computers

Review

Fake News Detection Using Machine Learning and Deep Learning Algorithms: A Comprehensive Review and Future Perspectives

Faisal A. Alshuwaier [1,2] and Fawaz A. Alsulaiman [1,*]

[1] Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; 445107903@student.ksu.edu.sa or shuwaier@kacst.gov.sa
[2] AI & Robotics Institute, KACST, Riyadh 11543, Saudi Arabia
* Correspondence: falsulaiman@ksu.edu.sa

Abstract

Currently, with significant developments in technology and social networks, people gain rapid access to news without focusing on its reliability. Consequently, the proportion of fake news has increased. Fake news is a significant problem that hinders societies today, as it negatively impacts many aspects, including politics, the economy, and society. Fake news is widely disseminated via social media through modern digital platforms. In this paper, we focus on conducting a comprehensive review on fake news detection using machine learning and deep learning. Additionally, this review provides a brief survey and evaluation, as well as a discussion of gaps, and explores future perspectives. Through this research, this review addresses various research questions. This review also focuses on the importance of machine learning and deep learning for fake news detection, by providing a comparison and discussion of how they are used to detect fake news. The results of the review, presented between 2018 and 2025, with the most commonly used publishers being IEEE, Intelligent Systems, EMNLP, ACM, Springer, Elsevier, JAIR, and others, can be used to determine the most effective algorithm in terms of performance. Therefore, articles that did not demonstrate the use of algorithms or performance were excluded.

Keywords: fake news detection; machine learning; deep learning; accuracy; feature engineering; algorithms; datasets

Academic Editors: Phivos Mylonas and Paolo Bellavista

Received: 10 July 2025
Revised: 2 September 2025
Accepted: 11 September 2025
Published: 16 September 2025

Citation: Alshuwaier, F.A.; Alsulaiman, F.A. Fake News Detection Using Machine Learning and Deep Learning Algorithms: A Comprehensive Review and Future Perspectives. *Computers* **2025**, *14*, 394. https://doi.org/10.3390/computers14090394

1. Introduction

In recent times, the world has become very fast-paced. Therefore, this rapid development, especially in the digital world, has several advantages and disadvantages. Due to the ease of accessing news without verifying its reliability, the prevalence of fake news has increased. One of the major drawbacks of the digital era is the rapid spread of misinformation. Individuals can unintentionally or deliberately disseminate fake news, potentially causing harm or offense to others or to organizations. Moreover, the spread of fake news can serve as a tool for propaganda against individuals through various online platforms [1–3]. On the contrary, machine learning and deep learning algorithms, which are part of artificial intelligence, have been utilized recently for the purpose of detecting fake news or prediction. The algorithms are first trained with a training dataset that contains both fake news and legitimate news. After training, those previously trained models are validated and tested. Then, the models are deployed to perform other tasks, such as predicting or revealing clues that aid in identifying fake news [1–5]. Online platforms prioritize delivering news in a convenient, accessible, and rapid manner. However, this speed and ease of access also create greater opportunities for the dissemination of fake news. As a result, efforts have been made by individuals and organizations to verify and expose false information. Detecting fake news remains a significant challenge. Numerous researchers are addressing this issue by employing machine learning and deep learning algorithms, training these models to identify fake content. Once adequately trained, these algorithms can automatically detect fake news with a certain degree of accuracy [6–8].

The accuracy of the classifier in detecting fake news must be observed in order for it to function properly, as failing to detect fake news might be harmful to different people. Some popular classifiers that are used for this purpose in machine learning are given below: naïve bayes, support vector machines (SVMs), random forests, k-nearest neighbors (KNNs), decision trees, and logistic regression. Some common deep learning algorithms used for this purpose are convolutional neural networks (CNNs), bidirectional long short-term memory networks (BI-LSTMs), recurrent neural networks (RNNs), and graph neural networks (GNNs) [9–16]. Figure 1 shows the concept of detecting fake news using machine or deep learning algorithms.

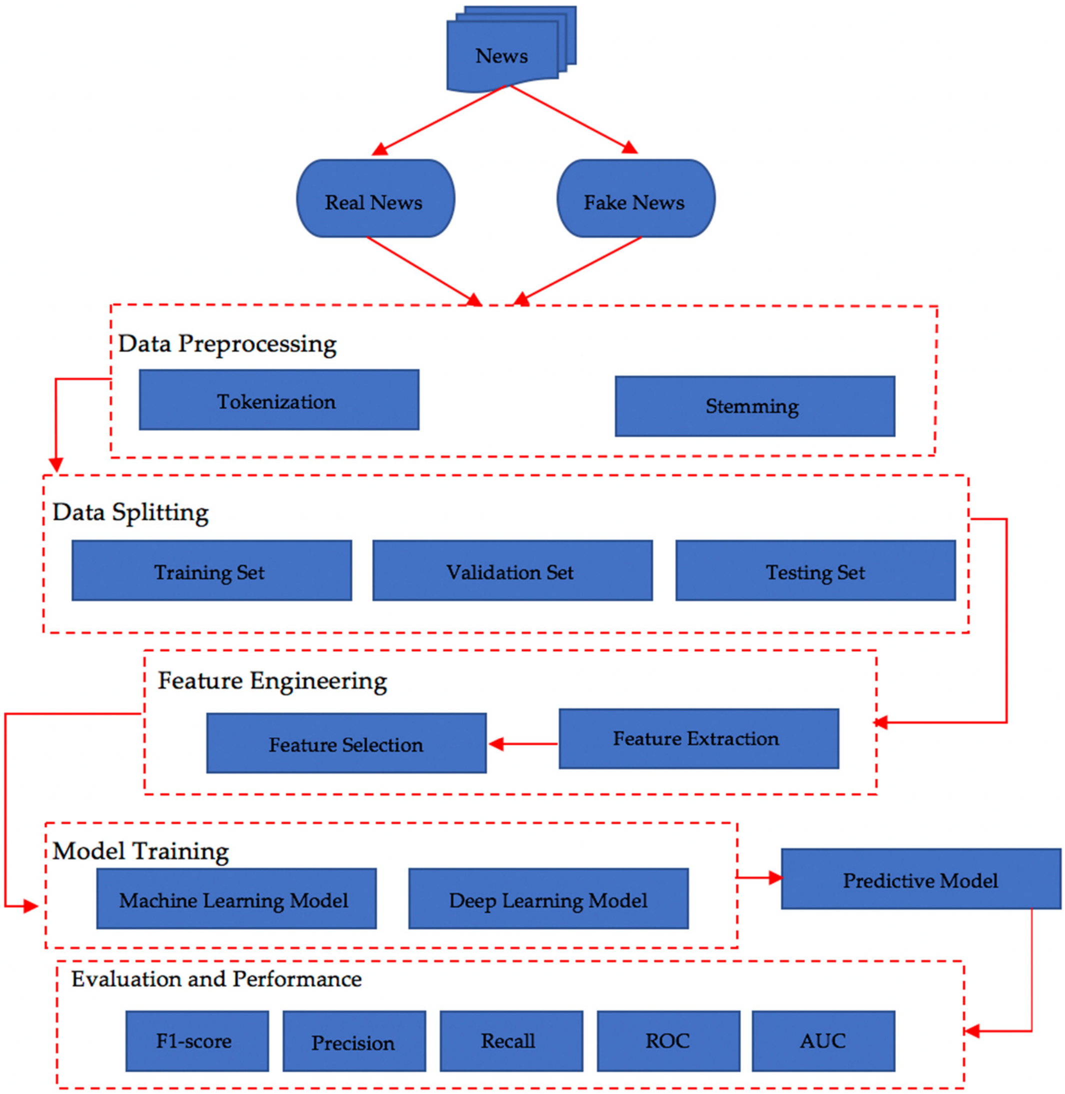

Figure 1. Detecting fake news using machine or deep learning algorithms.

The research questions of the literature review will be answered by focusing on machine learning and deep learning for fake news detection. They will also address how machine learning and deep learning can be utilized for fake news detection through examining the relevant work in the literature. This can serve as a stepping stone toward developing a methodology for this research. Papers from various databases will be presented, utilizing the inclusion and exclusion technique, which will be discussed in this literature review [17–21].

The quality of all literature reviews of the collected research papers will be evaluated based on the research presented in those papers. Papers in which researchers have demonstrated the use of machine learning and deep learning to detect fake news will be considered high-quality papers and included in this research.

Qualitative research methods will be used to collect data. Qualitative research uses non-numerical data to understand and interpret fake news detection experiences using machine learning and deep learning by making comparisons between previous scientific papers to extract results, for example, algorithms, datasets, years of publication, features, and accuracy. The rest of the paper is structured around the related works in Section 2. Section 3 explains the methodology and research questions. Section 4 presents the results and discussion, and the conclusion is presented in Section 5. Finally, references are provided for the papers discussed in this literature review.

2. Related Works

In this section, we will classify previous studies based on detecting fake news using machine learning, deep learning, or both.

2.1. Machine Learning

Aphiwongsophon and Chongstitvatana [1] employed three machine learning methods to detect fake news: naïve Bayes, neural networks, and support vector machines (SVMs). Moreover, with the use of Twitter API, they extracted twenty-two features. As a result, naïve Bayes achieved an accuracy of over 96%, while neural networks and SVMs yielded an accuracy of 99.90%.

Natural language processing (NLP) techniques were employed in this research to distinguish real news from “fake news”, which comes from unreliable sources. The authors relied on building a model based on a count vector (using word statistics) or TF-IDF matrix (term frequency–inverse document frequency) (word statistics for how often they are used in other articles in a given dataset). However, these models carry out important features such as word organization and context. Therefore, the probability that two articles with similar word counts may be completely different in meaning is high. The dataset used in this model is the Kaggle “Fake News Challenge”. So, the proposed work preprocessed the dataset of fake and real news of the articles and employed a naïve Bayes classifier to build a binary word-based model to classify the news correctly. As a result, it achieved an accuracy of 92.20% [2].

In the study by Ni et al. [3], the features of fake news were examined to detect any sudden changes in the news context by using propensity score matching (PSM) to extract document frequency features that include all variables in order to mitigate the effects of unwanted variables. The experimental data was from open-source FakeNewsNet, which consists of data from PolitiFact and GossiCop, and the results demonstrated that PSM is more applicable to fake news than solely raw PSM, which also performs better than relying on raw frequency for feature selection. They achieved an accuracy of 68%. With the PolitiFact dataset, various fake news classifiers, including logistic regression, random forest, and support vector machine, were considered to evaluate the performance and observe the improvements [3].

Singh et al. [4] have compared ensemble learning models to sort fake news by analyzing the quality of the report and knowing the truth of the news. The aim of the paper was to use natural language processing (NLP) and machine learning (ML) algorithms to detect fake news based on the context of the news. They employed decision trees, random forest, AdaBoost classification, and XGBoost as classifiers. They utilized TF, TF-IDF, and word embedding as features that are fed to the aforementioned classifiers. Thus, a web application was developed to reduce the challenges users face in distinguishing fake news.

In this paper, the authors relied on analyzing fake news as a two-dimensional classification approach using content and context features [8]. Therefore, experiments were performed on the tree-based ensemble machine learning framework (gradient boosting) with full content-based modeling to detect fake news. The experimental results demonstrated higher accuracy compared to existing benchmarks, with the gradient boosting algorithm (an ensemble machine learning framework) achieving 86% accuracy in multi-class fake news classification [8].

Albahr and Albahr [9] examined several traditional machine learning algorithms, namely random forests, naïve Bayes, neural networks, and decision trees, to verify the classification performance in detecting fake news based on unigram, bigram, and trigram features. Training was performed on one of the popular datasets known as LIAR, and the results showed that naïve Bayes significantly outperforms its counterparts, achieving an accuracy of 99.0%.

Goldani et al. [10] focused on using capsule neural networks in the fake news detection process. Various embedding models with different lengths were utilized. In the case of short-length news items, fixed word embeddings and n-gram features were used, but for medium-length or large news items, non-fixed word embeddings that support progressive training were used. Moreover, different levels of n-grams were applied to extract features. For the evaluation process, they were trained on two recently known datasets in this field, namely ISOT and LIAR. The study demonstrated strong performance, with the new methods passing 7.8% on ISOT, while achieving similar performance on LIAR dataset with more than 3% on the validation set, and 1% on the test set.

Birunda and Devi [13] used a textual feature model from authentic and fake news texts using a term frequency equation. To calculate the credibility rating of sources, they relied on the characteristics of the website's URL and top-level domain. By combining the TF-IDF, site_URL, and text-based features with the credibility rating of multiple sources, the credibility of the news was estimated. The experimental dataset collected from Kaggle contains 2050 news articles. The model was applied to machine learning (ML) classifiers to test its effectiveness in detecting fake news. Experimental results indicated that the proposed model achieved a maximum effectiveness of approximately 99.5%.

Mugdha et al. [14] demonstrated a model capable of detecting fake news based on news headlines by constructing a new dataset for the Bengali language. Using a Gaussian naïve Bayes algorithm, the model achieved acceptable performance. This algorithm used a TF-IDF-based text feature and an additional tree classifier for feature selection. The accuracy rate reached 87%, which is relatively better than any other algorithm used in this model.

Jardaneh et al. [16] used new features related to text containing user sentiment to detect fake news in Arabic. Sentiment analysis advanced the prediction process. Several machine learning algorithms were utilized to train classification models, including random forests, decision trees, AdaBoost, and logistic regression. As a result, they demonstrated that the system was able to detect fake news with an accuracy of 76%.

Tiwari and Jain [22] compared several machine learning algorithms, using decision tree classification, random forest classifiers, and logistic regression with the HSpam14 dataset, which contains a collection of 400,000 tweets and semantic features. The results demonstrated accuracy in identifying selected news items, with an accuracy rate between 98 and 99%.

Rampurkar and D.R [23] preprocessed the input texts to identify their features. The TF-IDF concept was used to estimate the importance of words in each article. The news items were then segmented using a naïve Bayesian algorithm to distinguish true news from fake news. The ISOT dataset contains 23,481 data pieces. This algorithm calculates the probability of classifying an article, assuming that the word is conditionally independent. The efficiency of the algorithms used was then determined using a confusion matrix to evaluate the validity of the model. The results showed that logistic regression performed well in detecting fake news, with an accuracy of 98.31%.

