International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-13, Issue-3, May 2025
https:/doi.org/10.55524/ijircst.2025.13.3.20
Article ID IRP-1626, Pages 127-137
www.ijircst.org

Detecting Fake News Using Hybrid Machine Learning Models

[1]Abdur Rahman, [2]Gulfam Khan, [3]Afzal Azad, [4]Ahmar Ejaz, and [5]Maruti Maurya

[1, 2, 3, 4] B. Tech Scholar, Department of Computer Science and Engineering, Integral University, Lucknow, India
[5]Assistant Professor, Department of Computer Science and Engineering, Integral University, Lucknow, India

Correspondence should be addressed to Abdur Rahman abdur243rahman@gmail.com

Received 18 April 2025; Revised 2 May 2025; Accepted 16 May 2025

ABSTRACT- The increasing diffusion of misinformation in online media has raised alarm as a significant threat to information credibility and societal trust. The ease of disseminating false information across social media platforms, news websites, and digital forums has led to severe consequences, including political manipulation, financial fraud, and public misinformation. This research outlines a robust strategy for detecting misinformation using Natural Language Processing. To take the detection process a step further, the study analyses the implementation of ensemble models. Ensemble learning combines multiple classifiers to improve generalization and robustness, reducing the likelihood of misclassification and helps the model focus on critical words and phrases that contribute to determining the authenticity of news articles.

The operational efficacy of the model is measured exploiting standard evaluation assessment metric fidelity, precision, recall, and F1-score to confirm consistent performance. Inclusion of ensemble learning further improves classification accuracy by reducing biases inherent in individual models. Future work in this domain can focus on Live Monitoring for Misinformation, multilingual analysis, and incorporation of context-aware models to further refine detection capabilities. By continuously evolving NLP-based approaches, researchers and technology developers can serve an important function in mitigating the effect of misinformation on social dynamics.

KEYWORDS: Machine Learning, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting Classifier, TF-IDF vectorization, NLP (natural language Processing), Neural Network, Tokenization, stemming & lemmatization, Word vectorization Fake news detection.

I. INTRODUCTION

The proliferation and Circulation of disinformation in cyberspace era present a significant obstacle to the accuracy and dependability of data sharing, affecting individuals, organizations, and societies at large The accelerated spread of deceptive news across collaborative digital environments and other online networks necessitates the innovation of sophisticated detection systems to mitigate its impact. Natural Language Processing (NLP) becomes very crucial technique in the automated identification of fake news, leveraging its ability to analyse and interpret human language to identify deceptive content.

Recent advancements in NLP, combined with deep learning techniques, has shown considerable success in enhancing both the fidelity and operational effectiveness of fabricated news detection systems. A study demonstrated that a hybrid model combining CNN and Bidirectional LSTM achieved an impressive accuracy 98.13%, highlighting the potential of these models to detect subtle textual cues indicative of fake news [1].

The challenge of Deceptive content detection is further compounded by the wide-ranging content of news, which can vary significantly in length, style, and topicality. To address this, researchers have developed end-to-end detection pipelines that utilize Natural language processing methods for automated retrieval of supporting information from web-based sources. These pipelines often employ ensemble models that combine different smart detection enhancement using machine learning accuracy and F1 scores [2]. Additionally, the application of remote supervision to produce weakly annotated training data has been validated as an effective Approach for model development and data acquisition [2]

Social media platforms, being primary conduits for news dissemination, have become focal points for fake news detection research. AI-assisted models that incorporate deep NLP techniques have been proposed to analyze social features of news, such as credibility scores of publishers and users, alongside traditional content analysis. These models have demonstrated high accuracy rates, with some achieving an average accuracy of 99.72% [3]. The use of datasets like Buzzface, FakeNewsNet, and Twitter has been instrumental in evaluating these models, providing a robust framework for testing and validation [3].

The application of capsule neural networks in NLP for fake news detection is another innovative approach that has gained attention. These networks, known for their success in computer vision, have been adapted to handle the complexities of language processing, using different embedding models and n-gram levels for feature extraction. The results have shown substantial gains compared to current best practices particularly in datasets like ISOT and LIAR [4].

Attention-based models, such as the attention-based convolutional bidirectional LSTM (AC-BiLSTM), have also been explored for their ability to classify fake news into multiple categories. These models leverage systems that isolate and amplify pertinent portions of the text,

thereby enhancing classification accuracy (Trueman et al., 2021). The use of benchmarked datasets has validated the effectiveness of these approaches, confirming their feasibility in real-world applications [5].

Despite these advancements, the task of fake news detection remains complex, requiring a multifaceted approach that addresses various subtasks such as lie detection, opinion classification, and automatic verification of information. A comprehensive review of AI applications in this field has identified these subtasks as critical components of a divide-and-conquer methodology, which aims to approach the issue from a computational standpoint. (Boró et al., 2020). This approach not only highlights the challenges involved but also provides a roadmap for future research avenues, highlighting the need for further exploration. fine-grained and practical detection models [6].

The emergence of sophisticated models that assemble multiple neural network architectures and attention mechanisms has improved detection accuracy and efficiency. However, ongoing research is essential to address the shifting landscape of misinformation and to develop more resilient configurations capable of adapting to new challenges. The insights gained from current studies provide a solid foundation for future innovations, offering hope for more effective solutions to combat the pervasive issue of misinformation in the digital age.

II. LITERATURE REVIEW

Multiple studies have utilized Artificial Intelligence for detecting misinformation, Implementing models like Naïve Bayes, SVM, and Random Forests. These models leverage NLP techniques to analyse linguistic patterns and classify news as fake or real, enhancing detection accuracy[8] [9] [7] [10].

Complex learning algorithms, effectively capture complex data relationships for Deceptive content analysis techniques like (LSTM) and (BERT) achieve high accuracy, with BERT reaching 98%. Their strength lies in understanding nuanced language and contextual meaning, enhancing detection precision.

