



Classifying Misinformation of User Credibility in Social Media Using Supervised Learning

Muhammad Asfand-e-Yar^{1,*}, Qadeer Hashir^{1,*}, Syed Hassan Tanvir¹ and Wajeeha Khalil²

¹Department of Computer Science, Center of Excellence in Artificial Intelligence (CoE-AI), Bahria University, Islamabad, 44000, Pakistan

²Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, 25000, Pakistan

*Corresponding Authors: Muhammad Asfand-e-Yar. Email: asfandeyar.buic@bahria.edu.pk; Qadeer Hashir. Email: hashir.isd@gmail.com

Received: 26 July 2022; Accepted: 07 January 2023

Abstract: The growth of the internet and technology has had a significant effect on social interactions. False information has become an important research topic due to the massive amount of misinformed content on social networks. It is very easy for any user to spread misinformation through the media. Therefore, misinformation is a problem for professionals, organizers, and societies. Hence, it is essential to observe the credibility and validity of the News articles being shared on social media. The core challenge is to distinguish the difference between accurate and false information. Recent studies focus on News article content, such as News titles and descriptions, which has limited their achievements. However, there are two ordinarily agreed-upon features of misinformation: first, the title and text of an article, and second, the user engagement. In the case of the News context, we extracted different user engagements with articles, for example, tweets, i.e., read-only, user retweets, likes, and shares. We calculate user credibility and combine it with article content with the user's context. After combining both features, we used three Natural language processing (NLP) feature extraction techniques, i.e., Term Frequency-Inverse Document Frequency (TF-IDF), Count-Vectorizer (CV), and Hashing-Vectorizer (HV). Then, we applied different machine learning classifiers to classify misinformation as real or fake. Therefore, we used a Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT), Gradient Boosting (GB), and K-Nearest Neighbors (KNN). The proposed method has been tested on a real-world dataset, i.e., "fakenewsnet". We refine the fakenewsnet dataset repository according to our required features. The dataset contains 23000+ articles with millions of user engagements. The highest accuracy score is 93.4%. The proposed model achieves its highest accuracy using count vector features and a random forest classifier. Our discoveries confirmed that the proposed classifier would effectively classify misinformation in social networks.

Keywords: Misinformation; user credibility; fake news; machine learning



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

In the last decade, online social networks have rapidly grown with great success. Social networks changed the way of News distribution as was used by traditional media. Social Media is the new ecosystem for spreading News or information. People are using social media mostly as compared to mainstream media. Through using online social platforms, unlike print media or electronic media, anyone can share or spread any kind of information or News [1]. Today, the misinformation dissemination rate is high on social forums and is very common [2]. Ease of access, low-cost internet facilities, high bandwidth, and quicker results have made social networks an essential part of our daily lives. Instead of switching on to TV, people are getting News alerts on their handheld gadgets [3]. Youth spend most of their time on social networks and other websites related to their interest. Hence, social networks have become the primary source of information. People are willing to like, comment, or share news and data on social websites. A survey was conducted on spreading News or information on social platforms in 2016 and 2012. The reports showed that in 2016, 62% of youth read News; however, it was 49% in 2012 [4]. It is very significant for users on a social network to get actual information. Today, misinformation is well-presented in articles or text so that the users believe in it. In 2020, EU DisInfoLab exposed the extensive network designed to dishonor Pakistan internationally by spreading misinformation on a social network. Social media growth created many challenges, and finding actual information is one of them. Social networks are full of News and information, and it is challenging to differentiate the real and fake information [5]. The spread of misinformation has affected people at different levels of life. Misinformation misleads the audience to gain profit, i.e., political purposes, misguidance, organizational benefits, and state conflicts, to implement their plan. There are many terms for misinformation, such as spam, misinformation, fake News, and others. These types of misinformation can be in the form of textual, numerical, numbers and images, or other formats. Misinformation can interrupt the validity balance of the News Ecosystem. It proves that misinformation or fake News is extensively being spread on Facebook compared to the general authentic mainstream News, e.g., US political voting in 2016 [6]. This research uses various supervised learning techniques and datasets to detect misinformation on a social platform. After identifying the misinformation using the datasets, we tried to categorize the user's credibility. We used Twitter data from the Fakenewsnet dataset. This dataset contains News content and context, for example, engagement of users in different articles and user relations. For the News context, we extracted data, i.e., Tweet, Retweet, Like, and Reply, as shown in Fig. 1. Then, we combined the News context with the News contents and applied the feature extractor and classifier; this helps to identify the misinformation on social media using supervised learning, as shown in Fig. 2. We calculated user credibility and combined this with the article's content. We used Machine Learning (ML) algorithms on our refined custom dataset to classify real and misinformation based on both article's content and the user's context.

2 Related Work

In the related work, various methods of detecting misinformation have been discussed based on the study of multiple research articles. The analysis of the following discussion is presented in Table 1.

