Performance Evaluation of Fake News Detection Using Artificial Intelligence Techniques

Syed Muhammad Hamza[1], Muhammad Hussain[2], Taimoor Zafar[3], Muhammad Zohaib Sohail[3], Tarique Aziz[4], Qamar Ud Din Memon[2]
[1]Department of Computer Science, NFC IEFR, FSD.
[2]Department of Software Engineering, Bahria University, Karachi Campus, Karachi
[3]Department of Electrical Engineering, Bahria University, Karachi Campus, Karachi
[4]Department of Electrical Engineering, Sir Syed University, Karachi
***Correspondence:** Qamar Ud Din Memon and qamaruddin.bukc@bahria.edu.pk
Citation | Hamza. S. M, Hussain. M, Zafar. T, Sohail. M. Z, Aziz. T, Memon. Q. U. D, "Performance Evaluation of Fake News Detection Using Artificial Intelligence Techniques", IJIST, Vol. 6 Issue. 2 pp 739-753, June 2024
DOI | https://doi.org/10.33411/ijist/202462739753
Received | May 26, 2024 **Revised |** June 13, 2024 **Accepted |** June 20, 2024 **Published |** June 25, 2024.

Introduction/Importance of Study: As the proliferation of fake news poses significant challenges to traditional fact-checking methods, there is a growing need for robust and automated approaches to combat misinformation.
Novelty statement: This study presents a comprehensive evaluation of artificial models for fake news detection, offering insights into their effectiveness and applicability in addressing the contemporary issue of misinformation.
Material and Method: The research employs various artificial algorithms, including logistic regression, gradient boosting, decision trees, random forest, AdaBoost, passive aggressive classification, XGBoost, naive Bayes, and support vector machines (SVM), to train datasets and evaluate the performance of each model.
Result and Discussion: Through rigorous evaluation, the study finds that XGBoost and AdaBoost classifiers exhibit the highest accuracy rates of 99.83% and 99.77%, respectively, in detecting fake news. Decision Tree, Support Vector Machine, and Gradient Boosting classifiers also demonstrate commendable performance. Conversely, the Naive Bayes classifier exhibits the lowest accuracy, suggesting its limitations in fake news detection.
Concluding Remarks: This research underscores the significance of ensemble methods such as XGBoost and AdaBoost in effectively identifying fake news, laying the groundwork for future advancements in combatting misinformation.
Keywords: Techniques, TD-IDF, Features, Artificial Techniques, True and Fake News.

Introduction:

In the current era, newspapers are being replaced by social media platforms such as Facebook, Twitter, TikTok, and many others. This shift is driven by the development of the World Wide Web and high-speed internet. Users find these platforms convenient and user- friendly, allowing information to spread rapidly, facilitating idea sharing, and garnering millions of views. Moreover, users can access the most recent information at their fingertips through these platforms. Currently, approximately 70% of information is sourced from social media tools like Twitter and Facebook [1]. During the COVID-19 pandemic, the spread of false information about the number of cases, its effects, and its transmission was notable. Similarly, during the 2016 US election, false information spread significantly. For instance, in 2012, only 49% of Americans reported seeing news on social media, but by 2016, this number had risen to 62% [2]. The coronavirus pandemic is a recent example where false information about the nature, origin, and characteristics of the virus spread widely on the internet. False information can take two forms: misinformation (incorrect information) and disinformation (deliberately misleading information). For example, a user might repost a link to a news story with a self-created headline, or the news piece itself might be entirely fabricated [3]. In Malaysia, the government introduced the Anti-Fake News Act (AFNA), which defines fabricated news content through its characteristics, audio, visual elements, or any form on social media platforms that convey suggestions or textual content [4]. Furthermore, WhatsApp reports that two million accounts are deleted every month due to the spread of fake news [5]. Abdelminaam et al. [6] reported an artificial model for detecting fake news related to COVID-19 using deep learning models, random forest, decision tree, and linear and logistic regression models.

Producing social media news quickly and in large quantities presents many challenges. A huge amount of data is generated every second, requiring substantial computing resources and efficient algorithms. Additionally, the data comes in various formats, such as audio, video, images, and text, making it difficult to use a single approach. The high speed of the data stream need real-time processing. On the other hand, Reliability is also a concern, as the data includes noise, misinformation, and other elements that make reliable filtering difficult. To address these problems, big data technologies like Apache Spark and MySQL databases are commonly used to process large amounts of data. These techniques require massive parallel processing and cloud computing resources. Further, Natural Language Processing (NLP) is also used to analyze the data. Also, Other AI-based techniques are employed to predict these data patterns. The issue of fake news is a complex problem that intersects with various domains, including journalism, psychology, sociology, and computer science. Traditional methods of fact-checking and source verification are often insufficient to keep pace with the sheer volume and speed at which fake news spreads across social media and online platforms [7]. Consequently, researchers have turned to artificial intelligence techniques to detect misinformation more effectively [1][2].

Objectives:

This research aims to apply machine learning techniques to detect false news. By using these algorithms, the goal is to improve understanding of the methods, challenges, and future developments related to addressing the growing problem of fake news. In this paper, we propose to evaluating the performance of various algorithms, such as support vector machines, naïve Bayes, XGBoost, random forest, gradient boosting, logistic regression, and decision trees, using majority voting to detect false news. Additionally, we aim to clarify the individual roles of these algorithms in the overall goal of identifying fake news. The paper offers a comprehensive exploration of machine learning algorithms utilized in fake news detection, delving into their underlying mathematical and conceptual frameworks. It elaborates on how these algorithms process and analyzes textual data to formulate predictions, providing a thorough understanding of their functionality. The primary objective of this research is to delineate the strengths and weaknesses of each algorithm under scrutiny. Additionally, for accuracy of model we evaluate

accuracy; precision, recall, and F1 score to assess the efficiency of these algorithms. The paper aims to facilitate the dissemination of information about tools that provide reliable and accurate information in our society.

