iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85907

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

A Comparative Study of Machine Learning Techniques for Fake News Detection

Pranav Sononi[1], Pritam Waghmale[2], Prof. Ashwini Takale[3]
Master of Computer Application (MCA), JSPM's Narhe Technical Campus, Savitribai Phule Pune University, Narhe, Pune 411041, India

Abstract: ***Fake news spreads quickly through social media and online news platforms, and manual fact-checking cannot keep pace with the volume of content. This paper is a literature-based comparative study of machine-learning (ML) techniques for detecting fake news from text. It covers six families of methods: Naive Bayes, Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and deep-learning and transformer-based models. It also reviews the datasets and text features most often used (Bag of Words, TF-IDF, n-grams, word embeddings, linguistic features and contextual embeddings) and compares reported results by accuracy, precision, recall and F1-score. No new experiments were run, so every figure comes from the cited studies. In the reviewed work, a linear SVM with simple frequency-based features reached 99.81% accuracy on the ISOT dataset in a head-to-head test, against 99.98% for a fine-tuned BERT model. The SVM trained in under 3 minutes on a CPU, whereas BERT needed about 1 hour 45 minutes on a GPU. On the harder six-class LIAR dataset, accuracy stayed below 30%. This gap suggests that very high scores on some benchmarks may partly reflect dataset-specific cues and not an understanding of truthfulness, a point we flag as a hypothesis. We summarise the advantages and limitations of each technique, identify research gaps, and suggest directions for future work.***

Keywords: Fake News Detection, Machine Learning, Text Classification, TF-IDF, Support Vector Machine, Random Forest, BERT, Comparative Study, Misinformation.

I. INTRODUCTION

A. Background and Motivation

Social media, online news portals and messaging applications let information reach millions of people within minutes. That reach has a cost: false and misleading content travels alongside genuine reporting. A large study of Twitter found that false news spread farther and faster than true news, and that the difference came mainly from how people share, not from automated accounts [1]. During the 2016 US presidential election, fake news stories were widely shared on social media, and researchers have asked how far they shaped what voters believed [2]. Reviews of the topic describe fake news as a problem for societies, institutions and individuals, not just for platforms [3].

The consequences are practical. Fake news can steer public opinion, damage reputations, create confusion in emergencies and cause social and economic harm. The study by Karim et al. points to political unrest linked to false content in Bangladesh, mob violence linked to messages spread over WhatsApp in India, and false claims about the virus and vaccines during the COVID-19 pandemic [4]. Because Indian users rely heavily on messaging and social platforms, a tool that can flag suspicious text early is of direct local interest.

Human fact-checkers are accurate but slow, and they cannot read every article. Machine learning offers a way to automate part of the work: a model learns the textual patterns that separate fake articles from genuine ones and then labels new articles in a fraction of a second. Many algorithms and feature representations have been proposed, but published results come from different datasets, splits and preprocessing, so a newcomer cannot easily tell which approach suits which situation. This paper compares the main techniques side by side, explains how each works, and examines what the published evidence does and does not show.

Three difficulties recur across this literature:

- Benchmark dependence. Results near 99% on one dataset sit beside results below 30% on another, so a high score alone says little about real-world performance.
- Cost versus gain. The most accurate models, usually transformers, need far more data, hardware and time than classical models, and the gain can be small.
- Scattered evidence. Comparisons are spread across many papers with different settings, which makes a clear picture hard to build.

©IJRASET: All Rights are Reserved

This study responds by organising the techniques, features, datasets and metrics into one structured comparison, by reporting the within-study comparisons that are most reliable, and by stating plainly where the evidence is thin.

B. Statement of the Problem

The large-scale spread of fake news through digital platforms makes manual verification difficult and time-consuming. There is a need to understand how different ML techniques can automatically identify fake news, and which techniques work best under different conditions.

The difficulty is not only technical. A detector that is accurate on one dataset but fails on articles from other sources gives a false sense of security, and one that wrongly flags genuine news erodes trust. A newcomer who tries to choose a method from the literature meets papers that use different datasets, different features and different metrics, and so cannot tell whether a gap between two numbers reflects the algorithm or the setting. Practitioners also need to know the cost side: how long a model takes to train, what hardware it needs and how readable its decisions are.

The problem addressed here is the lack of a clear, structured comparison of the main ML techniques for text-based fake news detection that sets their working principles, feature choices, reported performance, strengths and limitations side by side, and that is honest about how far published numbers can be compared.

C. Objectives of the Research

The aim is a comparative study of the ML techniques used to detect fake news, analysing their effectiveness, advantages, limitations and challenges. The study sets seven specific objectives:

- Study the problem of fake news and its impact on digital communication.
- Study existing ML techniques used for fake news detection.
- Examine commonly used datasets and textual features for fake news detection.
- Compare ML algorithms based on accuracy, precision, recall and F1-score.
- Identify the advantages and limitations of the different approaches.
- Identify existing research gaps and challenges.
- Suggest directions for improving future fake news detection systems.

D. Research Questions and Propositions

The study addresses four research questions. RQ1: Which ML techniques are most used for text-based fake news detection, and how do they differ in how they work? RQ2: How do feature representations (Bag of Words, TF-IDF, n-grams, embeddings) affect reported results? RQ3: How do the techniques compare on accuracy, precision, recall and F1-score, and at what computational cost? RQ4: What limitations and research gaps remain? Three propositions, drawn from the literature, guide the analysis in Section VI:

P1: On binary, source-separated benchmarks such as ISOT, classical linear models with frequency-based features are competitive with deep models.

P2: Transformer models give the highest reported scores but at a much higher computational cost.

P3: Performance falls sharply on fine-grained, short-statement benchmarks such as LIAR.

Because this is a review, the propositions are examined by comparing reported evidence and not by a statistical test. Section VI states how far the evidence supports each one.