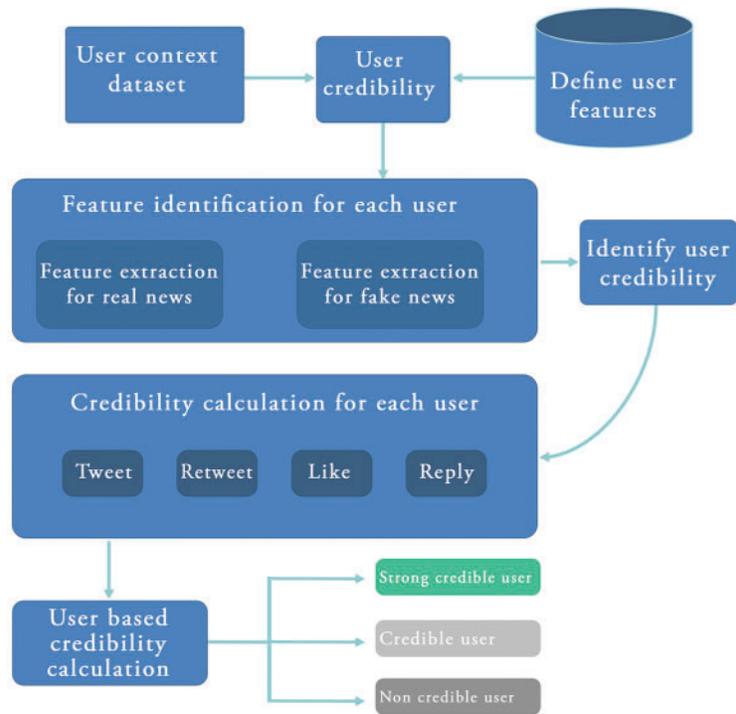


Figure 1: Context features extraction model

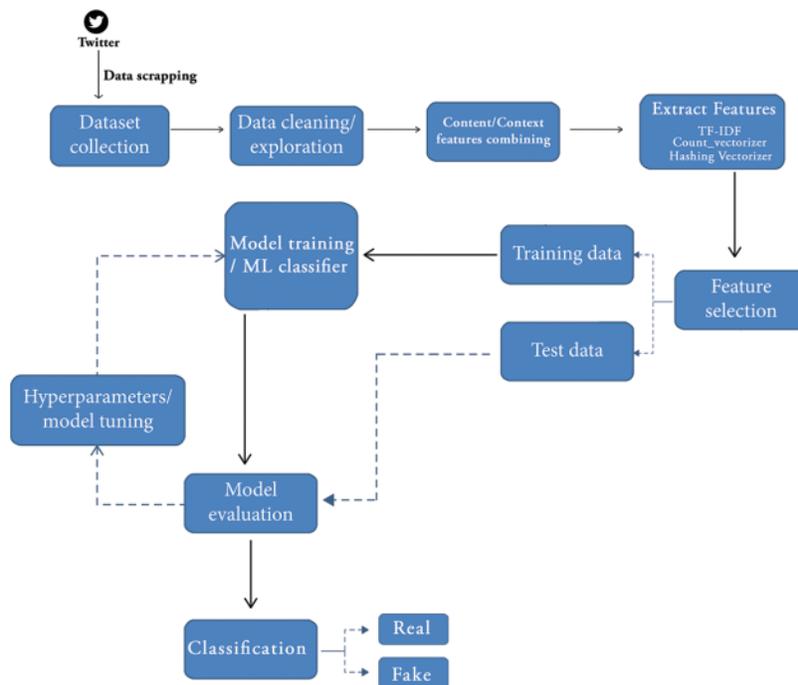


Figure 2: Misinformation classification model

2.1 Machine Learning Classifiers

Smitha et al. [1] used Natural Language Processing (NLP) and Machine Learning algorithms. They applied TF-IDF and Count Vector for feature extraction and seven different ML algorithms used for classification with the Kaggle dataset. SVM and Linear classification algorithm with TF-IDF feature extraction combined achieved more accurate results. In [5], the authors worked on a content-based method and proposed graph-based semi-supervised misinformation recognition, which showed better results. The authors used KNN with the Facebook News posts dataset and showed an accuracy of 79%. Dementieva et al. [6] follow different hypotheses and target five languages, i.e., English, Spanish, French, German, and Russian. In [7], authors used an RF, DT, and Adaboost classifier to detect fake News from social media. Random forest gives better results, and all their results accuracy is under 80%. In another study [8], they worked on ML ensemble approaches on four different real-world datasets. The following algorithms were used for the experiment, i.e., Linear Regression (LR), Multilayer Perceptron (MLP), SVM, KNN, Ensemble Learners, Bagging Ensemble Classifiers, RF, Voting Ensemble Classifier, Boosting Ensemble Classifiers, Benchmark Algorithms and Convolutional Neural Network (CNN). KNN classifier gets 88% accuracy on dataset one, but dataset two only gets 28% accuracy. The paper [9] uses one class of SVM for discovering real and fake News, and they use the Latent Semantic Analysis (LSA) technique to reduce the original data. Research shows 86% accuracy and 94% precision. In [10], the method depends on unsupervised, semi-supervised, and supervised learning. The paper also used a Deep Neural Network (DNN) for false information classification. Nyow et al. [11] altered social media Twitter data using a DT, NB, LR, and SVM for false News discovery. The paper [12] used an extra tree classifier, RF, and DT for fake News discovery and used ISOT and Lair datasets. The linguistic-based textual features were used for the results. Pranav et al. [13] used five linguistic features groups and investigated with Recurrent Neural Network (RNN) vs. Naïve Bayes and RF with the TF-IDF and Kaggle dataset. In [14], the author used a manually labeled twitter dataset for content and context features. The paper selected four categories of six models to represent the week-learning to compare the results. In the first category, NB was included; in the second category, the KNN model was included; for the third category, DT and RF; and for the fourth SVM model was used. In [15], researchers examined the possessions of content signifiers in assisting the human factor counter to disinformation attacks through its acknowledgment. This research showed the changing outcome of contextual evidence in people recognizing fake News. In [16], an obvious fake News test was conducted to measure the users how they use these forms as contextual signs in classifying some contents as either genuine or ambiguous.

2.2 Supervised Learning

