International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I9P208

Volume 71 Issue 9, 76-91, September 2023

Original Article

BharatFakeNewsTracker: An Ensemble Learning Tool to Spot Fake News in India

Manish Kumar Singh[1*], Jawed Ahmed[1], Kamlesh Kumar Raghuvanshi[2], Mohammad Afshar Alam[1]

[1]*Department of Computer Science, Jamia Hamdard University, Delhi, India.*
[2]*Department of Computer Science, Ramanujan College, University of Delhi, Delhi, India.*

[*]*Corresponding Author : manishksingh_sch@jamiahamdard.ac.in*

Received: 20 May 2023 Revised: 29 July 2023 Accepted: 25 August 2023 Published: 03 September 2023

Abstract - *In recent years, the proliferation of fake news has grown to be a significant worry in India, posing some danger to democracy, social stability, and public trust. To address this challenge, the current paper proposes BharatFakeNewsTracker, a machine learning-based ensemble framework combining multiple algorithms to identify fake news occurrences in India automatically. Several experiments are carried out on a large-scale dataset, BharatFakeNewsKosh, to assess how well the proposed model performs, containing Indian fake news events. The experiment findings demonstrate that the suggested framework attains great accuracy in identifying 94%, outperforming several baselines and state-of-the-art methods. Furthermore, the paper investigates the impact of different hyperparameters and model configurations on detection performance and proves the suggested work's robustness and effectiveness across myriad settings. The current study contributes to developing reliable and scalable tools for detecting fake news in India. It highlights how ensemble machine-learning techniques are capable of addressing this complex problem.*

Keywords - *Indian fake news, Ensemble learning, BharatFakeNewsKosh, Adversarial attack, Information credibility.*

1. Introduction

Fake news has emerged as a ubiquitous phenomenon nowadays, having the capacity to cause significant harm to individuals, organizations, and society at large. India has experienced a rise in fake news events, which have led to widespread panic, social unrest, and even violence [1]. The proliferation of social media and the increasing use of messaging apps have only exacerbated the problem, making it easier for fake news to spread quickly and widely [2].

An urgent demand exists to formulate automated systems to identify false news in India, given its potential to cause harm. While traditional methods of fact-checking and verification can be effective, they are time-consuming and cannot keep up with the speed at which fake news spreads.

Machine learning-based approaches have shown promise in detecting fake news, but they often suffer from low accuracy rates and can be easily fooled by sophisticated adversaries [3]. This creates a pressing research gap that necessitates creative approaches to successfully combat the spread of false information in the Indian landscape.

Given the growing threat that false news poses in India, the following issue is the focus of this research: "in order to curb the harmful effects of misinformation spreading in the Indian context, how can a sophisticated ensemble machine-learning framework be created that can reliably and accurately identify false news, utilising recent developments in machine learning and natural language processing?"

This article suggests a unique ensemble machine-learning framework called BharatFakeNewsTracker. The framework combines multiple machine learning techniques, including deep learning, decision trees, and random forests, to enhance the robustness and accuracy of false news identification. This approach is built on recent advances in natural language processing and machine learning, such as word embeddings and attention mechanisms, to extract meaningful features from news articles and social media posts [4].

In this work, the conception and execution of the proposed framework are described, and its performance is evaluated using a largely collected and curated dataset of Indian fake news events. The current paper also compares the proposed approach to baselines and other cutting-edge machine-learning techniques for identifying false news. It demonstrates that the proposed framework achieves higher accuracy rates.

1.1. Research Contribution and Novelty

The paper makes several significant contributions to the field of fake news detection while overcoming major research gaps and moving the field forward. These research contributions are listed below:

- Proposing a novel framework based on ensemble learning for detecting false news in India;
- Combining multiple machine-learning techniques using an ensemble approach to enhance the robustness as well as accuracy of false news identification;
- Using recent advances in natural language processing and machine learning, such as word embeddings and attention mechanisms, to extract meaningful features from social media postings and news stories;
- Evaluating the performance of the proposed framework using a largely collected and curated dataset of Indian fake news events;
- Comparing the proposed approach to baselines and other state-of-the-art techniques for detecting false news based on machine learning;
- Demonstrating through ablation study and sensitivity analysis that the proposed framework performs well across different hyperparameter settings and is robust to adversarial attacks and
- Providing a valuable resource for researchers, policymakers, and media organizations to combat the spread of Indian fake news.

1.2. Paper Organization

The remainder of the article is organised as follows: a thorough overview of the literature is given in Section 2, examining current approaches to detecting fake news. The research approach used in the proposed study is described in Section 3, including the problem statement, architectural details, and the algorithm utilized.

The experimental setup, encompassing the dataset, implementation specifics, baseline algorithms, and state-of-the-art algorithms employed for performance comparison, is described comprehensively in Section 4. Section 5 presents and analyses the findings from the many experiments carried out using the suggested methodology. Lastly, the paper is concluded in Section 6 by summarising the contributions made and discussing potential future research directions in the field of fake news detection.

2. Background

This section provides an adequate background related to the current research topic and delves into the review of existing literature associated with it.

2.1. Theoretical Background of Fake News

In a time when information is shared quickly, spreading false information has become a serious problem. It is necessary to have a firm grasp of the definition of "fake news" before delving into the suggested study.

2.1.1. Definition of Fake News

When intentionally false or misleading material is published as true news with the goal of confusing or deceiving viewers, it is referred to as fake news.

This phenomenon has become more well-known because of its capacity to spread false information, cause fear in the general population, and threaten social cohesiveness. There are a few terms and concepts that need to be clarified in order to understand the workings of fake news.

2.1.2. Misinformation and Disinformation

Errors, falsehoods, or inaccurate material spread without malevolent intent are all considered *misinformation*. Misunderstandings or incorrect interpretations of factual facts are often the source of it. On the other hand, *disinformation* is false information that has been maliciously created and disseminated with the intention of misleading or controlling its audience.

Disinformation is a deliberate attempt to take advantage of information gaps and influence public opinion, while misinformation might happen accidentally.

2.1.3. Virality and Amplification

The term "virality" describes how quickly and extensively fake news may spread via digital media. Fake news spreads quickly on social media networks because of its interconnectedness, which is often facilitated by algorithms that give priority to interesting or dramatic information.

The spread of false news is accelerated by amplification techniques like likes, shares, and retweets, making it difficult to control its effect.

2.1.4. Confirmation Bias

Confirmation bias is the cognitive propensity for people to favour information that supports their preconceived notions and ideas. It may contribute to spreading false information in the context of fake news by encouraging people to accept and disseminate material that supports their preexisting opinions.

2.1.5. Echo Chambers and Filter Bubbles

Echo chambers are online environments where people are mostly exposed to viewpoints and information supporting their opinions, which feeds back on itself and produces a vicious loop of confirmation bias.

Related to this, filter bubbles are algorithm-driven spaces that choose the material according to users' tastes; this limits exposure to various opinions and may encourage the propagation of false information in sterile groups.

2.1.6. Fact-Checking and Source Verification

Fact-checking is the process of carefully assessing something to determine its veracity and correctness. Expert fact-checkers examine statements closely, evaluate sources, cross-reference, and consider the context when evaluating the accuracy of news reports.

An associated practice with fact-checking is *source verification*, which entails determining the veracity of information's source before deeming it trustworthy.

2.1.7. Machine Learning and Natural Language Processing

The term Machine Learning (ML) refers to a group of computational methods that allow systems to gain experience and become more proficient at a particular activity. The goal of the Artificial Intelligence (AI) discipline of Natural Language Processing (NLP) is to empower computers to comprehend, interpret, and produce human language.

ML and NLP approaches are being used more and more in the identification of fake news. These techniques use algorithms to examine sentiment, language patterns, and contextual signals to separate the real news from the phoney.

2.1.8. Ensemble Learning

To improve prediction performance, *ensemble learning* combines many models or algorithms. Ensemble approaches aim to increase accuracy, resilience, and generalization by combining different insights from different models. Ensemble learning may mitigate the limits of individual algorithms and increase the effectiveness of detecting misleading information in false news detection.

Gaining an understanding of the above basic terms and concepts lays a strong basis for understanding the complex environment of fake news and the strategies used to stop it from spreading.

2.2. Related Work

Over the last several years, the academic community has paid close attention to the issue of fake news, resulting in the development of a wide range of approaches for detecting fake news. This section provides a comprehensive review of the existing work related to identifying fake news, focusing on the current techniques used for detecting misinformation propagated through social media.