E. Significance of the Study

- *For students and new researchers:* a single organised overview of techniques, features, datasets and metrics shortens the path into the field and shows which questions are already well answered and which are not.
- *For developers and institutions:* the comparison shows what each family costs in data, hardware and training time, which helps when choosing a model for a newsroom tool, a campus project or a low-resource setting. A fast linear model may be the sensible first choice where a transformer is out of reach.
- *For research:* the study lists specific gaps (cross-dataset testing, inconsistent evaluation, label quality, explainability, language coverage) that later work can address. It also separates what is established from what is only a hypothesis, which keeps later studies from building on shaky numbers.
- *For society:* better detection tools support healthier public information, but only if their limits are understood. A review that is explicit about those limits supports careful use.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

F. Scope and Delimitations

The study focuses on detecting fake news from textual news content. It covers Naive Bayes, Logistic Regression, SVM, Decision Tree, Random Forest and deep-learning and transformer-based approaches. It is literature-based: it analyses existing research, datasets, algorithms and reported results and does not build a commercial application. Approaches that rely on images, user networks and propagation patterns are mentioned only where they help place the text-based methods in context.

The study is a selective review, not a systematic survey, and it runs no new experiments. Reported figures come from different datasets and settings and are therefore indicative. These delimitations keep the questions narrow enough to be answered from the sources used and mark clearly what the numbers can and cannot show.

II. REVIEW OF LITERATURE

Research on automated fake news detection has grown quickly since the mid-2010s. This section reviews work in five areas that bear on the objectives in Section I: perspectives and definitions, benchmark datasets, classical ML and feature-based studies, deep-learning and transformer models, and comparative studies. It closes with the research gap.

A. Perspectives on Fake News Detection

Shu et al. surveyed fake news detection on social media from a data-mining perspective. They describe how fake news is characterised and separate detection approaches into those that use news content, those that use the social context of users and sharing, and those that use networks [5]. Zhou and Zafarani organised the field by four perspectives: the knowledge a claim conflicts with, the writing style of the article, the way it propagates, and the credibility of its source [6]. The present study sits in the writing-style and content part of this map: it asks what can be learned from the text alone.

B. Benchmark Datasets

Datasets have shaped what the field can measure. Wang introduced LIAR, a set of short political statements labelled by a fact-checking source on a six-point truthfulness scale, and showed that fine-grained classification is hard: the baseline models, including a hybrid convolutional network that used metadata, reached test accuracy below 30% [7]. Shu et al. built FakeNewsNet, a repository that pairs news content with social context for articles checked by fact-checking sites [8]. The ISOT dataset, distributed on Kaggle as the “Fake and Real News Dataset”, contains full-length articles, with real articles gathered from Reuters and fake articles from sites flagged by fact-checkers [9], [4]. It is binary and has been used in many comparisons.

C. Classical ML and Feature-Based Studies

Ahmed et al. compared two feature extraction methods (term frequency and TF-IDF) and six classifiers (KNN, SVM, Logistic Regression, Linear SVM, Decision Tree and Stochastic Gradient Descent) using n-gram features. Their best configuration was TF-IDF with a Linear SVM, which reached an accuracy of 92% [10]. Pérez-Rosas et al. built new datasets and studied lexical, syntactic and semantic features, reporting F1 scores around 0.78 for the strongest feature sets [11].

Several studies looked at style. Rashkin et al. analysed the language of different kinds of news and of political fact-checking, and showed that stylistic cues differ between reliable and unreliable content [12]. Horne and Adali found that the titles and body text of fake news differ in measurable ways from real news, with fake articles often using simpler and more repetitive text [13]. These findings explain why even simple word-frequency features can separate classes on some datasets, and they also point to a risk, which is that the model may be learning style or source habits.

D. Deep Learning and Transformer Models

Deep models learn features from text directly. Kim showed that a simple convolutional network over word vectors works well for sentence classification [15], and long short-term memory (LSTM) networks model word order over longer spans [16]. Ruchansky et al. proposed CSI, a hybrid deep model that combines the article text, the responses it draws and the characteristics of its source [14]. The transformer architecture replaced recurrence with attention [17], and BERT showed that a model pre-trained on large text collections can be fine-tuned for many tasks with strong results [18]. For fake news, such models read each word in the context of the whole article, at the price of heavy computation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Comparative and Benchmark Studies

Kanavos et al. compared ML algorithms and text vectorisation methods for fake news detection [19]. Karim et al. compared an SVM with Bag of Words, TF-IDF and Word2Vec features, with both linear and RBF kernels, against a fine-tuned BERT-base model on the same ISOT data. This is the most direct head-to-head comparison used in this paper, because the data, split and metrics are shared [4]. Verma et al. proposed WELFake, which combines word embeddings with linguistic features [20].

F. Research Gap Summary

The reviewed work shows real progress, yet four gaps remain. Table I sets each against the focus of the present study.

TABLE I
Research Gap Summary

Aspect	Existing research	Identified gap	Present research focus
Evaluation setting	Most studies test on one dataset and report one score.	Scores on one benchmark may not carry over to other sources.	Report results by dataset and flag where cross-dataset evidence is missing.
Comparability	Studies use different splits, preprocessing and metrics.	Numbers across studies cannot be ranked directly.	Compare mainly within studies that share data and settings.
Cost	Accuracy is the headline measure.	Training time, hardware and interpretability are seldom reported.	Include cost and interpretability in the comparison.
Benchmark difficulty	Binary datasets give near-ceiling scores; fine-grained ones give low scores.	The two regimes are rarely discussed together.	Contrast the regimes and discuss why they differ.

The contribution is therefore not a new detector. It is a structured comparison that places the main ML families, their features, datasets, metrics, costs and limits in one frame, and that is explicit about the limits of comparing published numbers. The review is selective and not a systematic survey, so any claim of novelty applies only to the studies reviewed here.

G. Comparative Review and Implications

Table II summarises how each study relates to this work.