Methodology and Techniques:

Flow of Methodology:

The proposed framework for detecting fake news using machine learning encompasses several crucial steps aimed at effectively distinguishing between fake and real news data. The first step involves data collection, where a diverse and representative dataset of labeled news articles is gathered. These datasets are collected in the form of text from various platforms, including fact-checking websites and other reliable sources such as Kaggle [8], the University of Victoria, Fact Checker, Politifact, and Factcheck.org. Following this, the data undergoes pre-processing to cleanse the text, ensuring coherence by converting it into words or sentences and eliminating noise and irrelevant information. Further, Feature extraction is performed to convert the text data into numerical vectors using technique such as term frequency-inverse document frequency (TF-IDF) [9], capturing relevant features for analysis. Each news article is treated as possessing distinct features, which are subsequently inputted into artificial models, with each model elaborated upon in the subsections of this section. Data Collection

Proposed Framework:

Before applying the proposed model to classify real and fake news, a labeled dataset is necessary. Collect over 44,000 news articles from reliable sources like Kaggle [9], ensuring a mix of real real (47.7 %) and fake (52.2 %) articles. Preprocess the data by cleaning, removing duplicates, and normalizing the text. Convert the text into numerical vectors using techniques like TF-IDF. The dataset was divided into training and testing sets. Initially, the models were trained using the training set, and their performance was evaluated using various metrics such as accuracy, recall, precision and F1-score discussed in subsequent section. After training, unknown data from the testing set was transformed into features and given as input to determine the status of the news. If performance criteria are met, stop; otherwise, update and optimize the parameters. This procedure enables real-time classification of news articles, emphasizing continuous model improvement through updates with new data and refinements based on user feedback and new research findings. The procedure for predicting whether an unknown news article is fake or real is summarized in Figure 2.

Data Preparation:

Social media data presents a significant challenge due to its highly unstructured nature, characterized by informal communication, including typos, slang, and poor grammar. Improving performance and reliability necessitates developing techniques that leverage resources to make informed decisions [8]. Before employing the data for predictive modeling, it is essential to clean it to yield more meaningful results for machine learning algorithms. This involves identifying and removing stop words, commonly occurring words like "the," "a," and "on", during index construction. Comprehensive lists of such words can be readily sourced from various websites. In this research, data preparation involves extracting the title and content of the news and categorized data into true or fake news as shown in Figure 1. The first preprocessing technique is tokenization, which divides the text into individual words. Subsequently, a word reduction approach is used, such as removing prepositions and pronouns, followed by the removal of punctuation and special characters. [10]. The final step involves obtaining keywords and transforming them into vectors, making them suitable for use in machine learning algorithms. This typically includes using techniques like word embeddings (e.g., Word2Vec, GloVe) or other vectorization methods (e.g., TF-IDF) to convert the processed text into numerical representations that can be used for predictive modeling.

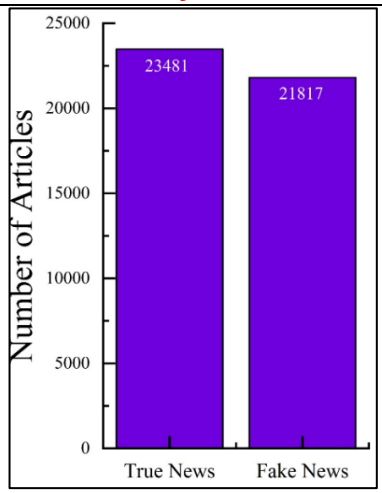

Figure 1: Distribution of True and Fake News.

