Fake News Detection Using Machine Learning Techniques

Author: Sanjiv Sharma[1], Vidit Upreti[1], Shwetank Pratap Tiwari[1], Lakshya Yadav[1], Umang Rastogi[1], Dr. B. Barani Sundaram[2]

1. **Department of Computer Science & Engineering, KIET Group of Institutions Ghaziabad, India**
2. **Department of Computer science, College of Informatics, Bule hora University, Bule hora ,Ethiopia**

Abstract

The proliferation of fake news has become a major concern in recent years, making it imperative to develop methods for detecting and combatting it. This research paper investigates the effectiveness of machine learning techniques in detecting fake news. In this study three datasets were used to evaluate the performance of various algorithms, including linear support vector machines (SVM), boosting algorithms like Adaboost and XGboost, K-Nearest Neighbors, Naive Bayes, and ensemble learning algorithms. The results show that the Linear SVM and Bagging Classifier algorithms performed best among ensemble learners, with an average accuracy of 96.4% and 97.6%, respectively, while K-Nearest Neighbors performed the worst with an accuracy of 65.8%. Logistic regression also had a high average accuracy of 95% across the three datasets. The study also analyzed the precision, recall, and F1-score of each algorithm on the three datasets. Overall, the findings suggest that machine learning techniques can effectively detect fake news and assist in combating its spread.

Keywords: Fake news, Social media, Authenticity, Artificial Intelligence, Logistic regression, Support vector machine, Naïve Bayes algorithm, Random Forest algorithm,

Introduction

False information spreading among the masses is nothing new and has been around since ancient times, long before the advent of the Internet. Misleading and false information is disseminated for financial gain through the use of fake news. A couple of years ago the news used to disseminate from one person to another by word of mouth, but in the current scenario, the dissemination of information has come a long way since then. Because of the development of the World Wide Web and the rapid proliferation of online social media platforms (such as Facebook, Twitter, and WhatsApp), it is now possible for information to be transmitted in a manner that has never been witnessed before in the annals of human history. As a result of increased connectivity and intelligent automation, now more than a million users get their news and other information primarily from social networking sites and other social platforms. Contrary to other internet applications, news organizations benefited from the extensive usage of social media platforms by updating their subscribers' news in almost real-time. Newspapers, tabloids, and magazines are the prime sources of online news, and blogs, social media feeds, and other digital media formats are the latest news media. With the help of social media, any idea can quickly spread to millions of people all over the world. These social media platforms, in their present phase, are very successful and helpful in providing users with the opportunity to debate, exchange, and discuss themes such as democracy, education, and health. However, some people or organizations also utilize these platforms negatively, frequently to obtain financial advantage, occasionally to influence public opinion, and people's attitudes, or to propagate satire or outrageousness. In the past ten years, false news has spread quickly.

A lot of issues have arisen in politics, sports, health, and science, as a result of the widespread circulation of fake news online. The financial markets are one of these areas where fake news is a problem. A rumor can have terrible results or even stop the market altogether. Someone's ability to make decisions is significantly influenced by the information they take in. There are mounting pieces of evidence that people have behaved unwisely in response to news that afterward turned out to be false. One recent example is the propagation of the new coronavirus, in which false information regarding the virus's origin and behavior is propagated across the Internet, which further leads to chaos and unrest in society.

It would take a tremendous amount of time and resources for people to verify the veracity of every social media post or article. Therefore, more effort should be put into developing automated applications and techniques for determining whether or not news stories shared on social media are authentic. There is an irresistible need for a low-cost, high-performing model that can accurately categorize the proliferation of fake news or articles while requiring minimal manual effort.

This research article is organized as follows: Section 1 gives a brief introduction to the research area and its importance in our society; Section 2 gives an insight into the contemporary research work available in the literature. Further section 3 discusses the proposed framework along with the data sets, data pre-processing, selected machine learning algorithms for the study, and different performance matrices for performance evaluation. Section 4 discusses the comparative analysis of the result achieved. In the last section, 5 presents a conclusion of this research work.

2. Related works

Machine learning is a tool that is often used to spot fake news. Many researchers have tried to come up with different methods and frameworks for figuring out whether the news is real or fake. This section has talked about some of these ways of doing things.

Shahbazi Z et al. [1] proposed a method for fake media detection using natural language processing and blockchain approaches where they collected data from online sources and social media networks which include Twitter, BBC, and Facebook. From this data, they have extracted 5 features that have been used to train blockchain-based reinforcement learning. The proposed solution states that proof of authority and used validation are useful tools for dealing with fake news.

Mridha M et al. [2] reviewed the use of deep learning algorithms in fake news detection using a combination of various NLP techniques and deep learning algorithms on a different dataset which is available for benchmarking. As a result, they have tried to uncover the weaknesses and strengths of these algorithms.

Seddari N et al. [3] created a technique that incorporates linguistic and fact-checking elements into the process of training a model that employs four different machine learning algorithms logistic regression, random forest, additional trees discriminant, and XGBoost. The result shows that Random Forest achieves an accuracy of 94% which outperforms the other algorithms used.

Jiang T et al. [4] proposed a Novel stacking approach for the accurate detection of fake news. They used 5 machine learning and 3 deep learning algorithm and evaluated their performance on two different datasets in terms of recall, accuracy, f1-score, and precision. Then they proposed a stacking approach model and used McNemar’s test to determine if the performance is any different.

Rohera D et al. [5] gave a taxonomy of fake news classification techniques. They trained 4 machine learning models to detect fake news. Later they improved its performance by hyper tuning parameters like drop-out factor, smoothening, and batch size. This new mode, when implemented gave better results in the evaluation phase.