TABLE II
Comparative Analysis of Reviewed Studies

Study	Main approach	Key contribution	Relevance to this study
Shu et al. [5]	Survey	Content, social-context and network perspectives	Frames the content-based scope
Zhou and Zafarani [6]	Survey	Knowledge, style, propagation and source perspectives	Places text-style methods in the wider field
Wang [7]	Dataset + baselines	LIAR, six-class labels	Shows how hard fine-grained labels are
Shu et al. [8]	Data repository	FakeNewsNet, content plus social context	Dataset for later cross-source work
Ahmed et al. [10]	N-grams + six classifiers	Linear SVM with TF-IDF reached 92%	Evidence for linear models on sparse text
Pérez-Rosas et al. [11]	Lexical, syntactic, semantic features	New datasets; F1 about 0.78	Shows value of feature diversity
Rashkin et al. [12]	Language analysis	Style differs by news type	Explains why simple features work
Horne and Adali [13]	Title and body analysis	Measurable style differences	Warns of style and source cues
Ruchansky et al. [14]	Hybrid deep model (CSI)	Text + response + source	Shows value of non-text signals
Kim [15]; Hochreiter and Schmidhuber [16]	CNN; LSTM	Neural text classification	Background for deep models

Study	Main approach	Key contribution	Relevance to this study
Vaswani et al. [17]; Devlin et al. [18]	Transformer; BERT	Attention; pre-train then fine-tune	Basis of the transformer family
Kanavos et al. [19]	ML + vectorisation comparison	Compares algorithms and features	Close in aim to this study
Karim et al. [4]	SVM variants vs BERT on ISOT	Same-data comparison with cost	Main source of head-to-head figures
Verma et al. [20]	Embeddings + linguistic features	WELFake	Supports hybrid feature design

III. RESEARCH METHODOLOGY

A. *Research Design*

The study follows a literature-based comparative design with six stages: (1) literature review, (2) identification of techniques, (3) dataset and feature analysis, (4) comparative analysis, (5) identification of research gaps and (6) future recommendations (Fig. 1). The techniques are the *independent* factor in the comparison. The *dependent* measures are accuracy, precision, recall and F1-score as reported in the sources, together with the computational cost where it is given.

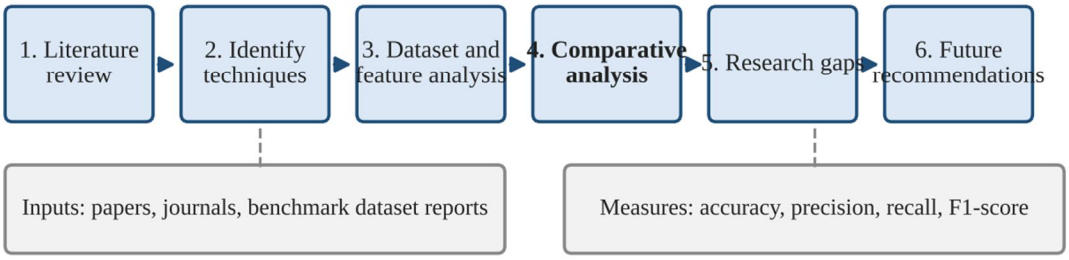


Fig. 1 Research workflow of the comparative study

B. *Review Procedure and Source Selection*

Sources were research papers, journals, conference papers, preprints and dataset descriptions on text-based fake news detection. A source was kept when it (a) described an ML or deep-learning method for classifying news text, (b) named its dataset and features, and (c) reported at least one standard metric. Foundational papers for each algorithm (support vector machines, decision trees, random forests, TF-IDF, word embeddings, BERT) were included so that the working principles are described from their origin. Algorithm 1 gives the procedure.

Algorithm 1: Comparative Review Procedure

Input: a set of candidate papers and dataset reports.

Output**:** a structured comparison.

1. Collect candidate sources on text-based fake news detection.
2. Screen each source against the three inclusion criteria (method, dataset and features, metric).
3. Record, for each kept source, the dataset, features, classifier, metrics and any cost figures.
4. Group the records by classifier family and by dataset.
5. Compare results within each study first; compare across studies only with stated caveats.
6. List strengths, limitations and gaps for each family; check each proposition against the evidence.

C. *Comparison Criteria*

Each technique is compared on six criteria: how it works, the features it suits, reported accuracy and F1-score, training and inference cost, interpretability, and typical weaknesses. Reported performance is compared within a single study wherever possible, because only there do the data, split and preprocessing match.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

D. Evaluation Metrics

The reviewed studies use four standard metrics, defined from the confusion matrix in Fig. 2. TP and TN are fake and real articles classified correctly; FP is a real article flagged as fake; FN is a fake article classified as real. Table III defines the metrics.

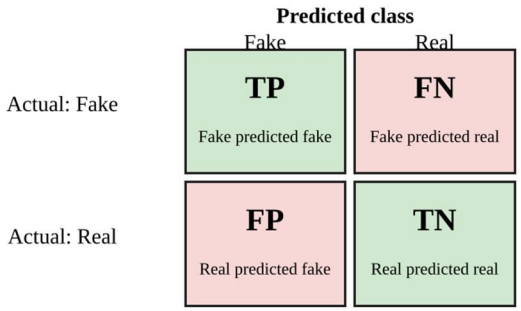


Fig. 2 Confusion matrix used to define the evaluation metrics

TABLE III
Evaluation Metrics

Metric	Formula	What it tells us
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Share of all articles classified correctly
Precision	TP / (TP + FP)	Of the articles flagged as fake, the share that really are fake
Recall	TP / (TP + FN)	Of all fake articles, the share that were found
F1-score	2 × Precision × Recall / (Precision + Recall)	Balance of precision and recall

Accuracy alone can mislead when the classes are imbalanced, so precision, recall and F1-score should be read together. The two error types also carry different costs: a false negative lets misinformation spread, while a false positive wrongly labels genuine news and raises concerns about censorship.

E. Threats to the Validity of the Comparison

- Different data and splits. Scores from different studies are not on a common scale.
- Selective coverage. The review is not systematic, so relevant studies may be missing.
- Secondary reporting. Some figures reach this paper through a later study's summary and should be checked against the originals.
- No replication. No result was reproduced by the authors of this paper.