Mutri et al. [24] focused on developing a method for detecting fake news by sorting and analyzing past data using machine learning. Various machine learning methods have been used, including the proposed KNN and SVM algorithm as an effective solution for detecting fake news. KNN is a machine learning algorithm that classifies texts based on proximity to known data in features such as categorical and datetime. This method was used due to its ability to handle nonlinear data and its ease of use. Applying the KNN can increase the efficiency of identifying fake news by leveraging the characteristics of nearby text. In a study conducted using the FakeNewsDetection dataset, the KNN algorithm performed better than other models, achieving a mean absolute error (MAE) of 0.011, which measures the average size of false detections in a set of predictions without taking their direction into account, and a root mean square error (RMSE) of 0.077, which tells the square root of the mean squared difference between the predicted and observed outcomes of data.

2.2. Deep Learning

Gereme et al. [6] presented several models, including the Amharic fake news detection model, an Amharic language dataset (GPAC), the ETH_FAKE dataset, and Amharic FastText word embedding features (AMFTWE). Thus, the model developed using the ETH_FAKE dataset achieved superior accuracy, above 99% using the 300-and 200-dimension embedding.

Detecting fake news is a challenge for many researchers, especially when news is being circulated through social media platforms. This helps to identify false and misleading stories across social media. One of the key challenges in this area of research is the limited availability of data for training detection models. A novel method for automatically generating misleading (and possibly fake) Arabic news stories was presented by Nagoudi et al. [25]. Part of speech (POS) tagging and word embedding features were used. To facilitate future research, this requirement will be completely eliminated by providing a ready-to-use dataset called AraNews. Finally, models were developed for Arabic fake news detection, achieving an accuracy exceeding 70% [25].

Hamed et al. [26] focused on extracting features, specifically for sentiment analysis of news articles, which includes user comments about this news and emotion analysis features. These features, along with the news content feature, were added to a bidirectional short-term memory model for fake news detection. The standard Fakeddit dataset with published headlines was used to train and test the proposed model. The detection accuracy was high, at 96.77%, representing the highest percentage compared to other recent studies.

Verma et al. [27] proposed a two-step standard model called WELFake based on word embedding (WE) by introducing linguistic features to detect fake news using machine learning classification. The first step pre-processes the dataset and verifies the news content using linguistic features. The second step is to embed linguistic feature sets with WE and apply voting classification. To validate the effectiveness of their approach, a new WELFake dataset consisting of nearly many articles was selected, which contains different datasets to produce unbiased classification. As a result, the WELFake model demonstrated an accuracy of 96.73% in fake news detection.

Ivancova et al. [28] focused on detecting fake news from Slovak-language news articles based on Word2Vec, GloVe, and morphological analysis features. A dataset was created to train models on political news. Two architectures, CNNs and LSTM neural networks, were trained on the generated training data. The first model (Model 1) was a CNN, which achieved an overall accuracy of 92.38%. The second model (Model 2) was a recurrent neural network, in which an LSTM layer containing 128 neurons was fed by the output of the embedding layer. This model achieved an accuracy of 93.56% on the Slovak dataset.

Wang et al. [11] presented SemSeq4FD, a novel graph-based neural network model designed for the early detection of fake news using modified text structures. SemSeq4FD employs graphs to model the global semantic representations of sentences, and the global sentence representations are trained using a graph convolutional network. Sentence features were considered, using a one-dimensional convolutional network to train internal sentence classifiers using SLN and LUN data. For the optimized sentences, an LSTM-based network was used, producing the final document representation for fake news recognition using training data in both English and Chinese. An accuracy of 92.6% is achieved.

Subramanian et al. [29] detected fake content in Malayalam on social media platforms. The screening process consists of two subtasks: the first classifies the content as either fake or non-fake using contextual embedding and sequential features, while for the second subtask, the classification was expanded to five categories (false, half-true, mostly false, partially false, and mostly true) with the utilization of multilingual contextual embedding features. For the first task, machine learning methods such as SVM, naïve Bayes, and SGD, along with BERT-based algorithms, were used. Among these algorithms, XLM-RoBERTa achieved a high performance of 89.80%. For the second task, models using LSTM, GRU, XLM-RoBERTa, and SVM were used. XLM-RoBERTa again performed well over the other algorithms, achieving the highest overall F1 score of 62.83%.

Jingyuan et al. [30] focused on improving graph detection through significant improvements to language models, frameworks, and training models in the fake news literature. Building on several successful approaches, the potential for real-time cross-platform fake news detection will be highlighted. Context and Symantec features were used for misinformation detection knowledge integration, fake news detection with multimodal large language models, domain adaptive few-shot fake news detection, and a style-agnostic detection framework. All these models were built on graph neural networks (GNNs). Moreover, their experiment utilized the FakeNewsNet, PolitiFact fact, PAN2020, and COVID-19 datasets. Fake news detection using large multi-modal language models on the PolitiFact dataset yielded a high accuracy of 95.10%.

Tan and Bakir [31] presented a model based on the transformer algorithm, which has multiple uses for processing longer texts more reliably. A hybrid bidirectional long-term text processing unit with the transformer algorithm in the model was performed. To facilitate the identification of fake tweets (TruthSeeker), the researchers added a class-specific balancing factor to the dataset using word embedding. The TomekLinks algorithm was utilized for the purpose of enchanting prediction performance. In order to achieve this goal, a parameter set was considered, and grid search was performed to identify the parameters that yielded optimal results. As for the test results, the model achieved high performance, reaching 99.91% accuracy.

Alsuwat, E. and Alsuwat, H. [32] focused on a new proposal for fake news detection, termed Multi-Modal Fake News Detection (MM-FND). In their experiments, they relied on three datasets, namely the ISOT fake news dataset, the LIAR dataset, and the COVID-19 fake news dataset. For feature generation, they employed Word2Vec and term frequency–inverse document frequency (TF–IDF) to extract temporal features. Bi-LSTM was used to extract temporal features using bidirectional long short-term memory networks. Furthermore, spatial features were extracted using named entity recognition (NER) combined with global vector embeddings for word representation (GloVe). The results showed that the proposal achieved 96.3% accuracy with testing on the ISOT dataset. On the LIAR dataset, the algorithm achieved 95.6% accuracy. On the COVID-19 fake news dataset, the algorithm achieved an accuracy of 97.1%.

2.3. Machine Learning and Deep Learning

Jiang et al. [5] applied two approaches. First, five machine learning models were evaluated, and second, three deep learning models were tested. For evaluation, cross-validation was conducted using two fake news datasets of distinctly different sizes. In addition, term frequency–inverse document frequency (TF–IDF) features and word embeddings were extracted as inputs for the machine learning and deep learning models, respectively. They then proposed a stacking model, which, when tested on the ISOT and KDnugget datasets, achieved accuracies of nearly 99.95% and 96%, respectively.

Pardamean and Pardede [7] worked on identifying inaccurate news by using Bidirectional Encoding Representations from Transformers (BERT). BERT is a deep learning language model and is highly effective in language processing. Experiments have shown that the representations using hyperparameters features can achieve an accuracy of 99.23% by the Kaggle dataset.

Mouratidis et al. [33] conducted a comparative experiment on traditional machine learning classifiers including naïve Bayes, SVMs, and random forests, in addition to deep learning models, such as CNNs, LSTMs, and BERT. The study generated features including TF-IDF, Word2Vec, and contextual embeddings. Moreover, they conducted various tests based on multiple datasets. The researchers found that BERT-based models achieve strong performance, represented by an improvement in the accuracy of fake news detection. They achieved a performance of 98.40% when the BERT algorithm was applied.

Al-Tarawneh et al. [34] found that TF-IDF can potentially extract features exhibiting discrimination features from content. Furthermore, TF-IDF improves CNNs by effectively extracting local features and patterns within the content of text when the Truthseeker dataset is utilized, which contains news articles and social blogs labeled for this purpose. On the other hand, they demonstrated that Word2Vec and FastText embeddings did not perform well in capturing semantic and syntactic nuances, which is not always beneficial for traditional machine learning models, including multilayer perceptron (MLP) or SVMs. This study highlights the importance of carefully choosing the proper embedding techniques based on model algorithm to achieve strong predication performance on the fake news detection task. For TF-IDF embedding, CNN 1 and CNN 3 demonstrated a comparable performance, with an accuracy of 98.77% and 98.99%, respectively, demonstrating the necessity of using these two models for embedding.

Shen et al. [35] developed GAMED, a multi-media modeling algorithm that primarily generates distinct and distinctive features through media sorting to enhance interconnectedness, thus improving overall detection performance. Multiple parallel expert networks are leveraged to extract distinctive and discriminative features and incorporate semantic knowledge into GAMED. The feature distribution is then systematically adjusted. GAMED explains difficult decisions and performs a new classification to dynamically manage contributions from different media. Experimental results on the Fakeddit and Yang datasets show that GAMED performed better than state-of-the-art models, with an accuracy of 93.90%.

2.4. Optimization Techniques

Ozbay and Alatas [12] proposed a new approach to detecting fake news (FND) spread through social media. In this approach, the FND problem was formulated as an optimization problem, supported by the generation of features such as term frequency (TF) and document vectors. To address it, the authors proposed two metaheuristic algorithms, namely Grey Wolf Optimization (GWO) and Negative Swarm Optimization (SSO). The FND approach involves three stages, including data preprocessing, followed by adapting GWO and SSO to train a new FND model. The final stage is testing using the FND model. The results showed that the GWO algorithm has superior performance compared to SSO and other AI algorithms. In the evaluation process, they utilized a public fake news detection (FND) dataset, namely the LIAR benchmark, and achieved an accuracy of 96.5%.

Al-Ahmad et al. [15] presented a model that incorporates a feature selection process aimed at reducing redundancy among similar features, in addition to generating features using Bag of Words (BOW), term frequency (TF), and term frequency–inverse document frequency (TF-IDF). Furthermore, they employed metaheuristic algorithms for classification, namely Particle Swarm Optimization (PSO), genetic algorithms (GAs), and negative swarm algorithms (SSAs). To evaluate their approach, the generated models were tested on the Koirala dataset, achieving an accuracy of 75.4%.

3. Methodology

This section focuses on presenting a comprehensive discussion of the research methodology, where the research strategy, the purpose of the research, how data was collected and analyzed, quality standards, and ethical considerations of the research are discussed. In this research, qualitative research methods are used, based on the analysis of literature reviews extracted from various available research databases. Qualitative research is a research approach with a deep and interpretive focus on phenomena, relying on the context and complexity of the situations under study. In this research, the aim is not only to answer specific questions, but also to delve deeper into understanding the meanings, expectations, and experiences of the individuals or groups concerned. Qualitative methods often include data collection through observations or document analysis, which helps researchers and participants interact quickly with each other. Systematic literature reviews (SLRs) have been increasing in the field of management research. They focus on reviews between journals and researchers, as well as comprehensive searches of scientific databases for research data and application of inclusion/exclusion criteria, thus leading to theoretically and methodologically accurate results to build a reliable foundation for scholars and researchers.

In order to have comprehensive coverage of the relevant work, this review is conducted based on the guidelines provided by Kitchenhamy et al. [19], which contain several stages: "research questions", "search process", and compliance with PRISMA 2020 guidelines [36]. The flow diagram is presented in Figure 2, and the completed checklist is provided in the Supplementary Materials.

In this study, key results are presented through summary tables showing the characteristics and outcomes of included studies. Moreover, current challenges and future trends are highlighted based on the identification of research gaps.

3.1. Research Questions

This section outlines the research questions that defined the direction of this study:

- RQ1: What is the accuracy of the primary techniques employed to detect fake news?
- RQ2: What datasets are used?
- RQ3: Do gaps affect model performance?

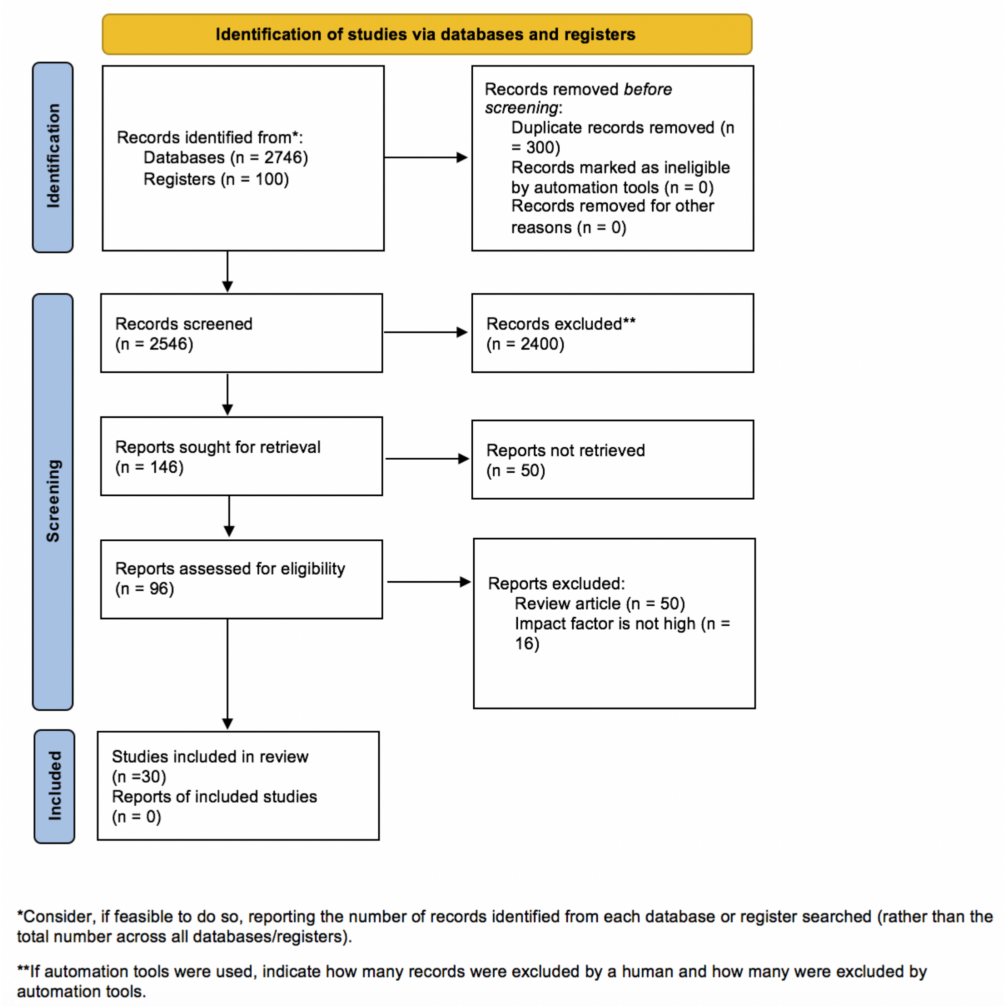

Figure 2. PRISMA flow diagram to include papers captured by this research.