Ensemble methods integrate multiple machine learning models enhance fraudulent news identification by exploiting the advantages of different algorithms. This approach mitigates individual model limitations, enhancing accuracy and robustness. Research shows that ensemble learners consistently outperform single models, making them a more reliable solution for detecting misinformation [7].

Effective fake news detection systems utilize advanced feature engineering and data processing techniques, including tokenization, lemmatization, and TF-IDF for text preprocessing. Emotional analysis and sentiment detection further improve accuracy by identifying misleading content. These methods enhance model training, ensuring better classification of deceptive information [10].

Effective fake news detection relies on advanced feature engineering, including tokenization, lemmatization, and TF-IDF for preprocessing. Sentiment analysis and emotional detection enhance accuracy by identifying misleading content. These techniques improve model training, leading to better classification of deceptive information and ensuring reliable detection [10].

Disinformation and harmful rhetoric can spread misinformation and also incite division among communities. The internet's anonymity makes detection crucial. Machine Learning and NLP help classify such content, but identifying harmful narratives remains challenging due to varied formats. Misleading information can fuel societal chaos, emphasizing the need for advanced detection techniques. One article may contain hate speech, profane language, and cyberbullying, all classified as toxic. Such content negatively impacts society.

Online social hubs attract millions of daily users, with numbers rising from 654 million to 823 million. However, detecting fake accounts in real-time remains a challenge. Key issues include data collection, stream management, and instant user response. This research explores fake news, hate speech, and detection techniques.

L. M. Jupe et al. explored identity manipulation, focusing on verification and annexation. Participants engaged in choice lies, coerced lies, and truth scenarios. Recordings were analysed using the Verifiability Technique to assess truthfulness in their statements.[11]

Y. Li, O. et al. introduced a hierarchical machine evolutionary system, ensuring computational efficiency, ease of processing, and parallel implementation.[12].

They proposed a robust malicious URL detection system using Random Forests, leveraging diverse phishing website features for high accuracy and low error rates.[13][14] .

Yong Zhang put forward holistic focus with Recurrent Neural Networks (CA-RNN), integrating past, present, and local context for fake news detection. Bidirectional Recurrent Neural Networks (BRNN) analyse future insights, while a convolutional layer captures location features. Rumour detection relies on distinguishing real from fake news using tweets and social context-based features.[15][16].

A. Textual content based

Early fake news detection studies primarily focused on textual features and user metadata. Researchers analysed writing styles, emotional tones, and network connections to identify misinformation. Machine learning models, including convolution filters, distinguished varying text granularities. One study analyses writing style and reader impact.[16]

B. Social context based

Customer-generated activity provides valuable context for detecting fake news. Researchers used knowledge graphs and graph-kernel approaches to analyse content propagation patterns. Despite challenges in gathering social context data, A novel method for delivering consumer health information attained 84% accuracy utilizing the SVM algorithm. highlighting machine learning's effectiveness in classification tasks.[16]

C. Stance detection overview

Stance detection involves determining an author's perspective on a specific topic, headline, or individual using machine learning-based categorization techniques. In December 2016, industry and academic volunteers launched the Fake News Challenge to develop AI-driven fact-checking tools. The competition required participants to classify news text perspectives into four categories—

"agree," "disagree," "discuss," or "unrelated"—by analysing headline and body text relationships. This initiative aimed to enhance automated fake news detection and assist human fact-checkers in combating misinformation more effectively.[16]

This research integrates semantic analysis with machine learning classifiers to improve fake news detection. It reviews hybrid deep learning approaches, rumour detection, and semantic analysis. Kumar et al. (2021) emphasized NLP methods, while Wu et al. (2020) proposed a GNN with attention for rumour detection. Bharti and Jindal (2020) focused on automatic rumour detection. Additionally, Konkobo. (2020) and Umer (2020) developed CNN for enhanced detection.[19]

The proposed system employs an integrated hybrid algorithm, coupling with machine learning, NLP, clustering to enhance fake news detection. NLP extracts key content using logic and semantic analysis, while deep learning models like CNN and LSTM identify subtle misinformation patterns. Adversarial learning strengthens resilience against evolving strategies, and continuous learning ensures model adaptability. By integrating multiple approaches, including reinforcement techniques, the system enhances accuracy, reliability, and adaptability, making it highly effective in identifying and mitigating misinformation in digital media.[20]

This paper addresses Uncovering false information as a binary classification problem. When presented with a news story title or article, the objective is to classify it as real or fake. We represent textual content as word embeddings and train A dual neural network setup (CNN + RNN) for classification. Our proposed framework consists of four steps: Preprocessing of input data classification. The preprocessing stage removes noise, punctuation, and alphanumeric characters, ensuring clean input for deep learning models. Tokenized words are converted into meaningful vectors for classification.[17]

Following preprocessing, the text data was translated into vectorized word embeddings. using a combination of two embedding techniques. These representations were fed into a hybrid CNN-BILSTM model, where CNN extracted fake news features, and BILSTM captured dependencies. Regularization via dropout layers mitigated overfitting, improving classification accuracy.[17]

Effective fake news detection relies heavily on diverse datasets. The FEVER dataset, with 185,445 claims verified against Wikipedia, supports tasks like evidence retrieval and fact verification. FakeNewsNet aggregates multimodal data from BuzzFeed and PolitiFact, including social media interactions. The FNC-1 dataset aids stance detection with labelled article pairs, while the LIAR dataset offers high-quality, categorical labels, widely used in research. Our study also introduces the LIAR2 dataset, enhancing reliability and classification accuracy for advanced fake news detection and verification methodologies.[18]

III. METHODOLOGY

Social media influences education, business, and mass communication, enabling online learning and job recruitment. Researchers analyse its effects on users and how businesses leverage these platforms. This study proposes an automatic classification mechanism to detect fake accounts, enhancing identity protection. By analysing factors like publishing time, language, and geolocation, we improve social media security using advanced learning algorithms.