In [17], authors identified three challenges, false News classification, domain identification, and bot identification. Deep Learning (DL) model is used to recognize stylistic differences between legitimate and false News articles. In another paper [18], a semantic model was developed, which determined the complete social context of an article on social media. Monther et al. [19] proposed a solution for a search engine to remove those websites on search engines that spread misinformation. The paper [20] introduced the misinformation definition and examined the difference between context-based and content-based features. The study [21] is based on group behaviors that selected features such as claim similarity, account created time, messages timesteps, number of followers, number of follows, and reputation. RF, SVM, DT, and neural networks are used for classification. The authors [22] used a DNN to do a user classification on the Twitter covid dataset and used the Bidirectional Encoder Representations from Transformers (BERT) encoder to handle the vocabulary word splitting. De et al. [23] studied past inscription impressions on a parallel point from constant ages and recovered

the results by counting the portions of rational inspection and past outcomes that did not do before this search. In [24], the authors applied only article text and used BERT as a neural network, and different ML classifiers were also used, such as SVM, NB, and KNN classifiers.

2.3 *Semi-Supervised Learning*

Benamira et al. [2] work on a content-based method for recognizing false information. They planned a semi-supervised graph-based fake News exposure method; this method is based on a graph neural network. The experiments showed better results as compared to other recognition methods. The model was based on a semi-supervised false News discovery technique, an illustration learning technique. In [25], authors labeled the unlabeled dataset using CNN for low-level feature extraction. The authors used shared CNN for unlabeled & labeled data and Word CNN based on the Semi Supervised Learning (SSL) model to test two different datasets, i.e., PHEME and LIAR datasets.

2.4 *Neural Network CNN, RNN-LSTM*

Goonathilake et al. [3] used the NLP approach, a combination of DL networks such as CNN and Long short-term memory (LSTM). The authors scored an accuracy of 92%. Ghinadya et al. [4] proposed a model built on RNN and Bidirectional Long Short-term Memory (b-LSTM) using content-based feature. The model showed 0.2096 and 0.2423 F1 scores. In [8], Linear SVM is used for comparison and auto-fake News detection; the authors used CNN. Xishuang et al. [25] used CNN with unlabeled dataset low-level feature extraction. To test two different datasets authors used shared CNN and Word CNN for unlabeled & labeled data, i.e., PHEME and LIAR datasets. Verma et al. [26] used LSTM, RNN, and Grated Recurrent Units (GRU) with the tensor board to visualize the neural network. The LSTM gave 94% accuracy, and the GRU model showed 91% accuracy. Kaliyar et al. [27] designed the DNN containing a diverse number of filters with five dense layers for their model. The results showed an accuracy of 92.30% with the dataset of BuzzFeed and PolitiFact. Barua et al. [28] used an ensemble model, Bidirectional Recurrent Neural Network (B-RNN), with LSTM and GRU for determining the sanctity. The experiment showed 80.2% accuracy. Ahmed et al. [29] described text-based detection approaches, used TF-IDF with N-Gram for feature extraction, and used different ML classifiers, i.e., Stochastic Gradient Descent (SGD), SVM, LSVM, KNN, and DT. Abdullah et al. [30] used CNN and LSTM to identify the misinformation within the Kaggle dataset based on three features (i.e., author's name, domain name, and article title). In [31], authors included both social content and social context features for generating comments Collective Learning Systems (CLS) and the Generative Pre-trained Transformer (GPT) methods were used. Initially, the authors used a sample dataset to create comments. The authors removed original comments and replaced them with generated comments in the classification part.

