable baseline for binary classification.
- Decision Tree: learns rule-like splits, but can overfit when the tree becomes too complex.
- Random Forest: combines multiple decision trees to reduce instability compared with a single tree.
- Naive Bayes: uses class-conditional feature probabilities and is often computationally efficient for text classification.
- K-Nearest Neighbours (KNN): classifies a sample based on nearby training examples; performance can be affected by high-dimensional sparse text.
- XGBoost: uses boosted trees and may capture non-linear patterns, with tuning needed to control complexity.
- LSTM: a sequence model that may be explored if the dataset size and computational resources support it.

Models should be compared using the same training/test split and consistent preprocessing. Cross-validation and hyperparameter tuning should be performed only on training data. A final test set should remain untouched until model selection is complete.

5. Evaluation
Accuracy is the proportion of all predictions that are correct. Precision measures how many items predicted as fake are actually labelled fake; recall measures how many labelled fake items are detected. The F1-score combines precision and recall through their harmonic mean. A confusion matrix shows true positives, true negatives, false positives, and false negatives. For an imbalanced dataset, accuracy alone can be misleading, so class-wise precision, recall, F1-score, and macro-averaged results should also be reported.

The experimental report should state the dataset version, class mapping, train/test proportions, random seed, feature settings, model parameters, and software versions. No numerical scores are presented here because the supplied seminar documents do not provide a complete, verifiable results table for the proposed experiment.

V. SYSTEM ARCHITECTURE

The proposed system follows a sequential pipeline:

- Input: receive a news statement, headline, or article text.
- Preprocessing: clean and normalise the input using the saved training pipeline.
- Vectorisation: transform the text using the fitted CountVectorizer or TF-IDF model.
- Classification: pass the feature vector to the selected trained classifier.
- Output: display the predicted class and, where supported, a calibrated confidence score.
- Review: encourage users to consult trustworthy sources and human fact-checking for consequential claims.

In an implementation, Python and scikit-learn can support preprocessing, vectorisation, model training, and evaluation. A lightweight web interface may be added to accept text and display the result. The trained vectoriser and classifier should be saved together so inference uses exactly the same feature mapping as training.

VI. RESULTS AND DISCUSSION

The supplied PPT and seminar report describe the planned dataset, candidate algorithms, feature extraction techniques, and evaluation through confusion matrices. However, they do not provide a complete set of numerical results that can be independently tied to a documented data split and reproducible experiment. For that reason, this paper does not claim a particular accuracy or declare a best-performing model.

Once experiments are completed, results should be reported in a comparison table containing each model's accuracy, precision, recall, F1-score, and false-positive/false-negative counts. The best model should be selected according to the project's use case rather than accuracy alone. For example, a fact-checking queue may prioritise recall to surface more potentially misleading items, while also monitoring false positives to avoid unfairly flagging genuine reporting. Performance should be checked across topics, publishers, time periods, and writing styles to estimate how well the model generalises.

A key risk is learning shortcuts in the dataset instead of signals of credibility. If certain publishers, topics, or repeated phrases are disproportionately associated with one label, the model may rely on those patterns. Dataset documentation, duplicate checks, error analysis, and out-of-domain testing are therefore essential.

Advantages and Limitations

Advantages

- Can screen large amounts of text more quickly than manual review alone.
- Offers a repeatable process for comparing classification methods.
- Can help journalists, educators, and platform moderators prioritise content for verification.
- Provides an educational application of NLP, feature engineering, and supervised learning.

Limitations

- Predictions may be wrong, producing both false alarms and missed misinformation.
- Language, events, and deception strategies change over time, so models require monitoring and updates.
- Training data may contain political, demographic, or source-related biases.
- Text-only models may miss context, satire, images, source credibility, and external evidence.
- Automated labels can affect trust and freedom of expression if treated as definitive judgements.

Applications and Future Scope

Potential applications include assisting journalists during fact-checking, prioritising posts for social-platform moderation, supporting media-literacy education, and helping news aggregators review content. In cybersecurity, related text-analysis techniques may help flag deceptive messages, although fake-news detection and phishing detection are distinct tasks and require appropriate datasets.

Future work can evaluate multilingual models, transformer-based language representations, source and context features, and evidence-retrieval methods. A useful extension would return supporting sources and explain which parts of an article require verification rather than presenting only a binary label. Further work should include temporal testing, calibration, fairness analysis, and a human-review workflow. Any live deployment should make clear that a model prediction is probabilistic and can be challenged or corrected.

VII. CONCLUSION

This paper presents a practical framework for fake-news detection and classification using NLP and supervised machine learning. It combines labelled data, careful text preprocessing, CountVectorizer or TF-IDF features, several candidate classifiers, and evaluation using precision, recall, F1-score, accuracy, and confusion matrices. The approach can support early screening of potentially misleading content, but it cannot independently verify every claim. Reliable use depends on dataset quality, transparent evaluation, ongoing monitoring, and human fact-checking. The next step is to implement the pipeline, run controlled experiments, publish reproducible results, and assess

errors before making claims about real-world effectiveness.

REFERENCES

1. N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1–4, 2015.
2. E. C. Tandoc, Z. W. Lim, and R. Ling, "Defining 'fake news': A typology of scholarly definitions," Digital Journalism, vol. 6, no. 2, pp. 137–153, 2018.
3. M. L. D. Vedova et al., "Automatic online fake news detection combining content and social signals," in Proceedings of the 22nd Conference of Open Innovations Association (FRUCT), pp. 272–279, 2018.
4. N. Seddari et al., "A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media," IEEE Access, vol. 10, pp. 62097–62109, 2022.
5. A. Galli, E. Masciari, V. Moscato, and G. Sperlí, "A comprehensive benchmark for fake news detection," Journal of Intelligent Information Systems, vol. 59, no. 1, pp. 237–261, 2022.
6. F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, "A survey of information cascade analysis: Models, predictions, and recent advances," arXiv:2005.11041, 2020.
7. S. Gaillard, Z. A. Oláh, S. Venmans, and M. Burke, "Countering the cognitive, linguistic, and psychological underpinnings behind susceptibility to fake news," Frontiers in Communication, vol. 6, 2021, Art. no. 661801.
8. 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 Proceedings of EMNLP, pp. 2931–2937, 2017.
9. Scikit-learn Developers, "Feature extraction: Text feature extraction," scikit-learn documentation. Available: https://scikit-learn.org/stable/modules/feature_ex traction.html