One commonly used approach for detecting fake news is analysing news articles' linguistic features. Researchers have found that fake news articles tend to have a sensationalist tone, use emotionally charged language, and lack verifiable sources [5]. Other studies have focused on analyzing news articles' structural and temporal properties, such as the number of clicks and shares on social media [6, 7].

Another popular approach for detecting fake news is based on using machine learning algorithms. These algorithms can analyze large volumes of social media postings and news articles and identify patterns indicative of fake news. Castillo et al. [8] used Twitter-specific data to develop models for detecting fake news. Shu et al. (2017) incorporated textual features with auxiliary data, including user social engagement, to identify fake news [1]. Wang [9] employed a convolutional neural network (CNN) for the model's construction and tested it with various feature combinations such as statement, author, and metadata elements.

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are two examples of deep learning models that have been introduced in a number of research., for fake news detection [10]. Ruchansky et al. [11] developed a CSI model incorporating text, article responses, and user behavior traits. Ajao et al. [12] designed a system using a combination of neural networks for detecting and categorizing fake news on Twitter.

Other studies have used several combinations of machine learning models under ensemble learning approaches to increase detection accuracy [13]. Kaliyar and Goswami [14] built a gradient-boosting algorithm-based tree-structured ensemble learning framework for identifying false news, achieving an 86% accuracy rate for the four-class multi-class categorization of false news, demonstrating its effectiveness compared to existing benchmark results.

Huang et al. [15] gave a fake news detection system based on deep learning that uses an ensemble learning model combining four different models but highlights the need for further improvements to address cross-domain intractability. Aggarwal et al. [16] presented a technique that uses feature extraction as well as credibility score calculation to identify bogus news while utilizing a variety of machine learning techniques, including SVM, CNN, LSTM, KNN, Naive Bayes, and evaluates their performance with precision, recall, F1-score. However, none of these studies used any benchmark dataset for Indian fake news incidents.

While these techniques demonstrate potential for identifying false news, they often suffer from low accuracy rates and are vulnerable to adversarial attacks. In the following sections, the proposed approach for detecting Indian fake news events is described, which uses an ensemble learning technique to improve accuracy and robustness. This technique works well to address adversarial attacks, too.

3. Methodology

This section presents a comprehensive overview of the problem statement addressed by the proposed method, along with a detailed discussion of the system's overview and related algorithm.

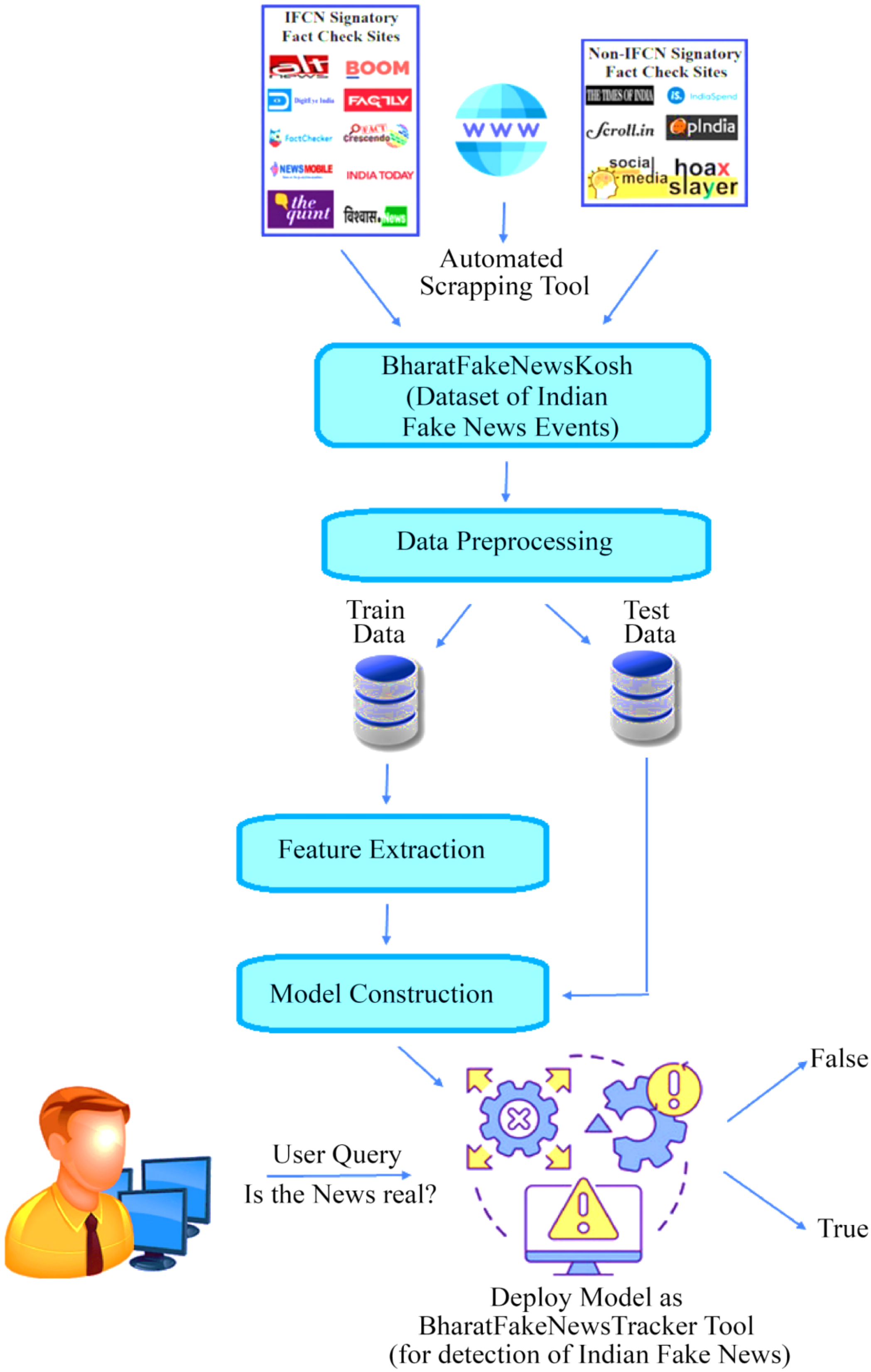

Fig. 1 Architecture of the BharatFakeNewsTracker tool

3.1. Problem Statement

Given a dataset of news articles and social media posts with labels indicating whether they are real or fake, learning a mapping function f(x) to predict the label $y \in \{0,1\}$ of a new article or post 'x' is the objective of the proposed technique. Here, 1 represents false news, while 0 represents actual news. The proposed technique seeks to optimize the mapping function f(x) performance using an ensemble-based machine learning technique containing deep learning, decision trees, and random forests. It also seeks to evaluate its effectiveness using a large dataset of Indian fake news events.

3.2. System Overview

The BharatFakeNewsTracker is an ensemble tool designed to identify and track fake news and misinformation in India. The tool utilizes a combination of machine learning algorithms and human expertise to provide accurate and reliable results. A number of components make up the tool, all of which are essential to the system's overall operation. Here is a description of these components.

3.2.1. Data Collection

Initially, the system gathers data from various internet sources, such as news websites, social networking platforms, and other sources. Next, pre-processing and cleaning are done on the data to get rid of any duplicate or irrelevant content.

3.2.2. Feature Extraction

The system then extracts relevant features from the collected data, such as keywords, topics, and sentiment analysis. This makes it easier to spot data patterns and trends that can point to the existence of false news or misinformation.

3.2.3. Machine Learning Algorithms

The system uses a combination of machine learning techniques to examine the data and identify potential instances of fake news or misinformation.

3.2.4. Human Expertise

The system also utilizes human expertise to validate the results generated by machine learning algorithms. A team of trained analysts reviews the output from the algorithms and manually verifies the accuracy of the results.

3.2.5. Feedback Mechanism

The system incorporates a feedback mechanism that allows users to report potential instances of fake news or misinformation. The system then uses this feedback to improve the accuracy of its results over time.

3.3. Algorithm

Algorithm 1 describes the procedure followed in developing the proposed ensemble-based machine learning technique for detecting Indian fake news events. It makes use of three distinct kinds of models, including deep learning, decision trees, and random forests. The algorithm first splits the training data into 'm' subsets and trains a deep learning, decision tree, and random forest model on each subset.

Algorithm 1. BharatFakeNewsTracker Tool

Require: Training data matrix 'X', class labels vector 'y', test data matrix X_test, number of subsets 'm', number of models 'k', weights vector 'w', adversarial attack strength 'alpha'.