Tariq A et al. [6] proposed adversarial training to regularize fake news classification tasks. This paper evaluates the performance of two transformed modes on publicly available datasets. The results of the evaluation process show that on a long text, the model performs better than the baseline in accuracy, precision, and recall but degrades for short text classification.

Shishah W [7] proposes a joint BERT application to perform automatic fake news detection on Arabic news. The proposed framework is tested on 4 real-world Arabic datasets and the test are compared to other Arabic fake news techniques such as Qarib, and araGPT2. This joint BERT model works better than all the others in terms of recall, precision, and F-1 score.

Umer M et al. [8] gave a fake news stance detection architecture using Deep learning. They use a combination of deep learning algorithms such as CNN and LSTM and neural networks. This approach gave an accuracy of 97.8% which they consider to be better than the previous study.

Ali H et al. [9] analyzed fake news detection models using black box testing where they used 4 deep learning methods which include Multi-layer Perceptron, Convoluted Neural Network, Recurrent Neural Network, and a hybrid CNN-RNN-based model for robustness. In their research, they have tried to give various length inputs and use different loss functions that prove that CNN is the most robust model followed by RNN, but the MLP and hybrid model drop in accuracy after an increase in the input size.

Carter M et al. [10] have tried to uncover the strengths of various machine learning and deep learning methods which are used in fake content detection. Their research provides insight into the fields in which supervised and unsupervised learning methods provide better results. They point out the key loopholes in these methods and ways we can try to improve the performance.

Qureshi K et al. [11] have created a model that used the information regarding the sources and the connectivity of the news propagators to classify the news present on Twitter and used it on the COVID-19-based news dataset. They used a series of algorithms in their evaluation and concluded that CatBoost and RNN were the most effective with an accuracy score of 98%.

Elsaeed E et al. [12] have used voting classifiers for the problem in which they have given a framework to pre-process the data using lemmatization and vectorization algorithms and built a model using their voting classifier on 3 datasets. The proposed model provides improved accuracy over the existing solutions on the ISOT dataset by 0.2%.

Xu K et al. [13] put forth a framework that is built around the domain of the news to be classified. They have studied features including domain popularity, domain ranking, and the probability of the news disappearing for classification. Other key components in their model were TF-IDF vectorizer and LDA topic modeling as a result they concluded that the fake and real news only had subtle differences in their data and have given a prospect to use doc2vec for vectorization.

Ravish et al. [14] have created a model that incorporates the use of Multi-layered Principal Component Analysis that determines the features to be used to train the model based on Multi-Support Vector Machines. They have used this model to classify news in 10 different datasets with varying features. They conclude based on their evaluation that the accuracy is only low in the case of the datasets having less number of features.

Sudhakar M et al. [15] suggested an automatic detection technique that makes use of algorithms for machine learning in order to identify false news stories in Indian media information and sources. The mechanism that is now being presented helps control bogus information. Classification models built using machine learning, such as the Naive Bayes, Support Vector Machine, and Logistic Regression algorithms. The LSTM Convolutional Neural Network model that was proposed has delivered much-improved accuracy.

Lorent S and Itoo A [16]evaluated the effectiveness of the Attention Mechanism for detecting fake news on two datasets, one named “Fake News Corpus” which contains articles from online sources, and the second “Liar-Liar Corpus” which is described as a collection of short news articles from various sources such as political interviews, TV ads. The results of the Attention Mechanism and results on both datasets are compared against Deep learning models and methods of machine learning. Word2vec Embedding is used along with the attention mechanism to enhance the performance of the approach.

Preda A [17] describes various techniques used in the classification of text and examines the performance of different machine learning and neural networks on a dataset containing articles written in Romanian. BERT used transformers alongside other machine learning techniques to categorize articles into real news, fake news, and other types. BERT used after parameter tuning performed much better than other machine learning techniques.

Dementieva D [18] proposed a new feature Multiverse, which uses cross-lingual evidence for the detection of unreliable news and enhances existing techniques. Firstly, a manual study was conducted on twenty news datasets to check whether a real-life user can detect fake news using cross-lingual evidence. Then, using two strategies, content similarity estimation was experimented on. First, based on cosine distance between news text embeddings. Second, based on Natural language inference scores when scraped news was used as hypothesis h, and original news was used as premise p. Lastly integration of cross-lingual evidence features into an automated pipeline for fake news detection. Cross-lingual features in combination with linguistic features bested both deep learning and statistical fake news classification systems.

Li D [19] proposed an autoencoder based on an unsupervised fake news detection method (UFNDA). This research studies various news kinds in social networks and incorporates text content, images, user information, and propagation to increase the effectiveness of fake news identification. UFNDA is used to assess fake news by utilizing it as anomalous data. Compared to unsupervised unreliable news detection techniques, UFNDA produces better outcomes. To perform in-depth categorization of news, UFNDA could yet be developed.

3. Material and methods

In this section, the proposed framework is explained in detail. The framework as reflected in *Figure-1* depicts, that the data is collected from various credible sources which include Kaggle [20] and the University of California[21]. The data is then preprocessed using the inbuilt library in Python NLTK (Natural Language Toolkit)[22] then feature extraction is performed on text using another inbuilt library of Python called TF-IDF Vectorizer[23]. Following this 9 machine learning algorithms have been used to train a model to classify if a news article is either reliable or non-reliable.