3.2. Search Process

The search process was conducted by manually searching for the facts of research papers in scientific journals from 2018 to 2025. The search process used in this review can be further detailed as follows:

3.2.1. Sources and Data Collection

The search method includes articles in journals and conference proceedings published between 2018 and 2025. The search was not limited to a single publisher and included leading sources such as IEEE, Intelligent Systems, EMNLP, ACM, Springer, Elsevier, JAIR, AAAI, and ACL. Furthermore, we extended the search to research-oriented databases, including Scopus, Web of Science, DBLP, and Google Scholar, to ensure comprehensive coverage of the relevant literature. Thus, the citations of all chosen articles were reviewed to find out which papers were not cited as relevant.

3.2.2. Search Keywords

The keywords discussed in the research questions of this research study are as follows:

Fake news, detection, machine learning, algorithms, deep learning, accuracy, features, dataset.

3.2.3. Expression of Research

The procedure described was implemented to enable the search terms in this review. Keywords are extracted from the search questions related to detecting fake news. The search expressions are made up of a set of target words, sorted using the AND logical operator, and a set of terms and synonyms, using the OR logical operator [19].

3.2.4. Inclusion and Exclusion Standards

For articles published between 2018 and 2025, we focused on the following topics:

- Detecting fake news;
- Using machine learning to detect the fake news;
- Using deep learning to detect the fake news.

Articles in which the literature review was the only component and articles in which the literature review was the main conclusion of the article were not included in this review:

- It does not present the use of algorithms to detect fake news.
- No performance has been provided in identifying fake news.

3.2.5. Quality Valuation

Each literature review was evaluated for review and publication in the database. Therefore, the quality valuation questions were listed based on several standards, including

- QV1: Did the study demonstrate the use of machine learning and deep learning methods/algorithms together to detect fake news?
- QV2: Is the dataset used in the model sufficient to achieve high performance?
- QV3: Does the model demonstrate high performance?

Regarding the questions, they were divided as follows:

- QA1 as described in QV1: Y (yes)—the study demonstrated both machine learning and deep learning methods for detecting fake news. P (partially)—the study demonstrated either machine learning or deep learning methods. N (no)—the study did not demonstrate clear methods for detecting fake news.
- QA2 as described in QV2: Y (yes)—the dataset is sufficient. P (partially)—the dataset is partially sufficient. N (no)—the study did not state a clear dataset.
- QA3 as described in QV3: The study showed a high performance of greater than or equal to 98%, with an RMSE of less than or equal to 0.75 and an MAE of less than 0.5. P (partial)—the study showed a performance of less than 98% and greater than or equal to 95%, with an RMSE of greater than 0.75 and less than or equal to 1 and an MAE of greater than 0.5 and less than or equal to 0.75. LP (less than partial)—the study showed a performance of less than 95%, with an RMSE of greater than 1 and less than or equal to 2 and am MAE of greater than 0.75 and less than or equal to 1.5.

The process of evaluating each paper was as follows: Y = 1, P = 0.5, LP = 0.25, and N = 0. When there was a conflict, opinions were discussed until an appropriate evaluation of the paper was reached [19].

Figure 2 displays the PRISMA flow diagram of the study. Out of 2746 citations retrieved by the electronic search, we found 30 eligible documents. We eliminated a total of 66 full-text articles for the following reasons: 50 articles represented review articles, and the impact factor of 16 articles was not high. The importance of a journal is measured by the number of times its selected articles are cited within the years specified in this study. Consequently, a lower impact factor corresponds to a lower journal ranking, and this metric was therefore adopted in our analysis.

This research focused on gaps in previous studies and compared algorithms, features, and performance, as well as datasets and performance. This is in contrast to previous literature reviews that did not focus on these points. Therefore, this research helps researchers quickly leverage machine learning and deep learning techniques for detecting fake news.

4. Results and Discussion

In most of the research conducted on classification to predict whether the obtained news is fake or real, the following algorithms have been used, whether in machine learning, deep learning, or optimization techniques. Machine learning algorithms include logistic regression classification, decision tree classification, gradient boosting classification, random forest classification, k-nearest neighbor classification, and naïve Bayes algorithm. On the other hand, deep learning algorithms include CNN, RNN, BI-LSTM, and GNN [18].

4.1. Machine and Deep Learning Algorithms

4.1.1. Logistic Regression Classification Algorithm

Logistic regression is typically used in two-class classification problems. The primary goal of classification algorithms is to classify objects based on the probability of the presence of the dependent variable. The relationship between the sigmoid function and the coefficients in this algorithm plays a key role in approximating the dependent variable [18].

4.1.2. Decision Tree Algorithm

Decision trees are a commonly utilized algorithm in machine learning. The algorithm works effectively on both classification and regression problems, making it easy for users to understand and interpret. To build a model, predictions based on test data are used in the first stage to determine whether the data is true or false. The algorithm works by splitting the dataset in the first stage and building a classification model for each subset. The model's efficiency is carefully evaluated, and a classification report reveals the results [18].

4.1.3. Random Forest Classification Algorithm

The random forest classification algorithm is an ensemble learning technique that incorporates the properties of decision trees. The algorithm trains each tree separately, and the final model is obtained by averaging the predictions of these trees. This algorithm achieves a more reliable model by reducing the tendency of a single decision tree to overfit. The algorithm's success is carefully evaluated [18].

4.1.4. Boosting Classification Algorithm

The concept of the progressive boosting algorithm is based on ensemble learning, combining weak decision trees to generate more accurate decisions. This algorithm thus improves the model's success by using a sequential error reduction strategy. For classification and regression problems, the progressive boosting algorithm prefers decision trees. The model's efficiency is evaluated and presented as a classification report [18].

4.1.5. K-Nearest Neighbor (KNN) Algorithm

The K-nearest neighbor (KNN) algorithm is a machine learning algorithm utilized in classification and regression problems. KNN is a simple and highly efficient algorithm that achieves high performance, especially for small datasets. The model's success is efficiently evaluated, and a classification report is generated based on the results [18].

4.1.6. Naïve Bayes Classification Algorithm

The naïve Bayes classifier algorithm is based on the probability of an event occurring given information from another context. The "naïve" statement is assumed to be independent and unrelated to any other attribute. Therefore, the absence of any attribute does not affect the presence of others. Features are extracted by extracting text data and then converting it to a feature using the concept of "term frequency—inverse document frequency." Thus, features in text documents can be either word frequencies or TF-IDF values. When testing text data, the naïve Bayes model calculates the probability that the data falls into each class. The data is then classified into the class with the highest probability. The model's success is efficiently evaluated, and a classification report is printed accordingly [18].

4.1.7. Support Vector Machine (SVM) Algorithm

The SVM algorithm is widely used in machine learning problems for text and news classification and regression. It creates a hyperplane to separate each class in a given dataset. Thus, in a binary classification task, the SVM aims to find the highest hyperplane to separate the dataset into two classes. The success of the SVM in classifying data points belonging to a particular class is based on determining their distance from the hyperplane. The algorithm's success is evaluated efficiently, and a classification report is printed based on its efficiency [18].

4.1.8. Convolutional Neural Network (CNN) Algorithm

This model evaluates and clarifies the adjustment of neural networks recognized for their effectiveness in sentiment analysis. The strongest feature of this model is that it allocates the highest total amount of information derived from texts through various layers [17].

4.1.9. Recurrent Neural Network (RNN) Algorithm

RNNs are now widely used for identifying fake news. The aim of RNN models is for a constrained-size vector to represent text by assigning each token a recurrent vector, allowing it to embody the crucial sequential nature of language [17].

4.1.10. BI-Directional Long Short-Term Memory (BI-LSTM) Algorithm

BI-LSTM is an extension of LSTM that reads in two directions through the input sequence. This allows the model to perform a richer understanding of the data, especially in tasks like detecting fake news [17].

4.1.11. Graph Neural Network (GNN) Algorithm

GNN are neural network models capable of working with graph data structures. GNNs are derived from CNNs and graph embedding in node and edge prediction and graph-based tasks [30].

4.2. Features Extraction

4.2.1. Term Frequency (TF)

TF measures how often a term appears in a text. It is the ratio of the number of times a word appears in a text to the total number of words in the text. The rule is shown in the TF formula [37]:

$$\mathrm{TF} = \frac{\text{number of times the term appers in th text}}{\text{total number of terms in the text}} \tag{1}$$

4.2.2. Term Frequency–Inverse Document Frequency (TF-IDF)

Inverse document frequency (IDF) scales down words that appear a lot across the corpus or the text. The rule is shown in the IDF formula of a term t:

$$\mathrm{IDF}(\mathrm{t}) = \log(\frac{\mathrm{N}}{\mathrm{df}(\mathrm{t})}) \tag{2}$$

where N represents the total documents in a collection, and df signifies the count of documents containing term t. The TF-IDF score of a word in a document is the product of its TF and IDF scores [37]. The rule is shown in the TF-IDF formula:

$$\mathrm{TF} - \mathrm{IDF}(\mathrm{t},\ \mathrm{d}) = \mathrm{TF}(\mathrm{t},\ \mathrm{d}) * \mathrm{IDF}(\mathrm{t}) \tag{3}$$

where t stands for term, and d for document.

4.2.3. Word2Vec Embedding

Word2Vec is a widely used technique for embedding words from text. A full text is scanned, and the vector is generated by identifying words that frequently occur with the target word [38].

4.2.4. FastText

FastText is a compact library that enables users to acquire text representations and text classifiers for text [38].

4.3. Performance

The research examines the identification of fake news employing machine learning, deep learning, and optimization techniques. Do et al. [20] introduced a system for assessing the evaluation and datasets for all contributors. The overall accuracy (OA) can be represented by ratios. F-score (*F*1) and Accuracy (*A*%) can be represented by ratios, while Precision (*P*) and Recall (*R*) can be expressed through ratios from the confusion matrix entries, as shown in Figure 3 [17,39].

$$P = \frac{TP}{TP + FP} \tag{4}$$

$$R = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = \frac{2PR}{P + R} \tag{6}$$

$$A\% = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where *TP*: true positive; *TN*: true negative; *FP*: false positive; and *FN*: false negative.

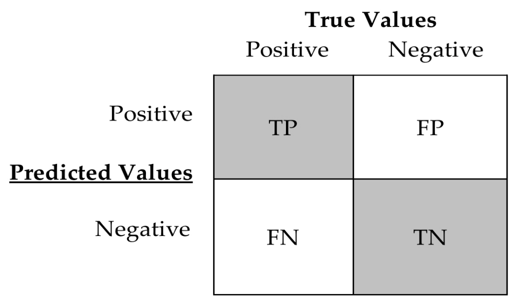

Figure 3. Confusion matrix.

Machine learning models may be evaluated using the mean absolute error (MAE) and root mean square error (RMSE) metrics to provide a clearer picture of their predictive performance. MAE measures the average absolute difference between the predicted and true values, giving an impression of the amount of error occurring on average without considering its direction. RMSE, on the other hand, provides a more accurate picture of the likelihood of significant errors because it squares difference between the predicted and true values, highlighting significant errors [24].

From Tables 1–4, it can be observed that deep learning algorithms achieve superior performance on average; however, some traditional machine learning algorithms outperform DL in detecting fake news in certain cases.