IV. MACHINE LEARNING TECHNIQUES COMPARED

A. General Pipeline

Fake news detection from text is usually treated as supervised classification. Each article is cleaned, converted into a numeric feature vector and passed to a classifier trained on labelled examples to predict fake or real (Fig. 3). The study compares the choices at the two middle stages, the features and the classifier.

©IJRASET: All Rights are Reserved

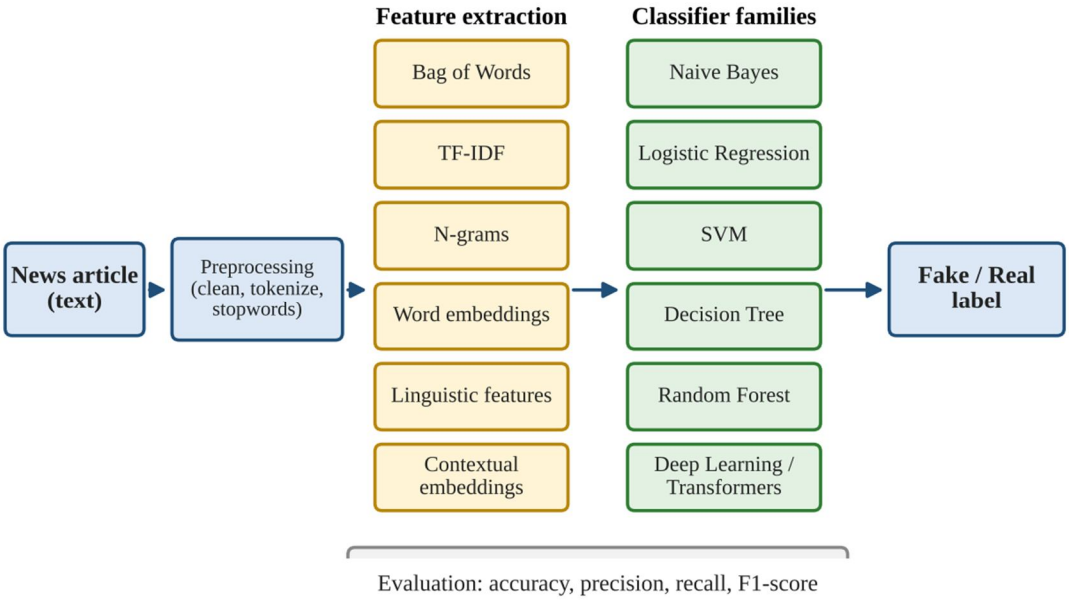


Fig. 3 Typical text-based fake news detection pipeline and the choices compared in this study

B. Preprocessing and Feature Extraction

Preprocessing usually lowercases the text, removes punctuation and stopwords, and splits it into tokens. In the ISOT experiments of Karim et al., stopwords were removed with a standard list and words shorter than three characters were dropped [4]. The cleaned text is then turned into numbers in one of several ways:

- Bag of Words (BoW) counts how often each word occurs. It is simple and effective but ignores word order and meaning.
- TF-IDF weights each word by its frequency in the document and lowers the weight of words that are common across the corpus [21]. It is the most widely used representation for classical models.
- N-grams count sequences of n consecutive words or characters and so capture some local order. Ahmed et al. varied the n-gram size and the number of features from 1,000 to 50,000 [10].
- Word embeddings such as Word2Vec give each word a dense vector so that words with similar meanings lie close together [22].
- Linguistic and stylistic features include punctuation, sentence length, sentiment and part-of-speech patterns, and capture how an article is written [11], [12].
- Contextual embeddings from transformer models give the same word different vectors in different contexts [18].

C. Naive Bayes

Naive Bayes applies Bayes' theorem and assumes that the words in a document are independent once the class is known. Training only needs counts, so it is very fast, works with little data and suits sparse word-count features. The independence assumption is unrealistic for language, which can limit accuracy when meaning depends on word order or context.

D. Logistic Regression

Logistic Regression estimates the probability that an article is fake from a weighted sum of its features. It is quick to train and easy to interpret, because the learned weights show which words push a prediction towards fake or real. It is a linear model, so it cannot capture complex non-linear patterns without extra feature engineering.

E. Support Vector Machine (SVM)

The SVM finds the boundary that separates the two classes with the widest margin [23]. With a linear kernel it handles high-dimensional sparse text features very well. With a kernel such as the radial basis function (RBF) it can model non-linear boundaries. The SVM objective balances a wide margin against classification errors, controlled by a regularisation parameter. In the studies reviewed, a linear SVM was either the best classical model [10] or came very close to the best overall model [4]. SVMs need tuning of the kernel and regularisation and become slower on very large datasets.

F. Decision Tree

A Decision Tree repeatedly splits the data on the feature that best separates the classes [24]. The resulting rules are easy to read and explain. A single tree tends to overfit, especially with the very large feature spaces that text produces.

G. Random Forest

Random Forest builds many decision trees on random subsets of the data and features and combines their votes [25]. Averaging reduces the overfitting of a single tree and usually improves stability and accuracy. The cost is higher memory use and training time, and the model is harder to interpret than one tree.

H. Deep Learning and Transformer-Based Models

Convolutional and recurrent networks learn their own feature representations from word embeddings [15], [16]. Transformers use attention to relate every word to every other word, and BERT is first pre-trained on a large text collection and then fine-tuned on the labelled news data [17], [18]. These models give the highest reported scores, but they need large amounts of data, GPU hardware and long training times, and their decisions are hard to explain.

I. Implementation Sketch

Listing 1 shows how a standard classical pipeline is usually assembled with a common ML library [26]. It is an illustration of the method, not code from any reviewed study, and no result in this paper comes from it.