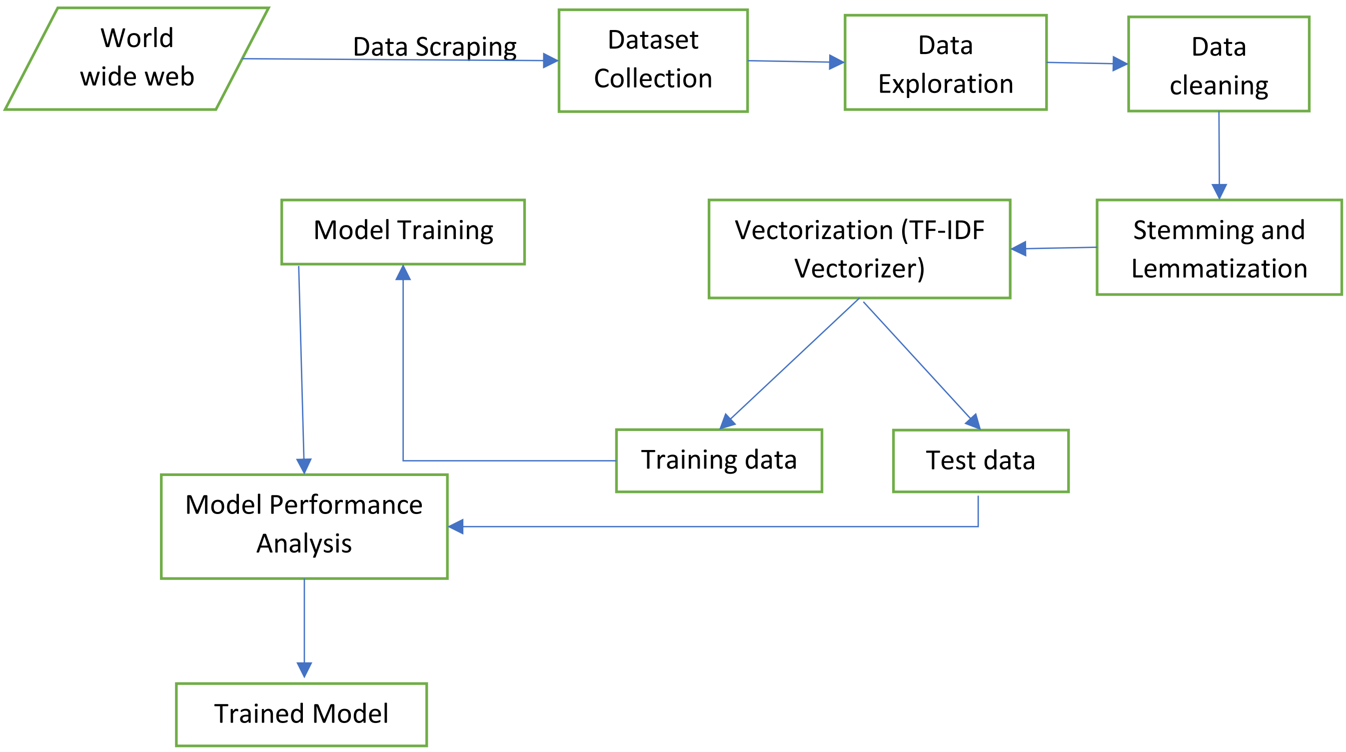

Figure 1 Proposed Framework

3.1. Datasets

This work includes 3 datasets that are used for training the model and finally testing it. The datasets consist of news from various genres which include news related to politics, entertainment, technology, and sports. The datasets are also having a mix of both true and fake news that is well labeled.

The first dataset is a dataset taken from Kaggle[24]. The data is consisting of 5 columns which include fields id, author, title, text, and a label. The dataset is divided into train and test data in which the training dataset contains 20387 rows of data, on the other hand, the testing data contains 5127 rows of data.

The second dataset (containing 3,352 articles) is accessible on Kaggle[25]. The authentic articles are sourced from The New York Times, CNN, Reuters, and other reputable online sources, whereas false news is sourced from news websites that have been known to publish unreliable information. This dataset covers a variety of topics including politics, entertainment, and sports.

The third dataset "ISOT Fake News Dataset"[21], is a dataset that is widely used as a benchmark for evaluating fake news classification problems. The dataset is a collection of articles that are present on a news website called reuters.com. On the other hand, there are fake articles that are a compilation of news from sources that Politifact.com has identified as unreliable. The articles are a mix of various genres but most of them focus on political news. A total of 44,898 articles are present of which 21,417 are true while the rest are false.

3.2. Data pre-processing

The data that is present in the dataset is in form of published articles hence it is still having many inconsistencies and comprises null values, non-ASCII characters, meaningless spaces, hashtags, and other symbols that are not useful in the evaluation process. The pre-processing includes 7 steps that ensure the removal of all inconsistencies and unwanted characters. The first step was removing all the null-valued rows, which are the rows that have no values entered in them. Null-valued rows provide a false output when any function is applied. The second step was to add an empty string to the columns where a value is missing to remove inconsistencies. The third step was to convert the data into lowercase so that it can be easily operated on and to maintain a standard for the data. The fourth step was removing all non-ASCII characters found in the data. The non-ASCII characters have no meaning in the evaluation since they cannot be handled through a vectorizer. The fifth step includes the removal of all mentions (names following @) and all the URLs in the data. The next step was to filter out hashtags, extra numbers, and spaces that have no significant meaning. All the above-mentioned pre-processing is done using a library called NLTK (Natural Language Toolkit)[26]. It has applications in semantic reasoning, categorization, tokenization, stemming, tagging, and parsing, among others.. The final step uses port stemmer to convert all the words and reduce them into their stem words or root words. Following stemming, the data at hand is a group of words that are in their root form, but a machine learning algorithm can only work on numerical data that is where the vectorizer comes in. The pre-processed data is then given to the vectorizer known as TF-IDF Vectorizer[27]. This commonly used technique transforms text into equivalent numerical representations to be used in machine learning algorithms. Machine-learning models are then constructed using the input features. Each dataset is split into a training set and a testing set, each with an 80/20 split.