Table 1. Performance comparison based on the machine learning algorithms. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

	Category		**Machine Learning**	
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S1	• Neural network • Naïve Bayes • SVM	• Twitter API 48,373 messages natural phenomena	• Raw data	• Acc: 99.90% • Acc: 96.08% • Acc: 99.90%
S2	• Naïve Bayes	• Kaggle (Fake News Challenge)	• TF-IDF matrix • Count vectorizer • Hash vector • Aggressive hash	• Acc: 85.70% • Acc: 89.30% • Acc: 90.20% • Acc: 92.20%
S3	• Logistic regression • Random forests • SVM	• FakeNewsNet: 1056 data; PolitiFact • GossipCop: 16,817 real and 5323 fake stories	• Document frequency • Document frequency	• Acc: 68.00% • Acc: 67.00%
S8	• Ensemble learning	• Multi-class	• Content and Context level	• Acc: 86.00%
S13	• Random forest • Naïve Bayes • Neural Network • Decision Tree	• LIAR, 12,836 short statements	• Unigram • Bigram • Trigram	• Acc: 91.00% • Acc: 99.00% • Acc: 92.00% • Acc: 90.00%
S14	• Capsule neural network	• ISOT, 44,898 articles • LIAR, 12.8 K short statements	• n-gram • Word embedding	• Acc: 99.80%
S17	• SVM • KNN • Naïve Bayes • logistic regression • Random forest • AdaBoost • Decision tree	• Kaggle, 2050 news articles	• TF-IDF • Site_Url • Text-based	• Acc: 64.00% • Acc: 70.60% • Acc: 72.30 • Acc: 80.70 • Acc: 88.30 • Acc: 96.00% • Acc: 98.00%
S18	• Gaussian naïve Bayes • SVM • Logistic regression • Multilayer perception • Random forest • VotingEnsemble • AdaBoost • Gradient boosting • Multimodal Naïve Bayes	• Bengali news, 538 instances	• TF-IDF • Extra tree classifier	• Acc: 57.32% • Acc: 78.62% • Acc: 72.93% • Acc: 61.14% • Acc: 76.29% • Acc: 87.42% • Acc: 71.53% • Acc: 64.93% • Acc: 62.43%

Table 1. *Cont.*

Category		Machine Learning		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S20	• Random forest • Random forest • Decision tree • Decision tree • Logistic regression • Logistic regression • AdaBoost • AdaBoost	• Twitter, API, non-credible Arabic tweets	• Content-based • User-based • Content-based • User-based • Content-based • User-based • Content-based • User-based	• Acc: 68.00% • Acc: 76.00% • Acc: 70.00% • Acc: 69.00% • Acc: 76.00% • Acc: 75.00% • Acc: 74.00% • Acc: 74.00%
S21	• Logistic regression • Decision tree • Random forest	• Articles, fact-checking websites (politifact.com and snopes.com)	• Semantic	• Acc: 98.00% • Acc: 98.00% • Acc: 99.00%
S4	• NLP • Decision trees • Random forests • AdaBoost classification • XGBoost	• Kaggle	• TF	• Acc: High Acc.
S22	• Naïve Bayes • Logistic Regression	• ISOT, 23,481 news articles	• TF-IDF	• Acc: 94.37% • Acc: 98.31%
S29	• SVM • KNN	• Fake news detection, 30,100 data	• Categorical feature • Datetime feature	• MAE: 0.725 • RMSE: 01.628 • MAE: 0.011 • RMSE: 0.077

Table 2. Performance comparison based on the Deep Learning Algorithms. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		Deep Learning		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S12	• CNN • CNN • Recurrent LSTM	• Articles from Slovak websites, 2278 articles	• Word2Vec • GloVe	• Acc: 92.38% • Acc: 92.38%
S15	• Neural network SemSeq4FD • CNN • LSTM	• SLN English, 360 news articles • LUN English, 24 K news articles for training and 1.5 K news articles for testing • Weibo Chinese, 7300 news articles • RCED Chinese, 2955 news articles	• Sentence encoding • Sentence rep. • Document rep.	• Acc: 88.42% • Acc: 93.78% • Acc: 81.74% • Acc: 90.34%
S24	• XLM-RoBERTa • BiLSTM with XLM-RoBERTa	• Task 1: news, 5091 news articles • Task 2: Malayalam news, 2100 news articles	• Task 1: Contextual embeddings and Sequential • Task 2: Multilingual contextual embedding	• F1: 89.80% • F1: 62.83%

Table 2. *Cont.*

	Category	Deep Learning		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S25	• GNN	• FakeNewsNet • PolitiFact • PAN2020 • COVID-19	• Context features • Semantic features	• Acc: 95.20% • Acc: 95.10% • Acc: 87.30% • Acc: 99.90%
S30 DL	• Bidirectional LSTM	• ISOT news • LIAR • COVID-19 Fake News S30	• Word2Vec • TF-IDF • Temporal features	• Acc: 96.30% • Acc: 95.60% • Acc: 97.10%
S6	• CNN	• GPAC, 121,071 documents	• PreTra emb. dim = 300 • Embedding dim = 50 • Embedding dim = 100 • Embedding dim = 200 • Embedding dim = 300	• Acc: 98.83% • Acc: 97.15% • Acc: 98.90% • Acc: 99.21% • Acc: 99.36%
S9	• mBERT • XLM-RBase • XLM-RLarge • AraBERT • mBERT • XLM-RBase • XLM-RLarge • AraBERT	• ATB, 2000 news stories • ATB, 2000 news stories • ATB, 2000 news stories • ATB, 2000 news stories • AraNews, 5, 187, 957 • AraNews, 5, 187, 957 AraNews, 5, 187, 957 AraNews, 5, 187, 957	• Word embedding • Word embedding • word embedding • Word embedding • Word embedding • Word embedding • Word embedding • Word embedding	• Acc: 77.16% • Acc: 81.72% • Acc: 82.41% • Acc: 83.19% • Acc: 79.39% • Acc: 82.77% • Acc: 82.12% • Acc: 87.21%
S10	• LSTM • LSTM • GRU • GRU • CNN • CNN • BI-LSTM • BI-LSTM	• Fakeddit	• Textual content • Text, titles, and comm. • Textual content • Text, titles, and comm. • Textual content • Text, titles, and comm. • Textual content • Text, titles, and comm.	• Acc: 89.99% • Acc: 90.16% • Acc: 91.65% • Acc: 92.60% • Acc: 94.14% • Acc: 96.05% • Acc: 94.65% • Acc: 96.77%
S11	• CNN • BERT • WELFake	• WELFake, 72,134 articles	• Linguistic • Word embedding	• Acc: 92.48% • Acc: 93.79% • Acc: 96.73%
S28 DL	• Bidirectional LSTM	• TruthSeeker, 134,198 tweets	• Word embedding	• Acc: 99.91%

Tables 5–8 demonstrate that datasets such as LIAR and ISOT, which contain a larger volume of news articles, in both training and testing datasets, yielded higher accuracy in fake news detection. A complete list of all studies and their results in ascending order (S1–S30) is provided in Appendix A, Tables A1–A3.

4.4. Current Challenges and Future Perspectives

This study helps raise awareness about the spread of fake news. The main goal of detecting fake news is to maintain the credibility of news in general. Previous studies have used machine learning, deep learning techniques, and optimization techniques to develop models that enhance the identification of misleading news. However, various challenges and gaps remain in each study. The most notable of these gaps are the following:

A major gap identified in various studies (S1 [1], S2 [2], S5 [5], S12 [28], S18 [14], S21 [22], S29 [24], and S30 [32]) concerns the applicability of the results to real news due to the limited data used for training. Therefore, it is important to expand the scope of data collection and attempt to apply the algorithm more widely in the future, as explained in the research. Therefore, in machine learning problems, obtaining sufficient data often significantly improves the algorithm's efficiency. The model in study S29 does not include different social media datasets for fake news detection [24]. Therefore, this model lacks a large dataset.

Table 3. Performance Comparison based on the Both Machine Learning and Deep Learning Algorithms. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		Both ML and DL		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S26	SVM	Truthseeker, 180,000 tweets	TF-IDF	Acc: 99.03%
	Multilayer perceptron		TF-IDF	Acc: 98.77%
	Logistic regression		TF-IDF	Acc: 97.58%
	Random forest		TF-IDF	Acc: 98.39%
	Decision tree		TF-IDF	Acc: 97.30%
	SVM		Word2Vec	Acc: 94.47%
	Multilayer perceptron		Word2Vec	Acc: 95.24%
	Logistic regression		Word2Vec	Acc: 85.42%
	Random forest		Word2Vec	Acc: 91.01%
	Decision tree		Word2Vec	Acc: 80.30%
	KNN		Word2Vec	Acc: 94.98%
	SVM		FastText	Acc: 90.41%
	Multilayer perceptron		FastText	Acc: 93.21%
	Logistic regression		FastText	Acc: 83.44%
	Random forest		FastText	Acc: 84.53%
	Decision tree		FastText	Acc: 72.42%
	KNN		FastText	Acc: 85.10%
	CNN Model 1		TF-IDF	Acc: 98.77%
	CNN Model 2		TF-IDF	Acc: 56.15
	CNN Model 3		TF-IDF	Acc: 98.99%
	CNN Model 1		Word2Vec	Acc: 94.25%
	CNN Model 2		Word2Vec	Acc: 90.73%
	CNN Model 3		Word2Vec	Acc: 94.92%
	CNN Model 1		FastText	Acc: 89.32%
	CNN Model 2		FastText	Acc: 85.26%
	CNN Model 3		FastText	Acc: 89.55%
S27	GAMED for multimodal modeling	Fakeddit, one million labeled Yang, 20,015 news articles	Distinctive features Discriminative features	Acc: 93.90%
S5	Logistic regression	ISOT, 44,894 data	TF-IDF	Acc: 99.63%
	SVM	ISOT, 44,894 data	TF-IDF	Acc: 99.63%
	K-NN	ISOT, 44,894 data	TF-IDF	Acc: 68.65%
	Decision tree	ISOT, 44,894 data	TF-IDF	Acc: 99.60%
	Random forest	ISOT, 44,894 data	TF-IDF	Acc: 99.87%
	Random forest	ISOT, 44,894 data	TF	Acc: 99.84%
	CNN	ISOT, 44,894 data	Embedding	Acc: 99.52%
	GRU	ISOT, 44,894 data	Embedding	Acc: 99.69%
	LSTM	ISOT, 44,894 data	Embedding	Acc: 99.74%
	Logistic regression	KDnugget, 6335 news articles	TF-IDF	Acc: 92.82%
	SVM	KDnugget, 6335 news articles	TF-IDF	Acc: 92.42%
	K-NN	KDnugget, 6335 news articles	TF-IDF	Acc: 82.56%
	Decision tree	KDnugget, 6335 news articles	TF-IDF	Acc: 79.87%
	Random forest	KDnugget, 6335 news articles	TF-IDF	Acc: 91.63%
	Random forest	KDnugget, 6335 news articles	TF	Acc: 91.48%
	CNN	KDnugget, 6335 news articles	Embedding	Acc: 89.50%
	GRU	KDnugget, 6335 news articles	Embedding	Acc: 91.32%
	LSTM	KDnugget, 6335 news articles	Embedding	Acc: 88.95%

Table 3. *Cont.*

Category		Both ML and DL		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S7	• BERT fine-tuning • Naïve Bayes SVM	• Kaggle 28, 711 news articles	• Hyperparameter Settings • Hyperparameter Settings	• Acc: 99.23% • Acc: 95.00%
S23	• Naïve Bayes • SVM • Random forest • BERT • CNN • LSTM	• Geroge McIntyre • UTK ML Kaggle • ISOT fake news • UTK ML Kaggle • Signalmedia • ISOT fake news	• TF-IDF • Word2Vec • Contextual embeddings	• Auc: 97.50% • Auc: 97.60% • Auc: 96.30% • Auc: 98.40% • Auc: 97.30% • Auc: 97.60%

Table 4. Performance comparison based on optimization techniques. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		Optimization Techniques		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S16	• SSO • GWO • Decision tree • Naïve Bayes • SVM, random • Gradient boost • Ridor • J48 • SMO • SSO • GWO • Decision tree • Naïve Bayes • SVM • Gradient boost • Ridor • J48 • SMO • SSO • GWO • Decision tree • Naïve Bayes • SVM • Gradient boost • Ridor • J48 • SMO	• Random political news • Random political news • Random political news • Random political news • Random political news • Random political news • Random political news • Random political news • Random political news • Random political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Buzzfeed political news • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements • Liar, 12,836 short statements	• TF • Document vector	• Acc: 71.30% • Acc: 92.60% • Acc: 63.40% • Acc: 76.20% • Acc: 70.00% • Acc: 71.70% • Acc: 64.20% • Acc: 65.40% • Acc: 68.00% • Acc: 80.30% • Acc: 87.50% • Acc: 63.40% • Acc: 69.60% • Acc: 59.00% • Acc: 62.10% • Acc: 56.20% • Acc: 65.50% • Acc: 61.90% • Acc: 78.00% • Acc: 96.50% • Acc: 79.80% • Acc: 72.60% • Acc: 83.60% • Acc: 79.80% • Acc: 82.00% • Acc: 82.20% • Acc: 82.30%

Table 4. *Cont.*

Category		Optimization Techniques		
Study	**Algorithms/Methods**	**Dataset**	**Features/Attributes**	**Performance**
S19	• KNN-BSSA • KNN-BPSO • KNN-BGA. • KNN • KNN-BSSA • KNN-BPSO • KNN-BGA • KNN • KNN-BSSA • KNN-BPSO • KNN-BGA • KNN	• Koirala, 6000 articles	• BOW • BOW • BOW • BOW • TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF • TF • TF • TF	• Acc: 72.64% • Acc: 72.58% • Acc: 73.48% • Acc: 70.53% • Acc: 61.61% • Acc: 66.39% • Acc: 67.64% • Acc: 70.53% • Acc: 73.32% • Acc: 73.48% • Acc: 73.84% • Acc: 70.53%

Table 5. Performance comparison based on the Twitter/X API dataset. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		Twitter/X API		
Study	**Dataset**	**Algorithms/Methods**	**Features/Attributes**	**Performance**
S1	• Twitter API • 948,373 messages • Natural phenomena	• Neural network • Naïve Bayes • SVM	• Raw data	• Acc: 99.90% • Acc: 96.08% • Acc: 99.90%
S20	• Twitter API, non-credible Arabic tweets	• Random forest • Random forest • Decision tree • Decision tree • Logistic regression • Logistic regression • AdaBoost • AdaBoost	• Content-based • User-based • Content-based • User-based • Content-based • User-based • Content-based • User-based	• Acc: 68.00% • Acc: 76.00% • Acc: 70.00% • Acc: 69.00% • Acc: 76.00% • Acc: 75.00% • Acc: 74.00% • Acc: 74.00%
S28	• TruthSeeker, 134,198 tweets	• Bidirectional LSTM	• Word embedding	• Acc: 99.91%
S26	• Truthseeker, 180,000 tweets	• SVM • Multilayer perceptron • Logistic regression • Random forest • Decision tree • SVM • Multilayer perceptron • Logistic regression • Random forest • Decision tree • KNN • SVM • Multilayer perceptron • Logistic regression • Random forest • Decision tree • KNN • CNN Model 1 • CNN Model 2 • CNN Model 3 • CNN Model 1 • CNN Model 2 • CNN Model 3 • CNN Model 1 • CNN Model 2 • CNN Model 3	• TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF-IDF • Word2Vec • Word2Vec • Word2Vec • Word2Vec • Word2Vec • Word2Vec • FastText • FastText • FastText • FastText • FastText • FastText • TF-IDF • TF-IDF • TF-IDF • Word2Vec • Word2Vec • Word2Vec • FastText • FastText • FastText	• Acc: 99.03% • Acc: 98.77% • Acc: 97.58% • Acc: 98.39% • Acc: 97.30% • Acc: 94.47% • Acc: 95.24% • Acc: 85.42% • Acc: 91.01% • Acc: 80.30% • Acc: 94.98% • Acc: 90.41% • Acc: 93.21% • Acc: 83.44% • Acc: 84.53% • Acc: 72.42% • Acc: 85.10% • Acc: 98.77% • Acc: 56.15 • Acc: 98.99% • Acc: 94.25% • Acc: 90.73% • Acc: 94.92% • Acc: 89.32% • Acc: 85.26% • Acc: 89.55%