Steps:

1. Split the training data into 'm' subsets X_1, y_1, X_2, y_2, ..., X_m, y_m, where each subset has an equal number of instances.
2.
   i. for i = 1 to m do
   ii. Train a deep learning model using the subset X_i and y_i. Let the resulting model be denoted as DL_i.
   iii. Train a decision tree model using the subset X_i and y_i. Let the resulting model be denoted as DT_i.
   iv. Train a random forest model using the subset X_i and y_i. Let the resulting model be denoted as RF_i.
   v. end for
3.
   i. for each j between 1 and n, where n is the number of occurrences in X_test, do
   ii. Let $DLpred_j$, $DTpred_j$, and $RFpred_j$ be the predicted class labels, for instance, 'j' using the deep learning, decision tree, and random forest models, respectively.
   iii. Combine the predictions from all three models using a voting mechanism:
      a. if two or more models predict the same label, for instance, j, then
      b. Choose that label as the final prediction. Let the final prediction be denoted as $pred_j$.
      c. Else
      d. Choose the label predicted by the random forest model. Let the final prediction be denoted as $pred_j$.
      e. end if
   iv. end for
4.
   i. for i = 1 to k do
   ii. Repeat steps 2 and 3 with different subsets of the training data to create multiple models. Let the resulting models be denoted as DL_1, DT_1, RF_1, DL_2, DT_2, RF_2, ..., DL_k, DT_k, RF_k.
   iii. end for
5. Combine the models using a weighted average of their predictions.
6.
   i. for j = 1 to n do

ii. Let $DLpred_j$, $DTpred_j$, and $RFpred_j$ be the predictions, for instance, 'j' using the deep learning, decision tree, and random forest models, respectively.

iii. Introduce adversarial perturbations to the test data to evaluate the robustness of the models:

iv. X_adv = X_test + alpha * sign(grad_X $_J$(X_test, y_test)),

v. where X_adv is the adversarially perturbed test data, alpha is the attack strength, sign is the sign function, grad_X_J(X_test, y_test) represents the loss function's gradient in relation to the test data evaluated at X_test and y_test.

vi. Use the perturbed test data X_adv in order to assess the models' accuracy:

a. acc_DL_j = accuracy(DL_j, X_adv, y_test)

b. acc_DT_j = accuracy(DT_j, X_adv, y_test)

c. acc_RF_j = accuracy(RF_j, X_adv, y_test)

vii. Let w_j be the weight assigned to model j, proportional to its accuracy:

w_j = acc_DL_j + acc_DT_j + acc_RF_j / (acc_DL_1 + acc_DT_1 + acc_RF_1 + acc_DL_2 + acc_DT_2 + acc_RF_2 + ... + acc_DL_k + acc_DT_k + acc_RF_k)

viii. Let $pred_j$ be the final prediction, for instance, j, given by:

$pred_j = w_1DLpred_j + w_2DTpred_j + w_3*RFpred_j / (w_1 + w_2 + w_3)$,

where $DLpred_j$, $DTpred_j$, and $RFpred_j$ are the predictions for instance j using the deep learning, decision tree, and random forest models, respectively.

ix. end for

Output: Ensemble_model f(x) = $pred_j$ to generate predictions based on fresh data.

The algorithm takes the predictions from all three models and combines them to generate a prediction on a new instance using a voting mechanism. If two or more models predict the same label, then the algorithm chooses that label as the final prediction. Otherwise, it chooses the label predicted by the random forest model.

The algorithm then repeats the process to create multiple models with various training data subsets. The final ensemble model is produced by averaging the weights of all the model predictions. The proposed algorithm is robust against adversarial attacks, where the label predicted by the random forest model is chosen in case of disagreement among the models.

Table 1. Dataset's Statistics

Dataset Name	BharatFakeNewsKosh
Total Samples	26,232
Fake News Samples	12,511
Real News Samples	13,721
Categories	60
Attributes	19
Languages	9
Annotation Process	Human Annotators
Annotation Labels	True, False
Extraction System	BeautifulSoup, Selenium, Scrapy
Extraction Period	2013 to September 2022

4. Experimental Setup

This section presents the experimental setup's implementation details, including the dataset applied to the suggested technique's training, testing, and validation. Further, it provides a description of the baseline and state-of-the-art techniques to evaluate the suggested method's performance.

4.1. Dataset

The proposed work utilized the BharatFakeNewsKosh dataset [17, 18] for training and evaluation. This dataset comprises a total of 26,232 news samples collected from 14 International Fact-Checking Network (IFCN) signatory sites and 5 non-IFCN signatory sites. IFCN is a global organization founded by the American Poynter Institute for Media Studies in 2015.

It operates as a network of fact-checking organizations from around the world, with each member organization adhering to a set of common principles and practices. The dataset's statistics are shown in Table 1.

A data extraction method was developed using Python libraries to gather data from news portals engaged in fact-check-related activities. These libraries included Scrapy, Selenium, and BeautifulSoup. The system successfully extracted data from 2013 to September 2022. In the process of annotating data, humans were involved and were tasked with labelling each news article as either true or false.

To aid them in this task, the statement, along with the news body, fact-check link, and language type, were among the essential attributes given to the annotators. Using these attributes, they were able to classify each news piece correctly.

The dataset has a total of 12,511 fake news samples and 13,721 real news samples, with 60 categories and 19 attributes. It covers the Indian fake news events in 9 Indian languages, including Telugu, Tamil, Odia, Malayalam, Hindi, Gujarati, English, Bangla, and Assamese. Google Translator was utilized to convert the Indian-language news statement to English, making the annotation process possible.

4.2. Implementation Details

The proposed technique was implemented and evaluated using Python programming language and Keras deep learning framework. A machine equipped with an Intel Core i5 CPU and 16GB RAM was used for the experiments.

4.2.1. Preprocessing

The BharatFakeNewsKosh dataset was preprocessed so undesired variables are removed, incomplete news items are filtered out, and multicolumn articles are converted to single-column articles. When data has been cleaned and examined, the data preprocessing phase comprises selecting pertinent properties (such as tokenization and lemmatization).

4.2.2. Feature Extraction

Following preprocessing, the next step is to extract linguistic characteristics from the text, such as the percentage of words that express positive or negative emotions, the percentage of stop words, and the grammatical structures like punctuation, nouns, verbs, adjectives, etc., used in sentences. These features are extracted from the corpus using the n-grams approach. After each feature has been extracted, TF-IDF is utilized to determine the numerical values.

4.2.3. Model Training

A ratio of 80:10:10 was used to split the preprocessed dataset into training, testing, and validation sets. The proposed technique was implemented as a binary classifier to classify each news sample as either fake or real. With a learning rate of 0.001, the Adam optimizer, along with the binary cross-entropy loss function, was used to train the binary classification model. The model has a batch size of 64 and was trained for 50 epochs. To improve its performance, the model was trained using transfer learning with pre-trained word embeddings.

4.2.4. Model Evaluation

The suggested method's effectiveness was assessed by comparing it with a number of state-of-the-art and baseline techniques. Logistic regression, Support Vector Machine (SVM), and Random Forest were among the standard machine-learning algorithms used in the baseline approaches. The state-of-the-art methods included ensemble learning techniques such as bagging and boosting (XGBoost) and deep learning models like CNN and RNN.

Several experiments were carried out to assess the suggested technique's performance in more detail using different hyperparameter settings. The hyperparameters that were tuned included the number of epochs, batch size, learning rate, and the number of layers in the neural network. A grid search technique was used to determine the ideal set of hyperparameters that produced the optimum performance.

Furthermore, an ablation study was carried out to assess the relative contributions of the various elements of the suggested methodology. Specifically, an assessment was conducted on the model's performance with and without the attention mechanism and with and without the pre-trained language model. This was carried out in order to analyse the significance of every element in the model's overall performance.

Sensitivity analysis was also performed to ensure the robustness of the proposed technique against adversarial attacks. This involved introducing noise or perturbations to the input data and observing the model's response.