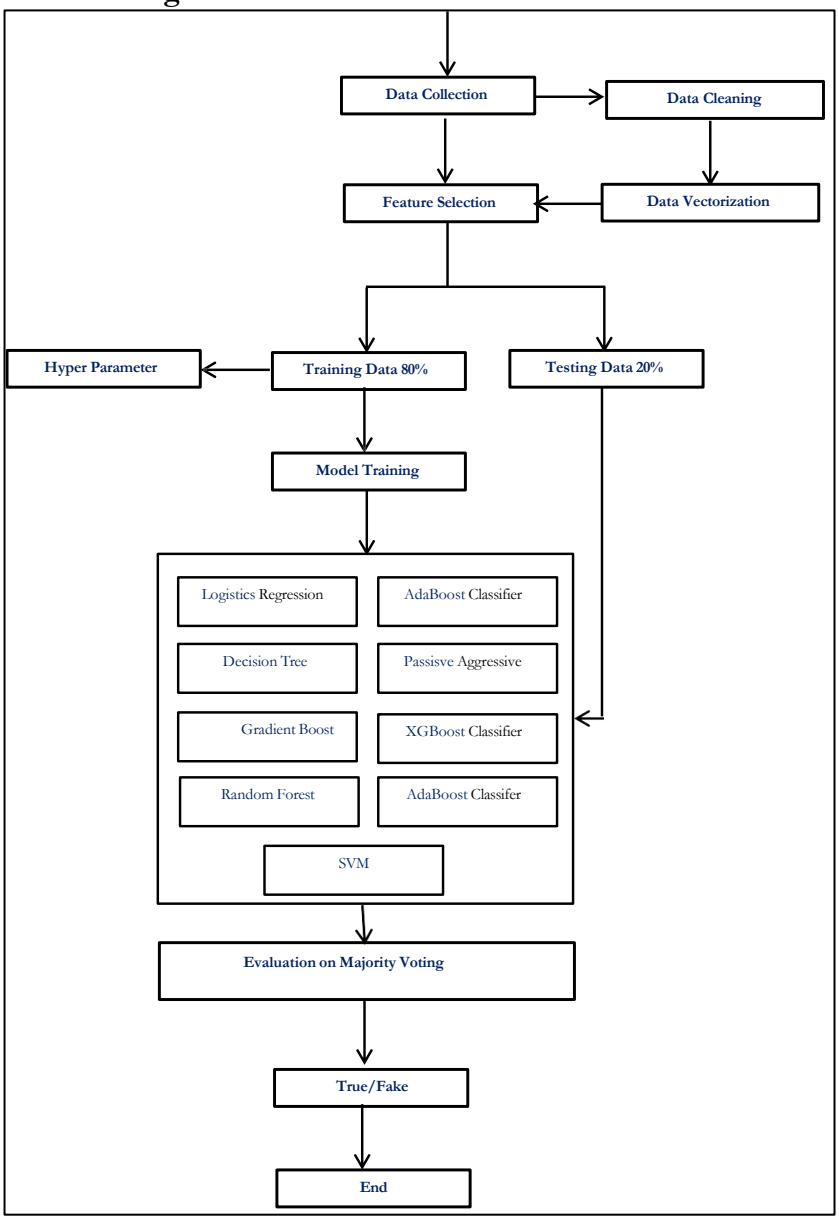

Figure 2: Performance Evaluation through majority voting using Artificial Intelligence.

Features Extraction: To identify whether a news article is real or fake, each model was fed with

specific input parameters. These parameters typically include various features that represent the characteristics of the text. Here are the input parameters for each model:

TF-IDF (Term Frequency-Inverse Document Frequency): Used to measure the importance of words within a document relative to a corpus. TF-IDF scores highlight significant words while reducing the weight of common words.

Logistic Regression: Numerical vectors representing the relationship between the input features (likely derived from TF-IDF scores) and the binary target variable (real or fake news).

Decision Tree: Features such as word count and average sentence length, among possibly other text-based metrics.

Gradient Boosting Classifier: TF-IDF scores and additional feature parameters derived from the text.

Random Forest: TF-IDF scores and other features such as word counts and sentiment scores.

AdaBoost Classifier: Initially equal weights for all training instances, adjusted weights based on misclassified instances, and features like TF-IDF scores and word counts.

Passive Aggressive Classifier: TF-IDF scores and other textual features that are dynamically adjusted during training with new data.

XGBoost Classifier: Residuals of TF-IDF scores and other features, with adjustments to correct errors in previous iterations.

Naïve Bayes: Probabilities of words (features) occurring in the class (real or fake) based on word frequency distributions.

Support Vector Machine (SVM): High-dimensional feature vectors representing textual content, such as TF-IDF scores.

The frequency of words in a document is a crucial statistic used for information acquisition. It helps increase the percentage of significant words appearing in the text and, in some situations, reveals common words typically found in news articles [11]. Since machines can only understand binary numbers, encoding is required to convert words into digital vectors, as discussed in an earlier section. Fake news content often masquerades as real news, making it difficult to distinguish between the two. The basic characteristics of real news content are complex. To address this problem, researchers use TF-IDF techniques to identify important features in media content. This is vital for storing information. According to Wu et al. [8][12], extracting valuable features from real news content is challenging due to the ability of fake news content to mimic genuine news.

$$W_{x,} = tf_{x,y} \times \log\left(\frac{N}{df}\right) \quad (1)\ x$$

Where, Term x within document y, $tf_{x,y}$ = frequency of x in y, df_x = number of documents containing x, and N is total number of documents. The above equation is used to count the importance of words in a document relative to a corpus. It aims to highlight words that are significant within a document while reducing the weight of common words such as a, an, or and etc. TF measures the frequency of a term within a document, while IDF measures its importance across the corpus. The TF-IDF score for a term in a document is the product of its TF and IDF scores. This numerical representation helps in text mining tasks like classification and information retrieval, enhancing the analysis of textual data.

Artificial Intelligence Technique:

This paper presents several supervised learning techniques developed to predict whether news articles are real or fake. Researcher reported many algorithms to detect false information [6], however here; we evaluated performance of each model. Initially, it requires selecting appropriate artificial intelligence architecture, and defining news feature parameters along with their associated characteristics. Once the model is designed, the accuracy of the model can be assessed based on known characteristics of the news articles. After evaluating the error between known and predicted data, the techniques are tuned to achieve an optimized solution. During the training

phase, the models learn to identify patterns in the text data that distinguish between fake and real news. Finally, when an optimal solution is found, the trained model can be used to predict the category of new, unseen news articles as either fake or real based on given features, the procedural steps are given in Figure 2.

Logistic Regression:

Logistic regression is a widely used supervised learning algorithm applicable to binary classification problems, making it suitable for fake news detection. The objective of this study is to determine whether a given news article is real or fake based on its content. This algorithm operates by learning the relationship between the input features (typically represented as numerical vectors) and the binary target variable (real or fake news) from a labeled dataset [13]. It functions by modeling the probability that a given news article belongs to a certain class, which in this case could be "fake" or "real" news. The logistic function (also known as the sigmoid function) is employed to transform the output of a linear equation into a value between 0 and 1, representing a probability. The formula for the logistic function is:

Equations:

$$P\,(y = 1 \mid x) = \frac{1}{1 + e^{-\beta}} \qquad (2)$$

Where: P (y = 1 | x) is the probability of the news, e is a natural logarithm and finally β is the linear combination of the input features and their corresponding weights which can be represented as of logistic function:

$$\beta = a_0 + a_1X_1 + a_2X_2 + \cdots + a_nX_n \qquad (3)$$

Where: a_n is associated weight of corresponding features, X_i, are the input parameter represented as feature vector. This technique learns the optimal weights of the training dataset through the gradient descent algorithm. Once the desired accuracy is achieved, the weights obtained at that accuracy level are used for the testing data to compute the probability. If the probability meets a threshold value, it is interpreted as real news; otherwise, it is classified as false news.