Tokenization refers to segmenting textual data into discrete linguistic components such as tokens, syntactic structures, or semantic units to enable more advanced computational processing. It plays a crucial role in sorting, data mining, and text summarization. In linguistic and scientific research, tokenization aids in knowledge extraction. However, challenges remain, such as handling punctuation, brackets, and hyphens. Proper validation ensures accuracy in parsing text, making tokenization essential for mobile-friendly formats and advanced NLP applications. It serves as a fundamental step in data processing and language understanding. It converts text into machine-readable vectors, enabling efficient processing in natural language processing tasks.

The removal of stop words discards commonly used words that don't carry significant meaning, such as "and" "are," and "this. "Which hold little significance in document classification. Although constructing stop word lists varies across sources, removing them enhances text processing efficiency. This process improves the quality of text analysis by refining content, reducing noise, and enhancing model performance.

Stemming and lemmatization are crucial techniques that Linguistically reduce words to their foundational form Stemming is a heuristic process that trims word endings to normalize variations, while lemmatization relies on linguistic analysis to find the root form of words. These processes enhance text preprocessing by improving consistency in language modelling.

Text preprocessing in machine learning refers to a series of systematic techniques applied to raw textual data to transform it to an optimized, structured, and machine-readable Template suitable for simulation and analysis. It reduces noise and standardizes input data, thereby improving the performance and generalization capability of learning algorithms.

Table 1: Advantages and disadvantages of Word Vector Models

Method	Advantages	Disadvantages
TF-IDF	The TF-IDF model includes information on both the more significant and less important words.	Slow for large vocabularies. Does not capture position in text, semantics, co-occurrences in different documents, etc.
Bag-of-words	The ease of implementation.	It ignores the ordering of the words in a given document. Ignores the semantic relations among words.
Word2Vec	Maintains the semantic meaning of various words in a text. The context information is preserved. The size of the embedding vector is very small.	Inability to deal with unfamiliar words. There are no common representations at the sub-word level.
Doc2Vec	A numeric representation of a document, regardless of its length. Faster than Word2Vec.	The benefit of using Doc2Vec is diminished for shorter documents.
GloVe	GloVe, unlike Word2Vec, does not rely solely on local statistics (Words local context information).	In order to obtain word vectors, global statistics (word co-occurrence) are used.
BERT [89]	Identify and capture contextual meaning in a sentence or text.	Compute-intensive at inference time.

This (Table 1) compares various text representation techniques in NLP based on their strengths and limitations. TF-IDF captures word importance but lacks semantic understanding. Bag-of-words is simple but ignores word order and meaning. Word2Vec creates compact vectors that retain semantic context but struggles with out-of-vocabulary words. Doc2Vec enhances Word2Vec by encoding full documents, though it's less effective on short texts. BERT uses deep contextual embeddings to grasp sentence meaning but requires significant computational power, especially during inference. Each method serves different use cases based on accuracy, complexity, and resource needs.

Word vectorization is crucial in fake news detection, transforming text into numerical representations. Though they lose semantic meaning and word context. TF-IDF delegates importance to terms according to their occurrence rate, while BoW counts occurrences in documents. However, these methods ignore word relationships and contextual meaning. Neural network-based models, using word embeddings like Word2Vec and GloVe, overcome these limitations. These embeddings map words into continuous vector spaces, capturing relationships and improving detection accuracy. Pre-trained models like GloVe enable large-scale training, enhancing deep learning applications.

Feature Extraction Term Frequency (TF) acts as a primary feature extraction method that captures the distribution of terms in a text corpus, enabling statistical representation of language data that measures importance of a term's relevance within a document is contingent upon its frequency of occurrence. It encodes each document as a vector that includes the frequency of word occurrences, which are then normalized to sum to one, converting them into probabilities. Given a document d within a corpus D, where w is a word appearing nw(d)n times, the document size is represented as |d|=∑_(w∈d)⁣ ⟦n_w (d) ⟧ . The normalized TF value Considering word was part of the textual content of document d is then computed as:

$$TF(w)_d = \frac{n_w(d)}{|d|}$$

We utilized the Term Frequency-Inverse Document Frequency (TF-IDF) technique to transform linguistic information into quantitative feature vectors suitable for classification models. TF-IDF is a widely used statistical technique that measures the significance of a term within a document by considering both its frequency within the document and its inverse frequency across a larger corpus of texts. This technique generates a numerical weight for each term, indicating its significance by considering how often it appears. The Inverse Document Frequency (IDF) measure assesses the rarity or specificity of a term across a collection of documents.

$$IDF(w)_D = \left\{1 + log\left(\frac{|D|}{\{|d: D|_{w \in d}|\}}\right)\right\}$$

The final TF-IDF score:

$$TF - IDF = TF(t, D) \times IDF(t)$$

This helps highlight relevant words while reducing the influence of commonly occurring terms.

TF-IDF transforms raw text into a numerical representation, making it usable for machine learning algorithms. Converts news articles into numerical feature vectors. Helps the model focus on discriminative terms that distinguishing amongst fabricated and authentic news

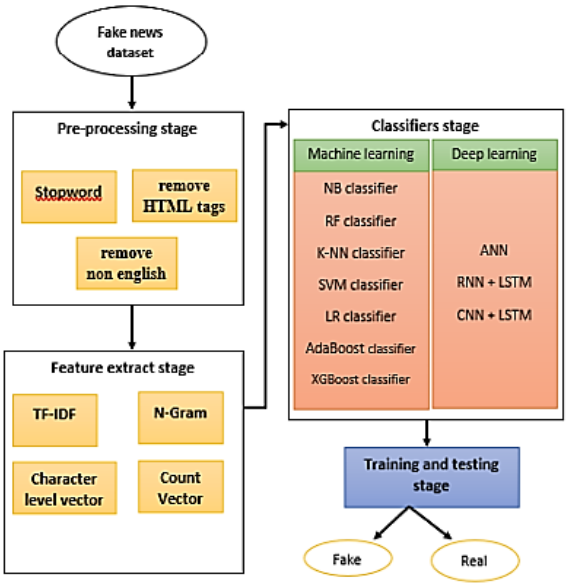

Figure 1: Fake news detection pipeline

The diagram (Figure 1) illustrates a fake news detection pipeline. It begins with pre-processing steps like stopword removal, HTML tag stripping, and filtering non-English text. Next, features are extracted using techniques such as TF-IDF, N-Gram, count vectors, and character-level vectors. These features feed into classifiers—either traditional machine learning (e.g., SVM, XGBoost) or deep learning models (e.g., RNN, CNN). Finally, the model is trained and tested to classify news as fake or real.