The problem at hand puts forth a scenario where two classes are there, and machine learning is used to classify the news articles into one of these classes. The first class is reliable and the other is unreliable or fake news. In these types of scenarios, we use an algorithm that can perform the task of binary classification which include Logistic regression, K-nearest neighbors, Linear SVM, Random Forest, Bagging classifier, Decision trees, AdaBoost, XGBoost, and Naive Bayes Classifier.

3.3. Algorithms

3.3.1. Logistic Regression.

Logistic Regression[28] is one of the most frequently applied classification methods. It is a supervised classification algorithm that takes a set of inputs as features and maps the data into classes that are our required outputs. The algorithm although called regression is used in classification because it uses a sigmoidal function to model the data and divide it into classes depending on a threshold value.

$$sig(x) = \frac{1}{1+e^{-x}} \tag{1}$$

Using the probability value, we aim at minimizing the cost function that results in the optimal probability.

$$C(sig(x), y) = log\big(sig(x)\big), \quad y = 1 \tag{2}$$

$$C(sig(x), y) = -\log\big(1 - sig(x)\big), \quad y = 0$$

Here C (x, y) states the cost when the sigmoidal function returns the value x.

The tuning we require is to determine the outcome based on the requirement. We can have either a high Precision or a high Recall result. The difference between these results determines the number of false positives and false negatives we will get in our results.

3.3.2. Linear Support Vector Machine Algorithm.

The linear support vector machine (LSVM) is an example of a supervised learning method that is useful for solving classification issues [29]. It does so by plotting the data points into the n-dimensional graph for the supplied set of n characteristics. Support vectors are known to be the most extreme points on a graph, which is where the term "support vector machine" comes from. The data is classified based on the hyperplane that runs through the middle of the two categories. When working with data that can be linearly divided into two classes, linear support vector machines, or linear SVMs, perform exceptionally well. This program makes use of the RBF kernel because it is the one that performs the best when dealing with huge applications such as a corpus of news items. Given these two samples, x and x', the Radial Basis function can be written as:

$$f(x, x') = e^{\frac{||x-x'||^2}{2a^2}} \tag{3}$$

3.3.3. K-Nearest Neighbour(KNN) Algorithm

KNN algorithm[30] is a supervised machine learning algorithm that is used for solving a classification problem. KNN is an instance-based learning algorithm that uses the complete data for training the model. It classifies each new case into a category based on the knowledge gained from previous cases. This algorithm maintains the data throughout the training phase and then classifies the data based on the value of K at the time of classification. The number of neighbours we consider when determining the most similar class is K. The value of K can be determined by trying out different values.

To find the k most similar instances we use Euclidian's distance formula (5). The Euclidian distance is the square root of the sum of squares of the difference between the testing data and training data.

$$Euclidean\ distance\ = \sqrt{\sum_{i=1}^{i=n}(x_i - y_i)^2} \tag{4}$$

3.3.4. Naïve Bayes Classifier Algorithm

Naïve Bayes Classifier Algorithm[31] is a supervised learning algorithm and is used for solving classification problems. This Classifier uses the probability of data to predict its class. The probability is calculated using Bayes theorem (6). In this algorithm, all the features of an instance contribute independently to the determination of its class and that is why it is named Naïve. This algorithm is useful for text classification over a large dataset.

Bayes theorem is used to find the posterior probability p(a/x), using p(a),p(x), and p(x/a). Given a predictor x, the posterior probability of the class (a, target) is denoted by the symbol P(a|x) (x, attributes). The prior probability of class is denoted by the symbol P(a). The likelihood, denoted by the symbol P(x|a), is the probability of a predictor being given a class. The prior probability of the predictor is denoted by the symbol P(x).

$$P\left(\frac{a}{x}\right) = P\left(\frac{x}{a}\right).P(a)/P(x) \tag{5}$$

3.3.5. Decision Tree Classification Algorithm

Classification issues can be tackled with the help of the decision tree classification method [32], a supervised learning approach. After inferring the decision rules from the training data, the resulting decision tree may accurately predict the target class. Each branch of a decision tree is either a decision node or a leaf node. The decision nodes are where the data is divided, and the leaves are the decisions or the results. It offers a top-down strategy in terms of its organizational structure.

The tree divides the dataset into a small set of records by applying the inferred decision rules. These small sets of records are the different classes that are predicted by the tree.

3.3.6. Random Forest Algorithm.

Random forest[33] is a supervised machine learning algorithm that is used to solve classification problems as well as regression problems. It is an ensemble of a decision tree that gives better performance for a complex problem compared to a decision tree. Instead of relying on one decision tree for predictions, Random Forest uses predictions from many decision trees and the result depends on the majority, among the predictions of all the trees.

The accuracy of this algorithm depends on the number of decision trees used in the algorithm. Increasing the decision trees in the forest also prevents the problem of overfitting. The random forest algorithm works because it uses many relatively uncorrelated trees operating together. If similar trees are used in the forest, then the result will be no different from the decision tree algorithm. We can get uncorrelated trees can be obtained by using feature randomness and bootstrap aggregation.