Decision Tree:

This technique operates recursively, splitting the feature space into subspaces represented as the root and child nodes of a tree, also referred to as data classes. Each class of data pertains to the context of real or fake news detection. Once the tree is constructed, the leaf nodes represent the final decisions or classifications of fake or real news based on the majority class of the training samples in that node. Consider a scenario with two features: X_1 representing word count and X_2 representing average sentence length. Based on these features, a tree will be developed. This technique selects the feature that divides the data to meet certain criteria, such as Gini impurity [14]. The chosen feature is then used as a decision node, and the data is divided into subsets accordingly. This procedure is repeated recursively until it reaches a leaf node, at which point the recursion stops.

Gradient Boosting Classifiers:

In this algorithm, equal weights are initially assigned to all samples, and the training process begins. When samples are predicted incorrectly, they are known as weak learners, and their weights are gradually adjusted. This process continues until the desired accuracy is achieved. In the context of fake news detection, a gradient-boosting classifier can learn to make decisions about whether an article is fake or real based on certain feature parameters [15]. Initially, the model acts as a weak learner on the training dataset. An error is calculated by subtracting the actual and predicted data of true labels from the training dataset. The goal is to minimize the error of those samples that were not predicted well. In the next iteration, the weights of the incorrectly predicted samples are adjusted, giving them more importance in the training process.

In the next iteration, the weights of the weak learner are updated by scaling them with a certain factor, known as the learning rate. This process repeats until the defined number of

iterations is reached or until it meets convergence criteria. In each iteration, weak models are trained to achieve high accuracy between predicted and actual values. The final prediction of the Gradient Boosting Classifier is obtained by summing the predictions of all weak learners in the ensemble.

Example Formulation:

Assuming we have a simple dataset with a single feature x represent word count and binary labels y

- Initialize model: $F_0(x) = 0$
- Compute residuals: $r_0 = y - F_0(x)$
- Fit new weak learner to residuals: $h_1(x)$
- Update model: $F_1(x) = F_0(x) + learning\ rate * h_1(x)$
- Compute new residuals: $r_1 = y - F_1(x)$
- Iterate: Fit subsequent weak learners and update the model using the residuals.
- Final prediction: $F(x) = F_T(x) = F_{T-1}(x) + learning\ rate * h_T(x)$

This technique gradually improves the accuracy in each iteration by minimizing the error of weak trained models to make strong mode.

Random Forest:

Random Forest (RF) is a type of ensemble learning technique in which multiple decision trees are trained on different subsets of data. Their predictions are combined to make a final decision about whether the news is false or true. A majority vote among the trees determines the final prediction. The strength of RF lies in its low error rate, which outperforms other models. This improvement can be attributed to the low correlation among the constituent trees [16].

This technique randomly selects subsets (with replacement) of the original training data for each decision tree, known as bootstrap sampling. Each bootstrap sample is used to grow a decision tree. Each tree is grown deep, without pruning. Each decision tree independently predicts the label (fake/real) of a news article. The final prediction is determined through a majority vote (for classification) or averaging (for regression) of the individual tree predictions.

Example Formulation:

Assuming we have a dataset with features $x_0, x_1, x_2, \ldots\ldots x_n$ (e.g., word counts, sentiment scores) and binary labels y (0 for Fake News, 1 for Real News). For a new news article with feature values $x = x_1, x_2, \ldots\ldots x_n$:

- Pass x through each decision tree to get individual predictions $y_1, y_2, \ldots, y_T$.
- Aggregate the predictions through majority voting (for classification) or averaging (for regression) to obtain the final prediction y for the Random Forest.

Random Forest leverages the diversity of individual decision trees to reduce over fitting and improve generalization. By combining the predictions of multiple trees, it provides more robust and accurate results compared to a single decision tree.

ADA Boost Classifier:

It is an ensemble learning method that combines several weak learners, typically decision trees, to construct a more robust model. This algorithm assigns equal weights $w_i = \frac{1}{N}$ to all training instances i and number of instances N. Train a weak learner h_t (e.g., a decision tree) on the training data with the assigned weights w_i. The weak learner aims to minimize the weighted classification error. It focuses on instances that were previously misclassified [18].

It calculates the weighted error e_t and alpha ($a_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t})$) of the weak learner's vote in the final prediction. High accuracy leads to a lower weight. The predictions of all weak learners are combined by weighting them based on α. The final prediction is based on the weighted sum of weak learner predictions. Update the weights w_i to give more importance to misclassified instances:

$$w_i = w_i \cdot e^{-\alpha t y_i h t x_i} \quad (4)$$

Where y_i is the true label of instance i and $h(x_i)$ is the prediction of h_t. The final prediction is determined by combining the weak learner predictions through weighted majority voting based on their respective α\alphaα values [17]:

$$(x) = (\Sigma^T a_t h(x)) \quad t=1 \quad (5)$$

Where T is the total number of weak learners?

Passive Aggressive Classifier:

The Passive Aggressive Classifier is a machine learning algorithm particularly useful for online learning tasks where new data arrives continuously. It aims to make correct predictions while keeping the model's parameters as unchanged as possible. In the context of fake news detection, the Passive Aggressive Classifier works by adapting to new information while making minimal adjustments to its existing knowledge [18]. We have a dataset with features $x_1, x_2, \ldots, x_n$ e.g., word counts, sentiment scores and binary labels y for fake news and real news. The algorithm initializes the weight vector w and bias term b and receive a new instance x with features and its true label y, compute the prediction

$$y = (w \cdot x + b) \quad (5)$$

Calculate the hinge loss:

$$hinge\ loss = (0, 1 - y \cdot (y + b)) \quad (6)$$

Update parameters using a learning rate η:

$$w \rightarrow w + \eta \cdot y \cdot x \quad (7)$$

$$b \rightarrow b + \eta \cdot y \quad (8)$$

The Passive Aggressive Classifier dynamically adjusts its parameters to learn from new data while aiming to maintain stability in its existing knowledge. It strikes a balance between making conservative updates for correct predictions and more aggressive updates for incorrect predictions, facilitating efficient adaptation to evolving information streams.