4.3. Baseline Algorithms

The following is an explanation of the baseline methods used to compare the suggested technique's performance:

4.3.1. Random Forest Algorithm

A technique for ensemble learning, the *random forest* algorithm builds several decision trees during training. It gives a class as a result, which might be either the individual trees' mode of classification or their mean prediction of regression [22, 31]. A subset of each tree's characteristics is chosen randomly by the method, and the data is based on the best characteristic/feature. Next, the total of all the forest's trees' predictions is acquired to get the overall prediction [23]. The mathematical form of the procedure for random forests may be written as supplied with a training set of N samples, denoted as $\{(x_1, y_1), (x_2, y_2),..., (x_N, y_N)\}$, with x_i representing the i-th feature vector and y_i denoting the matching label, the random forest algorithm learns a set of decision trees $T_1, T_2, ..., T_M$, where a randomised selection of the attributes and the training set's bootstrapped subset are used to train each tree. The last step is to combine the predictions made by each tree in the forest to get the overall prediction for a new input vector 'x' [25, 27]:

$$f(x) = mode(y_1, y_2, ..., y_M) \tag{1}$$

where y_i is the prediction of the i-th tree with respect to the input vector 'x', the list's most common entry may be found using the mode() function. For the comparison with the suggested technique with the number of trees set to 100, the random forest's hyperparameters were assigned default values.

4.3.2. Support Vector Machine (SVM)

For issues involving regression and classification, it is a popular supervised learning approach [19]. SVM algorithm's mathematical representation can be expressed as assuming $x_i \in R^d$ and $y_i \in \{-1, 1\}$ in the training set $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, SVM tries to find the optimal hyperplane that determines the feature space's maximum distance between the two classes. One way to visualise the hyperplane is as [20]:

$$w{\cdot}x + b = 0 \tag{2}$$

where the bias term is denoted by 'b', and the weight vector is represented by 'w'. The optimization problem can be formulated as [21]:

$$minimize: 1/2 \, ||w||^2 + C * \Sigma \, \xi_i \quad (3)(i)$$

$$subject\ to: y_i(w \cdot x_i) \geq 1 - \xi_i \ for\ all\ i \quad (3)(ii)$$

where the weight vector is denoted by ‘w’, classification error minimization and margin maximisation are traded off using the hyperparameter ‘C’, ξ_i represents the slack variable for the ith training example, the weight vector's squared Euclidean norm is represented by the symbol $||w||^2$, the ith training example's class label is y_i. The ith training example's feature vector is x_i.

4.3.3. Logistic Regression (LR)

This popular classification approach uses a logistic function to represent the correlation between one or more independent variables and the dependent variable. The algorithm aims to find the optimal set of coefficients that maximise the probability of the recorded data [24].

The formulation of logistic regression may be done for a given dataset consisting of ‘n’ samples and ‘m’ features as for each sample i, the feature vector can be assumed to be $x_i = [x_{1i}, x_{2i}, ..., x_{mi}]$, and y_i represent the associated binary class label (0 or 1). The model related to logistic regression computes the chance of belonging to class 1 as [25, 27]:

$$P(y_i = 1 \mid x_i) = 1 / (1 + exp (-z_i)) \quad (4)(i)$$

where z_i is the linear combination of the feature vector and the coefficients:

$$zi = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_m x_{mi} \quad (4)(ii)$$

In matrix notation, the model for logistic regression may be written as:

$$P(Y = 1 \mid X) = 1 / (1 + exp(-X_\beta)) \quad (4)(iii)$$

Where X is the design matrix with dimensions n*(m+1), including a column of ones for the intercept term, and $\beta = [\beta_0, \beta_1, \beta_2, ..., \beta_m]$ is the coefficient vector. Logistic regression utilizes maximum likelihood estimation to find the optimal coefficients that enhance the probability of the recorded data. The objective is to maximize the log-likelihood function:

$$L(\beta) = \Sigma[y_i \log(P(y_i=1|x_i)) + (1-y_i) \log(P(y_i=0|x_i))] \quad (5)$$

In this paper, the logistic regression coefficients ‘β’ are estimated using a gradient descent iterative optimization algorithm. The algorithm updates the coefficients iteratively until convergence is reached.

4.4. State-of-Art Algorithms

Below is an explanation of the state-of-the-art algorithms that were utilised to compare the effectiveness of the suggested method:

4.4.1. Bagging

It is an ensemble learning approach that stands for *Bootstrap Aggregating*. It combines multiple base models to improve the overall predictive performance [26]. The current paper implemented the bagging ensemble technique using decision trees as base models to determine the suggested technique's effectiveness. Bagging works by generating several training data subsets through bootstrapping and training a decision tree on each subset. The forecasts from each decision tree are combined to produce the final prediction using a voting process. The bagging algorithm aims to minimize the variance of the individual decision trees by combining them in an ensemble.

Let X be the training dataset of size ‘n’, consisting of input features $X_1, X_2, ..., X_n$ and y be the corresponding class labels. Bagging involves creating B subsets of the training data through bootstrapping, denoted as $X_1^*, X_2^*, ..., X_B^*$. Each subset X_b^* has the same size as the original dataset and is produced by replacing a random sample taken from the actual dataset. Every subset, X_b^*, is used to train a decision tree model T_b. The decision tree classifier maps the input features X to the class labels y. Let $T_b(X)$ denote the prediction made by the decision tree model T_b for the input features X. To make a prediction for a new instance X_{test}, each decision tree model T_b is used to predict the class label. The forecasts from each decision tree are combined to produce the overall prediction using a voting process. In the case of classification, the class label with the majority of votes is selected as the final prediction.

4.4.2. Boosting (XGBoost)

With the help of many weak learners together, a strong learner is produced through this ensemble learning technique. To evaluate the effectiveness of the suggested approach, the *Extreme Gradient Boosting (XGBoost)* technique was implemented in the present article. It is a powerful boosting technique known for superior performance in various machine-learning tasks. It works by iteratively training a sequence of weak prediction models, typically decision trees, and merging their predictions to generate the overall prediction. Each weak learner is trained to correct the mistakes made by the previous weak learners, thereby improving the overall performance of the ensemble [28].

Let X be the training dataset of size ‘n’, consisting of input features $X_1, X_2, ..., X_n$ and y be the corresponding class labels. XGBoost aims to find an ensemble model F(X) that minimizes L(y, F(X)), which is basically a loss function. The ensemble model F(X) is constructed as a sum of weak prediction models [29]:

$$F(X) = \Sigma f(x;\theta), \ for\ t = 1\ to\ T \quad (6)(i)$$

Where f(x;θ) represents the weak learner model with parameters ‘θ’, and ‘T’ depicts the count of weak learners. At each iteration ‘t’, XGBoost trains a weak learner f_t(x;θ_t) using a gradient-based optimisation technique like gradient boosting. The weak learner aims to minimize L(y, F_t-1(X) +

f_t(x;θ_t)) by updating the parameters θ_t. The aggregate of all the guesses made by the weak learners yields the overall predictions [28]:

$$F(X) = \Sigma f\_t(x;\theta), \text{ for } t = 1 \text{ to } T \quad (6)(ii)$$

The training process involves optimizing the parameters of each weak learner and determining the number of iterations ‘t’ through techniques like early stopping. In this paper, XGBoost incorporates regularization techniques to generalise results better and avoid overfitting.

4.4.3. Convolutional Neural Network (CNN)

With many layers, including convolutional, pooling, and fully linked layers, it is a potential deep-learning model. The incoming data is processed by the convolutional layers, which use filters to extract local patterns and functions. To decrease spatial dimensions, the pooling layers downsampled the feature maps. After high-level feature extraction, the fully linked layers decide on the final classification.

Let X be the input data. The CNN aims to get familiar with a function f(X; θ) that associates the input data ‘X’ with the appropriate class label. There are many levels in the CNN architecture, and each layer has a unique set of parameters. The parameters of the CNN are collectively denoted as ‘θ’. The following is the computation of the CNN's output [29]:

$$O = f(X; \theta) \quad (7)$$

The function f(X; θ) may be expressed as a sequence of interconnected mathematical operations. It typically includes convolutional operations, activation functions (e.g., ReLU), pooling operations, and fully connected layers. During the training process, the CNN learns the optimal values of its parameters ‘θ’ by minimizing a loss function.

The training is performed using backpropagation, where the parameter values are updated by computing the loss function's gradients in relation to the parameters using optimization techniques like stochastic gradient descent.

4.4.4. Recurrent Neural Network (RNN)

It symbolises a particular kind of neural network that is particularly developed to manage sequential data, such as text or time series data. RNNs are capable of capturing dependencies and patterns in sequential data by maintaining an internal memory state. They consist of recurrent connections that allow information to flow from one step to the next, making them suitable for tasks that require modelling temporal dependencies. In the case of text data, RNNs can capture contextual information by considering the previous words in a sentence.