In [32], a dataset was used to pick 200 articles for the model. In [33], the TF-IDF and count vectorization were used for feature extraction after Deep Semantic Similarity Model (DSSM). Word hashing was done in the first layer of DSSM for the feature extraction, and improved RNN characterized the classification task. In [34], authors found medical misinformation on social networking websites; CNN model was used for two features. The first feature was the News content, i.e., comment date publication status, and the second feature was author identity, i.e., login name, gender, and other. In [35], BiLSTM subtasks and Capsule Neural Network (Capsnet) models were used for article content and title modeling. The UTK ML club was used to collect the dataset for the experiment. In [36], authors proposed the framework named Embeddings from Multilingual-Encoder Transformer (EMET). Their technique controls text input from bilingual encoder modifiers. Jiang et al. [37] used ML classifiers, such as LR, SVM, k-NN, DT, and RF, and also used Deep Networks, i.e., CNN, GRU,

and LSTM, for misinformation detection. Shu et al. [38] propose a verdict comment co-consideration sub-network to achieve user comments and new article content. In [39], authors used Capsnet for short News, and static word embedding was used for medium News on two datasets. An analytical review with limitation of previous work is discussed in the [Table 1](#).

Table 1: Analytical table of recent studies

Ref.	Techniques	Results	Limitations
[2]	AGNN (Attention Graph-Neural Network, Fast Belief-Propagation)	They discover that the GCNN and AGNN better when used for new articles, with 85% accuracy.	The results can be improved more using more baseline approaches with a large dataset.
[5]	K-Nearest-Neighbor classifier	The authors present a simple approach to detect misinformation on social sites by using KNN and get an accuracy of 79%.	More classifiers can improve the results paper only using one classifier for the classification.
[11]	SVM, Naïve Bayes, DT	The paper uses a different machine-learning model to classify misinformation on the bases of Twitter data	This paper has limitations due to the restraints set by Twitter and its Application Programming Interface (API).
[12]	DT, RF, extra tree classifier	The results show that ET and RF classifiers have improved prediction accuracy before feature extraction than DT classifiers.	The accuracy of different datasets improves with large datasets and t hyperparameter tuning models based on meta-heuristic algorithms.
[13]	Bigram, random forest, TF and TFIDF, N-grams, Naive Bayes, GLOVE, RNN	TF and TFIDF result in similar accuracies RNN with GloVe outperforming quad-grams but with lower accuracy than bi-grams and trigrams.	Semantic features may be joined with other linguistic cues and metadata to expand the detection results.
[27]	Tensor factorization. DNN	Results show this model with the accuracy of 92.30% with the dataset of BuzzFeed and PolitiFact.	The dataset is minimal to check the model results, and the results deviate with a large dataset.
[28]	LSTM, GRU	The highest accuracy achieved by the ensemble model is 80.2%.	Need to run the model on some other dataset to check more reliability.

(Continued)

Table 1: Continued

Ref.	Techniques	Results	Limitations
[31]	Neural network, NLP, deep learning	The grover model provides better results with a neural network.	Generating comment classification provides some irrelevant comments.
[34]	Pre-trained CNN	The model detects automatic medical misinformation and achieves 91% accuracy.	Only use WordPress website data research can be improved by using a different dataset.

3 Dataset

The fakenewsnet dataset repository was used in this research for the experiment. We require a larger dataset with user engagements in the form of user tweets, re-tweet, like, and replies. The data repository provided all these features; hence we downloaded the raw form of the dataset via provided twitter script. Afterward, the data was refined according to our requirements.

3.1 Refining Dataset

The dataset was in script format, and only script code was provided. Twitter API keys were used to collect the dataset from Twitter. Therefore, the Python script was used to manage the dataset. The dataset repository contains two types of data, i.e., Politifact and Gossipcop. After downloading the datasets, they were not completing our requirements; therefore, preprocessing of the dataset was done. The downloaded dataset contained different files and folders.

For example, an article contains sub-files and folders containing news content, user context, and user relation files. Each article has a News content.json file, retweets folder, tweets folder, like folder, and reply folder. Similarly, each article has user context-related folders containing different JSON files for each user who engage with the article. All information of different users involved in tweets is stored in the user timeline tweets folder. The folders downloaded are; News content, Tweets folder, retweets folder, like, reply, user profiles folder, and user timeline tweets.

3.2 Finalized Dataset

To experiment on our dataset, we did preprocessing, and the downloaded dataset was not in proper form. We removed unnecessary data from the raw dataset and combined fake and real article content. We also combined the user engagement in the form of tweets, re-tweet, like, and replies in one .csv file. Our finalized dataset contains the following fields (Article id, Title, Text, Tweet ID, User Id, Verified, Tweet, Retweet, Reply, Forward, Label). The finalized dataset contains 23000+ articles with millions of user engagement.