Listing 1 Python-style sketch of a TF-IDF and linear SVM pipeline (illustrative)

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# texts: list of article strings, labels: 1 = fake, 0 = real
X_tr, X_te, y_tr, y_te = train_test_split(
    texts, labels, test_size=0.2, stratify=labels, random_state=42)

vec = TfidfVectorizer(stop_words='english', max_features=50000)
clf = LinearSVC().fit(vec.fit_transform(X_tr), y_tr)
pred = clf.predict(vec.transform(X_te))
print(classification_report(y_te, pred, digits=4))
```

The vectoriser is fitted on the training split only, so the test articles never influence the features. This matters because leakage between training and test data is one common way to inflate a reported score.

V. DATASETS AND TEXTUAL FEATURES

A. Commonly Used Datasets

TABLE IV
Datasets Appearing in the Reviewed Literature

Dataset	Description	Typical use
ISOT / Kaggle “Fake and Real News”	Full-length articles; real ones from Reuters, fake ones from sites flagged by fact-checkers. After preprocessing, Karim et al. used 21,477 real and 23,421 fake items [4], [9].	Binary classification; classical ML and BERT comparisons
LIAR	About 12,800 short political statements with six truthfulness labels from a fact-checking source [7].	Fine-grained classification; much harder than binary tasks
FakeNewsNet	News content with social context for articles checked by fact-checking sites [8].	Studies that combine text with social signals
FakeNewsAMT and Celebrity	Crowd-written and web-collected articles built for linguistic analysis [11].	Studying lexical, syntactic and semantic cues

B. Why the Dataset Matters

The same classifier can look excellent on one dataset and weak on another because the datasets pose different tasks. ISOT asks for a two-way decision on full articles from very different sources. LIAR asks for a six-way decision on a single short sentence, where little text is available. The dataset also fixes how the label was produced: in ISOT the label largely follows the source of the article, whereas in LIAR it follows a fact-checker's judgement of one claim.

VI. RESULTS AND DISCUSSION

A. Results Reporting Principle

This section reports what the reviewed studies found. No new experiments were run, so every figure below is taken from the cited source and is indicative, not a controlled benchmark. Rows should be compared within a study and not across studies.

B. Reported Results

TABLE V
Reported Results from the Reviewed Studies

Study	Dataset	Features	Model	Reported result
Ahmed et al. [10]	Compiled news dataset (6 classifiers tested)	TF-IDF, unigrams	Linear SVM (best)	Accuracy 92%
Pérez-Rosas et al. [11]	FakeNewsAMT / Celebrity	Lexical, syntactic, semantic	Linear classifier	F1 about 0.78
Wang [7]	LIAR (6 classes)	Text and metadata	Logistic Regression, CNN baselines	Accuracy below 30%
Karim et al. [4]	ISOT	Bag of Words	SVM, linear kernel	Accuracy 99.81%, F1 0.9980
Karim et al. [4]	ISOT	Bag of Words	SVM, RBF kernel	Accuracy 99.62%, F1 0.9961
Karim et al. [4]	ISOT	TF-IDF	SVM, linear kernel	Accuracy 99.52%, F1 0.9949
Karim et al. [4]	ISOT	TF-IDF	SVM, RBF kernel	Accuracy 99.31%, F1 0.9928
Karim et al. [4]	ISOT	Word2Vec	SVM, RBF kernel	Accuracy 97.75%, F1 0.9767
Karim et al. [4]	ISOT	Word2Vec	SVM, linear kernel	Accuracy 96.54%, F1 0.9644
Karim et al. [4]	ISOT	Subword tokens	BERT-base, 3 epochs	Accuracy 99.98%, F1 0.9998

Fig. 4 plots the within-study ISOT results of Karim et al. Simple BoW and TF-IDF features gave the best SVM results, Word2Vec was lower, the RBF kernel changed results only slightly and not in one direction, and BERT scored highest [4]. On the test split of 8,980 articles, BERT made 7 false positives and 19 false negatives [4].

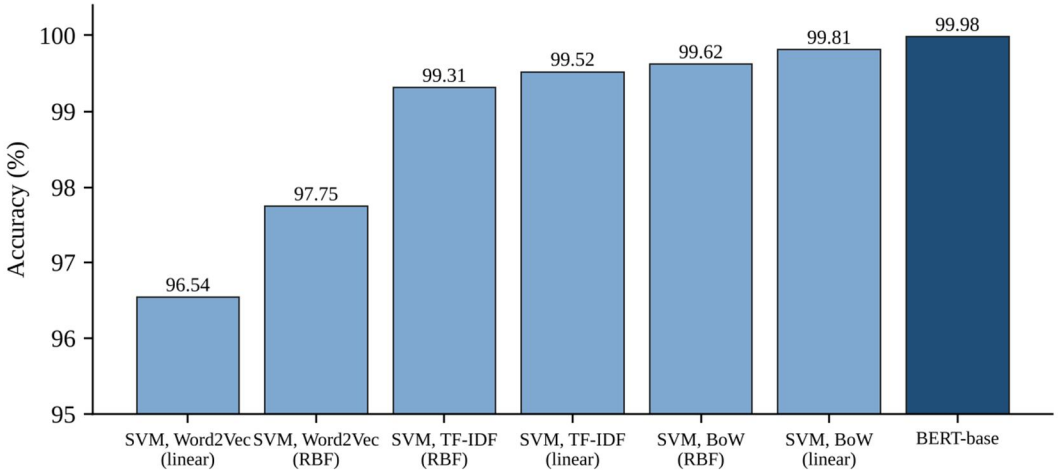


Fig. 4 Accuracy on ISOT for SVM variants and BERT-base (data from Karim et al. [4]); the vertical axis starts at 95% to make small differences visible

C. Computational Cost

The same study reports that BERT-base took about 1 hour 45 minutes to train for three epochs on a 16 GB NVIDIA Tesla T4 GPU, while the SVM trained in under 3 minutes on an ordinary Intel Core i7 CPU with 16 GB of RAM [4] (Fig. 5). The accuracy gain from BERT is about 0.2 percentage points over the best SVM. On this dataset the gain is small relative to the cost, although the cost may be justified where even a few errors matter or where the data are harder.

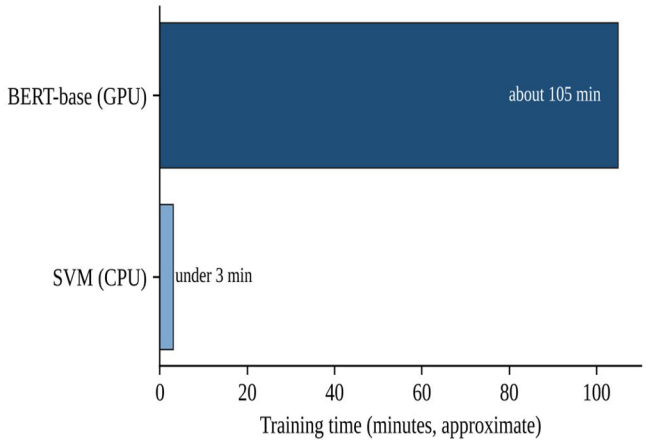