Table 6. Performance comparison based on Kaggle dataset. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		**Kaggle**		
Study	**Dataset**	**Algorithms/Methods**	**Features/Attributes**	**Performance**
S17	• Kaggle, 2050 news articles	• SVM • KNN • Naïve Bayes • Logistic regression • Random forest • AdaBoost • Decision tree	• TF-IDF • Site_Url • Text-based	• Acc: 64.00% • Acc: 70.60% • Acc: 72.30 • Acc: 80.70 • Acc: 88.30 • Acc: 96.00% • Acc: 98.00%
S2	• Kaggle (Fake News Chall.)	• Naïve Bayes	• TF-IDF matrix • Count vectorizer • Hash vector • Aggressive hash	• Acc: 85.70% • Acc: 89.30% • Acc: 90.20% • Acc: 92.20%
S7	• Kaggle 28, 711 news articles	• BERT fine-tuning • Naïve Bayes SVM	• Hyperparameter settings • Hyperparameter settings	• Acc: 99.23% • Acc: 95.00%
S23	• Geroge McIntyre • UTK ML Kaggle • ISOT fake news • UTK ML Kaggle • Signalmedia • ISOT fake news	• Naïve Bayes • SVM • Random forest • BERT • CNN • LSTM	• TF-IDF • Word2Vec • Contextual embeddings	• Auc: 97.50% • Auc: 97.60% • Auc: 96.30% • Auc: 98.40% • Auc: 97.30% • Auc: 97.60%
S4	• Kaggle website	• NLP • Decision trees • Random forests • AdaBoost classification • XGBoost	• TF	• Acc: High Acc.
S3	• FakeNewsNet: 1,056 data • PolitiFact and GossipCop: 16,817 real stories and 5323 fake stories	• Logistic regression • Random forests • SVM	• Document frequency • Document Frequency	• Acc: 68.00% • Acc: 67.00%
S29	• FakeNewsDetection, 30,100 data	• SVM • KNN	• Categorical feature • Datetime feature	• MAE: 0.725 • RMSE: 01.628 • MAE: 0.011 • RMSE: 0.077
S25	• FakeNewsNet • PolitiFact • PAN2020 • COVID-19	• GNN	• Context features • Semantic features	• Acc: 95.20% • Acc: 95.10% • Acc: 87.30% • Acc: 99.90%

Table 7. Performance comparison based on the LIAR and ISOT datasets. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		LIAR and ISOT		
Study	**Dataset**	**Algorithms/Methods**	**Features/Attributes**	**Performance**
S13 ML	• LIAR, 12,836 short statements	• Random forest • Naïve Bayes on • Neural network • Decision tree	• Unigram • Bigram • Trigram	• Acc: 91.00% • Acc: 99.00% • Acc: 92.00% • Acc: 90.00%
S14	• ISOT, 44,898 articles. • LIAR, 12.8 K short statements	• Capsule neural network	• n-gram • Word embedding	• Acc: 99.80%
S22	• ISOT, 23,481 news articles	• Naïve Bayes • Logistic regression	• TF-IDF	• Acc: 94.37% • Acc: 98.31%
S30	• ISOT News • LIAR • COVID-19 Fake News	• Bidirectional LSTM	• Word2Vec • TF-IDF • Temporal features	• Acc: 98.00% • Acc: 98.00% • Acc: 99.00%
S5	• ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • ISOT, 44,894 data • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles • KDnugget, 6335 news articles	• Logistic regression • SVM • K-NN • Decision tree • Random forest • Random forest • CNN • GRU • LSTM • Logistic regression • SVM • K-NN • Decision tree • Random forest • Random forest • CNN • GRU • LSTM	• TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF • Embedding • Embedding • Embedding • TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF • Embedding • Embedding • Embedding	• Acc: 99.63% • Acc: 99.63% • Acc: 68.65% • Acc: 99.60% • Acc: 99.87% • Acc: 99.84% • Acc: 99.52% • Acc: 99.69% • Acc: 99.74% • Acc: 92.82% • Acc: 92.42% • Acc: 82.56% • Acc: 79.87% • Acc: 91.63% • Acc: 91.48% • Acc: 89.50% • Acc: 91.32% • Acc: 88.95%
S16	• BuzzFeed political news • Random political news • Liar, 12,836 short statements	• SSO • GWO • Decision Tree • Naïve Bayes • SVM • Gradient boost • Ridor • J48 • SMO • SSO • GWO • Decision tree • Naïve Bayes • SVM • Gradient boost • Ridor • J48 • SMO • SSO • GWO • Decision tree • Naïve Bayes • SVM • Gradient boost • Ridor • J48 • SMO	• TF • Document vector	• Acc: 71.30% • Acc: 92.60% • Acc: 63.40% • Acc: 76.20% • Acc: 70.00% • Acc: 71.70% • Acc: 64.20% • Acc: 65.40% • Acc: 68.00% • Acc: 80.30% • Acc: 87.50% • Acc: 63.40% • Acc: 69.60% • Acc: 59.00% • Acc: 62.10% • Acc: 56.20% • Acc: 65.50% • Acc: 61.90% • Acc: 78.00% • Acc: 96.50% • Acc: 79.80% • Acc: 72.60% • Acc: 83.60% • Acc: 79.80% • Acc: 82.00% • Acc: 82.20% • Acc: 82.30%

Table 8. Performance comparison based on the different datasets. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Category		Different Datasets		
Study	**Dataset**	**Algorithms/Methods**	**Features/Attributes**	**Performance**
S18	• Bengali news, 538 instances	• Gaussian Naïve Bayes • SVM • Logistic regression • Multilayer perception • Random forest • VotingEnsemble • AdaBoost • Gradient boosting • Multimodal naïve Bayes	• TF-IDF • Extra tree classifier	• Acc: 57.32% • Acc: 78.62% • Acc: 72.93% • Acc: 61.14% • Acc: 76.29% • Acc: 87.42% • Acc: 71.53% • Acc: 64.93% • Acc: 62.43%
S19	• Koirala, 6000 articles	• KNN-BSSA • KNN-BPSO • KNN-BGA. • KNN • KNN-BSSA • KNN-BPSO • KNN-BGA • KNN • KNN-BSSA • KNN-BPSO • KNN-BGA • KNN	• BOW • BOW • BOW • BOW • TF-IDF • TF-IDF • TF-IDF • TF-IDF • TF • TF • TF • TF	• Acc: 72.64% • Acc: 72.58% • Acc: 73.48% • Acc: 70.53% • Acc: 61.61% • Acc: 66.39% • Acc: 67.64% • Acc: 70.53% • Acc: 73.32% • Acc: 73.48% • Acc: 73.84% • Acc: 70.53%
	• Articles, fact-checking websites like politifact.com and snopes.com	• Logistic regression • Decision tree • Random forest	• Semantic	• Acc: 96.30% • Acc: 95.60% • Acc: 97.10%
S12	• Articles from Slovak websites, 2278 articles	• CNN • LSTM	• Word2Vec • GloVe	• Acc: 92.38% • Acc: 92.38%
S15	• SLN English, 360 news articles • LUN English, 24 K news articles for training and 1.5 K news articles for testing • Weibo Chinese, 7300 news articles • RCED Chinese, 2955 news articles	• Neural network SemSeq4FD • CNN • LSTM	• Sentence encoding • Sentence rep. • Document rep.	• Acc: 88.42% • Acc: 93.78% • Acc: 81.74% • Acc: 90.34%
S24	• Task 1: news, 5091 news articles • Task 2: Malayalam news, 2100 news articles	• XLM-RoBERTa • BiLSTM with XLM-RoBERTa	• Task 1: Contextual embeddings and sequential models • Task 2: Multilingual contextual embedding	• F1: 89.80% • F1: 62.83%

Table 8. *Cont.*

	Category	Different Datasets		
Study	**Dataset**	**Algorithms/Methods**	**Features/Attributes**	**Performance**
S6	• GPAC, 121,071 documents • GPAC, 121,071 documents • GPAC, 121,071 documents • GPAC, 121,071 documents • GPAC, 121,071 documents	• CNN	• PreTra emb. dim = 300 • Embedding dim = 50 • Embedding dim = 100 • Embedding dim = 200 • Embedding dim = 300	• Acc: 98.83% • Acc: 97.15% • Acc: 98.90% • Acc: 99.21% • Acc: 99.36%
S8	• Multi-class	• Ensemble learning	• Content and context level	• Acc: 86.00%
S9	• ATB, 2000 news stories • ATB, 2000 news stories • ATB, 2000 news stories • ATB, 2000 news stories • AraNews, 5, 187, 957 • AraNews, 5, 187, 957 • AraNews, 5, 187, 957 • AraNews, 5, 187, 957	• mBERT • XLM-RBase • XLM-RLarge • AraBERT • mBERT • XLM-RBase • XLM-RLarge • AraBERT	• Word embedding • Word embedding • Word embedding • Word embedding • Word embedding • Word embedding • Word embedding • Word embedding	• Acc: 77.16% • Acc: 81.72% • Acc: 82.41% • Acc: 83.19% • Acc: 79.39% • Acc: 82.77% • Acc: 82.12% • Acc: 87.21%
S10	• Fakeddit	• LSTM • LSTM • GRU • GRU • CNN • CNN • BI-LSTM • BI-LSTM	• Textual content • Text, titles, and comm. • Textual content • Text, titles, and comm. • Textual content • Text, titles, and comm. • Textual content • Text, titles, and comm.	• Acc: 89.99% • Acc: 90.16% • Acc: 91.65% • Acc: 92.60% • Acc: 94.14% • Acc: 96.05% • Acc: 94.65% • Acc: 96.77%
S11	• WELFake, 72,134 articles	• CNN • BERT • WELFake	• Linguistic • Word embedding	• Acc: 92.48% • Acc: 93.79% • Acc: 96.73%
S21	• Logistic regression • Decision tree • Random forest	• Articles, fact-checking websites (politifact.com and snopes.com)	• Semantic	• Acc: 98.00% • Acc: 98.00% • Acc: 99.00%
S27	• Fakeddit, one million labeled • Yang, 20,015 news articles	• GAMED for multimodal modeling	• Distinctive features • Discriminative features	• Acc: 93.90%

Furthermore, the issue of datasets is not limited to their size but rather expands to the importance of the proper selection of datasets and their category set, based on the gap identified in S13 [9]. Therefore, building the model requires several fine-tuning operations on different datasets during testing to obtain high accuracy in the results, and then relying on those results in future studies [9].

Another important consideration on datasets was identified by the gap in study S10, which lies in the difficulty of dealing with an imbalanced dataset with an uneven representation of categories, where one or more categories contain fewer examples than others [26].

As for the studies S19 [15] and S20 [16], they lack the ability to leverage Twitter responses to improve overall accuracy. To close this gap in research, achieving high performance requires larger datasets.

A shortcoming was found in study S25 [30], in which the current models were unable to adapt to the dynamic trends of social media due to the lack of features described in this research. Consequently, some models may provide inaccurate information and are difficult to scale to include all types of fake news.

A research challenge in study S24 [29] concerns the need to improve the model's natural language processing (NLP) capabilities by adding features to enhance accuracy. The gap in the aforementioned studies [14–16,29] highlights the importance of expanding the feature extraction and generation process during the formation of datasets [14]. Similarly, study S3 [3] observed that the PSM model only considers biases resulting from observed variables and does not consider unobserved variables.

One of the challenges in study S4 [4] is that when using the AdaBoost algorithm, the number of iterations is excessively large, and, therefore, the model overfits the training data [4].

A limitation observed in study S6 [6] is the absence of a word embedding algorithm; this gap could be addressed by using other word embedding algorithms, such as BERT (Bidirectional Encoder Representations from Transformers), which may help train word embeddings better than AMFTWE. However, BERT requires a large amount of data. However, creating a dataset of Amharic fake news and providing its transcripts will be a significant challenge. As for the gap found in study S26 [34], word embedding was not sufficiently considered, so the choice of word embedding technique significantly impacts the model's accuracy in detecting fake news.

One of the gaps in the S15 study is the need to extract most of the text structure information. Similarly, text modeling methods require further improvements in their accuracy to achieve the desired results and enable their application in other applications [11].

One of the challenges in study S27 [35] is that the model did not include all fake news from media outlets, such as audio or video, to obtain a systematic and comprehensive analysis.

One limitation observed in the S7 study is that BERT is a highly computational model and takes longer to train, so there is a need to reduce its computational load [7].

Various studies S8 [8], S14 [10], S17 [13], and S22 [23] suffered from not achieving high accuracy performance of classifying fake news into multiple categories, and the chosen models did not achieve high efficiency. Therefore, further training is needed [8]. Also, there was a loss of accuracy in the location and pose of objects in an image when the image was not fully classified. Location and pose were classified based on the content of the image and the perspective from which it was captured [10].

The gap in study S9 is that the model was limited to only one language and faced a significant challenge in text processing during training. Therefore, it must be applied to languages other than Arabic. The model also faced difficulties in text processing [25].

One of the limitations in study S11 is that the WELFake model did not address knowledge graph factors, such as the number of labels [27].

Most supervised learning algorithms applied to fake news detection are black-box approaches, as observed in S16 study [12], which does not facilitate the interpretation of the key factors contributing to the model's predictions.

One of the challenges in study S23 involves the limited use of machine learning algorithms, which negatively impacted the model's performance. Therefore, it is necessary to add more labels and leverage transfer learning techniques [33].

Based on the limitations in study S28 [31], it requires a more comprehensive study to enhance its ability to counter fake news on social media.