IV. MACHINE LEARNING CLASSIFICATION MODELS

A. Logistic regression (lr)

Logistic Regression is a fundamental, yet powerful learning algorithm widely utilized for classification problems. It estimates the likelihood that a given input text falls into one of two categories—"real" or "fake." Prior to model training, the textual data must undergo preprocessing through Natural Language Processing (NLP) methods to convert it to a numerical configuring competent for assessment. The model is trained on annotated data, where it iteratively updates its parameters (weights) using gradient descent in order to reduce the binary cross-entropy loss.

It operates by estimating the likelihood that a given article is either fake (labeled as 1) or authentic (labeled as 0) using the sigmoid function:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(wX+b)}}$$

Where:

X = the feature vector (TF-IDF scores or word embeddings)

w = learned weights

b = bias

The sigmoid function maps predictions between 0 and 1. Therefore, the output of logistic regression is constrained within the range 0 to 1, ensuring the prediction remains probabilistic. The sigmoid function, which defines the hypothesis in logistic regression, is mathematically represented as:

$$h\theta(x) = g\left((\theta^T X)\right)$$

$$\text{where } g(z) = 1/(1 + x^{-z}) \text{ and } h\theta(x) = 1/(1 + x^{-\theta^T X})$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost(h\theta(x^{(i)}), y^{(i)})$$

Similarly, the objective function for logistic regression is mathematically expressed using an equation follows:

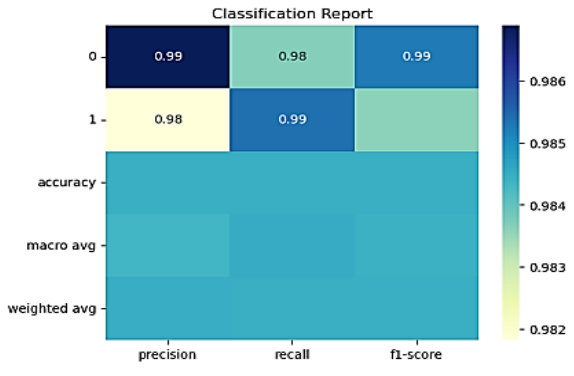

Figure 2: Classification report

Figure 2 shows the Classification report of logistic regression Using heatmap representation.

B. Decision tree

A Decision Tree is a form of supervised machine learning model that partitions data into progressively smaller segments using a series of rule-based splits. It is commonly applied in fake news detection tasks because of its clear decision-making structure and effectiveness in managing text classification problems.

A Decision Tree splits data into branches using if-else conditions based on word frequency scores (TF-IDF or Count Vectorization). It follows these steps:

The algorithm chooses the word (feature) that best separates fake and real news.

The dataset is partitioned into multiple subsets according to the chosen attribute.

The process repeats recursively, forming branches and nodes.

The splitting stops when:

- A node contains only one class (pure node).
- A maximum depth is reached.
- Information gain is too low.

The tree assigns a label (Fake or Real) to new data based on the path followed in the tree. Decision Trees are powerful tools for fake news detection, especially when combined with NLP techniques like TF-IDF. They provide clear decision rules, making them useful for real-world applications in automated fact-checking systems.

The Gini Index is a statistical metric utilized in decision trees, particularly in the CART (Classification and Regression Tree) algorithm, to evaluate the optimal way to partition data at each node in the tree. It quantifies the purity or impurity of nodes, where a node is deemed "pure" if all its elements belong to the same category. The core aim of the developmental a decision tree is to identify the optimum feature and threshold that minimize the impurity of the resulting child nodes after a split.

This work seeks to systematically explore and interpret the application of decision tree algorithms in detecting fake news. The system learns to distinguish between authentic and misleading content.

Decision tree classification using Gini index

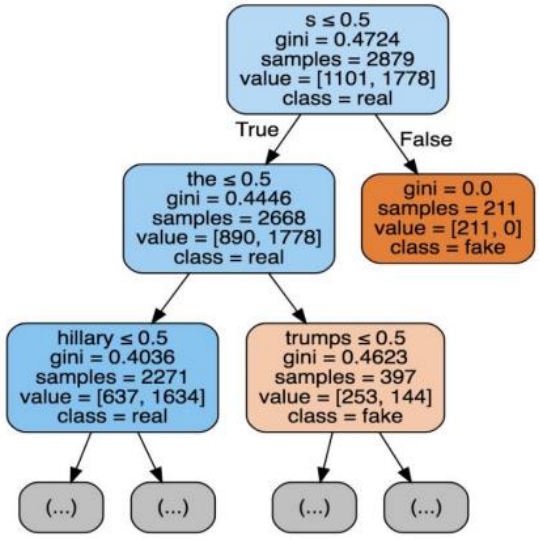

Figure 3: Decision Tree Visualization for Fake News Classification

The Figure 3 depicts a decision tree used for classifying news as real or fake. The decision tree initiates a split at the root node contingent upon the occurrence of the term 's' within the feature set. If absent (True), it moves left; otherwise, right. The right node has zero Gini impurity, indicating all 211 samples are fake. The left side continues splitting using keywords like “the”, “hillary”, and “trumps”. Gini index measures impurity, and class labels are determined by majority samples. The tree systematically learns patterns from word occurrences to differentiate between fake and real news articles with hierarchical binary decisions.