XGBoost Classifier:

Extreme Gradient Boosting (XGBoost) is an ensemble learning algorithm particularly effective for classification tasks like fake news detection. XGBoost iteratively builds new trees to correct the errors of the previous ones, ultimately producing a powerful ensemble model [19]. The working principles of this classifier are based on the assumption that we have N instances with M features in our dataset. Initialize the initial predictions for all instances as $F_0(x_i)$, usually set to the mean of the target labels. Train a decision tree $h_1(x)$ to predict residuals:

$$h_1(x_i) = y_i - F_0(r_i) \quad (9)$$

The residuals of each iteration are calculated where yi are actual and F0 are prediction of ensemble model

$$F_1(x_i) = F_0(x_i) - h_1(x_i) \quad (10)$$

For each iteration t, train a decision tree $h_1(x)$ predict the negative gradient of loss function with respect to current prediction

$$h_t(x_i) = \frac{L(y_i, F_{t-1}(x_i))}{F_{t-1}(x_i)} \quad (11)$$

Update the predictions:

$$F_t(x_i) = F_{t-1}(x_i) + \eta \cdot h(x_i) \quad (12)$$

Where η is learning rate, the final ensemble prediction is given by $F_t(x_i)$, where t is the total number of iterations (trees). XGBoost uses a combination of regularization techniques, gradient boosting, and a novel split-finding algorithm to improve model performance and prevent over fitting. This algorithm stands out among boosting algorithms for several reasons, making it a preferred choice for tasks like fake news detection. Compared to other boosting algorithms like AdaBoost and Gradient Boosting Classifier (GBC), it offers superior

performance and scalability. Its efficient split-finding algorithm and regularization techniques prevent over fitting, ensuring robust generalization to new data. Also, it provides better handling of missing values and is less prone to bias due to its use of differentiable convex loss functions. These advantages get higher accuracy and reliability in fake news detection tasks.

Naïve Bayes:

This algorithm can quickly and accurately make predictions because it is based on the idea that the features of the input data are conditionally independent given the class. Naive Bayes classifiers are straightforward probabilistic input data are independent of one another as assumed by the Naive Bayes classifier. In complex scenarios where the data distribution is not well-defined, the classifier's performance can be improved by estimating the probability density function of the input data using a kernel function. Consequently, the Naive Bayes classifier is an effective machine learning tool, especially for text classification, spam filtering, and sentiment analysis, among other applications.

$$P(C|X) = \frac{P(X|C)P(C)}{p(X)} \quad (13)$$

Consider C as the class labels and x is a feature vector of news represented by a set of words X1, X2, … , Xn. P(C): Prior probability of class C, calculated as the percentage of occurrences in class C in the training data. P(Xi | C) is the probability of observing word Xi given class C. It is calculated as the frequency of xi in instances of class C divided by the total number of words in instances of class C. P(C | X): Posterior probability of class given the news article, P(C | X): calculated for each class using Bayes' theorem and class with the highest P(C | X) is the predicted class for the news article.

Support Vector Machine (SVM)

It is a robust classification technique used in various applications, including detecting whether an article is fake or real. This algorithm creates a hyper plane between different classes and separates data points of different classes within a high-dimensional space. It is used in binary classification problems and offers a wide variety of kernel functions [20] to establish a hyper plane or decision boundary based on the given features [21]. Training dataset $\{(x_i, y_i)\}$, where x_i is the feature vector of the I[th] news article and y_i is the corresponding class label 1 for real news, -1 for fake news, the objective of SVM is to find the weight vector w and bias term b that minimize the following equation:

$$min_{w,} \quad \frac{1}{-} \| w \| 2^{-} + C \sum_{n=1}^{N} (0,1 - yi\ (w \cdot xi + b)) \quad (14)$$

Here:

- The C serves as the regularization parameter in support vector machine, which have a capable to maximize the margin distance and reduce the classification error.
- The part of above equation $(0, 1 - yi\ (w \cdot xi + b))$ is represented as hinge loss, which measures distance of a different data point from the margin.

The research article explores various machine learning techniques, this study discuss over all summaries of all techniques highlighting their working principles and relevance to fake news detection. Logistic regression models the probability of news being real or fake using the logistic function and gradient descent to optimize features weights, making it effective for binary classification. Decision trees recursively split the feature space, constructing a hierarchical model that classifies news based on learned patterns. Gradient boosting classifiers iteratively adjust weights of misclassified samples, enhancing the model's robustness by combining weak learners, which is crucial for improving accuracy in detecting fake news. Random forests, by training multiple decision trees on different data subsets and combining their predictions through majority voting, reduce overfitting and improve generalization. AdaBoost focuses on misclassified instances, adjusting weights iteratively to emphasize challenging samples, thus increasing detection

performance. The passive aggressive classifier is suitable for online learning, making minimal adjustments to adapt to new data, ensuring timely and accurate fake news detection. XGBoost employs efficient gradient boosting with regularization to prevent overfitting, making it highly effective for classification tasks. Naive Bayes, assuming feature independence, uses Bayes' theorem for quick and accurate predictions, ideal for text-based fake news detection. Lastly, SVMs create a hyperplane to separate classes in high-dimensional space, offering robust classification capabilities for distinguishing fake from real news.

In machine learning, we often train complex models that require significant time and resources to build. To avoid retraining these models every time we want to use them, we can save them to a file and load them later when needed. Pickling is the process of converting Python objects, such as trained machine learning models, into a stream of bytes that can be written to a file. These files can then be easily transported or stored, and the objects can be reconstructed from the bytes at any time. This is particularly useful for machine learning models, as training can be time-consuming and resource-intensive. Being able to save and reuse trained models as needed is highly efficient.