Fig. 5 Approximate training time for SVM and BERT-base on ISOT (from Karim et al. [4])

D. Results Across Datasets

Fig. 6 sets the best scores reported on different datasets side by side. The bars are not on a common scale, because the tasks, labels and metrics differ, and the figure is meant only to show the contrast between regimes. Binary, source-separated ISOT gives scores near 99%, the Ahmed et al. data gave 92%, the feature study of Pérez-Rosas et al. gave an F1-score near 0.78, and six-class LIAR stayed below 30% [4], [7], [10], [11].

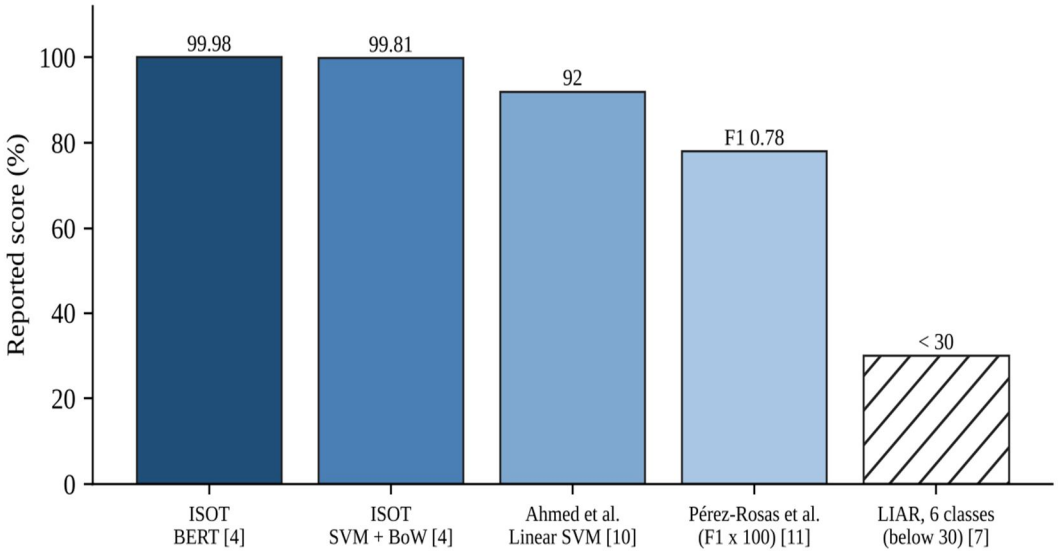


Fig. 6 Best reported scores by dataset; tasks and metrics differ, so bars are not directly comparable

Studies summarised in [4] also report 92.4% for a CNN-LSTM model, 89.2% for CSI, 94.1% for a multimodal variational autoencoder and 96.7% for a CNN with a BiLSTM. These come from differing datasets, so they are not ranked against the results above.

E. Advantages and Limitations

TABLE VI
Summary of the Techniques

Technique	Advantages	Limitations
Naive Bayes	Very fast; works with small data; simple	Independence assumption; weaker on context
Logistic Regression	Fast; interpretable weights; strong baseline	Linear; limited with complex patterns
SVM	Excellent with high-dimensional sparse text; strong results with TF-IDF and BoW [4], [10]	Needs tuning; slower on very large data
Decision Tree	Easy to interpret; little preprocessing	Overfits; unstable
Random Forest	More stable and accurate than one tree; handles many features	More memory and time; less interpretable
Deep learning / transformers	Learns context; highest reported scores [4]	Needs large data and GPUs; costly; hard to interpret

Table VI is a qualitative summary built from the algorithm descriptions and the reviewed studies. Only the SVM and BERT rows are backed by head-to-head figures in the sources used here; the other rows describe general properties of the algorithms, and no reviewed result was found that ranks Naive Bayes, Logistic Regression, Decision Tree and Random Forest against the others under identical conditions.

F. Discussion

Four observations follow from the reviewed work.

- Feature choice matters as much as classifier choice. Frequency-based features (BoW, TF-IDF) with a linear SVM were highly competitive in the studies reviewed [4], [10]. Word2Vec was lower on ISOT, so richer features are not automatically better.
- Cost changes the ranking. BERT scored highest, but a 0.2-point gain for more than 35 times the training time is a real trade-off [4].
- The benchmark decides the picture. Near-99% on ISOT against below 30% on six-class LIAR shows that the task, not only the model, drives the score [4], [7].
- High scores may partly reflect dataset cues. In ISOT the real articles come from one news agency while the fake ones come from many websites [4]. Our interpretation, which we flag as a hypothesis and not an established finding, is that part of the very high accuracy may come from source-specific style and formatting cues and not from understanding whether a claim is true. Style differences between fake and real news are documented [12], [13], which makes this plausible. Testing on sources not seen in training would show how much carries over.

Table VII sets the propositions against the evidence.

©IJRASET: All Rights are Reserved

TABLE VII
Interpretation of Reported Evidence Against the Propositions

Proposition	Evidence in the reviewed studies	Reading
P1: classical linear models are competitive on binary benchmarks	SVM with BoW reached 99.81% against 99.98% for BERT on ISOT [4]; Linear SVM with TF-IDF reached 92% in [10].	Supported on ISOT; evidence is from few studies and one dominant dataset.
P2: transformers score highest at much higher cost	BERT 99.98% vs SVM 99.81%; about 1 h 45 min on GPU vs under 3 min on CPU [4].	Supported in the one study that reports both; needs more studies.
P3: performance falls on fine-grained benchmarks	LIAR baselines below 30% [7].	Supported by the original benchmark paper; newer models were not reviewed here.

In summary, the reviewed evidence supports all three propositions in a limited way. Each rests on a small number of studies, and no result was reproduced for this paper, so the findings should be read as a well-organised picture of the literature and not as new measurements.

VII. FINDINGS AND SUGGESTIONS

A. *Findings*

- Linear models with simple features are strong. A linear SVM with BoW or TF-IDF features was within a fraction of a percentage point of BERT on ISOT [4].