In Decision Tree (DT) learning, selecting the best attribute is crucial for building an efficient model. Various algorithms use different metrics to determine the best attribute. The ID3 algorithm employs Information Gain as a criterion for attribute selection, whereas the C4.5 algorithm enhances this approach by introducing refinements and improvements to the attribute evaluation process by using Gain Ratio.

Given a discrete attribute A with n possible values, let D be the training dataset, and Di represent the subset of D where attribute A takes the ith value.

- Information Gain (IG) evaluate the decrease in ambiguity or disorder in a data model that results from partitioning it based on attribute A.
- Gain Ratio (GR) normalizes IG by dividing it by the intrinsic value of A, preventing bias towards attributes with many distinct values.

These metrics guide attribute selection for effective tree construction.

$$Gain(A, D) = Entropy(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} Entropy(D_i)$$

$$GainRatio(A, D) = \frac{Gain(D, A)}{IV(A)}$$

where intrinsic value of attribute A can be calculated as:

$$IV(A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log2 \frac{|D_i|}{|D|}$$

C. Random forest classifier

Random Forest rely on ensemble-based machine learning technique that aggregates the outputs of several Decision Trees to enhance predictive performance and minimize the risk of overfitting. It is particularly effective in fake news detection tasks because of its robustness in managing large-scale textual data and its resilience to noise within datasets.

Building the Random Forest Model

- Bootstrapping: Random Forest selects random samples from the dataset to train multiple Decision Trees.
- Feature Randomness: Each tree uses a random subset of features (words) to make decisions, ensuring diversity among the trees.
- Decision Trees Training: Each tree is trained independently using different samples.
- Majority Voting: An ensemble of decision trees contributes to the final output through a collective voting or averaging mechanism. If most trees classify an article as "Fake," the model outputs "Fake News"; otherwise, it outputs "Real News."

The random forest algorithm can be expressed as

$$F(x) = \arg\max_{l} \left\{ \sum_{i=1}^{z} T(A(B, \theta_k)) \right\}$$

Random Forest is a powerful and reliable algorithm for fake news detection, offering high accuracy and robustness. It performs well with TF-IDF and word embeddings, making it a preferred choice for detecting misinformation on social media and news platforms.

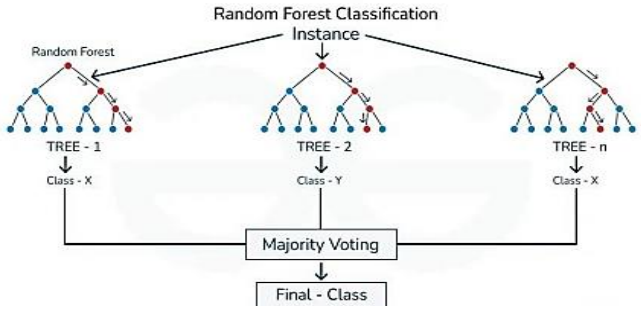

Figure 4: Random Forest Classification Process

The Figure 4 illustrates the working of a Random Forest classifier. It shows multiple decision trees (TREE-1 to TREE-n) trained on random data subsets. Each tree independently predicts a class (e.g., X or Y). The final classification is determined through majority voting among the trees. This ensemble approach improves accuracy and reduces overfitting compared to individual decision trees.

D. Gradient boosting classifier

Gradient Boosting Classifier (GBC) rely on ensemble method that iteratively constructs a series of weak learners, typically ensemble methods such as decision tree, wherein each successive model is trained to correct the residual errors of its predecessor. This technique focuses on minimizing the prediction errors through sequential model training, enhancing the overall accuracy of the classifier. This method is highly effective in fake news detection, where textual data needs to be classified as fake or real based on extracted features.

Prior to applying Gradient Boosting, the text data undergoes preprocessing through various Natural Language Processing (NLP) methods. Subsequently, textual representations are derived by applying modalities like. TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings such as Word2Vec, GloVe, or BERT.

E. Building the Gradient Boosting Model

Gradient Boosting works in an iterative manner, refining predictions over multiple rounds. The process follows these steps:

- Start with a Weak Model – The algorithm initiates with a basic predictive model, typically a low-depth decision tree, to generate the first set of estimations.
- Compute Errors (Residuals) – The difference between predicted and actual labels (fake or real) is measured.
- Implement and optimize a new predictive model to identify and rectify anomalies – A new tree-based predictive model has been generated through training to

focus on misclassified samples, correcting mistakes from the previous iteration.

- Minimize Loss Function – Gradient Boosting employs an objective function (like cross-entropy in the context of classification models) and utilizes gradient-based optimization techniques to iteratively update model parameters, weights and improve performance.
- Combine Models into an Ensemble – Each tree contributes to the final decision, with higher weights given to models that reduce classification errors.
- Final Prediction – The weighted sum of all decision trees is used to classify a news article as real or fake.

F. Gradient Boosting Works Well for Fake News Detection

- Handles High-Dimensional Data – Text data has thousands of features, and Gradient Boosting efficiently selects the most relevant ones.
- Captures Complex Patterns – Fake news often follows unique language patterns, and Gradient Boosting can recognize them.
- Reduces Overfitting – Regularization techniques (e.g., learning rate tuning) prevent the model from memorizing training data.
- Boosts Accuracy – By continuously refining weak models, it achieves high precision and recall in classifying fake news.