For future directions, this review has analyzed and thoroughly explained the previous literature. It demonstrates that fake news detection algorithms using machine learning and deep learning require large datasets to obtain highly accurate results. Therefore, there is significant scope for further research in this area.

A key recommendation is to expand the feature extraction and feature generation process to capture features that might assist and provide potential clues to fake news prediction process. For example, in the case of analyzing Twitter/X tweets, the incorporation of responses and related features can improve fake news detection.

The combination of sufficient data, effective feature extraction and generation, and appropriate machine learning techniques is a major contributing factor to fake news detection. An essential future direction is the development of interpretable prediction models, which can enhance understanding of the significance of the features selected or generated in the detection process. Few studies have addressed the purpose of ambiguous information, while extensive studies have used explicit information as a criterion for assessing fake news. One approach involves carefully selecting features and adding a large dataset. Table 9 presents the results obtained by displaying the gaps for each study.

Table 9. The gaps for each study.

Study	Gap
S1	There is a gap in the applicability of this study's findings to real-life news. Therefore, it is important to expand the range of data gathering and attempt to apply the algorithm more broadly in the future, as explained in the research.
S2	For this research, the gap can be bridged by using more data for training. Therefore, in machine learning problems, obtaining more data often significantly improves the efficiency of the algorithm.
S3	The gap in this research, which was mentioned by the researchers, is that the PSM model only considers biases resulting from observed variables and does not consider unobserved variables.
S4	The gap in this research is that when the AdaBoost algorithm is used, the number of iterations is too big, so the model will overfit the training data.
S5	The gap in this research is that decision trees, support vector machines, logistic regression, RNN, GRU, and LSTM had poor performance on small data.
S6	To fill this research gap, utilizing alternative word embedding algorithms, such as BERT (Bidirectional Encoder Representations from Transformers), may help train word embeddings better than AMFTWE, but BERT requires a large amount of data. However, creating an Amharic fake news dataset and providing transcripts will be a significant challenge.
S7	The gap in this research is that BERT is a very computational model, so there is a need to reduce the computational load of BERT.
S8	A gap in this research is that the accuracy of classifying fake news into multiple classes is not high, reaching 86%. Therefore, more training is needed.
S9	The gap in this research is that the model is only applicable to one language and needs to be applied to languages other than Arabic. The model also has difficulty processing texts.
S10	The gap in this research is the difficulty in dealing with an unbalanced dataset.
S11	As for the gap in this research, the WELFake model does not deal with the factors of knowledge graphs.
S12	To address the gap in the paper, the model needs to be improved by expanding and collecting more datasets. Therefore, researchers need to create more datasets based on specific topics.
S13	The gap in this research is the dataset and class set. The model built requires a number of fine-tuning operations on different datasets during testing.
S14	The gap in this research is the loss of accuracy in the location and pose of objects in the image when the image is not fully classified.
S15	A gap in research is that most text structure information needs to be extracted. Similarly, text modeling methods require further improvements in their accuracy.

Table 9. *Cont.*

Study	Gap
S16	The gap in this research is that most of the supervised algorithms applied in fake news detection are black-box approaches.
S17	The gap in this research is that to increase accuracy, other deep learning techniques must be used, with a focus on expanding datasets that include more articles.
S18	The gap in this research is to increase the dataset to extract more features.
S19	To address the gap in research, achieving high performance requires larger datasets.
S20	The gap in this research is that the model needs to leverage Twitter responses to enhance the overall precision of the model.
S21	The gap in this research is that the model needs to increase the number of datasets to enhance the accuracy of the model.
S22	The gap in this research is that the model must contain complex correlation management to increase the accuracy of the model.
S23	The gap in this research involves further improving the model, in terms of adding its labels and making use of transfer learning techniques.
S24	For the gap in this research, the model needs improvement in NLP to enhance the accuracy.
S25	Regarding the gap in this research, current models cannot adapt to the dynamic trends of social media. Some models may provide inaccurate information, and they are difficult to scale to include all types of fake news.
S26	Regarding the gap in this research, the decision of word embedding technique significantly affects the model's accuracy in detecting fake news.
S27	In identifying the gap in this research, this model does not include all fake news from media such as audio or video to obtain a systematic and comprehensive analysis.
S28	Regarding the gap in this research, it needs a more comprehensive study to strengthen its resilience to fake news in social media.
S29	A gap in this research is that the model does not include a deep learning algorithm using different social media dataset to detect fake news.
S30	For the gap in this research, more datasets need to be added.

Table 10 presents the bibliometric assessment regarding authors' names, author institutions, author countries, citations and accessibility.

Table 10. Bibliometric analysis in terms of author.

Study	Names	Institutions	Country	Citation Access
S1	Supanya Aphiwongsophon	Chulalongkorn Uni.	Thailand	220—Open Access
	Prabhas Chongstitvatana	Chulalongkorn Uni.	Thailand	
S2	I.M.V.Krishna	PVP Siddhartha Ins.	India	N/A—Open Access
	Dr. S.Sai Kumar	PVP Siddhartha Ins.	India	
S3	Bo Ni	Uni. of Notre Dame	USA	17—Open Access
	Zhichun Guo	Uni. of Notre Dame	USA	
	Jianing Li	Uni. of Notre Dame	USA	
	Meng Jiang	Uni. of Notre Dame	USA	
S4	Devanshi Singh	Delhi Tech. Uni.	India	238—Closed Access
	Ahmad Habib Khan	Delhi Tech. Uni.	India	
	Shweta Meena	Delhi Tech. Uni.	India	

Table 10. *Cont.*

Study	Names	Institutions	Country	Citation Access
S5	TAO JIANG	UESTC	China	210—Open Access
	JIAN PING LI	UESTC	China	
	AMIN UL HAQ	UESTC	China	
	ABDUS SABOOR	UESTC	China	
	AMJAD ALI	University of Swat	Pakistan	
S6	Fantahun Gereme	UESTC	China	52—Open Access
	William Zhu	UESTC	China	
	Tewodros Ayall	UESTC	China	
	Dagmawi Alemu	UESTC	China	
S7	Amsal Pardamean Hilman F. Pardede	STMIK Nusa Mand. Indo. Ins. of Sci.	Indonesia Indonesia	12—Open Access
S8	Rohit Kumar Kaliyar	Bennett University	India	75—Closed Access
	Anurag Goswami	Bennett University	India	
	Pratik Narang	Bits Pilani	India	
S9	El Moatez Billah Nagoudi	The Uni. of Brit. Colu.	Canada	66—Open Access
	AbdelRahim Elmadany	The Uni. of Brit. Colu.	Canada	
	Muhammad Abdul-Mageed	The Uni. of Brit. Colu.	Canada	
	Tariq Alhindi	Columbia University	Canada	
	Hasan Cavusoglu	The Uni. of Brit. Colu.	Canada	
S10	Suhaib Kh. Hamed	UKM	Malaysia	66—Open Access
	Mohd Juzaiddin Ab Aziz	UKM	Malaysia	
	Mohd Ridzwan Yaakub	UKM	Malaysia	
S11	Pawan Kumar Verma	GLA University	India	270—Open Access
	Prateek Agrawal	Lovely Prof. Uni.	Austia	
	Ivone Amorim	University of Porto	Portugal	
	Radu Prodan	Uni. of Klagenfurt	Austia	
S12	Klaudia Ivancová	Tech. Uni. Kosˇice	Slovakia	24—Close Access
	Martin Sarnovský	Tech. Uni. Kosˇice	Slovakia	
	Viera Maslej-Kresˇnˇáková	Tech. Uni. Kosˇice	Slovakia	
S13	Abdulaziz Albahr	KSAUHS	Saudi Arabia	47—Closed Access
	Marwan Albahar	Umm Al Qura Uni.	Saudi Arabia	
S14	Mohammad Hadi Goldani	Amirkabir Uni. of Tec.	Iran	170—Closed Access
	Saeedeh Momtazi	Amirkabir Uni. of Tec.	Iran	
	Reza Safabakhsh	Amirkabir Uni. of Tec.	Iran	
S15	Yuhang Wang	Taiyuan Uni. of Tec.	China	73—Closed Access
	Li Wang	Taiyuan Uni. of Tec.	China	
	Yanjie Yang	Taiyuan Uni. of Tec.	China	
	Tao Lian	Taiyuan Uni. of Tec.	China	
S16	Feyza Altunbey Ozbay	Firat University	Turkey	73—Closed Access
	Bilal Alatas	Firat University	Turkey	
S17	S. Selva Birunda	Kalasalingam ARE	India	31—Closed Access
	Dr. R. Kanniga Devi	Kalasalingam ARE	India	

Table 10. *Cont.*

Study	Names	Institutions	Country	Citation Access
S18	Shafayat Shabbir Mugdha	United Intern. Uni.	Bangladesh	47—Closed Access
	Saayeda Muntaha Ferdous	United Intern. Uni.		
	Ahmed Fahmin	United Intern. Uni.		
S19	Bilal Al-Ahmad	The Uni. of Jordan	Jordan	108—Open Access
	Ala' M. Al-Zoubi	The Uni. of Jordan	Jordan	
	Ruba Abu Khurma	The Uni. of Jordan	Jordan	
	Ibrahim Aljarah	The Uni. of Jordan	Jordan	
S20	Ghaith Jardaneh	An-Najah Nati. Uni.	Palestine	65—Closed Access
	Hamed Abdelhaq	An-Najah Nati. Uni.	Palestine	
	Momen Buzz	An-Najah Nati. Uni.	Palestine	
	Douglas Johnson	Uni. of Colorado	USA	
S21	Shreya Tiwari	Amity University	India	100—Open Access
	Sarika Jain	Amity University	India	
S22	Mr. Vyankatesh Rampurkar	BIHER	India	23—Open Access
	Dr. Thirupurasundari D.R.	BIHER	India	
S23	Despoina Mouratidis	Ionian University	Greece	1—Open Access
	Andreas Kanavos	Ionian University	Greece	
	Katia Kermanidis	Ionian University	Greece	
S24	Malliga Subramanian	Kongu Eng. College	India	13—Open Access
	Premjith B	Amrita School of AI	India	
	K. Shanmugavadivel	Kongu Eng. College	India	
	Santhiya Pandiyan	Kongu Eng. College	India	
	Balasubramanian Palani	Indian Inst. of IT.	India	
	Bharathi Raja Chakravarthi	Uni. of Galway	Ireland	
S25	Jingyuan Yi	Carnegie Mellon Uni.	USA	27—Open Access
	Zeqiu Xu	Carnegie Mellon Uni.	USA	
	Tianyi Huang	Uni. of California	USA	
	Peiyang Yu	Carnegie Mellon Uni.	USA	
S26	Mutaz A. B. Al-Tarawneh	Am. Uni. of the ME	Kuwait	8-Open Access
	Omar Al-irr	Am. Uni. of the ME	Kuwait	
	Khaled S. Al-Maaitah	Mutah University	Jordan	
	Hassan Kanj	Am. Uni. of the ME	Kuwait	
	Wael Hosny Fouad Aly	Am. Uni. of the ME	Kuwait	
S27	Lingzhi Shen	Uni. of Southampt.	UK	7—Open Access
	Yunfei Long	University of Essex	UK	
	Xiaohao Cai	Uni. of Southampt.	UK	
	Imran Razzak	M. bin Z. Uni. of AI	UAE	
	Guanming Chen	Uni. of Southampt.	UK	
	Kang Liu	Uni. of Southampt.	UK	
	Shoaib Jameel	Uni. of Southampt.	UK	
S28	Muhammet TAN	Sivas Uni. of S.&T.	Turkey	N/A—Open Access
	Halit BAKIR	Sivas Uni. of S.&T.	Turkey	

Table 10. *Cont.*

Study	Names	Institutions	Country	Citation Access
S29	Hari Murti	Uni. Stikubank	Indonesia	1—Open Access
	Sulastri	Uni. Stikubank	Indonesia	
	Dwi Budi Santoso	Uni. Stikubank	Indonesia	
	Dwi Agus Diartono	Uni. Stikubank	Indonesia	
	Kristiawan Nugroho	Uni. Stikubank	Indonesia	
S30	Emad Alsuwat	Taif University	Saudi Arabia	674—Closed Access
	Hatim Alsuwat	Umm Al-Qura Uni.	Saudi Arabia	

Each literature review was evaluated for review and publication in the database. Therefore, the quality valuation questions were listed based on several standards, as shown in Table 11.

Table 11. The quality valuation for each study.

Study	Study Type	QA1	QA2	QA3	Total Score
S1	Experiment	P	P	Y	2
S2	Experiment	P	P	LP	1.25
S3	Experiment	P	Y	LP	1.75
S4	Experiment	P	Y	Y	2.5
S5	Experiment	Y	P	Y	2.5
S6	Experiment	P	P	Y	2
S7	Experiment	Y	Y	P	2.5
S8	Experiment	P	P	LP	1.25
S9	Experiment	P	Y	LP	1.75
S10	Experiment	P	P	P	1.5
S11	Experiment	P	Y	P	2
S12	Experiment	Y	P	LP	1.75
S13	Experiment	P	P	Y	2
S14	Experiment	Y	Y	Y	3
S15	Experiment	P	Y	LP	1.75
S16	Experiment	P	Y	P	2
S17	Experiment	P	P	Y	2
S18	Experiment	P	P	LP	1.25
S19	Experiment	P	P	LP	1.25
S20	Experiment	P	Y	LP	1.75
S21	Experiment	P	P	Y	2
S22	Experiment	P	Y	Y	2.5
S23	Experiment	Y	Y	Y	3
S24	Experiment	Y	Y	LP	2.25
S25	Experiment	Y	Y	Y	3
S26	Experiment	Y	Y	Y	3
S27	Experiment	P	P	LP	1.25
S28	Experiment	P	P	Y	2
S29	Experiment	Y	P	Y	2.5
S30	Experiment	Y	P	P	2

The chart shows the rating of each study in the literature review, as shown in Figure 4. From Figure 4, we see that in studies number S14, S23, S25, and S26, both deep learning and machine learning algorithms were used, and the datasets were sufficient to train the data with the features used. Therefore, the accuracy demonstrated by each study was above 98%.