- Transformers score highest but cost much more. The gain in the one direct comparison was small relative to the added training time and hardware [4].
- Features matter. Frequency-based features beat Word2Vec in the same SVM setting on ISOT [4].
- Results depend on the benchmark. Binary, source-separated data give near-ceiling scores; fine-grained data do not [4], [7].
- Comparisons are fragile. Different datasets, splits and metrics make cross-study ranking unreliable.
- Interpretability varies. Linear models and single trees can be inspected; deep models are hard to explain.

B. *Suggestions*

- Start simple. Use TF-IDF with a linear SVM or Logistic Regression as a baseline before moving to heavier models.
- Test across sources. Train on one source or dataset and test on another before trusting a score.
- Report cost. State training time, hardware and model size beside accuracy.
- Report more than accuracy. Give precision, recall and F1-score, and the confusion matrix.
- Consider hybrids. Use a transformer for contextual embeddings and a light classifier for the final decision [4].
- Check for leakage and source cues. Fit vectorisers on training data only and inspect which words drive predictions.

C. *Implications for Practice*

For a student project or a small organisation, a classical pipeline is cheap to build, quick to train and easy to explain, and the reviewed evidence suggests it can match much heavier models on clean binary data. For high-stakes use, a transformer may add robustness, but only with testing on articles from sources it has never seen.

VIII. FUTURE SCOPE

1) *Cross-dataset and cross-source evaluation.* Models should be trained on one source and tested on others, so that reported accuracy reflects real use.
2) *Hybrid pipelines.* Transformer embeddings paired with light classifiers such as SVM or Logistic Regression could combine contextual understanding with low cost [4].
3) *Multimodal and social signals.* Adding source credibility, user behaviour, propagation patterns and images could improve robustness beyond text alone [14].
4) *Explainable detection.* Methods that show which words or claims led to a label would make predictions easier to trust and audit.

©IJRASET: All Rights are Reserved

5) *Indian languages and code-mixed text.* Most work targets English. Datasets and models for Hindi, Marathi and code-mixed social-media text would serve Indian users directly.
6) *Standard benchmarks and reporting.* Shared splits, metrics and cost reporting would make future comparisons direct.
7) *Current large language models.* Newer transformer models were outside the scope of this review and deserve their own comparison.

IX. LIMITATIONS OF THE STUDY

TABLE VIII
Limitations, Their Effect, and a Possible Way Forward

Limitation	Effect	Possible way forward
Literature-based, no experiments	No result was reproduced by the authors	Replicate key comparisons on shared data
Selective, not systematic, review	Relevant studies may be missing	Follow a formal systematic-review protocol
Different datasets and settings	Cross-study numbers are not comparable	Compare within studies; run a common benchmark
Head-to-head evidence from few studies	SVM versus BERT rests mainly on one study	Add more same-data comparisons
Secondary reporting of some figures	Possible transcription or context errors	Check each figure against the original paper
Text-only scope	Ignores source, user and image signals	Include multimodal work
Mostly English-language sources	Little guidance for other languages	Add regional-language studies

The most important limit is that the numbers here come from other people's experiments. They show how the techniques behaved in those settings, and they should not be taken as a guarantee of performance on new data.

X. CONCLUSION

This paper presented a literature-based comparative study of ML techniques for fake news detection, covering Naive Bayes, Logistic Regression, SVM, Decision Tree, Random Forest and deep-learning and transformer models, together with the features, datasets and metrics used to evaluate them. In the reviewed evidence, simple frequency-based features with a linear SVM were strong and efficient, a fine-tuned BERT model scored slightly higher at a much greater computational cost, and performance dropped sharply on the harder six-class LIAR benchmark. Important gaps remain in cross-dataset generalisation, consistent evaluation, label quality, explainability and language coverage. We hope this structured overview gives students and practitioners a clear starting point and helps later work to build more reliable and practical misinformation-detection systems.