Table 2: Dataset used for training fake news detection

Index	Title	Text	Subject
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn't wish all Americans ...	News
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News

Table 3: Dataset used for training fake news detection

Index	Title	Text	Subject
0	As U.S. budget fight looms, Republicans flip to...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews

This dataset shown in Table 2.1 and Table 2.2, used for training fake news detection systems, provides a structured collection of news articles categorized by their title, full text, and subject. For model training, the 'Text' column serves as the primary input feature, containing the content of the news article that the model learns to analyze for linguistic patterns, sentiment, and factual consistency. The 'Title' can be used as supplementary information, offering a concise summary that might also contain clues about the article's veracity. Crucially, although not explicitly labelled as such in this snippet, such datasets typically include a target variable (often a separate column not shown here) indicating whether each article is real or fake. The 'Subject' column can be valuable for creating more nuanced models that consider the topic of the news, as fake news might exhibit different characteristics across various subjects like 'politicsNews' as seen in the examples. By learning the relationships between the text, title, subject, and the real/fake label, machine learning models can be trained to classify new, unseen articles.

V. LIBRARIES USED

A. NumPy

NumPy (Numerical Python) is a core a Python package offering high-performance utilities and optimized functionalities for performing numerical and scientific computations. It enables efficient handling of large multidimensional arrays and matrices, offering a variety of mathematical operations that can be applied to these structures. NumPy arrays serve as the foundation for most Python-based data analysis tools, making them essential for computations and data processing.

B. Pandas:

Pandas is a robust Python library for data analysis. Its main features include data structures like DataFrame and Series, which efficiently manage structured data. It provides tools for reading and writing data from various file formats and memory sources. Additionally, it supports data merging and handling missing values. Pandas enables reshaping, rotating, slicing, indexing, and subsetting of large datasets. It also includes specialized time series functions, such as date manipulation and frequency adjustments.

C. Seaborn

Seaborn is a statistical data visualization library in Python that extends the functionality of Matplotlib, offering a high-level interface for creating informative and aesthetically pleasing graphical representations. is highly effective for generating detailed and complex plots using structured data stored in a Data Frame. It supports multiple plot grids, making it easier to construct sophisticated visualizations. Additionally, it offers predefined settings to enhance and style Matplotlib figures seamlessly.

D. Matplotlib

Matplotlib is a comprehensive plotting library for Python, enabling the creation of static, animated, and interactive visualizations with extensive customization and control over graphical output. Its primary goals involve modifying graphs and charts while supporting various formats and interactive elements across multiple platforms. It seamlessly integrates with backend servers, various graphical user interface (GUI) frameworks, Python-based scripts, interactive environments such as Python and IPython shells, Jupyter notebook platforms, and browser-driven applications.

E. Scikit-learn (sklearn)

Scikit-learn, commonly known as sklearn, is a versatile and high-performance machine learning library in Python, built upon core scientific computing libraries such as NumPy, SciPy, and Matplotlib. It provides a unified and reusable framework for implementing a broad spectrum of supervised and unsupervised learning algorithms across various analytical and predictive modeling tasks.

Key Features of Scikit-learn:

- Implementation, restoration, merging, and dimensionality reduction
- Model training, preprocessing, selection, and evaluation
- Comprehensive tools for statistical analysis and performance measurement
- Preloaded datasets for experimentation and testing

Scikit-learn is widely utilized for machine learning applications, making It serves as a comprehensive reference for both novice and seasoned professionals in the field.

VI. EVALUATION METRICS

Evaluating algorithmic effectiveness of model is a fundamental process for validating its accuracy, reliability, and overall effectiveness in addressing the intended task. A model may show high accuracy, but its real performance depends on how well it handles different situations. Classification accuracy alone is not enough—other evaluation metrics help in assessing a model's reliability.

A fundamental logical tool in model evaluation is the confusion matrix, which offers a granular classification performance analysis of prophecy issues.

- True Positive (TP) – Instances where the model accurately identifies fraudulent transactions (true positives).
- True Negatives (TN) – Negative instance recognition via model prediction.
- False Positive (FP) – legitimate transactions incorrectly flagged as fraudulent.
- False Negatives (FN) – Misclassification of fraudulent activity as non-fraudulent

In addition to the confusion matrix, evaluative criteria analogous as delicacy(A), Precision(P), and Recall(R) are considerably employed to quantify model performance. The selection of applicable criteria is largely told by the specific nature of the model and its will-be operation sphere.

VII. ACCURACY

The accuracy score, also known as classification accuracy, quantifies the proportion of true positives and true negatives among the total number of predictions. made by the model. It serves as a global indicator for the model's effectiveness in correctly classifying input data.

The accuracy (A) can be expressed mathematically using the following formula in Equation:

$$A = \frac{TruePositive + TrueNegative}{TotalNumberOfPredictions}$$

While this metric offers a general assessment of model performance, it may yield misleading results in the presence of class imbalance, it does not incorporate the class distribution, potentially leading to biased performance metrics in imbalanced datasets.

A. Precision

Precision (P) denotes the proportion of correctly predicted positive instances relative to all instances classified as positive., encompassing the two, correct and incorrect classifications. It assesses the model's accuracy in correctly assigning positive classifications. Precision metric is computed using the following expression:

$$P = \frac{TruePositive}{Positive + FalsePositive}$$

This metric assesses the fraction of predicted Instances exhibiting positive outcomes which are accurately classified, thereby indicating the model's capability to reduce the occurrence of incorrect positive predictions, making it crucial for operations, where incorrect positives need to be minimized, such as spam detection or medical diagnoses.

B. Recall

Recall (R) denotes the fraction of correctly identified positive instances compared to the overall count of true positive samples within the data-set. It reflects the model's proficiency to successfully entire positive instances. It measures how well a model detects positive samples,

ensuring that relevant cases are not missed. The computation behind recall is given below:

$$R = \frac{TruePositive}{TruePositive + FalseNegative}$$

A higher recall value signifies that the model successfully identifies the majority of true positive instances, whereas a lower recall suggests that a significant number of positive cases are overlooked. This metric is particularly critical in domains where failing to detect positive instances could lead to severe repercussions, such as in medical diagnostics or fraud detection.