4 Methodology

Nowadays, we have a lot of text data on social networks, and it is difficult to read and recommend whether it is real or fake. In Section 4, we will discuss different models and feature extraction techniques used to predict Misinformation on social media and describe the architecture of the models. We deployed three extraction features and different ML classifiers. Initially, we discuss the refinement of the dataset. Then discussed, the proposed methodology in detail, i.e., context feature extraction and information preprocessing, including feature extraction and classification.

4.1 Models Implementation

We have used nine different models in our study: NB, Multinomial Naive Bayes, Gaussian Naive Bayes, DT, K-NN, LR, GB, SVM, and RF, for the Misinformation classification. We used different bagging and boosting techniques to check different model performances with hyperparameter changes.

4.2 Evaluation Measure

For the results, we generate a classification report in our study, and the evaluation measures include accuracy score, F1-score, precision, and recall on both content and context results. All the above measurements are used for the model results evaluations.

4.3 Testing & Training Dataset

The training dataset is downloaded from Twitter (details mentioned in Section 3). After pre-processing, all the News datasets used for training contain 23000+ articles; the division is shown in Table 2. Two datasets used for this experiment have fake and real articles with millions of user engagements. To analyze the proposed methodology, two distinctive datasets are used. The motivation to utilize two datasets is that each dataset has its challenges which assist us with analyzing the proposed methodology better. The dataset is divided into training and test cases, 67% is used to train the models, and 33% is used to test the proposed model.

Table 2: The articles dataset division

Training set	15410 articles
Testing set	7590 articles

4.4 Context Features Extraction

For context features, we calculate user credibility based on user engagement with real and misinformation. We calculate the user credibility score for every user engaged with articles based on article tweets, retweets, likes, and replies. As discussed below:

- If the user engages in real articles, their score increases by 0.1.
- If the user retweets the article, their score increases by 0.2.
- If the user liked the article, then their point increased by 0.1.
- If the user replies the article, then their point increase by 0.1.
- If a user tweets a fake article, their point increase by 0.1.
- If retweets the fake article, their point increase by 0.1.
- If, like the fake article, their point increase by 0.1.
- If the user replies the fake article, their point increase by 0.1, and so on.

Some users lay among strong, credible, and non-credible users. The strong, credible users are shown with the green color, credible users (who were engaged with most of the real articles) are shown with gray color, and non-credible users are shown with dark gray color. High-credible users are less likely to spread misinformed data, and less credible users are more likely to spread misinformation [40]. In our research, we observe that 54.3% of users are strongly credible, 15.4% of users are credible, and 30.3% are not credible, as shown in Table 3. This reveals that the user context is a significant feature

for the classification of real and misinformation. After identifying the credibility of users, we combine these features with News content and classify real and misinformation as shown in Figs. 1 and 2 (i.e., Fig. 2).

Table 3: Percentage of user credibility score

Strong credible	54.3%
Credible	15.4%
Non-credible	30.3%

4.5 Input Preprocessing

The dataset repository contains two sub-datasets which include 23000 articles from different publications. In our research, we used different content files containing the title and description of an article. Initially, we preprocessed to clean the websites' text, email addresses, and punctuation and converted the text into lowercase. Then, we used the function that removes all stop words from the text. Afterward, we used the data from the content and text columns because other columns were not required. The next step was to tokenize the words, which is a process to make the text a useful word. After preprocessing, we did feature extraction to extract useful features for classifiers.

4.6 Feature Extraction

In ML or NLP, to solve any text base problem, it needs to convert into a form that can be demonstrated. The ML models do not work with raw datasets, and the text needs to be converted into numbers to apply classification. This type of text conversion is called text classification. We used different techniques to extract valuable features in content.

We use different Bag Of Words (BOW) techniques for feature extraction for our model. BOW is a popular method used for feature extraction on the content dataset. We can characterize a verdict as a BOW vector. We first build a vocabulary from all unique words in the dataset, take each of them, and mark their existence. We used N-gram because our dataset is very large, and the vocabulary size increased. The corpus is very large, and the length of the vector contains many positions; hence to decrease vocabulary, we converted the data to the vocabulary of grouped words by using N-gram. This allows the BOW to capture a bit more meaning from the document. We used three different feature extraction methods HV, CV, and TF-IDF. TF-IDF [41] is the calculation of the dot product of TF, and IDF, TF is the counting of the occurrence of the word in the current article, and IDF Illustrations how a term is relevant. It is a marking of how infrequent the word is across the article. CV [42] is used to alter a group of text documents to a vector of term token counts. HV is a well-organized way of mapping terms to features, every token straight map to a column location in a matrix.