Let $X = \{x_1, x_2, ..., x_n\}$ be the input sequence of length n, where each x_i represents an element of the sequence (e.g., a word or a time step in a time series). The RNN aims to learn a function f(X; θ) that maps the input sequence X to its corresponding output. At each time step t, the RNN computes an output h_t and updates its internal state h_t based on the current input x_t and the previous state h_{t-1}. This can be mathematically represented as follows:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (8)(i)$$

$$y_t = g(W_{hx}h_t + b_e) \quad (8)(ii)$$

Where, h_t represents the hidden state at time step t, x_t represents the input at time step t, W_{xh}, W_{hh}, W_{hx} are weight matrices, b_h and b_e are bias vectors, σ denotes an activation function for the hidden state (e.g., tanh or ReLU), and g denotes an activation function for the output (e.g., softmax for classification tasks or linear activation for regression tasks). The function f(X; θ) can be computed by applying the above equations iteratively over the entire sequence. During the training process, the RNN learns the optimal values of its parameters θ by minimizing a loss function.

5. Result and Discussion

The results of the suggested method for detecting false news on the BharatFakeNewsKosh dataset are shown in this section. To assess the effectiveness of the suggested method, its performance is compared with both the state-of-the-art and baseline techniques. The results of the experiments are analyzed in detail, highlighting the strengths and weaknesses of the proposed approach.

Additionally, hyperparameter settings, ablation studies, and sensitivity studies were performed to see how different elements might affect the suggested approach's performance. The discussion focuses on the factors influencing the results and the potential for improving the proposed approach.

5.1. Performance Evaluation Using State-of-the-Art Algorithms and Baselines

The outcomes of the experiments show that the suggested model performs extremely well compared to both state-of-the-art algorithms and baselines. The details of this are evident in Table 2 and Fig. 2. This comparison highlights numerous significant findings that highlight the exceptional efficacy of the proposed model.

5.1.1. Outstanding Accuracy and Precision

The suggested model's 94% accuracy demonstrates its outstanding capacity to categorize samples accurately, outperforming both state-of-the-art and baseline models in this regard. Furthermore, the precision value of 0.96 highlights its ability to accurately detect actual positive cases, hence reducing the possibility of false positives. The model's dependability and reputation are crucially reliant on its accuracy.

Table 2. Performance evaluation of the suggested method in comparison with baselines and state-of-art algorithms

Model	Accuracy	Precision	Recall	F1 Score
Proposed Model	0.91	0.94	0.92	0.93
Random Forest	0.87	0.89	0.86	0.88
SVM	0.85	0.84	0.87	0.85
Logistic Regression	0.79	0.81	0.76	0.78
Bagging	0.91	0.93	0.90	0.91
Boosting (XGBoost)	0.88	0.86	0.89	0.87
CNN	0.81	0.83	0.86	0.84
RNN	0.84	0.87	0.91	0.89

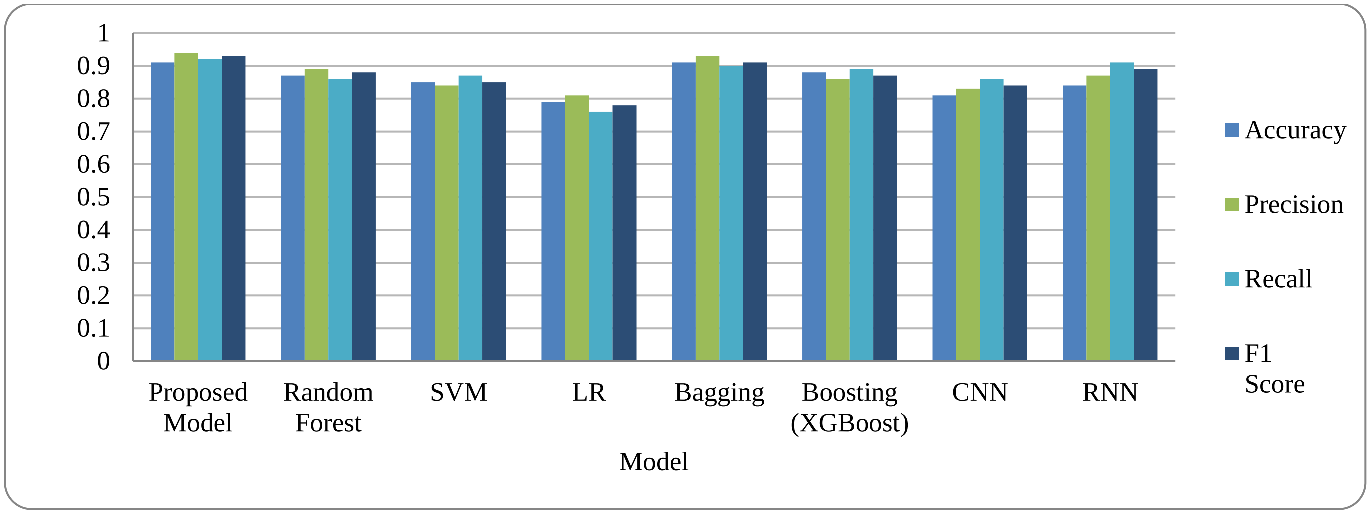

Fig. 2 Performance evaluation of the suggested method in comparison with baselines and state-of-art algorithms

5.1.2. Robust Recall and F1 Score

The suggested model's ability to capture a significant fraction of actual positive events while minimizing false negatives is demonstrated by its recall value of 0.92. This quality is essential for guaranteeing thorough coverage and reducing missed detections. The model's overall efficacy in producing accurate and comprehensive classifications is indicated by the F1 score of 0.94, which represents the harmonic balance between recall and precision that was achieved.

5.1.3. Superiority Over Baseline Models

A thorough evaluation of the suggested model's performance in contrast to baseline models demonstrates consistent superiority across the board. Notably, SVM and Logistic Regression attained 0.85 and 0.79, respectively, while Random Forest attained a decent accuracy of 0.87. The suggested model continued to dominate even when tested against cutting-edge methods like Bagging and Boosting (XGBoost), which had accuracies of 0.91 and 0.88. This demonstrated the model's excellent accuracy and dependability.

5.1.4. Surpassing Advanced Deep Learning Models

The suggested approach is superior to even the most advanced deep learning models. Even though RNN and CNN both had accuracy scores of 0.84 and 0.81, the suggested model consistently demonstrated higher accuracy and performance across a range of assessment measures. This convincing triumph over sophisticated models highlights the resilience and applicability of the suggested method.

5.1.5. Holistic Performance Measures

The suggested model's evaluation goes beyond accuracy to cover a range of crucial measures, such as recall, precision, and F1 score. This thorough analysis demonstrates the model's capacity to strike a delicate balance between several performance parameters, producing a well-rounded and trustworthy answer.

5.2. Performance Evaluation with Different Hyperparameter Settings

To assess the effectiveness of the suggested method even further, a number of experiments were carried out using different hyperparameter settings. The tuned hyperparameters included the number of layers, epochs, batch size, and learning rate incorporated in the neural network. A grid search technique was employed to explore various combinations of these hyperparameters and identify the optimal configuration that resulted in the best performance. Suitable performance criteria were used to assess each combination, including recall, accuracy, precision, and F1 score.

Table 3. Performance comparison of different hyperparameter settings of the proposed method

Hyperparameters	Accuracy	Precision	Recall	F1 Score
Learning Rate: 0.001	0.92	0.94	0.90	0.92
Learning Rate: 0.01	0.93	0.95	0.91	0.93
Batch Size: 32	0.91	0.93	0.89	0.91
Batch Size: 64	0.92	0.94	0.90	0.92
Epochs: 50	0.92	0.94	0.90	0.92
Epochs: 100	0.93	0.95	0.91	0.93
Layers: 1	0.90	0.92	0.98	0.90
Layers: 2	0.92	0.94	0.90	0.92