C. F1-Score

The F1-score, a holistic evaluation measure, synthesizes precision and recall into a singular metric for assessing classification model efficacy that evaluates classification accuracy of a model by integrating the one precision and other recall for each class. It has advantage in scenarios involving class imbalance, where traditional accuracy may convey a fallacious impression. In the circumstance of erroneous information detection, F1-score is frequently employed as a primary metric to assess the effectiveness of the classification model.

F1-score is derived using a mathematical formula that harmonized precision and recall in a single performance metric:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

where Precision and Recall are key factors in model evaluation. This metric effectively accounts for both Type I and Type II errors, offering a more dependable assessment than overall accuracy, especially in situations involving skewed class distributions.

D. ROC Curve and AUC

The performance of a binary diagnostic test is visualized by the Receiver Operating Characteristic curve, which plots the probability of detection against the probability of a false alarm. This method serves to gauge the efficacy of a classification model across a spectrum of decision boundaries. It graphically depicts the interplay between the hit rate (recall) and the false alarm rate, thus appraising the model's ability to distinguish between instances of the target class and other instances.

The Area Under the Curve (AUC) functions as a condensed quantitative indicator of the aggregate performance of the classifier — with greater AUC values indicating stronger discriminatory power between classes.

The probability of a false alarm is established using the following formula:

$$FPR = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

This metric helps evaluate how often the model incorrectly predicts negative cases as positive.

VIII. RESULT AND DISCUSSION

This research explores the identification of deceptive information via the implementation of Machine Learning (ML) methodologies by developing robust classification models. The assessment and interpretation of experimental outcomes are vital for measuring the predictive accuracy and consolidated performance metrics for the integrated models. In this ML-based fake news detection, a comprehensive evaluation and clear presentation of results are essential to understand the practical effectiveness and implications of the employed techniques.

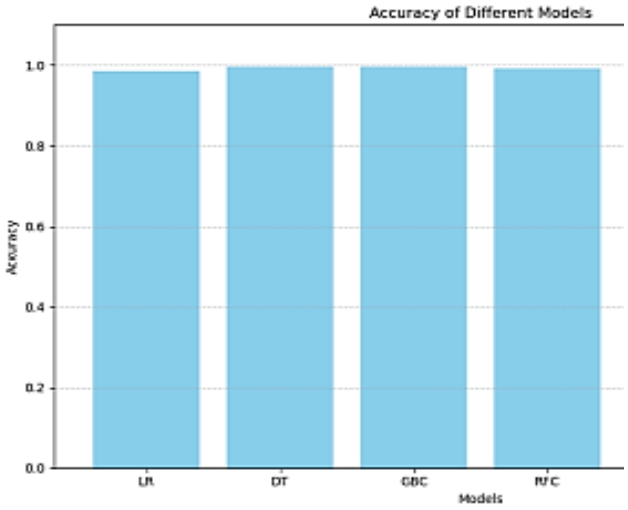

Figure 5: Model Evaluation: A Bar Chart Comparison

The bar plot shown in Figure 5 compares the classification accuracy of four distinct models: Logistic Regression (LR), Decision Tree (DT), Gradient Boosting Classifier (GBC), and Random Forest Classifier (RFC). The ordinate represents the accuracy metric, ranging from 0.0 to 1.0. Each bar corresponds to a specific model, with its height indicating the achieved accuracy. Visual inspection suggests comparable, near-perfect classification performance across all evaluated models on the given dataset.

Apart from accuracy, other essential metrics include precision, Positive predictive value represents the proportion of correctly identified positive instances among all instances flagged as positive by the classifier. Conversely, sensitivity quantifies the fraction of actual positive instances that were correctly detected. The F-measure, calculated as the harmonic mean of positive predictive value and sensitivity, offers a comprehensive assessment by integrating both aspects. Furthermore, the macro-averaged or weighted average of positive predictive value and sensitivity is essential for evaluating the model's holistic performance across all classes, especially in imbalanced datasets. These evaluation metrics are particularly significant in scenarios where the misclassification rates of false positives (authentic news mislabelled as fake) and false negatives (fake news overlooked by the model) differ. A thorough understanding and application of these performance indicators are instrumental in the advancement of a more resilient and dependable Computational Identification of Misleading Information.

The output shown in Figure 6 presents a comparative analysis of classification accuracy for Logistic Regression, Decision Tree Classifier, Gradient Boosting Classifier, and Random Forest Classifier. Two sets of accuracy metrics are displayed: direct model outputs and a tabular representation. The tabular data reveals slightly varying, high-performing accuracy scores, ranging from

approximately 0.987 to 0.996, indicating robust predictive capabilities across all evaluated algorithms on the specific dataset.

```
Accuracy of Logistic Regression: 0.99
Accuracy of Decision Tree Classifier: 1.00
Accuracy of Gradient Boosting Classifier: 1.00
Accuracy of Random Forest Classifier: 0.99
                        Model  Accuracy
0         Logistic Regression  0.986667
1     Decision Tree Classifier  0.996190
2  Gradient Boosting Classifier  0.995193
3     Random Forest Classifier  0.989116
```

Figure 6: Quantitative Comparison of Model Accuracy

IX. CONCLUSION

Identifying fraudulent news content is essential for mitigating the dissemination of misinformation across modern digital communication platforms. The rise of false information can significantly impact data analysis, affecting users, journalists, and researchers who rely on accurate content. Various machine learning algorithms, NLP techniques, and
integration approaches have been used to develop efficient solutions throughout this project.

This research combines multiple techniques to enhance adaptability, ensuring the system can counter evolving fake news strategies. The literature review offers valuable insights into existing detection methods. By building upon this foundation, the study introduces new strategies to improve model performance.

The project covers key aspects like data ingestion, NLP, integration, deep learning, adversarial learning, continuous model updates, and a user-friendly interface to enhance accuracy and usability. Ethical considerations, including user privacy, data responsibility, and transparency, guide this initiative.