Bootstrap Aggregation

Decision trees are extremely sensitive to the data that they are trained on, and even the smallest of modifications to the data set that they are trained on can result in dramatically different tree architectures. The random forest algorithm makes use of this fact by enabling each individual tree to randomly pick data from the dataset while maintaining replacement, which ultimately results in a variety of trees. Bagging, or bootstrapping, is another name for this process.

Feature Randomness

When it comes time to split a node in a typical decision tree, we take into account all of the alternative features and select the one that creates the greatest amount of differentiation between the observations included in the left node and those contained in the right node. In contrast, only a random subset of characteristics is available for selection by each individual tree in a random forest. Because of this, there is an even greater degree of variety among the trees in the model, which ultimately leads to a weaker correlation between the trees and a greater degree of diversification.

3.3.7. Boosting Classifier

In this method of ensemble learning, we initially have stumps (1-level trees) these are the basic tree that we initially use in the training process. The training data is then classified, and the results are checked the data that has been misclassified is then given a higher weight and we focus on reducing these misclassified points, then we further use these base-level trees to create a classifier that can handle the classification problem more efficiently. Some examples of boosting include AdaBoost[34] and XGBoost[35]. AdaBoost (Adaptive boosting) functions by changing the sample distribution by updating the weights during each iteration of its execution. The 3 main steps in this are using stump (1-level tree) as weak learners then influencing the next stump based on modification on the first stump and finally a weighted vote to select the stump that will be used in the final class. XGBoost (eXtreme Gradient Boosting) is another widely used boosting algorithm that is based on Gradient boosting and uses more regularized model formalization to control overfitting which yields better results. Sequential patterns are used to construct decision trees.

3.4. Performance Metrics

The performances of the algorithms (in section 2.3) for fake news detection have been compared using several assessment metrics. Evaluation metrics are widely utilized to assess the performance of supervised machine learning algorithms as they enable us to compare the efficiency of all the algorithms. In the proposed model, As can be seen in Table 1, an evaluation of the effectiveness of the false news detection model was carried out with the use of a confusion matrix. Samples are determined to be either real or fake using this matrix.TP, FP, TN, and FN concepts are explained as follows:

• True Positive (TP): The predicted fake news is actually fake.

• False Positive (FP): The predicted fake news is actually real.

• True Negative (TN): The predicted real news is actually real.

• False Negative (FN): The predicted real news is actually fake.

Table 1 Confusion Matrix

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True Positive	False Positive
Predicted Negative Class	False Negative	True Negative

Accuracy: Accuracy measures how often a prediction turns out to be true or false for a set of observations. In order to determine how well a model performs, Equation (6) can be used to measure accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Recall: The recall measure is the proportion of correct classifications that correspond to the actual class. Number of projected true articles divided by the total number of true articles in the suggested model. Equation (7) is used to find out the value of the recall:

$$\mathit{Recall} = \frac{TP}{TP+FN} \quad (7)$$

Precision: The precision score is the ratio of the number of true positives to the total number of events that were correctly forecasted as true. In the model that has been proposed, it indicates the percentage of articles that have been determined to be accurate out of the total number of articles that have been accurately predicted. Equation 8 can be used to find out the recall value.

$$\mathit{Precision} = \frac{TP}{TP+FP} \quad (8)$$

F1-Score: The F1 score is a representation of the balance that was made between precision and recall. It performs the calculation necessary to determine the harmonic mean between the two. This method takes into consideration both the false positive and the false negative observations. The equation 9 below can be used to get an individual's F1 score.:

$$\mathit{F1\text{-}score} = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Accuracy refers to the rate at which news is correctly expected for all samples when discussing the difficulty of identifying false news. In relation to the overall number of fake news items, the recall value shows what percentage of false news stories may have been successfully predicted. Based on the overall amount of fake news that can be foreseen, the Precision measure assesses the amount of fake news that can be reliably predicted. The harmonic average of the precision value and recall value is used to calculate the F-measure, a method for detecting instances of fake news.

4. Results

The accuracy of each method on the three datasets under consideration is summarised in Table 2. The Linear SVM attained a maximum accuracy of 96.5% on DS1 (Kaggle Fake News Dataset 1). The accuracy of boosting algorithms like Adaboost and XGboost was 95%. The least accurate models were K-Nearest Neighbors and Naive Bayes, with scores of 45% and 71%, respectively. On DS1, ensemble learning algorithms achieve an accuracy of 93.5% on average, compared to an accuracy of 66.7% for individual learners. Individual and Ensemble learners differ significantly, with a 32.2% absolute difference between the two groups. The algorithms with the best results on Dataset 2 are Linear SVM and Bagging Classifier, which achieve an accuracy of 97%. Although individual learners reported an average accuracy of 94.3%, the average accuracy of the ensemble learners was 95.6%, making the difference between the two non-significantly different by 1.3%. On Dataset 3, logistic regression. and bagging algorithms both achieved an accuracy of 98%, while Linear SVM reported an accuracy of 99%. K-Nearest Neighbors reported the lowest accuracy, at 57.5%. The average accuracy for individual and ensemble learners was 88.7% and 97.1%, respectively.