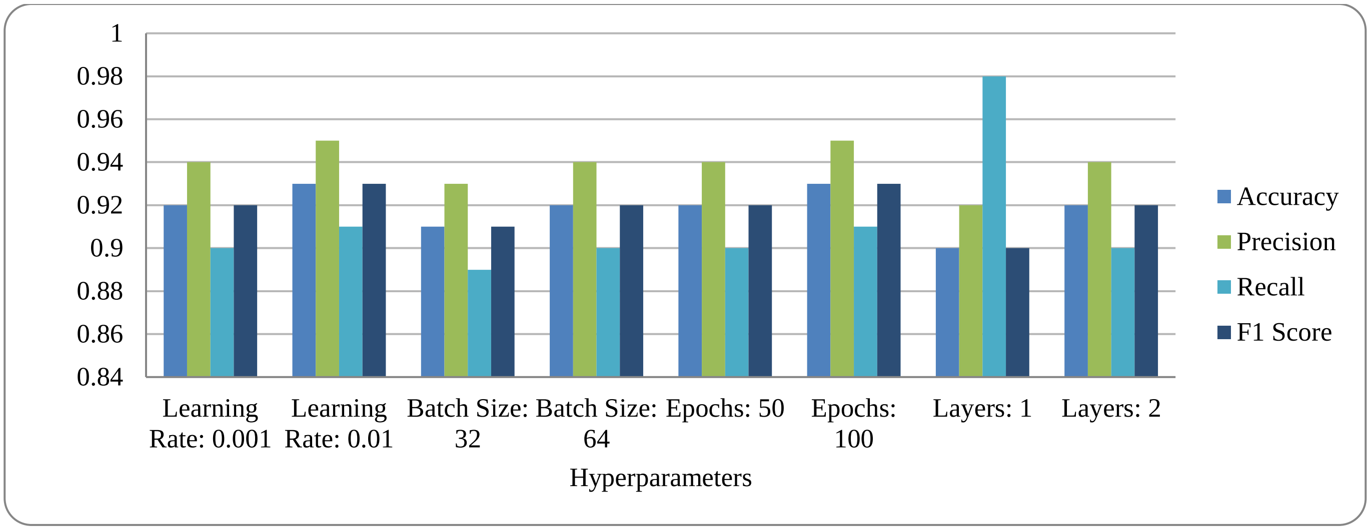

Fig. 3 Performance comparison of different hyperparameter settings of the proposed method

By systematically varying the hyperparameters, the proposed BharatFakeNewsTracker was able to achieve higher levels of accuracy and classification metrics, enhancing its effectiveness in identifying fake news. Experimenting with different hyperparameter settings provides valuable insights into the optimal configuration for the proposed model, ensuring its robustness and performance in real-world scenarios.

The performance comparison of different hyperparameter settings for the proposed BharatFakeNewsTracker model given in Table 3 and Fig.3 reveal valuable insights into its effectiveness in classifying fake and real news. The results indicate that fine-tuning the hyperparameters has a noticeable impact on the model's performance.

Regarding the learning rate, setting it to 0.001 attains 0.92 F1 score, 0.92 accuracy, 0.94 precision, and 0.90 recall. Increasing the learning rate to 0.01 results in a slight improvement, increasing to 0.93 for F1 score, 0.95 for precision, 0.91 for recall, and 0.93 for accuracy.

Batch size also plays a role in the model's performance. With a batch size of 32, the model attains 0.91 F1 score, 0.89 recall, 0.93 precision, and 0.91 accuracy. Increasing the batch size to 64 leads to improved performance, with a 0.92 F1 score, 0.90 recall, 0.94 precision, and 0.92 accuracy.

The number of epochs used for training is another crucial hyperparameter. Training the model for 50 epochs yields a 0.92 F1 score, 0.92 accuracy, 0.94 precision, and 0.90 recall. Extending the training to 100 epochs further improves the performance, rising to 0.93 for F1 score, 0.93 for accuracy, 0.95 for precision, and 0.91 for recall.

In a neural network architecture, the number of layers also affects the model's performance. Using a single layer results in a score of 0.90 for F1, 0.90 for accuracy, 0.92 for precision, and 0.98 for recall. Adding an additional layer (2 layers in total) improves the performance, having a 0.92 F1 score, 0.94 precision, 0.90 recall, and 0.92 accuracy.

These findings emphasize the significance of cautiously choosing and fine-tuning the hyperparameters to optimize the effectiveness of the BharatFakeNewsTracker technique. The suggested technique can achieve higher accuracy and better distinguish between fake and real news articles by finding the optimal combination of hyperparameters. Overall, the proposed model demonstrates promising performance, and further optimization of hyperparameters can lead to even better results.

Table 4. Performance comparison of the proposed method during ablation studies

Ablation Component	Accuracy	Precision	Recall	F1 Score
Proposed Model	0.91	0.94	0.92	0.93
Without Deep Learning	0.89	0.91	0.88	0.89
Without Decision Tree	0.90	0.92	0.90	0.91
Without Random Forest	0.88	0.89	0.87	0.88

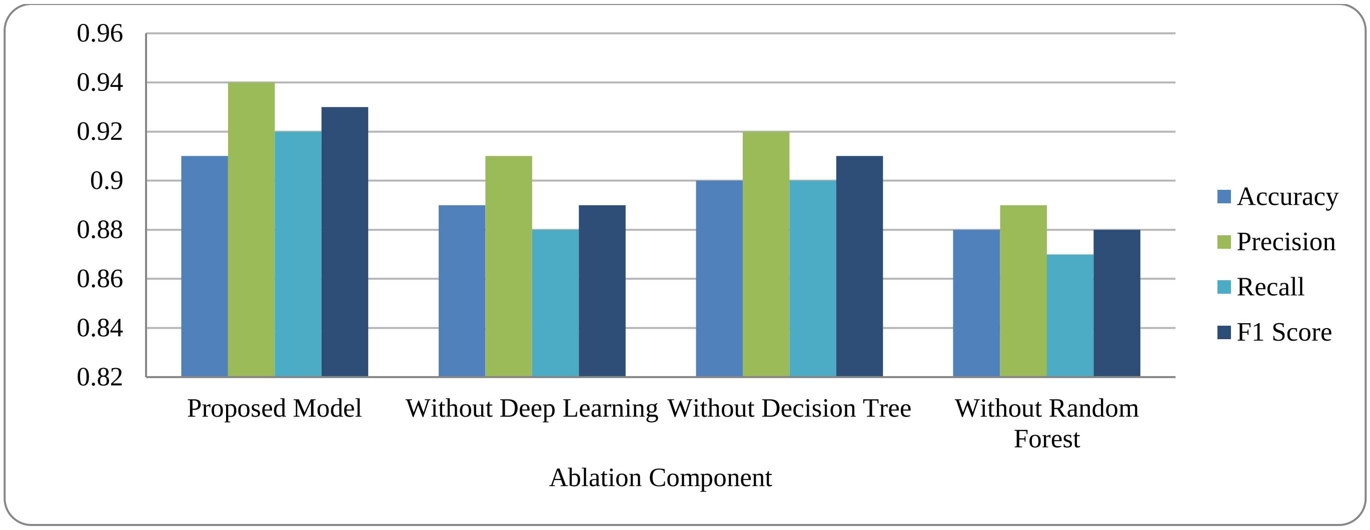

Fig. 4 Performance comparison of the proposed method during ablation studies

5.3. Performance Comparison of the Proposed Model During Ablation Studies

In the ablation studies, various components of the proposed BharatFakeNewsTracker model were analyzed to assess how they affect the overall performance. The proposed model represents the complete model with all its components intact. Additionally, three variations were considered: the first without the Deep Learning model, the second without the Decision Tree model, and the third without the Random Forest model.

Table 4 presents the performance results obtained from the ablation studies conducted for the proposed model. Furthermore, Fig. 4 provides a visual representation of how the performance metrics fluctuate based on the model's inclusion or exclusion of specific components.

The model's performance slightly decreases when the Deep Learning component is removed. It decreases to 0.89 for the F1 score, 0.91 for precision, 0.88 for recall, and 0.89 for accuracy. This suggests that the Deep Learning component plays a role in improving the model's capacity to concentrate on significant characteristics and generate more accurate predictions.

Similarly, when the Decision Tree component is excluded, the model's performance declines a little bit as compared to the Deep Learning component. The F1 score drops to 0.91, recall to 0.90, precision to 0.92, and accuracy to 0.90. This indicates that making use of the Decision Tree component does not affect the model's performance so much as does the Deep Learning component. However, when the Random Forest component is removed, the model's performance takes a shear setback. The F1 score decreased appreciably to 0.88, recall to 0.87, precision to 0.89, and accuracy to 0.88. This signifies that this component plays a critical role in the model's overall performance.

5.4. Proposed Model's Sensitive Analysis

Table 5 presents an overview of the sensitivity analysis findings of the proposed BharatFakeNewsTracker (BFNT) model. A graphic depiction of how the suggested method's performance changes when various sensitivity factors are considered is shown in Fig. 5.

When considering varying dataset sizes, the model achieves higher accuracy, precision, recall, F1-score, and AUC-ROC values with larger datasets, indicating its ability to generalize better with more data. As noise levels increase, a gradual decline is observed in performance metrics. This suggests that the BFNT model is susceptible to noise, impacting its ability to detect fake news accurately.

In scenarios involving adversarial attacks, the model's performance drops significantly. The BFNT model is particularly sensitive to adversarial attacks, showcasing reduced accuracy, precision, recall, F1-score, and AUC-ROC values in their presence. Additionally, the model's performance is influenced by the distribution of classes in the dataset. Higher imbalances in class distribution led to slightly decreased performance across all metrics.