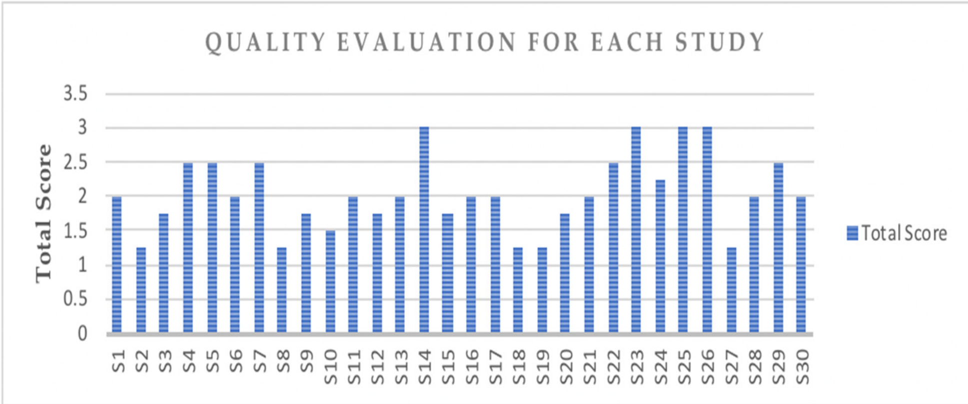

Figure 4. The quality evaluation.

5. Conclusions

This research provided a review of machine learning and deep learning algorithms for detecting fake news. It also presented the datasets used in this research, along with the features used to extract important data. It also presented gaps identified in each study and how to fill them. Studies number S14, S23, S25, and S26 used both deep learning and machine learning algorithms, and the datasets were sufficient to train the data with the features used. Therefore, the accuracy demonstrated by each study was high. The performance and quality evaluation of each study were also presented. Finally, this review concluded with a discussion of challenges, highlighting future perspectives on the topic of fake news detection.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/computers14090394/s1, Table S1: PRISMA 2020 Checklist.

Author Contributions: Conceptualization, F.A.A. (Faisal A. Alshuwaier) and F.A.A. (Fawaz A. Alsulaiman); formal analysis, F.A.A. (Faisal A. Alshuwaier); methodology, F.A.A. (Faisal A. Alshuwaier) and F.A.A. (Fawaz A. Alsulaiman); project administration, F.A.A. (Fawaz A. Alsulaiman); resources, F.A.A. (Faisal A. Alshuwaier); supervision, F.A.A. (Fawaz A. Alsulaiman); writing—original draft, F.A.A. (Faisal A. Alshuwaier); writing—review and editing, F.A.A. (Faisal A. Alshuwaier) and F.A.A. (Fawaz A. Alsulaiman). All authors have read and agreed to the published version of the manuscript.

Funding: There is no funding for this research.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1 (S1–S30) presents the results obtained through the analyzed articles, features, datasets, and algorithms, reported by them.

Table A1. Results obtained through the analyzed articles.

Study	Author	Year	Dataset	Algorithms/Methods
S1	Aphiwongsophon and Chongstitvatana [1]	2018	Twitter API	• Neural network • Naïve Bayes • SVM
S2	Krishna and Kumar [2]	2021	Kaggle	• Naïve Bayes
S3	Ni et al. [3]	2021	Open-source FakeNewsNet PolitiFact and GossipCop	• Logistic regression • Random forests • SVM
S4	Singh et al. [4]	2023	Kaggle	• NLP • Decision trees • Random forests • AdaBoost classification • XGBoost
S5	Jiang et al. [5]	2021	ISOT KDnugget	• Logistic regression • SVM • K-NN • Decision tree • Random forest • CNN • GRU • LSTM
S6	Gereme et al. [6]	2021	GPAC ETH_FAKE AMFTWE	• CNN
S7	Pardamean and Pardede [7]	2021	Kaggle	• BERT • NBSVM
S8	Kaliyar et al. [8]	2019	Multi-class	• Ensemble learning
S9	Nagoudi et al. [25]	2020	Arabic TreeBank AraNews	• mBERT • AraBERT • XLM-RBase • XLM-RLarge
S10	Hamed et al. [26]	2023	Fakeddit news	• LSTM • GRU • CNN • BI-LSTM
S11	Verma et al. [27]	2021	WELFake articles	• CNN • BERT • WELFake
S12	Ivancova et al. [28]	2021	Articles from Slovak websites	• CNN • LSTM
S13	Albahr and Albahr [9]	2020	LIAR	• Random forest • Naïve Bayes • Neural network • Decision tree
S14	Goldani et al. [10]	2021	ISOT LIAR	• Capsule neural network

Table A1. *Cont.*

Study	Author	Year	Dataset	Algorithms/Methods
S15	Wang et al. [11]	2021	LUN English SLN English Weibo Chinese RCED Chinese	• Neural network SemSeq4FD • CNN • LSTM
S16	Ozbay and Alatas [12]	2019	BuzzFeed political news Random political news LIAR	• Grey Wolf Optimization • Salp Swarm Optimization
S17	Birunda and Devi [13]	2021	Kaggle	• SVM • KNN • Naïve Bayes • Logistic regression • Random forest • AdaBoost • Decision tree • Gradient boosting
S18	Mugdha et al. [14]	2020	Bengali news	• Gaussian naïve Bayes • SVM • Logistic regression • Multilayer perception • Random forest • VotingEnsemble • AdaBoost • Gradient boosting • Multimodal naïve Bayes
S19	Al-Ahmad et al. [15]	2021	Koirala	• KNN-BGA • KNN BPSO • KNN BSSA
S20	Jardaneh et al. [16]	2019	Twitter API	• Random forest • Decision tree • AdaBoost • Logistic regression
S21	Tiwari and Jain [22]	2024	Articles	• Logistic regression • Decision tree • Random forest
S22	Rampurkar and D.R [23]	2024	ISOT	• Naïve Bayes • Logistic regression
S23	Mouratidis et al. [33]	2025	• Geroge McIntyre • UTK ML Kaggle • ISOT fake news • Kaggle + Signalmedia	• Naïve Bayes • SVM • Random forest • CNN • LSTM • BERT
S24	Subramanian et al. [29]	2025	• Task 1: news • Task 2: Malayalam news	• XLM-RoBERTa • BiLSTM with XLM-RoBERTa
S25	Jingyuan et al. [30]	2025	• FakeNewsNet • PolitiFact • PAN2020 • COVID-19	• GNN
S26	Al-Tarawneh et al. [34]	2024	• Truthseeker	• SVM • Multilayer perceptron • CNN

Table A1. *Cont.*

Study	Author	Year	Dataset	Algorithms/Methods
S27	Shen et al. [35]	2025	• Fakeddit • Yang	• GAMED for multimodal modeling
S28	Tan and Bakir [31]	2025	• TruthSeeker	• Bidirectional LSTM
S29	Mutri et al. [24]	2025	• FakeNewsDetection	• SVM • KNN
S30	Alsuwat, E. and Alsuwat, H. [32]	2025	• ISOT news • LIAR • COVID-19 fake news	• Bidirectional LSTM

Table A2 presents the results obtained through the features analyzed and the languages used in the literature review.

Table A2. Results obtained through the features analyzed and languages.

Study	Features/Attributes	Language
S1	• Raw data from Twitter API	Thailand
S2	• Count vectorizer • TF-IDF matrix	English
S3	• Document frequency	English
S4	• TF	English
S5	• TF and TF-IDF • Embedding	English
S6	• Word embedding	Amharic (African)
S7	• Hyperparameter settings	English
S8	• Content and context level	English
S9	• Word embedding	Arabic
S10	• Emotion analysis • Sentiment analysis • Text classification	English
S11	• Linguistic • Word embedding	English
S12	• Word2Vec • GloVe • Morphological analysis	Slovak
S13	• Unigram • Bigram • Trigram	English
S14	• n-gram • Word embedding	English
S15	• Sentence encoding • Sentence representation • Document representation	English + Chinese

Table A2. *Cont.*

Study	Features/Attributes	Language
S16	• TF • Document vector	English
S17	• TF-IDF • Site_Url • Text-based	English
S18	• TF-IDF • Extra tree classifier	Bengali
S19	• BOW • TF • TF-IDF	English
S20	• Content-based • User-based	Arabic
S21	• Semantic	English
S22	• TF-IDF	English
S23	• TF-IDF • Word2Vec • Contextual embeddings	English
S24	• Task 1: Contextual embeddings and sequential models • Task 2: Multilingual contextual embedding	Malayalam
S25	• Context features • Semantic features	English
S26	• TF-IDF • Word2Vec • FastText embedding	
S27	• Distinctive features • Discriminative features	English
S28	• Word Embedding	English
S29	• Categorical feature • Datetime feature	English
S30	• Word2Vec • TF-IDF • Temporal features	English

Table A3 presents the results obtained by displaying the models with their performances.

Table A3. The models and performances. The Performance column indicates the performance measure used in each study, followed by its corresponding value.

Study	Model	Performance
S1	• Neural network • Naïve Bayes • SVM	• Acc: 99.90% • Acc: 96.08% • Acc: 99.90%

Table A3. *Cont.*

Study	Model	Performance
S2	• Naïve Bayes TF-IDF vector • Naïve Bayes count vector • Naïve Bayes hash vector • Passive aggressive hash	• Acc: 85.70% • Acc: 89.30% • Acc: 90.20% • Acc: 92.20%
S3	• DF (PolitiFact) • DF (GossiCop)	• Acc: 68.00% • Acc: 67.00%
S4	• NLP • Decision trees • Random forests • AdaBoost classification • XGBoost	• Acc: High Acc.
S5	• Logistic regression TF-IDF on ISOT dataset • SVM TF-IDF on ISOT dataset • K-NN on ISOT dataset • Decision tree TF-IDF on ISOT dataset • Random forest TF-IDF on ISOT dataset • Random forest TF on ISOT dataset • CNN embedding on ISOT dataset • GRU embedding on ISOT dataset • LSTM embedding on ISOT dataset • Logistic regression TF-IDF on KDnugget dataset • SVM TF-IDF on KDnugget dataset • K-NN on KDnugget dataset • Decision tree TF-IDF on KDnugget dataset • Random forest TF-IDF on KDnugget dataset • Random forest TF on KDnugget dataset • CNN embedding on KDnugget dataset • GRU embedding on KDnugget dataset • LSTM embedding on KDnugget dataset	• Acc: 99.63% • Acc: 99.63% • Acc: 68.65% • Acc: 99.60% • Acc: 99.87% • Acc: 99.84% • Acc: 99.52% • Acc: 99.69% • Acc: 99.74% • Acc: 92.82% • Acc: 92.42% • Acc: 82.56% • Acc: 79.87% • Acc: 91.63% • Acc: 91.48% • Acc: 89.50% • Acc: 91.32% • Acc: 88.95%
S6	• cc_am_300 with pretrained embedding dim = 300 • AMFTWE with Amharic embedding dim = 50 • AMFTWE with Amharic embedding dim = 100 • AMFTWE with Amharic embedding dim = 200 • AMFTWE with Amharic embedding dim = 300	• Acc: 98.83% • Acc: 97.15% • Acc: 98.90% • Acc: 99.21% • Acc: 99.36%
S7	• BERT fine-tuning • Naïve Bayes SVM	• Acc: 99.23% • Acc: 95.00%
S8	• Gradient boosting	• Acc: 86.00%
S9	• mBERT on ATB dataset • XLM-RBase on ATB dataset • XLM-RLarge on ATB dataset • AraBERT on ATB dataset • mBERT on AraNews dataset • XLM-RBase on AraNews dataset • XLM-RLarge on AraNews dataset • AraBERT on AraNews dataset	• Acc: 77.16% • Acc: 81.72% • Acc: 82.41% • Acc: 83.19% • Acc: 79.39% • Acc: 82.77% • Acc: 82.12% • Acc: 87.21%
S10	• LSTM textual content features • LSTM textual content, title, and comment features • GRU textual content features • GRU textual content, title, and comment features • CNN textual content features • CNN textual content, title, and comment features • BI-LSTM textual content features • BI-LSTM textual content, title, and comment features	• Acc: 89.99% • Acc: 90.16% • Acc: 91.65% • Acc: 92.60% • Acc: 94.14% • Acc: 96.05% • Acc: 94.65% • Acc: 96.77%

Table A3. *Cont.*

Study	Model	Performance
S11	• CNN • BERT • WELFake	• Acc: 92.48% • Acc: 93.79% • Acc: 96.73%
S12	• CNN on model 1 • CNN on model 2 • Recurrent LSTM on model 2	• Acc: 92.38% • Acc: 92.38% • Acc: 93.56%
S13	• Random forest • Naïve Bayes • Neural Network • Decision Trees	• Acc: 91.00% • Acc: 99.00% • Acc: 92.00% • Acc: 90.00%
S14	• Non-static capsule networks	• Acc: 99.80%
S15	• SemSeq4FD on SLN English dataset • SemSeq4FD on LUN-test English dataset • SemSeq4FD on Weibo Chinese dataset • SemSeq4FD on RCED Chinese dataset	• Acc: 88.42% • Acc: 93.78% • Acc: 81.74% • Acc: 90.34%
S16	• SSO on random political news dataset • GWO on random political news dataset • Decision Tree on random political news dataset • Naïve Bayes on random political news dataset • SVM on random political news dataset • Gradient Boost on random political news dataset • Ridor on random political news dataset • J48 on random political news dataset • SMO on random political news dataset • SSO on Buzzfeed political news dataset • GWO on Buzzfeed political news dataset • Decision Tree on Buzzfeed political news dataset • Naïve Bayes on Buzzfeed political news dataset • SVM on Buzzfeed political news dataset • Gradient Boost on Buzzfeed political news dataset • Ridor on Buzzfeed political news dataset • J48 on Buzzfeed political news dataset • SMO on Buzzfeed political news dataset • SSO on LIAR dataset • GWO on LIAR dataset • Decision Tree on LIAR dataset • Naïve Bayes on LIAR dataset • SVM on LIAR dataset • Gradient Boost on LIAR dataset • Ridor on LIAR dataset • J48 on LIAR dataset • SMO on LIAR dataset	• Acc: 71.30% • Acc: 92.60% • Acc: 63.40% • Acc: 76.20% • Acc: 70.00% • Acc: 71.70% • Acc: 64.20% • Acc: 65.40% • Acc: 68.00% • Acc: 80.30% • Acc: 87.50% • Acc: 63.40% • Acc: 69.60% • Acc: 59.00% • Acc: 62.10% • Acc: 56.20% • Acc: 65.50% • Acc: 61.90% • Acc: 78.00% • Acc: 96.50% • Acc: 79.80% • Acc: 72.60% • Acc: 83.60% • Acc: 79.80% • Acc: 82.00% • Acc: 82.20% • Acc: 82.30%
S17	• SVM • KNN • Multinomial Naïve Bayes • Logistic regression • Random forest • AdaBoost • Decision Tree	• Acc: 64.00% • Acc: 70.60% • Acc: 72.30 • Acc: 80.70 • Acc: 88.30 • Acc: 96.00% • Acc: 98.00%