APPENDIX

A. *Summary of Reported Results*

TABLE IX
Summary of Key Reported Results

Result	Value	Source
Best ISOT accuracy (BERT-base)	99.98%	[4], Table V, Fig. 4
Best ISOT accuracy for SVM (BoW, linear)	99.81%	[4], Table V, Fig. 4
BERT training time (T4 GPU)	About 1 h 45 min	[4], Fig. 5
SVM training time (CPU)	Under 3 min	[4], Fig. 5
Linear SVM with TF-IDF (Ahmed et al.)	92%	[10], Table V
LIAR six-class baselines	Below 30%	[7], Table V

B. Research Integrity and Acknowledgment Note

All quantitative results in this paper are quoted from the cited studies. No experiments were run by the authors, and no claim is made that the figures are reproducible without consulting the sources. Readers should check each figure and reference against the original publication. The authors thank their guide, Prof. Ashwini Takale, and JSPM's Narhe Technical Campus, Savitribai Phule Pune University, for their guidance and support.

C. Glossary of Terms and Abbreviations

TABLE X
Glossary of Terms and Abbreviations

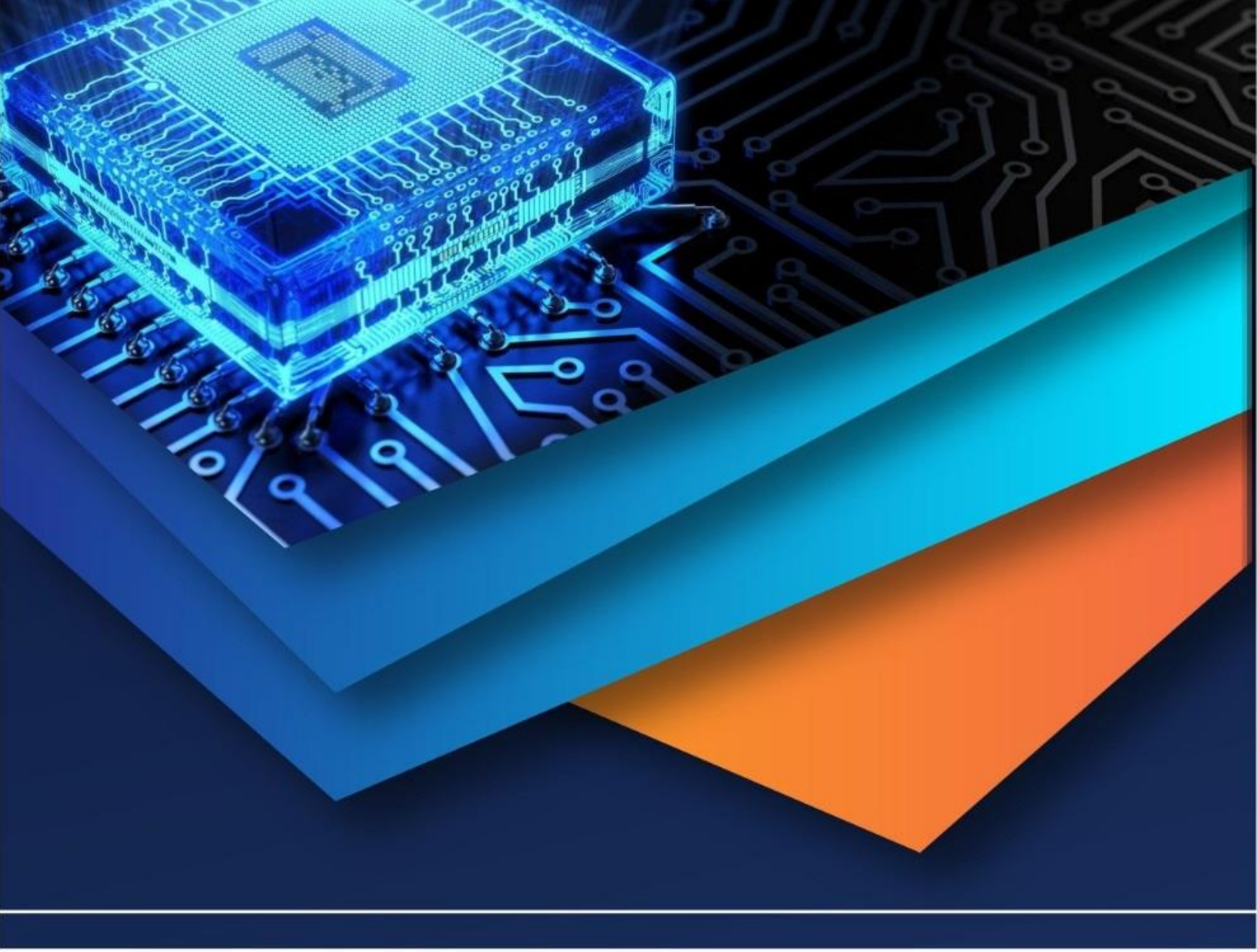
Term	Meaning
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag of Words
CNN	Convolutional Neural Network
FN / FP	False Negative / False Positive
ISOT	Information Security and Object Technology research group dataset
LSTM	Long Short-Term Memory
ML	Machine Learning
RBF	Radial Basis Function
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TP / TN	True Positive / True Negative

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] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," J. Econ. Perspect., vol. 31, no. 2, pp. 211–236, 2017.
[3] D. M. J. Lazer et al., "The science of fake news," Science, vol. 359, no. 6380, pp. 1094–1096, 2018.
[4] A. A. J. Karim, K. H. M. Asad, and A. Azam, "Strengthening false information propagation detection: Leveraging SVM and sophisticated text vectorization techniques in comparison to BERT," arXiv:2411.12703, 2024.
[5] 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.
[6] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," ACM Comput. Surv., vol. 53, no. 5, art. 109, 2020.
[7] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Short Papers), 2017, pp. 422–426.
[8] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," Big Data, vol. 8, no. 3, pp. 171–188, 2020.
[9] C. Bisaillon, "Fake and Real News Dataset," Kaggle, 2017. [Online]. Available: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset
[10] 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, 2017, pp. 127–138.
[11] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in Proc. 27th Int. Conf. Comput. Linguistics (COLING), 2018, pp. 3391–3401.
[12] 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. Conf. Empirical Methods Natural Language Process. (EMNLP), 2017, pp. 2931–2937.
[13] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," in Proc. Int. AAAI Conf. Web Social Media Workshops, 2017.
[14] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in Proc. ACM Conf. Inf. Knowl. Manage. (CIKM), 2017, pp. 797–806.
[15] Y. Kim, "Convolutional neural networks for sentence classification," in Proc. Conf. Empirical Methods Natural Language Process. (EMNLP), 2014, pp. 1746–1751.
[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
[17] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst. 30 (NIPS), 2017, pp. 5998–6008.

[18] 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.
[19] A. Kanavos, I. Karamitsos, A. Mohasseb, and V. C. Gerogiannis, "Comparative study of machine learning algorithms and text vectorization methods for fake news detection," in Proc. 14th Int. Conf. Inf., Intell., Syst. Appl. (IISA), 2023.
[20] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, "WELFake: Word embedding over linguistic features for fake news detection," IEEE Trans. Comput. Social Syst., 2021.
[21] K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," J. Documentation, vol. 28, no. 1, pp. 11–21, 1972.
[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
[23] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.
[24] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81–106, 1986.
[25] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[26] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)