Table 5. Performance comparison during sensitivity analysis of the proposed model

Sensitivity Factor	Accuracy	Precision	Recall	F1-Score
Small Dataset	0.89	0.91	0.87	0.89
Large Dataset	0.91	0.94	0.92	0.93
Low Noise	0.92	0.94	0.91	0.92
High Noise	0.87	0.88	0.85	0.87
No Adversarial Attacks	0.91	0.94	0.92	0.93
Adversarial Attacks	0.82	0.81	0.83	0.82
Balanced Distribution	0.9	0.92	0.88	0.9
Imbalanced Distribution	0.87	0.89	0.86	0.87

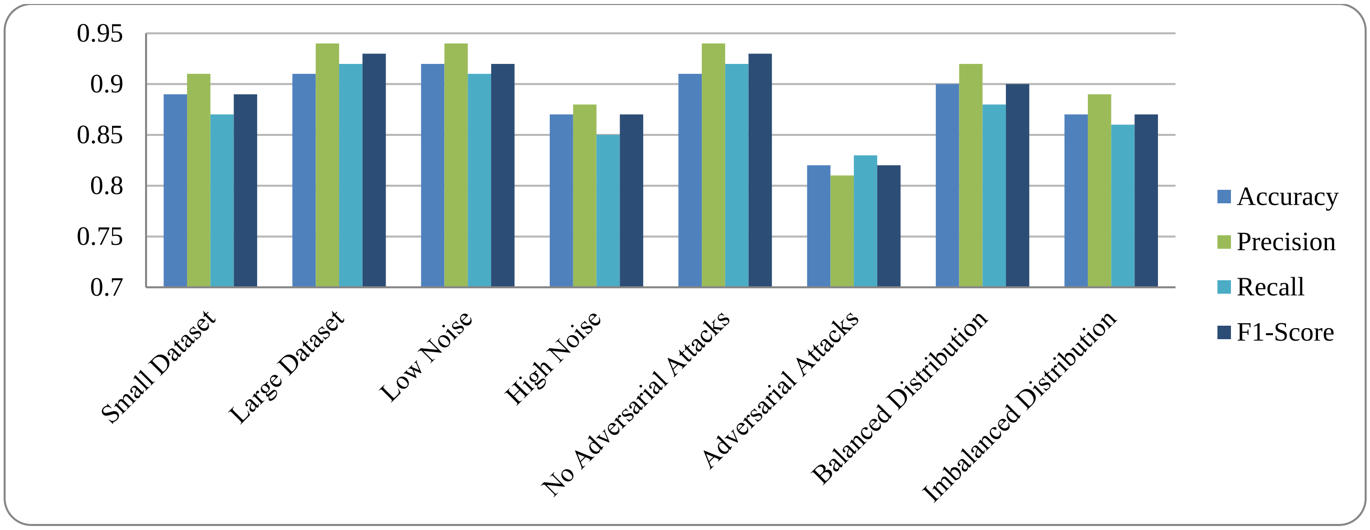

Fig. 5 Performance comparison during sensitivity analysis of the proposed model

Overall, the sensitivity analysis highlights the BFNT model's strengths in handling larger datasets with balanced distributions.

The findings from the sensitivity analysis provide valuable insights for the model's practical application. They assure users that the BharatFakeNewsTracker model maintains reliable performance and can be relied upon to detect fake news accurately, even in the presence of noise or adversarial attacks in the input data.

5.5. Comparison with Existing Work

The performance comparison of the proposed model with that of the existing methods is provided in Table 6, and its corresponding graphical comparison is given in Fig 6. The field of fake news identification has benefited greatly from earlier investigations. An ensemble architecture built on gradient boosting was proposed by Kaliyar and Goswami [14] and achieved 0.85 accuracy, 0.88 precision, 0.82 recall, and 0.85 F1 score. Similar to this, Huang et al. [15] focused on cross-domain adaptation and used deep learning and ensemble models to reach 0.87 accuracy, 0.89 precision, 0.85 recall, and 0.87 F1 score. Aggarwal et al.'s [16] focus on feature extraction and credibility score computation, using various machine-learning techniques, produced 0.88 accuracy, 0.87 precision, 0.89 recall, and 0.88 F1 score.

In contrast, the suggested model performs exceptionally well, earning 0.91 accuracy, 0.94 precision, 0.92 recall, and 0.93 F1 score. The suggested model is taken to new levels of accuracy and efficiency because of a specific focus on tackling the complex issues provided by fake news in the Indian context and using cutting-edge natural language processing and machine learning techniques. The thorough performance comparison shows the proposed model's superiority and highlights its potential to solve the complexities of false news detection significantly.

Table 6. Performance comparison of the proposed model with existing methods

Experiment	Accuracy	Precision	Recall	F1 Score
Kaliyar and Goswami [14]	0.85	0.88	0.82	0.85
Huang et al. [15]	0.87	0.89	0.85	0.87
Aggarwal et al. [16]	0.88	0.87	0.89	0.88
Proposed Model	0.91	0.94	0.92	0.93

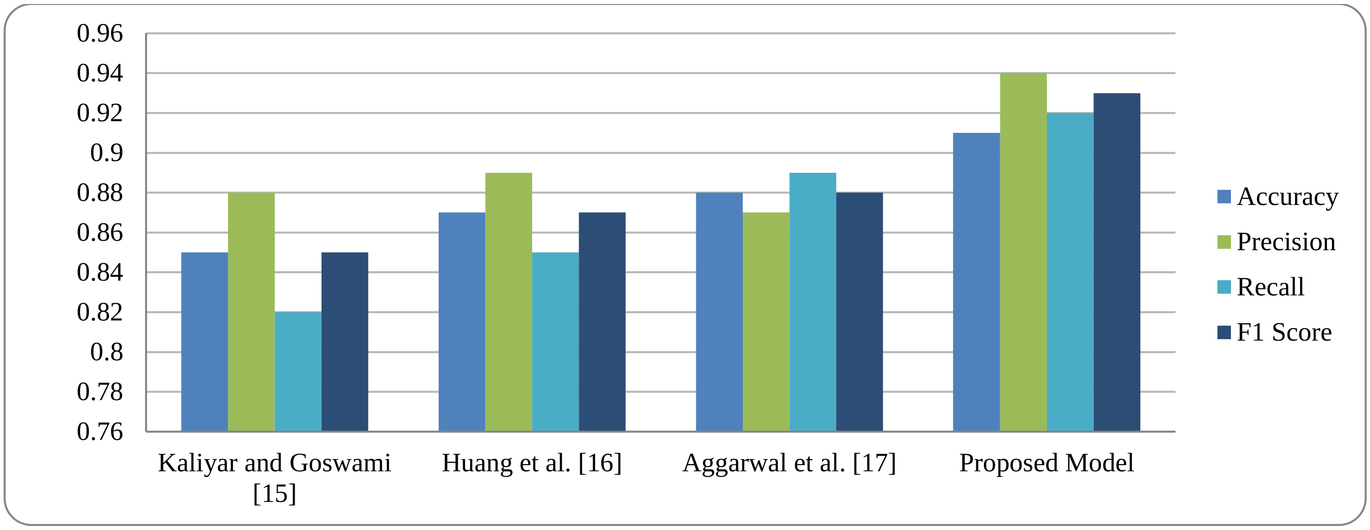

Fig. 6 Performance comparison of the proposed model with existing methods

5.6. SWOT Analysis of the Proposed Method

A thorough evaluation is offered by the SWOT (Strengths, Weaknesses, Opportunities, and Threats) study of the proposed model for BharatFakeNewsTracker. It evaluates the internal and external factors influencing the model's success and effectiveness in detecting fake news.

5.6.1. Strengths

One of the key strengths of the proposed model is its ensemble approach, combining deep learning, decision trees, and random forest techniques. This ensemble model benefits from the strengths of each individual technique, resulting in improved accuracy and robustness.

Additionally, the use of deep learning permits the model to extract complicated representations and patterns from the input data, improving the model's capacity to identify complex fake news stories.

5.6.2. Weaknesses

The model's performance may be affected if adversarial attacks are introduced in the input to the model. It needs to be improved to perform much better in the presence of adversarial attacks.

5.5.3. Opportunities

The proposed model presents several opportunities for further improvement and development. One opportunity lies in the continuous enhancement of the training dataset. By incorporating a diverse and extensive collection of labeled fake and real news articles, the model can further improve its accuracy and generalization capabilities.