Validation and Cross Validation:

To ensure robust model performance, the study likely employed cross-validation or validation techniques. Cross-validation divides the dataset into multiple subsets, using one subset for testing and the others for training iteratively. This technique helps assess the model's performance across different data partitions, reducing the risk of over fitting. Additionally, validation techniques like holdout validation or k-fold cross-validation are likely utilized. Holdout validation reserves a portion of the dataset for validation, while k-fold cross-validation divides the data into k subsets, using each subset for testing and the rest for training. These methods help evaluate the model's generalization ability and optimize hyper parameters for better performance.

Performance Metrics:

It is compulsory to assess the model's performance using its performance metrics which include accuracy, precision, recall and F1 score. The number of times our actual positive values match the expected positive values is known as the true positive or TP. TN stands for True Negative which is the proportion of times our actual and predicted negative values match. False Positive or FP is the quantity of times our model incorrectly forecasts negative values as positive ones. FN stands for False Negative which is the quantity of times our model interprets negative values as positives [22].

Accuracy:

Accuracy is often the most commonly used metric, representing the percentage of correctly predicted observations, both true positives and true negatives. A high accuracy value indicates a well-performing model. To calculate the accuracy of model performance, the following equation can be used:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (15)$$

Recall:

The recall represents the total number of positive classifications out of the true class. In our case, it represents the number of articles predicted as true out of the total number of true articles.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (16)$$

Precision:

Conversely, the precision score represents the ratio of true positives to all events predicted as true. In our case, precision shows the number of articles that are marked as true out of all the positively predicted (true) articles.

$$\text{Presision} = \frac{\text{TP}}{\text{TP+FP}} \quad (17)$$

F1-Score:

F1-score represents the trade-off between precision and recall. It calculates the harmonic meaning between each of the two. Thus, it takes both the false positive and the false negative observations into account. F1-score can be calculated using the following formula:

$$F1 - Score = 2\frac{Precision \times Recall}{Precision + Recall} \quad (18)$$

These metrics are not only applicable in fake news detection but also find utility in other domains such as antivirus software and email filtering. In the context of email filtering, for instance, binary classification techniques are commonly employed to determine whether emails are classified as spam or real. Similar way, these systems analyze various features of incoming emails to predict their classification. By employing machine learning algorithms, such as logistic regression or decision trees, these systems can effectively differentiate between spam and non-spam emails, thereby enhancing user experience and security. These applications highlight the flexibility and importance of utilizing these metrics to tackle various challenges in different fields.

For Examples, TP:

Predict correctly email have spam (350), FP: Predict incorrectly email have spam but actually have not (100), TN: Predict correctly that email have not spam (850), FN: Predict incorrectly that emails have not spammed actually it have. (200). When apply model then we find this metrics score and efficiency of the trained model Using equation of Accuracy 80%, using equation of precision we get 77.8 %, Recall factor was 63.6% and f1 score 70%.

Result and Discussion:

The research article explores various machine learning techniques, discussing their summaries and relevance to fake news detection. Logistic regression models the probability of news being real or fake using the logistic function and gradient descent to optimize feature weights, making it effective for binary classification. Decision trees recursively split the feature space, constructing a hierarchical model that classifies news based on learned patterns. Gradient boosting classifiers iteratively adjust weights of misclassified samples, enhancing the model's robustness by combining weak learners, crucial for improving accuracy in detecting fake news. Random forests train multiple decision trees on different data subsets and combine their predictions through majority voting, reducing overfitting and improving generalization. AdaBoost focuses on misclassified instances, iteratively adjusting weights to emphasize challenging samples, thus increasing detection performance. The Passive Aggressive Classifier is suitable for online learning, making minimal adjustments to adapt to new data, ensuring timely and accurate fake news detection. XGBoost employs efficient gradient boosting with regularization to prevent overfitting, making it highly effective for classification tasks. Naive Bayes, assuming feature independence, uses Bayes' theorem for quick and accurate predictions, ideal for text-based fake news detection. Lastly, SVMs create a hyperplane to separate classes in high-dimensional space, offering robust classification capabilities for distinguishing fake from real news. In this paper, we employed state-of-the-art artificial algorithms, including logistic regression, decision trees, gradient boosting, random forest, AdaBoost, passive-aggressive classification, XGBoost, Naive Bayes, and Support Vector Machines (SVM), to develop and evaluate the performance of the models for fake news detection.

Table 1: Performance metric evaluation of artificial techniques

S. #	Algorithm	Accuracy	Precision	F1 Score	Recall
1	Logistics Regression	98.91	98.89	98.80	98.78
2	Decision Tree	99.62	99.64	99.60	99.55
3	Gradient Boost	99.61	99.34	99.59	99.83
4	Random Forest	99.36	99.32	99.33	99.34
5	AdaBoost Classifier	99.77	99.32	99.76	99.88
6	Passisve Aggressive	99.57	99.59	99.55	99.53

7	XGBoost Classifier	99.83	99.74	99.82	99.9
8	Naïve Bayes	99.72	94.51	93.31	92.14
9	SVM	99.53	99.43	99.5	99.57

The accuracy performance of models are illustrated in Figure 3, Based on the accuracy results obtained from various classifiers used in the fake news detection, it can be concluded that the XGB Boost and AdaBoost classifiers exhibited the highest accuracy rates, achieving 99.83% and 99.77% accuracy, respectively. The Decision Tree classifier also performed exceptionally well, boasting an accuracy of 99.62%. Additionally, the Support Vector Machine and Gradient Boosting classifiers yielded commendable results, with accuracies of 99.53% and 99.61%, respectively. Conversely, the Naive Bayes classifier demonstrated the lowest accuracy at 93.72%, suggesting it may not be the optimal choice for fake news detection. Moreover, the Logistic Regression and Random Forest classifiers registered comparatively lower accuracies compared to the top-performing models, achieving 98.91% and 99.36%, respectively. In summary, the findings highlighted the useful ensemble methods such as XGB Boost and AdaBoost in identifying fake news, underscoring their importance in the development of robust fake news detection systems.