Future advancements may include multilingual support, real-time analysis, and partnerships with organizations to further combat misinformation.

ACKNOWLEDGEMENT

We extend our profound appreciation to the Department of Computer Science and Engineering, Integral University Lucknow, due to their essential resources and consistent support and expert guidance throughout the research endeavour. The department's intellectual contributions, access to essential resources, and continuous encouragement were instrumental in the successful execution and completion of this study.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] T. Mahmud, T. Akter, M. Aziz, M. Uddin, M. Hossain, and K. Andersson, "Integration of NLP and Deep Learning for Automated Fake News Detection," in *2024 Second International Conference on Inventive Computing and Informatics (ICICI)*, pp. 398–404, 2024. Available from: https://doi.org/10.1109/ICICI62254.2024.00072

[2] G. Fang and G. Tan, "NLP in Fake News Detection," pp. 71–83, 2021. Available from: https://doi.org/10.1007/978-981-15-9472-4_6

[3] G. Devarajan, S. Nagarajan, S. Amanullah, S. Mary, and A. Bashir, "AI-Assisted Deep NLP-Based Approach for Prediction of Fake News From Social Media Users," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, pp. 4975–4985, 2024. Available from: https://doi.org/10.1109/TCSS.2023.3259480

[4] M. Goldani, S. Momtazi, and R. Safabakhsh, "Detecting Fake News with Capsule Neural Networks," *ArXiv*, abs/2002.01030, 2020. Available from: https://doi.org/10.1016/j.asoc.2020.106991

[5] T. Trueman, A. J, P. Narayanasamy, and J. Vidya, "Attention-based C-BiLSTM for fake news detection," *Appl. Soft Comput.*, vol. 110, 107600, 2021. Available from: https://doi.org/10.1016/J.ASOC.2021.107600

[6] R. Oshikawa, J. Qian, and W. Wang, "A Survey on Natural Language Processing for Fake News Detection," *ArXiv*, abs/1811.00770, 2018. Available from: https://doi.org/10.48550/arXiv.1811.00770

[7] V. Srikanth, "Fake News Detection System Using Machine Learning and Deep Learning," *Int. J. Sci. Res. Eng. Manag.*, 2024. Available from: https://doi.org/10.55041/ijsrem30766

[8] A. Mushiba and G. Selvam, "FAKE NEWS DETECTION SYSTEM USING MACHINE LEARNING ALGORITHMS," *Int. J. Innovative Technol. Explor. Eng.*, vol. 8, no. 10, pp. 2019. Available from: https://doi.org/10.35940/ijitee.j9453.0881019

[9] N. Fahad et al., "Stand up Against Bad Intended News: An Approach to Detect Fake News using Machine Learning," *Emerging Sci. J.*, 2023. Available from: https://doi.org/10.28991/esj-2023-07-04-015

[10] N. Prachi, M. Habibullah, M. Rafi, E. Alam, and R. Khan, "Detection of Fake News Using Machine Learning and Natural Language Processing Algorithms," *J. Adv. Inf. Technol.*, vol. 13, no. 6, pp. 652–661, 2022. Available from: https://doi.org/10.12720/jait.13.6.652-661

[11] L. M. Jupe, A. Vrij, G. Nahari, S. Leal, and S. A. Mann, "The lies we live: Using the verifiability approach to detect lying about occupation," *J. Artic. Support Null Hypoth.*, vol. 13, no. 1, pp. 1–13, 2016. Available from: https://tinyurl.com/3s6r58tj

[12] Y. Li, O. Martinez, X. Chen, Y. Li, and J. E. Hopcroft, "In a world that counts: Clustering and detecting fake social engagement at scale," in *Proc. 25th Int. Conf. World Wide Web*, pp. 111–120, 2016. Available from: https://doi.org/10.1145/2872427.2882972

[13] B. P. R. Raju, B. V. Lakshmi, and C. V. L. Narayana, "Detection of Multi-Class Website URLs Using Machine Learning Algorithms," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 1704–1712, 2020. Available from: https://doi.org/10.30534/ijatcse/2020/122922020

[14] R. Purohit and B. S. Borkar, "Identification of Fake vs. Real Identities on Social Media using Random Forest and Deep Convolutional Neural Network," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 7347–7351, 2019. Available from: https://doi.org/10.35940/ijeat.A9739.109119

[15] Y. Zhang, M. J. Er, R. Venkatesan, N. Wang, and M. Pratama, "Sentiment classification using comprehensive attention recurrent models," in *2016 Int. Jt. Conf. Neural Netw. (IJCNN)*, pp. 1562–1569, 2016. Available from: https://tinyurl.com/mrxfp4dn

[16] A. Altheneyan and A. Alhadlaq, "Big Data ML-Based Fake News Detection Using Distributed Learning," 2023. Available from: https://tinyurl.com/3sers3jx

[17] M.-A. Ouassil, B. Cherradi, S. Hamida, M. Errami, O. El Gannour, and A. Raihani, "A Fake News Detection System

based on Combination of Word Embedded Techniques and Hybrid Deep Learning Model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 10, 2022. Available from: https://dx.doi.org/10.14569/IJACSA.2022.0131061

[18] X. Cheng and M.-T. Kechadi, "An Enhanced Fake News Detection System With Fuzzy Deep Learning," *IEEE Access*, 2024. Available from: https://doi.org/10.1109/ACCESS.2024.3418340

[19] M. Khan and M. Haroon, "Detecting Network Intrusion in Cloud Environment Through Ensemble Learning and Feature Selection Approach," *SN Comput. Sci.*, vol. 5, no. 1, p. 84, 2023. Available from: https://tinyurl.com/268sh9my

[20] W. Khan and M. Haroon, "An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks," *Int. J. Cogn. Comput. Eng.*, vol. 3, pp. 153–160, 2022. Available from: https://doi.org/10.1016/j.ijcce.2022.08.002