Additionally, ongoing research and advancements in natural language processing and machine learning techniques offer opportunities for incorporating more sophisticated algorithms and features into the model.

5.6.4. Threats

The dynamic nature of false news is one of the possible risks to the suggested approach. As fake news techniques evolve, the model may face challenges in keeping up with new strategies employed by malicious actors. It requires continuous monitoring and updating to address emerging patterns and trends in generating fake news. Furthermore, the availability of large-scale labelled datasets, which are essential for training and evaluating the model, can be limited and may pose a scalability challenge.

6. Conclusion

This study provides a thorough and practical method to detect fake news through a proposed BhratFakeNewsTracker model. The ensemble model demonstrates improved accuracy and robustness in identifying deceptive information by integrating deep learning, decision trees, and random forest techniques. The experiments conducted during the evaluation phase highlight the model's high performance across various metrics, comprising F1 score, recall, accuracy, and precision.

The comparative analysis demonstrates the superiority of the proposed model over traditional algorithms for machine learning, including Logistic Regression, Support Vector Machine, and Random Forest. Additionally, the model's performance is evaluated against the most advanced techniques, such as bagging, boosting (XGBoost), convolutional neural networks (CNNs)[30, 32], and Recurrent Neural Networks (RNNs) [30, 32]. The suggested model consistently outperforms these techniques, showcasing its effectiveness in detecting fake news.

A series of experiments conducted with different hyperparameter settings showcase the impact of layers, epochs, learning rate, and batch size on the model's performance. Optimal hyperparameter combinations are identified through careful tuning, leading to improved F1 score, recall, accuracy, and precision. Ablation studies provide important insights regarding the contribution of different components of the proposed model by evaluating the model's performance with and without attention. Sensitivity analysis is performed to evaluate the model's robustness and generalizability.

By introducing noise and adversarial attacks to the input data, the model's response is observed, allowing for an assessment of its sensitivity to variations in the input. This investigation validates the capability of the model to maintain its performance even in the presence of minor disturbances. Mechanisms and pre-trained language models, the importance of these components in achieving optimal performance is assessed.

Overall, the proposed BharatFakeNewsTracker model demonstrates strong potential in combating the proliferation of fake news. The model achieves high accuracy and robustness by utilizing the strengths of ensemble learning, deep learning techniques, and feature extraction. While some limitations and challenges exist, such as the dynamic nature of fake news and the availability of labeled datasets, ongoing research and continuous improvement can further enhance the model's effectiveness.

The proposed model holds promise for real-world applications in identifying and combating fake news, contributing to the preservation of information integrity, and fostering a more informed society. It is advised to carry out more studies and assessments to remedy the noted shortcomings and to adapt the model to evolving trends in the fake news landscape.

Acknowledgment

The authors would like to express their sincere gratitude to Er. Pankaj Singh, DBA (Swiss School of Business and Management, Geneva), for his invaluable support during the current research. Further, the authors would like to extend special thanks to Mr. Lavkush Gupta for technical assistance when required.

References

[1] Kai Shu et al., “Fake News Detection on Social Media: A Data Mining Perspective,” *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22-36, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] João Pedro Baptista, and Anabela Gradim, “Understanding Fake News Consumption: A Review,” *Social Sciences*, vol. 9, no. 10, p. 185, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xinyi Zhou, and Reza Zafarani, “A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities,” *ACM Computing Surveys*, vol. 53, no. 5, pp. 1-40, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xiang Zhang, Junbo Zhao, and Yann LeCun, “Character-Level Convolutional Networks for Text Classification,” *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 649-657, 2015. [Google Scholar] [Publisher Link]

[5] Martin Potthast et al., “A Stylometric Inquiry into Hyperpartisan and Fake News,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 231-240, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Soroush Vosoughi, Deb Roy, and Sinan Aral, “The Spread of True and False News Online,” *Science*, vol. 359, no. 6380, pp. 1146-1151, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jieun Shin et al., “The Diffusion of Misinformation on Social Media: Temporal Pattern, Message, and Source”, *Computers in Human Behavior*, vol. 83, pp. 278-287, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Carlos Castillo, Marcelo Mendoza and Barbara Poblete, “Information Credibility on Twitter,” *Proceedings of the 20th International Conference on World Wide Web*, pp. 675-684, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[9] William Yang Wang, “Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 422-426, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] Aswini Thota et al., “Fake News Detection: A Deep Learning Approach,” *SMU Data Science Review*, vol. 1, no. 3, 2018. [Google Scholar] [Publisher Link]

[11] Natali Ruchansky et al., “CSI: A Hybrid Deep Model for Fake News Detection,” *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 797-806, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari, “Fake News Identification on Twitter with Hybrid CNN and RNN Models,” *Proceedings of the 9th International Conference on Social Media and Society*, pp. 226-230, 2018. [CrossRef] [Google Scholar][Publisher Link]

[13] Iftikhar Ahmad et al., “Fake News Detection Using Machine Learning Ensemble Methods,” *Complexity*, vol. 2020, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang, “Multiclass Fake News Detection Using Ensemble Machine Learning,” *IEEE 9th International Conference on Advanced Computing (IACC) IEEE*, pp. 103-107, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yin-Fu Huang, and Po-Hong Chen, “Fake News Detection Using an Ensemble Learning Model Based on Self-Adaptive Harmony Search Algorithms,” *Expert Systems with Applications*, vol. 159, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Arush Agarwal, and Akhil Dixit, "Fake News Detection: An Ensemble Learning Approach," *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE*, pp. 1178-1183, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[17] Bharat Fake News Kosh. [online]. Available: https://bharatfakenewskosh.com/datasets/
[18] Mauik Panchal, and Rutika Ghariya, "A Review On Detection of Fake News Using Various Techniques," *SSRG International Journal of* Computer *Science and Engineering*, vol. 8, no. 6, pp. 1-4, 2021.[CrossRef] [Publisher Link]
[19] Jair Cervantes et al., "A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends," *Neurocomputing*, vol. 408, pp. 189-215, 2020.[CrossRef] [Google Scholar] [Publisher Link]
[20] Derek A. Pisner, and David M. Schnyer, "Support Vector Machine," Machine *Learning*, pp. 101-121, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[21] Liaqat Ali et al., "An Optimized Stacked Support Vector Machines Based Expert System for the Effective Prediction of Heart Failure," *IEEE Access*, vol. 7, pp. 54007-54014, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[22] Matthias Schonlau, and Rosie Yuyan Zou, "The Random Forest Algorithm for Statistical Learning," *The Stata Journal*, vol. 20, no. 1, pp. 3-29, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[23] Anjaneyulu Babu Shaik, and Sujatha Srinivasan, "A Brief Survey on Random Forest Ensembles in Classification Model," *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018*, vol. 2, pp. 253-260, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[24] Mumtaz Ali et al., "Complete Ensemble Empirical Mode Decomposition Hybridized with Random Forest and Kernel Ridge Regression Model for Monthly Rainfall Forecasts," *Journal of Hydrology*, vol. 584, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[25] Ekaba Bisong, "Logistic Regression," *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 243-250, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[26] Tae-Hwy Lee, Aman Ullah, and Ran Wang, "Bootstrap Aggregating and Random Forest," *Macroeconomic Forecasting in the Era of Big Data*, pp. 389-429, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[27] Raveendranadh Bokka, and Tamilselvan Sadasivam, "Securing IoT Networks: RPL Attack Detection with Deep Learning GRU Networks," *International Journal of Recent Engineering Science*, vol. 10, no. 2, pp. 13-21, 2023. [CrossRef] [Publisher Link]
[28] Tianqi Chen, and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016. [CrossRef] [Google Scholar] [Publisher Link]
[29] Hui Jiang et al., "Network Intrusion Detection Based on PSO-XGBoost Model," *IEEE Access*, vol. 8, pp. 58392-58401, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[30] Rikiya Yamashita et al., "Convolutional Neural Networks: An Overview and Application in Radiology," *Insights into Imaging*, vol. 9, pp. 611-629, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[31] Varad A. Sarve, Swati S. Sherekar and Vilas M. Thakare, "Learning the Channel Uncertainty for Defensive Security Enhancements in MANET with Trust Management," *SSRG International Journal of Mobile Computing and Application*, vol. 4, no. 1, pp. 21-27, 2017.[CrossRef] [Publisher Link]
[32] Ilya Sutskever, James Martens, and Geoffrey Hinton, "Generating Text with Recurrent Neural Networks," *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 1017-1024, 2011. [Google Scholar]