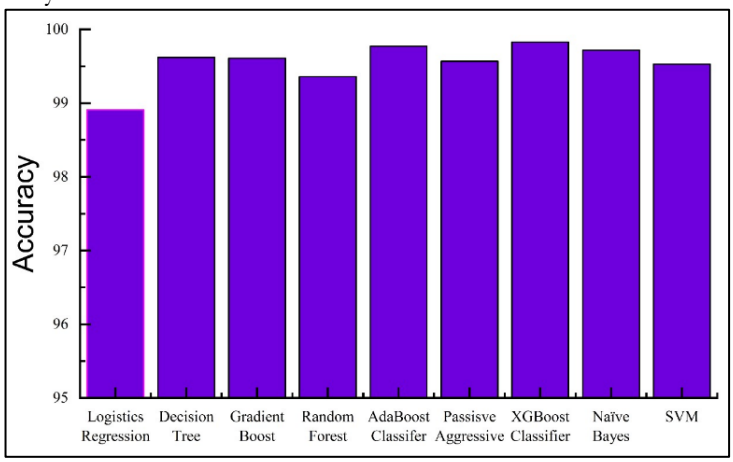

Figure 3: Accuracy Achieved through Artificial Intelligence Techniques.

For further validation, model accuracy was determined through performance metrics parameters using the expression in the preceding section. Based on the accuracy, precision, F1 score, and recall values, it is evident that all models performed reasonably well in detecting fake news. The Decision Tree model notably exhibited the highest precision and recall values, accompanied by a strong F1 score, indicating its adeptness in accurately classifying both positive and negative instances. Performance metrics values of each artificial model are evaluated and given in Table 1 and also pictorial representation are given in Figure 4. In summary, considering the precision, F1 score, and recall values, the Decision Tree, XGB Boost, and Logistic Regression models emerged as the top performers in detecting fake news within the provided dataset. Nevertheless, conducting further analysis and testing may be imperative to ascertain the most appropriate model for a particular use case.

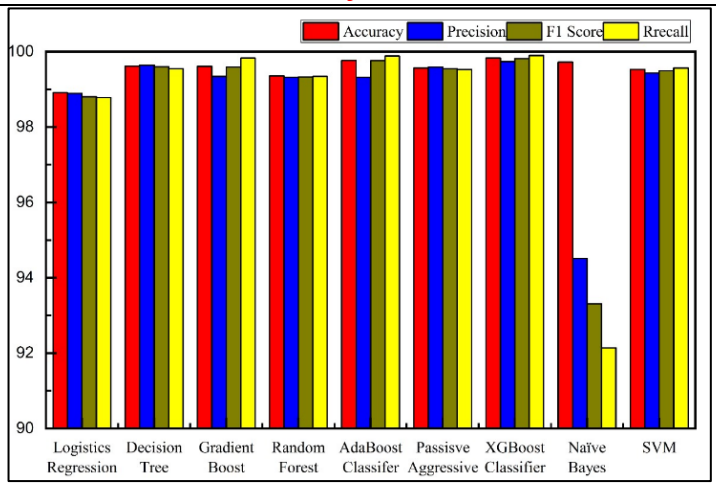

Figure 4: Performance Evaluation of Artificial Intelligence Techniques in terms of Metrics parameter.

Practical Implication:

This research offers practical tools for combatting fake news by employing machine learning algorithms like logistic regression, decision trees, and gradient boosting classifiers. These techniques empower users to critically assess news credibility, aiding fact-checking organizations and informing policymakers about effective strategies against misinformation. By enhancing media literacy and technological innovation in artificial intelligence, the findings contribute to a more informed and resilient information ecosystem. This research not only addresses the immediate challenge of fake news but also fosters broader societal implications in media, technology, education, and policy domains, ultimately bolstering the integrity of information dissemination in the digital age.

Practical Application:

The outcomes of this research can be practically applied in several ways. Firstly, the machine learning models developed can be integrated into existing news platforms and social media networks to automatically flag potentially false or misleading information, thereby providing users with warnings and prompts to critically evaluate content. Additionally, these models can inform the creation of browser extensions or standalone applications that offer real-time fact-checking capabilities, allowing users to verify news articles before sharing them. Furthermore, policymakers and regulatory bodies can utilize the findings to design more effective strategies for combating fake news, such as implementing legislation to hold purveyors of misinformation accountable. Overall, the practical applications of this research extend to both individual users and broader societal efforts to mitigate the spread of false information.

Authenticity of the Proposed Model:

The first step is choosing a hosting platform that best suits your needs and budget for the proposed model. Popular options include AWS and Google Cloud. Once you have chosen a platform, you need to set up a server to host your app. This typically involves creating a new instance or container and installing the necessary software, including Python, Flask, and any other dependencies your app requires. Next, you need to upload your app to the server using FTP or a similar file transfer protocol. You can then navigate to the URL of your app in a web browser to ensure it is running correctly.

Conclusion:

Currently, the propagation of fake news presents a challenge through traditional verification methods, so there is need of robust and automated approaches to combat misinformation. This paper conducts a comprehensive evaluation of artificial models for fake news detection, providing valuable insights into their effectiveness and applicability in addressing

these issues. Different artificial models are evaluated through majority voting assessment, also from performances evaluation, XGBoost and AdaBoost classifiers emerge as the most accurate in detecting fake news, with Decision Tree, Support Vector Machine, and Gradient Boosting classifiers also demonstrating reasonable performance. These findings underscore the potential of machine learning techniques to combat misinformation, offering a foundation for future advancements in this critical area. In conclusion, our approach offers a robust solution for identifying fake news in the digital age, highlighting the importance of a multi-faceted approach that includes education, critical thinking, and responsible information dissemination.