Table A3. *Cont.*

Study	Model	Performance
S18	• SVM (Linear) • Logistic regression • Multilayer perception • Random forest • VotingEnsemble Classifier • Gaussian Naïve Bayes • Multinomial Naïve Bayes • AdaBoost • Gradient Boosting	• Acc: 57.32% • Acc: 78.62% • Acc: 72.93% • Acc: 61.14% • Acc: 76.29% • Acc: 87.42% • Acc: 71.53% • Acc: 64.93% • Acc: 62.43%
S19	• KNN-BSSA with BOW features • KNN-BPSO with BOW features • KNN-BGA. with BOW features • KNN with TF-IDF features • KNN-BSSA with TF-IDF features • KNN-BPSO with TF-IDF features • KNN-BGA. with TF-IDF features • KNN with TF-IDF features • KNN-BSSA with TF features • KNN-BPSO with TF features • KNN-BGA. with TF features • KNN with TF features	• Acc: 72.64% • Acc: 72.58% • Acc: 73.48% • Acc: 70.53% • Acc: 61.61% • Acc: 66.39% • Acc: 67.64% • Acc: 70.53% • Acc: 73.32% • Acc: 73.48% • Acc: 73.84% • Acc: 70.53%
S20	• Random forest without sentiment features • Random forest with 4 sentiment features • Decision Tree without sentiment features • Decision Tree with 4 sentiment features • Logistic regression without sentiment features • Logistic regression with 4 sentiment features • AdaBoost without sentiment features • AdaBoost with 4 sentiment features	• Acc: 68.00% • Acc: 76.00% • Acc: 70.00% • Acc: 69.00% • Acc: 76.00% • Acc: 75.00% • Acc: 74.00% • Acc: 74.00%
S21	• Random forest • Logistic regression • Decision Tree	• Acc: 98.00% • Acc: 98.00% • Acc: 99.00%
S22	• Naïve Bayes • Logistic regression	• Acc: 94.37% • Acc: 98.31%
S23	• Naïve Bayes • SVM • Random forest • BERT • CNN • LSTM	• Auc: 97.50% • Auc: 97.60% • Auc: 96.30% • Auc: 98.40% • Auc: 97.30% • Auc: 97.60%
S24	• Taske 1: XLM-RoBERTa contextualized and sequential • Taske 2: BiLSTM with XLM-RoBERTa	• F1: 89.80% • F1: 62.83%
S25	• Misinformation detection knowledge integration • Fake news detection with multimodal large language models • Domain adaptive few-shot fake news detection • Style-agnostic detection framework	• Acc: 95.20% • Acc: 95.10% • Acc: 87.30% • Acc: 99.90%

Table A3. *Cont.*

Study	Model	Performance
S26	• SVM with TF-IDF • Multilayer perceptron with TF-IDF • Logistic regression with TF-IDF • Random forest with TF-IDF • Decision tree with TF-IDF • SVM with Word2Vec • Multilayer perceptron Word2Vec • Logistic regression with Word2Vec • Random forest with Word2Vec • Decision tree with Word2Vec • KNN with Word2Vec • SVM with FastText • Multilayer perceptron with FastText • Logistic regression with FastText • Random forest with FastText • Decision tree with FastText • KNN with FastText • CNN Model 1 with TF-IDF • CNN Model 2 with TF-IDF • CNN Model 3 with TF-IDF • CNN Model 1 with Word2Vec • CNN Model 2 with Word2Vec • CNN Model 3 with Word2Vec • CNN Model 1 with FastText • CNN Model 2 with FastText • CNN Model 3 with FastText	• Acc: 99.03% • Acc: 98.77% • Acc: 97.58% • Acc: 98.39% • Acc: 97.30% • Acc: 94.47% • Acc: 95.24% • Acc: 85.42% • Acc: 91.01% • Acc: 80.30% • Acc: 94.98% • Acc: 90.41% • Acc: 93.21% • Acc: 83.44% • Acc: 84.53% • Acc: 72.42% • Acc: 85.10% • Acc: 98.77% • Acc: 56.15 • Acc: 98.99% • Acc: 94.25% • Acc: 90.73% • Acc: 94.92% • Acc: 89.32% • Acc: 85.26% • Acc: 89.55%
S27	• GAMED	• Acc: 93.90%
S28	• Bidirectional LSTM	• Acc: 99.91%
S29	• SVM • KNN	• MAE: 0.725 • RMSE: 01.628 • MAE: 0.011 • RMSE: 0.077
S30	• Bidirectional LSTM on ISOT fake new • Bidirectional LSTM on LIAR • Bidirectional LSTM on COVID-19 fake news	• Acc: 96.30% • Acc: 95.60% • Acc: 97.10%

References

1. Aphiwongsophon, S.; Chongstitvatana, P. Detecting Fake News with Machine Learning Method. In Proceedings of the 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Rai, Thailand, 18–21 July 2018; pp. 528–531.
2. Krishna, I.; Kumar, S. Fake News Detection using Naïve Bayes Classifier. *Int. J. Creat. Res. Thought (IJCRT)* **2021**, *9*, e757–e761. Available online: https://ijcrt.org/papers/IJCRT2106550.pdf (accessed on 26 May 2025).
3. Ni, B.; Guo, Z.; Li, J.; Jiang, M. Improving Generalizability of Fake News Detection Methods using Propensity Score Matching. *arXiv* **2020**, arXiv:2002.00838. [CrossRef]
4. Singh, D.; Khan, A.H.; Meena, S. Fake News Detection Using Ensemble Learning Models. In Proceedings of the Data Analytics and Management. ICDAM 2023; Lecture Notes in Networks and Systems. Swaroop, A., Polkowski, Z., Correia, S.D., Virdee, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2023; Volume 78, pp. 55–63.
5. Jiang, T.; Li, J.P.; Haq, A.U.; Saboor, A.; Ali, A. A novel stacking approach for accurate detection of fake news. *IEEE Access* **2021**, *9*, 22626–22639. [CrossRef]
6. Gereme, F.; Zhu, W.; Ayall, T.; Alemu, D. Combating fake news in "low-resource" languages: Amharic fake news detection accompanied by resource crafting. *Information* **2021**, *12*, 20. [CrossRef]
7. Pardamean, A.; Pardede, H.F. Tuned bidirectional encoder representations from transformers for fake news detection. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 1667–1671. [CrossRef]

8. Kaliyar, R.K.; Goswami, A.; Narang, P. Multiclass Fake News Detection using Ensemble Machine Learning. In Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 13–14 December 2019; pp. 103–107. [CrossRef]
9. Albahr, A.; Albahar, M. An Empirical Comparison of Fake News Detection using different Machine Learning Algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 146–152. [CrossRef]
10. Goldani, M.H.; Momtazi, S.; Safabakhsh, R. Detecting fake news with capsule neural networks. *Appl. Soft Comput.* **2021**, *101*, 106991. [CrossRef]
11. Wang, Y.; Wang, L.; Yang, Y.; Lian, T. Sem-Seq4FD: Integrating global semantic relationship and local sequential order to enhance text representation for fake news detection. *Expert Syst. Appl.* **2021**, *166*, 114090.
12. Ozbay, F.A.; Alatas, B. A novel approach for detection of fake news on social media using metaheuristic optimization algorithms. *Elektron. Ir. Elektrotechnika* **2019**, *25*, 62–67.
13. Birunda, S.S.; Devi, R.K. A Novel Score-Based Multi-Source Fake News Detection using Gradient Boosting Algorithm. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 406–414. [CrossRef]
14. Mugdha, S.B.S.; Ferdous, S.M.; Fahmin, A. Evaluating machine learning algorithms for bengali fake news detection. In Proceedings of the 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 19–21 December 2020; pp. 1–6.
15. Al-Ahmad, B.; Al-Zoubi, A.M.; Abu Khurma, R.; Aljarah, I. An evolutionary fake news detection method for COVID-19 pandemic information. *Symmetry* **2021**, *13*, 1091. [CrossRef]
16. Jardaneh, G.; Abdelhaq, H.; Buzz, M.; Johnson, D. Classifying Arabic tweets based on credibility using content and user features. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 596–601.
17. Alshuwaier, F.; Areshey, A.; Poon, J. Applications and Enhancement of Document-Based Sentiment Analysis in Deep learning Methods: Systematic Literature Review. *Intell. Syst. Appl.* **2022**, *15*, 200090. [CrossRef]
18. Battal, B.; Yıldırım, B.; Dinçaslan, Ö.F.; Cicek, G. Fake News Detection with Machine Learning Algorithms. *Celal Bayar Univ. J. Sci.* **2024**, *20*, 65–83.
19. Kitchenhamy, B.; Brereton, O.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. *Systematic Literature Reviews in Software Engineering-A Systematic Literature Review*; Elsevier: Amsterdam, The Netherlands, 2009; Volume 51, pp. 7–15.
20. Do, H.H.; Prasad, P.; Maag, A.; Alsadoon, A. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Syst. Appl.* **2019**, *118*, 272–299. [CrossRef]
21. Toyer, S.; Thiebaux, S.; Trevizan, F.; Xie, L. Asnets: Deep learning for generalised planning. *J. Artif. Intell. Res. (JAIR)* **2020**, *68*, 1–68. [CrossRef]
22. Tiwari, S.; Jain, S. Fake News Detection Using Machine Learning Algorithms. In Proceedings of the KILBY 100 7th International Conference on Computing Sciences 2023 (ICCS 2023), Phagwara, India, 5 May 2024.
23. Rampurkar, M.V.; Thirupurasundari, D.D. An Approach towards Fake News Detection using Machine Learning Techniques. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 2868–2874.
24. Murti, H.; Sulastri, S.; Santoso, D.B.; Diartono, D.A.; Nugroho, K. Design of Intelligent Model for Text-Based Fake News Detection Using K-Nearest Neighbor Method. *Sinkron* **2025**, *9*, 1–7. [CrossRef]
25. Nagoudi, E.M.; Elmadany, A.; Abdul-Mageed, M.; Alhindi, T.; Cavusoglu, H. Machine Generation and Detection of Arabic Manipulated and Fake News. In *Workshop on Arabic Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020.
26. Hamed, S.K.; Ab Aziz, M.J.; Yaakub, M.R. Fake News Detection Model on Social Media by Leveraging Sentiment Analysis of News Content and Emotion Analysis of Users' Comments. *Sensors* **2023**, *23*, 1748. [CrossRef]
27. Verma, P.K.; Agrawal, P.; Amorim, I.; Prodan, R. WELFake: Word Embedding Over Linguistic Features for Fake News Detection. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 881–893. [CrossRef]
28. Ivancova, K.; Sarnovsky, M.; Krešňáková, V. Fake news detection in Slovak language using deep learning techniques. In Proceedings of the SAMI 2021, IEEE 19th World Symposium on Applied Machine Intelligence and Informatics, Herl'any, Slovakia, 21–23 January 2021; pp. 000255–000260.
29. Subramanian, M.; Premjith, B.; Shanmugavadivel, K.; Pandiyan, S.; Palani, B.; Chakravarthi., B. Overview of the Shared Task on Fake News Detection in Dravidian Languages-DravidianLangTech@NAACL 2025. In Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, Acoma, The Albuquerque Convention Center, Albuquerque, NM, USA, 3 May 2025; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2025; pp. 759–767.
30. Jingyuan, Y.; Zeqiu, X.; Tianyi, H.; Peiyang, Y. Challenges and Innovations in LLM-Powered Fake News Detection: A Synthesis of Approaches and Future Directions. *Comput. Lang.* **2025**, 87–93. [CrossRef]

31. Tan, M.; Bakır, H. Fake News Detection Using BERT and Bi-LSTM with Grid Search Hyperparameter Optimization. *Bilişim Teknolojileri Dergisi.* **2025**, *18*, 11–28.
32. Alsuwat, E.; Alsuwat, H. An improved multi-modal framework for fake news detection using NLP and Bi-LSTM. *J. Supercomput.* **2025**, *81*, 177. [CrossRef]
33. Mouratidis, D.; Kanavos, A.; Kermanidis, K. From Misinformation to Insight: Machine Learning Strategies for Fake News Detection. *Information* **2025**, *16*, 189. [CrossRef]
34. Al-Tarawneh, M.A.B.; Al-Irr, O.; Al-Maaitah, K.S.; Kanj, H.; Aly, W.H.F. Enhancing Fake News Detection with Word Embedding: A Machine Learning and Deep Learning Approach. *Computers* **2024**, *13*, 239. [CrossRef]
35. Shen, L.; Long, Y.; Cai, X.; Razzak, I.; Chen, G.; Liu, K.; Jameel, S. GAMED: Knowledge Adaptive Multi- Experts Decoupling for Multimodal Fake News Detection. In Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM '25), Hannover, Germany, 10–14 March 2025; pp. 586–595.
36. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Aki, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef] [PubMed]
37. Aizawa, A. An information-theoretic perspective of tf–idf measures. *Inf. Process. Manag.* **2003**, *39*, 45–65. [CrossRef]
38. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781. [CrossRef]
39. Matrix, C. Available online: https://h2o.ai/wiki/confusion-matrix/ (accessed on 26 May 2025).