Table 2: Overall accuracy for each dataset

Accuracy Score	Dataset 1	Dataset 2	Dataset 3
K- Nearest Neighbors	0.45	0.905	0.575
Linear SVM	0.965	0.973	0.99
Logistic regression	0.939	0.961	0.979

Naïve-Bayes	0.711	0.94	0.947
Decision Tree	0.913	0.936	0.946
Random Forest	0.885	0.943	0.976
Bagging	0.944	0.97	0.98
XG boost	0.957	0.945	0.963
Ada boost	0.956	0.966	0.965

The average accuracy of each algorithm across the three datasets is shown in Figure 2. Overall, the bagging classifier and linear SVM algorithms perform the best among ensemble learners (accuracy of 96.4% and 97.6%, respectively), while K-Nearest Neighbors perform the worst (accuracy of 65.8%). The accuracy of individual learners is 87.8%, whereas that of ensemble learners is 95.37%. Only dataset 2 (90.5%) had a high accuracy score with K-Nearest Neighbors, while datasets 1 and 3 received poor results. On datasets 2 and 3, naive Bayes also performed well (accuracy of 94%), but not on dataset 1 (accuracy of 71%). Recall, precision, and F1-score are additional metrics we use to evaluate the performance of learning models because the accuracy score alone is not a trustworthy measure of a model's performance.

Tables 3-5 review each algorithm's precision, recall, and f1-score for each of the three datasets. In terms of average precision (Table 3), Linear SVM performed best among individual learners, while the Bagging classifier performed best among ensemble learners. On all three datasets, the linear SVM and bagging classifiers' average precision is 97%. The average precision of random forest (RF) increased to 96.9% on the three datasets after deleting the dataset with the lowest score, DS2. Random forest (RF) had previously attained a precision of 95%. The K-Nearest Neighbors algorithm has the lowest precision score.

According to the recall performance metric, the Boosting Algorithm (Adaboost) outperforms the Ensemble Learners with a recall score of 0.96, while the Linear SVM performs best among Individual Learners with a recall score of 0.971. Following closely behind are the K-Nearest Neighbors technique and the boosting classifier (XGBoost), both of which had recall values of 0.953 and 0.963 respectively. On the F1- score, the algorithms performed similarly to how they did for precision. The best F1-score of any technique was attained by the linear SVM, which had an F1-score of 0.97. The bagging classifier and Adaboost learning algorithm came in second with F1-scores of 0.959.

Figure 3 shows the average performance of learning algorithms utilizing precision, recall, and f1-score across the three datasets. It is evident that the effectiveness of learning algorithms utilizing different performance criteria does not differ significantly with K- Nearest Neighbors and Naïve Bayes being exceptions.

On all performance criteria, the ensemble learner XGboost outperforms other learning models. The main reason for the increased performance of the software is XGBoost's operating philosophy, which efficiently discovers defects and minimizes them in each cycle. Using multiple classifications and regression trees (CART), which include numerous weak learners, to give misclassified data points higher weights, is the essential notion behind how XGBoost functions. For the model to reliably identify the erroneously classified points on each successive iteration, overfitting is reduced using regularisation parameters.

Despite being a relatively straightforward model, logistic regression has an average accuracy of nearly 95% across the three datasets. The high average accuracy can be attributed to a few factors. First, a thorough grid search with numerous hyperparameters is used to optimize the logistic regression model. Second, the similar writing styles of some authors in some datasets (like DS2) help explain the logistic regression model's 96% accuracy.

Table 3: Overall precision for each dataset

Precision	Dataset 1	Dataset 2	Dataset 3
K-Nearest Neighbors	0.44	0.902	0.55
Linear SVM	0.969	0.97	0.989
Logistic regression	0.96	0.95	0.978
Naïve-Bayes	0.998	0.903	0.933
Decision Tree	0.907	0.954	0.939
Random Forest	0.959	0.916	0.979
Bagging	0.967	0.97	0.983
XG Boost	0.957	0.962	0.964
Ada boost	0.953	0.929	0.963

Table 4: Overall Recall for each dataset

Recall	Dataset 1	Dataset 2	Dataset 3
K-Nearest Neighbors	0.998	0.893	0.998
Linear SVM	0.95	0.973	0.992
Logistic regression	0.89	0.967	0.981
Naïve-Bayes	0.334	0.975	0.969
Decision Tree	0.891	0.906	0.959
Random Forest	0.768	0.967	0.976
Bagging	0.902	0.965	0.979
XG Boost	0.943	0.954	0.964
AdaBoost	0.945	0.965	0.97

Table 5: Overall F1-Score for each dataset

F1-Score	Dataset 1	Dataset 2	Dataset 3
K-Nearest Neighbors	0.61	0.897	0.71
Linear SVM	0.959	0.971	0.99
Logistic regression	0.928	0.958	0.98
Naïve-Bayes	0.5	0.938	0.951
Decision Tree	0.89	0.93	0.949
Random Forest	0.85	0.941	0.977
Bagging	0.93	0.967	0.981
XG Boost	0.95	0.941	0.964
Ada boost	0.949	0.963	0.966