4.7 Machine Learning Classifiers

We used nine ML classifiers to check their performance on two different datasets and three different feature extraction models. The NB classifier assumes that the occurrence of a specific word or feature in a class is unconnected to the occurrence of any other word or feature. We used three variants of NB and Multinomial Naive Bayes. The NB classifier calculates each label probability for a given example and results from the label with the highest chance. Gaussian Naive Bayes [43] assumes that each class is generally distributed and supports continuous values. The algorithm supports a solid

hypothesis that analyses should be independent of each other. We also used K-NN because it is an essential algorithm for classification problems KNN algorithm assumes that similar things are near each other. KNN imprisons the idea of comparison, occasionally called proximity, distance, or closeness. We used LR [44] because it is quite fast and related to other classification methods, giving the best results when the dependent variable is binary. The DT [45] was used because it works based on a decision with a set of rules like humans make to solve a different problem. We used entropy values 0 to 1; for Gini impurity, the value was from 0 to 0.5. For node split, we used splitter value to best split, and for maximum depth of the tree, we used max depth. To perform ensemble-level tasks, we used RF, which contains many of the DTs. It's a fast process and has been proven successful in various fields. To best split a node, we used sklearn, provided max feature hyperparameters and used min sample leaf to regulate the minimum number of leaves required. We also explored the boosting technique and used Gradient Boosting [45] to combine different weak learners to improve the classification results.

The main idea of gradient boosting is to fit a new predictor to the error made by the previous classifier. We set n-estimators to 100 for boosting stages and changed different numbers in max depth to limit the number of nodes. SVM was also regularly used in classification difficulties, and we tested this to improve our results.

5 Experiments

We used to refine the fakenewsnet dataset and get data according to over requirements, as mentioned above in Sections 3.1 and 3.2. We used the columns from our dataset, i.e., Articles ID, Articles Title, Articles Text, and User credibility score label, as mentioned in Section 4.4.

We executed the ML mentioned above algorithms on our refined custom dataset to classify real and misinformation based on article content and user context. We started with input preprocessing with the help of Tokenization and stop word removal. Afterward, we used three different feature extraction techniques to get better features for our model. We extracted features using N-Gram, such as Unigram $n = 1$ after bigram $n = 2$, then progressively enlarged n by $n = 3$. Moreover, the individual n value was verified with a diverse number of features. We divide our dataset into 67% for training and 33% for testing. The different classifiers were used to create a learning model then these models were used to classify the labels allotted to testing data. Research outcomes were then analyzed and presented. We picked 23000 articles from our custom-created dataset with the help of the fakenewsnet data repository.

6 Results

We used three different feature mining approaches, TF-IDF, count vectorizer, and hashing vectorizer (described earlier), and diverse the number of the N-Gram from $n = 1$ to $n = 3$. Tables 4 to 12 shows the obtained results. In the obtained results according to our experiments, we observed that TF-IDF performs overall good results with all ML classifiers, but the count vector feature extractor provides better results with the DT, LR, RF, and GB. The maximum accuracy was with CV and RF on the PolitiFact dataset as 93.4%. Also, with the increase of N-gram, the system's accuracy declines. The lowest accuracy of 44.7% was achieved with Gaussian NB, as shown in Fig. 3.

Table 4: Results of multinomial NB

	Results	TF-IDF	Count vect.	Hashing vect.
Gossip cop dataset	Accuracy	0.86%	0.81%	0.79%
	Precision	0.85%	0.74%	0.72%
	Recall	0.74%	0.79%	0.77%
	F1-score	0.77%	0.75%	0.73%
Politi fact dataset	Accuracy	0.87%	0.83%	0.82%
	Precision	0.88%	0.83%	0.82%
	Recall	0.86%	0.84%	0.83%
	F1-score	0.87%	0.83%	0.82%

Table 5: Results of gaussian NB

	Results	TF-IDF	Count vect.	Hashing vect.
Gossip cop dataset	Accuracy	0.75%	0.54%	0.44%
	Precision	0.70%	0.62%	0.57%
	Recall	0.77%	0.67%	0.59%
	F1-score	0.71%	0.54%	0.44%
Politi fact dataset	Accuracy	0.66%	0.68%	0.56%
	Precision	0.67%	0.74%	0.54%
	Recall	0.62%	0.72%	0.54%
	F1-score	0.62%	0.69%	0.54%

Table 6: Results of bernoulli NB

	Results	TF-IDF	Count vect.	Hashing vect.
Gossip cop dataset	Accuracy	0.83%	0.83%	0.81%
	Precision	0.75%	0.75%	0.74%
	Recall	0.77%	0.77%	0.75%
	F1-score	0.76%	0.77%	0.74%
Politi fact dataset	Accuracy	0.68%	0.68%	0.65%
	Precision	0.74%	0.74%	0.70%
	Recall	0.72%	0.72%	0.68%
	F1-score	0.69%	0.69%	0.66%