References:

[1] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods," Complexity, vol. 2020, no. 1, p. 8885861, Jan. 2020, doi: 10.1155/2020/8885861.

[2] "Fake News Detection on Social Media: A Data Mining Perspective", [Online]. Available: https://dl.acm.org/doi/10.1145/3137597.3137600

[3] N. Kshetri and J. Voas Kshetri, "The Economics of "Fake News," IEEE IT Prof., vol. 19, no. 6, pp. 8–12, 2017, doi: 10.1109/MITP.2017.4241459.

[4] J. M. Fernandez, "Malaysia's Anti-Fake News Act: A cog in an arsenal of anti-free speech laws and a bold promise of reforms," Pacific Journal. Rev. Te Koakoa, vol. 25, no. 1&2, pp. 173–192, Jul. 2019, doi: 10.24135/PJR.V25I1.474.

[5] T. Thate, "Using Natural Language Processing to Detect Fake News." Jun. 22, 2021. Accessed: Jun. 15, 2024. [Online]. Available: https://www.academia.edu/82451181/Using_Natural_Language_Processing_to_Detect_Fake_News

[6] D. S. Abdelminaam, F. H. Ismail, M. Taha, A. Taha, E. H. Houssein, and A. Nabil, "CoAID-DEEP: An Optimized Intelligent Framework for Automated Detecting COVID-19 Misleading Information on Twitter," IEEE Access, vol. 9, pp. 27840–27867, 2021, doi: 10.1109/ACCESS.2021.3058066.

[7] I. Ali, M. N. Bin Ayub, P. Shivakumara, and N. F. B. M. Noor, "Fake News Detection Techniques on Social Media: A Survey," Wirel. Commun. Mob. Comput., vol. 2022, no. 1, p. 6072084, Jan. 2022, doi: 10.1155/2022/6072084.

[8] "Fake News | Kaggle." Accessed: Jun. 15, 2024. [Online]. Available: https://www.kaggle.com/c/fake-news/overview

[9] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," WSDM 2018 - Proc. 11th ACM Int. Conf. Web Search Data Min., vol. 2018-Febuary, pp. 637–645, Feb. 2018, doi: 10.1145/3159652.3159677.

[10] "Preprocessing Techniques for Text Mining - An Overview." Accessed: Jun. 15, 2024. [Online]. Available: https://www.researchgate.net/publication/339529230_Preprocessing_Techniques_for_Text_Mining_-_An_Overview

[11] A. Kathuria, A. Gupta, and R. K. Singla, "A Review of Tools and Techniques for Preprocessing of Textual Data," Adv. Intell. Syst. Comput., vol. 1227, pp. 407–422, 2021, doi: 10.1007/978-981-15-6876-3_31.

[12] "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents." Accessed: Jun. 15, 2024. [Online]. Available: https://www.ijcaonline.org/archives/volume181/number1/qaiser-2018-ijca-917395.pdf

[13] "BINARY LOGISTIC REGRESSION BASED CLASSIFIER FOR FAKE NEWS | Journal of Higher Education Research Disciplines." Accessed: Jun. 15, 2024. [Online]. Available: https://www.nmsc.edu.ph/ojs/index.php/jherd/article/view/98

[14] S. Lyu and D. C. T. Lo, "Fake News Detection by Decision Tree," Conf. Proc. - IEEE

SOUTHEASTCON, vol. 2020-March, Mar. 2020, doi: 10.1109/SOUTHEASTCON44009.2020.9249688.

[15] S. Selva Birunda and R. Kanniga Devi, "A Novel Score-Based Multi-Source Fake News Detection using Gradient Boosting Algorithm," Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021, pp. 406–414, Mar. 2021, doi: 10.1109/ICAIS50930.2021.9395896.

[16] B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," Stat. Comput., vol. 27, no. 3, pp. 659–678, May 2017, doi: 10.1007/S11222-016-9646-1/METRICS.

[17] L. Holla and K. S. Kavitha, "An Improved Fake News Detection Model Using Hybrid Time Frequency-Inverse Document Frequency for Feature Extraction and AdaBoost Ensemble Model as a Classifier," J. Adv. Inf. Technol., vol. 15, no. 2, pp. 202–211, 2024, doi: 10.12720/JAIT.15.2.202-211.

[18] S. Gupta and P. Meel, "Fake News Detection Using Passive-Aggressive Classifier," Lect. Notes Networks Syst., vol. 145, pp. 155–164, 2021, doi: 10.1007/978-981-15-7345-3_13.

[19] J. P. Haumahu, S. D. H. Permana, and Y. Yaddarabullah, "Fake news classification for Indonesian news using Extreme Gradient Boosting (XGBoost)," IOP Conf. Ser. Mater. Sci. Eng., vol. 1098, no. 5, p. 052081, Mar. 2021, doi: 10.1088/1757-899X/1098/5/052081.

[20] A. J. S. T. Hofmann, B. Schölkopf, "Kernel methods in machine learning," 2008.

[21] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," An Introd. to Support Vector Mach. Other Kernel-based Learn. Methods, Mar. 2000, doi: 10.1017/CBO9780511801389.

[22] F. K. ; Alarfaj, J. A. Khan, F. Khaled Alarfaj, and J. A. Khan, "Deep Dive into Fake News Detection: Feature-Centric Classification with Ensemble and Deep Learning Methods," Algorithms 2023, Vol. 16, Page 507, vol. 16, no. 11, p. 507, Nov. 2023, doi: 10.3390/A16110507.