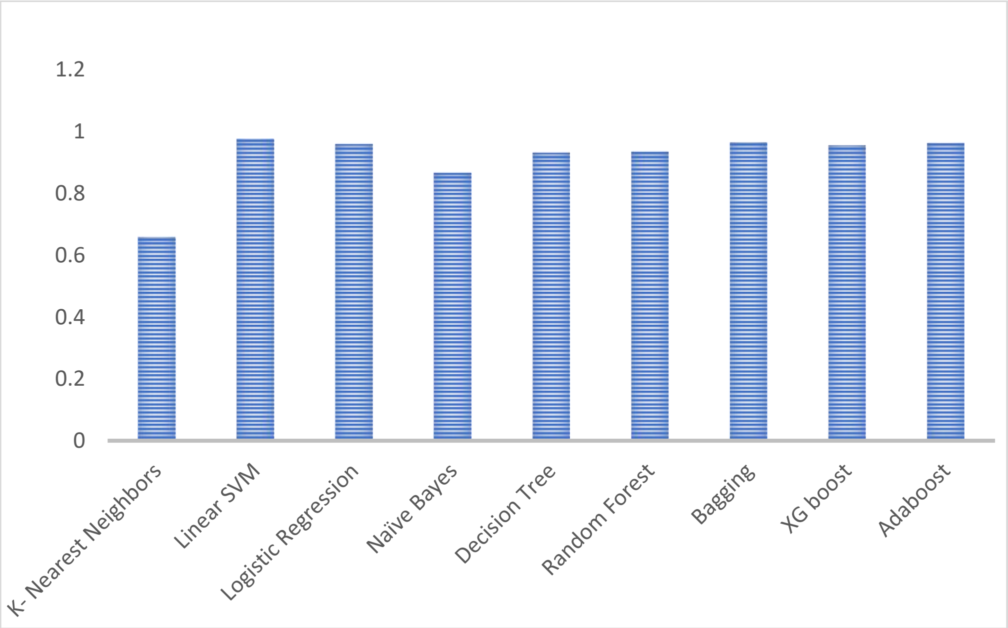

Figure 2: Average Accuracy Score over all datasets

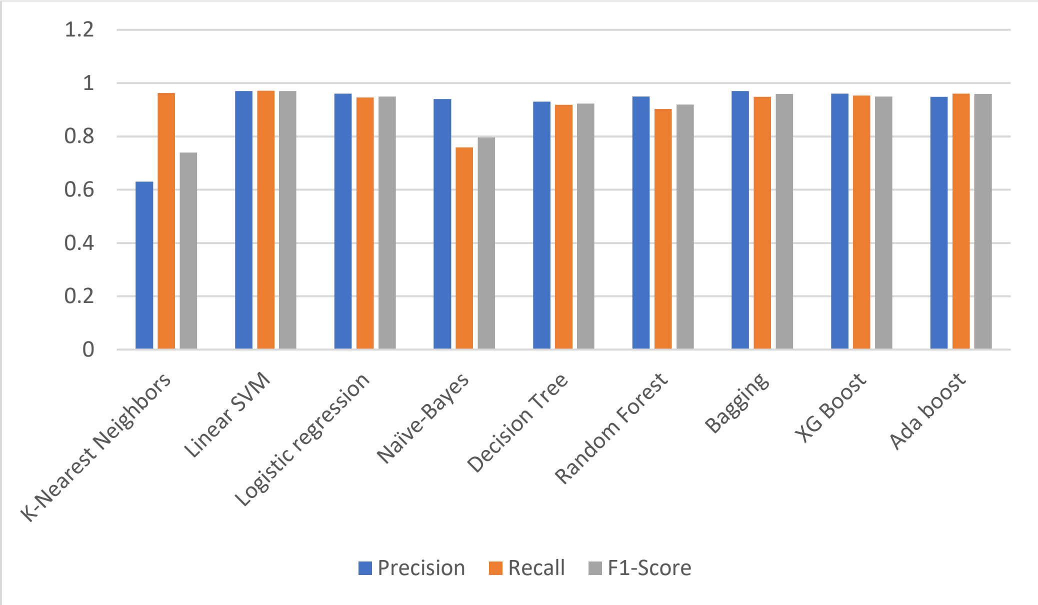

Figure 3: Overall Precision, Recall, and F1-Score over all datasets

5. Conclusion

Fake news identification and detection, when performed manually, require a lot of detailed knowledge of the field and proficiency to recognize abnormalities in the news article. This work tries to consider the problem of identifying fake news using machine learning algorithms. We used data from Kaggle and the ISOT Research Lab for our work, which contained news articles from various fields, to deal with the majority of the news and not any specific news. The research's main goal is to find textual characteristics that distinguish genuine news from fraudulent news. The different features required for classification were identified using the TF-IDF vectorizer, and the identified feature set was used as input to the machine learning models. Comparatively speaking, some models achieved higher accuracy than others. Results were compared using different performance metrics such as precision, recall, and f1-score for each algorithm. On overall performance, ensemble learning algorithms achieved a much better score than individual learning algorithms.

Fake news identification currently has many problems that require attention from research teams. Machine learning can be used to identify the causes of the spread of unreliable news. Future work can be done to identify fake news in live news articles and stop the spread of fake news.

6. References

[1] Z. Shahbazi and Y. C. Byun, "Fake Media Detection Based on Natural Language Processing and Blockchain Approaches," *IEEE Access*, vol. 9, pp. 128442–128453, 2021, doi: 10.1109/ACCESS.2021.3112607.

[2] M. F. Mridha, A. J. Keya, M. A. Hamid, M. M. Monowar, and M. S. Rahman, "A Comprehensive Review on Fake News Detection with Deep Learning," *IEEE Access*, vol. 9, pp. 156151–156170, 2021, doi: 10.1109/ACCESS.2021.3129329.

[3] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, "A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media," *IEEE Access*, vol. 10, pp. 62097–62109, 2022, doi: 10.1109/ACCESS.2022.3181184.

[4] T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, “A Novel Stacking Approach for Accurate Detection of Fake News,” *IEEE Access*, vol. 9, pp. 22626–22639, 2021, doi: 10.1109/ACCESS.2021.3056079.

[5] D. Rohera *et al.*, “A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects,” *IEEE Access*, vol. 10, pp. 30367–30394, 2022, doi: 10.1109/ACCESS.2022.3159651.