Table 7: Results of logistic regression

	Results	TF-IDF	Count vect.	Hashing vect.
Gossip cop dataset	Accuracy	0.87%	0.83%	0.82%
	Precision	0.86%	0.75%	0.74%
	Recall	0.75%	0.75%	0.75%
	F1-score	0.79%	0.75%	0.74%
Politi fact dataset	Accuracy	0.88%	0.89%	0.85%
	Precision	0.89%	0.89%	0.85%
	Recall	0.87%	0.90%	0.86%
	F1-score	0.88%	0.90%	0.86%

Table 8: Results of gradient boosting classifier

	Results	TF-IDF	Count vect.	Hashing vect.
Gossi p cop dataset	Accuracy	0.86%	0.87%	0.85%
	Precision	0.86%	0.87%	0.84%
	Recall	0.73%	0.75%	0.71%
	F1-score	0.77%	0.79%	0.75%
Politi fact dataset	Accuracy	0.87%	0.89%	0.86%
	Precision	0.89%	0.91%	0.87%
	Recall	0.86%	0.89%	0.85%
	F1-score	0.87%	0.89%	0.85%

Table 9: Results of random forest classifier

	Results	TF-IDF	Count vect.	Hashing vect.
Gossip cop dataset	Accuracy	0.87%	0.87%	0.86%
	Precision	0.87%	0.87%	0.87%
	Recall	0.75%	0.75%	0.70%
	F1-score	0.79%	0.79%	0.74%
Politi fact dataset	Accuracy	0.90%	0.93%	0.91%
	Precision	0.91%	0.94%	0.91%
	Recall	0.90%	0.90%	0.91%
	F1-score	0.90%	0.90%	0.91%

Table 10: Results of decision tree

	Results	TF-IDF	Count vect.	Hashing vect.
Gossi pcp dataset	Accuracy	0.80%	0.80%	0.79%
	Precision	0.71%	0.72%	0.70%
	Recall	0.72%	0.73%	0.71%
	F1-score	0.72%	0.72%	0.70%
Politi fact dataset	Accuracy	0.81%	0.83%	0.81%
	Precision	0.81%	0.84%	0.81%
	Recall	0.80%	0.83%	0.80%
	F1-score	0.80%	0.84%	0.80%

Table 11: Results of KNN

	Results	TF-IDF	Count vect.	Hashing vect.
Gossip cop dataset	Accuracy	0.84%	0.83%	0.82%
	Precision	0.80%	0.78%	0.75%
	Recall	0.70%	0.68%	0.67%
	F1-score	0.73%	0.71%	0.70%
Politi fact dataset	Accuracy	0.76%	0.71%	0.69%
	Precision	0.79%	0.71%	0.69%
	Recall	0.74%	0.72%	0.69%
	F1-score	0.75%	0.71%	0.69%

Table 12: Results of SVM

	Results	TF-IDF	Count vect.	Hashing vect.
Gossipcop dataset	Accuracy	0.87%	0.86%	0.86%
	Precision	0.87%	0.86%	0.86%
	Recall	0.76%	0.71%	0.71%
	F1-score	0.79%	0.75%	0.74%
Politifact dataset	Accuracy	0.89%	0.73%	0.73%
	Precision	0.90%	0.78%	0.76%
	Recall	0.90%	0.76%	0.73%
	F1-score	0.90%	0.73%	0.77%