[6] A. Tariq, A. Mehmood, M. Elhadef, and M. U. G. Khan, “Adversarial Training for Fake News Classification,” *IEEE Access*, vol. 10, pp. 82706–82715, 2022, doi: 10.1109/ACCESS.2022.3195030.

[7] W. Shishah, “JointBert for Detecting Arabic Fake News,” *IEEE Access*, vol. 10, pp. 71951–71960, 2022, doi: 10.1109/ACCESS.2022.3185083.

[8] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, “Fake news stance detection using deep learning architecture (CNN-LSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.

[9] H. Ali *et al.*, “Analyzing the Robustness of Fake-news Detectors under Black-box Adversarial Attacks,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3085875.

[10] M. Carter, M. Tsikerdekis, and S. Zeadally, “Approaches for Fake Content Detection: Strengths and Weaknesses to Adversarial Attacks,” *IEEE Internet Comput*, vol. 25, no. 2, pp. 73–83, Mar. 2021, doi: 10.1109/MIC.2020.3032323.

[11] K. A. Qureshi, R. A. S. Malick, M. Sabih, and H. Cherifi, “Complex Network and Source Inspired COVID-19 Fake News Classification on Twitter,” *IEEE Access*, vol. 9, pp. 139636–139656, 2021, doi: 10.1109/ACCESS.2021.3119404.

[12] E. Elsaeed, O. Ouda, M. M. Elmogy, A. Atwan, and E. El-Daydamony, “Detecting Fake News in Social Media Using Voting Classifier,” *IEEE Access*, vol. 9, pp. 161909–161925, 2021, doi: 10.1109/ACCESS.2021.3132022.

[13] K. Xu, F. Wang, H. Wang, and B. Yang, “Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding,” 2020.

[14] Ravish, R. Katarya, D. Dahiya, and S. Checker, “Fake News Detection System Using Featured-Based Optimized MSVM Classification,” *IEEE Access*, vol. 10, pp. 113184–113199, Nov. 2022, doi: 10.1109/access.2022.3216892.

[15] M. Sudhakar and K. P. Kaliyamurthie, “A Machine Learning Framework for Automatic Fake News Detection in Indian News,” 2022, doi: 10.21203/rs.3.rs-2268597/v1.

[16] S. Lorent and A. Itoo, “Fake News Detection Using Machine Learning,” 2018.

[17] A. Preda, S. Ruseti, S.-M. Terian, and M. Dascălu, “Romanian Fake News Identification using Language Models,” in *RoCHI - International Conference on Human-Computer Interaction*, 2022, pp. 73–79. doi: 10.37789/rochi.2022.1.1.13.

[18] D. Dementieva, M. Kuimov, and A. Panchenko, “Multiverse: Multilingual Evidence for Fake News Detection,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.14279

[19] D. Li, H. Guo, Z. Wang, and Z. Zheng, “Unsupervised Fake News Detection Based on Autoencoder,” *IEEE Access*, vol. 9, pp. 29356–29365, 2021, doi: 10.1109/ACCESS.2021.3058809.

[20] “Kaggle: Your Home for Data Science.” https://www.kaggle.com/ (accessed Nov. 23, 2022).

[21] “Fake News Detection Datasets - University of Victoria.” https://www.uvic.ca/ecs/ece/isot/datasets/fake-news/index.php (accessed Nov. 23, 2022).

[22] “NLTK :: Natural Language Toolkit.” https://www.nltk.org/ (accessed Nov. 23, 2022).

[23] “sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.1.3 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (accessed Nov. 23, 2022).

[24] "Fake News | Kaggle." https://www.kaggle.com/competitions/fake-news/overview (accessed Nov. 23, 2022).

[25] "Fake News detection | Kaggle." https://www.kaggle.com/datasets/jruvika/fake-news-detection (accessed Nov. 24, 2022).

[26] E. Loper and S. Bird, "NLTK," 2002, pp. 63–70. doi: 10.3115/1118108.1118117.

[27] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int J Comput Appl*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[28] C. Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *Journal of Educational Research*, vol. 96, no. 1, pp. 3–14, 2002, doi: 10.1080/00220670209598786.

[29] P. Hilaire Torr, ubor Ladick, and P. H. Torr, "Linear Support Vector Machines Learning Lightweight Neural Networks: Pruning and Quantization View project Continual Learning for Scene Understanding Agents View project Locally Linear Support Vector Machines," 2011. [Online]. Available: https://www.researchgate.net/publication/221345963

[30] G. Guo, H. Wang, D. A. Bell, Y. Bi, D. Bell, and K. Greer, "KNN Model-Based Approach in Classification," 2004. [Online]. Available: https://www.researchgate.net/publication/2948052

[31] P. Kaviani, B. Vidyapeeth, and M. S. Dhotre, "Short Survey on Naive Bayes Algorithm Study of Scheduling Algorithm to assess the performance of a processess in Operating Systems View project E-Learning System View project Short Survey on Naive Bayes Algorithm," *International Journal of Advance Engineering and Research Development*, vol. 4, no. 11, 2017, [Online]. Available: https://www.researchgate.net/publication/323946641

[32] H. H. Patel and P. Prajapati, "Study and Analysis of Decision Tree Based Classification Algorithms," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74–78, Oct. 2018, doi: 10.26438/ijcse/v6i10.7478.

[33] G. Biau and G. B. Fr, "Analysis of a Random Forests Model," 2012.

[34] C. Tu, H. Liu, and B. Xu, "AdaBoost typical Algorithm and its application research," in *MATEC Web of Conferences*, Dec. 2017, vol. 139. doi: 10.1051/matecconf/201713900222.

[35] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794. doi: 10.1145/2939672.2939785.