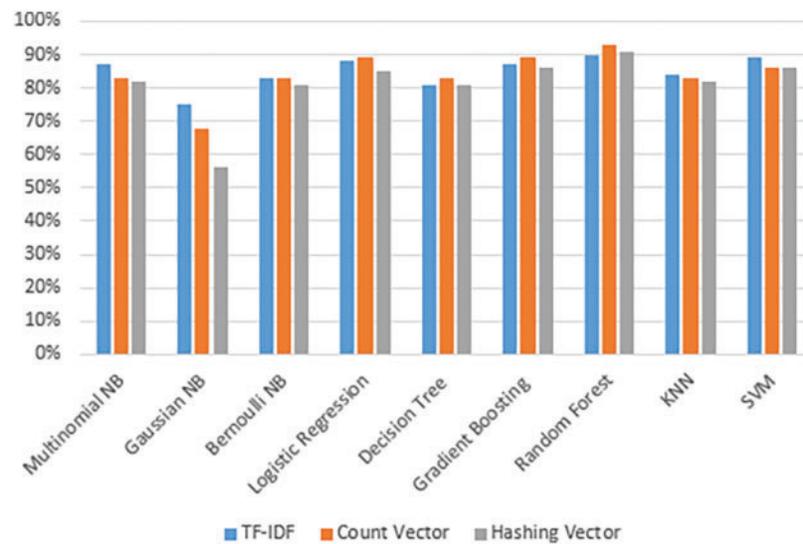


Figure 3: Best accuracy of classifiers

7 Discussion

We have compared our results with previously proposed ML and Neural Network Models methods. The comparison is made with the following models, i.e., [5], Echofaked [27–29,34,38]. Some of the methods used content, and others used content with context features. While in the context methods, the authors used only one feature, i.e., “share an article.” Our model uses both information, i.e., content and user context; furthermore, we add additional features to user context and predict user credibility based on the reply, tweets, re-tweets, and likes of content. As we analyzed in the literature review, some authors worked on the contents and user engagement. While the other authors experimented on text features, for example, contents, user comments, and tweets, the remaining study focuses only on content, for example, [4,8,11,13,15,24,28,29,35] to detect the misinformation. In [8], the authors used different datasets, but results vary on the same classifier; for example, their KNN classifier gives 83% accuracy on a dataset but on another dataset, it is 28% accuracy. Hence our classifier provides nearby results, i.e., KNN provides 84.3% accuracy on one dataset while on the other, it is 80.0% accuracy. Some of the comparison results are briefly discussed below and shown in Table 13.

- In [11], the authors analyzed tweets using a random forest. They achieve 84.0% accuracy, while our random forest model archives 93.4%.
- In [12], the authors analyzed content features using the decision tree ensemble model and archives 44.15% accuracy on the lair dataset, while our model with the decision tree achieves 83.9% accuracy.
- In [15], the authors used NB classifiers to analyze content features. They achieve 74.3% accuracy. In comparison, our model with the NB technique achieves 87.5%.
- In [24], the authors used SVM and NB. They achieved an accuracy of 73% in NB and 92% using SVM. In comparison, our model’s accuracy using SVM is 89.8%, and NB classifiers archives 87.5% accuracy.
- Some of the studies focus on News content and user feature, for example, [5,7,16,27] for misinformation detection.

- o In [5], the authors used content and context features and archives with 78% accuracy with boosting algorithm, while our GB model achieved 89.8% accuracy.
- o In [27], authors used content and context features with a deep neural network and achieved 92.3% accuracy with only 280 articles. We used 2300+ articles and reached 93.4% accuracy with ML. The experiment results of all the methods are shown in [Table 13](#).

Table 13: Different method performance

Ref.	Methods	Features	Dataset	Accuracy
[5]	KNN	Both content with context	Buzz feed	79%
[27]	Tensor factorization DNN	Both content with context	Politifact and buzz feed	92.3%
[28]	CNN, LSTM	Content	F-NAD	80%
[29]	TF-IDF, LSVM	Content	Kaggle	92%
[34]	Pre-trained CNN	Content	PubMed	91%
[38]	DNN	Content	PolitiFact & GossipCop	80%
Our	Count vectorization & random forest	Both content with context	Politifact and GossipCop	93.4%

Our accuracy increased because we used different hyper-parameters to implement these models. We added various features, for instance, and calculated user credibility with respect to user engagement with various real and misinformed articles.

8 Conclusion & Future Work

With the growth of social media, many people get daily updates from social media instead of old News media. A social network is used to spread misinformation, specifically affecting individual users and societies. In this research, we classify misinformation into information content and context. The content of misinformation is gathered from text of the article, and context is collected from social engagement, such as a tweet, retweet, like, and reply to an article. For classification, we used three feature extraction techniques, i.e., HV, CV, and TF-IDF. After feature extraction, we applied nine different ML classifiers. To evaluate the performance of our model, we used a Fakenewsnet data repository and created a dataset according to the requirement of content and context. We calculated the user credibility from user engagement with the content of articles. Then we combined both features for classification. The dataset contains 23000+ articles with millions of user engagements. Experimental results acquired using the misinformation datasets prove that content and context features are required to identify misinformation on social media. Our proposed method has enhanced the classification result in terms of accuracy and F1-score related to existing methods, as discussed in tables from [Tables 4 to 12](#), Section 6.

We used three feature extraction techniques with nine different machine-learning classifiers. We found that the TF-IDF with the tri-gram count is better for feature extraction with Multinomial NB, Gaussian NB, Bernoulli NB, KNN, and SVM achieved an accuracy of 87.5%, 75.7%, 83.0%, 84.3%, and 89.8% respectively. The CV provides better accuracy on LR and DT, respectively. The highest

accuracy score is 93.4%. The proposed model achieves its highest accuracy using CV features and an RF classifier. Our finding shows that user engagement is significant for misinformation detection; almost 30.3% of users share only fake articles, and 15.4% of users share both fake and real articles; this proves that user expectations are essential for misinformation detection, as shown in Table 3 Section 4.4. In the future, we will add more features to the social context and analyze the features using ML and DL techniques. The model will be more efficient if we divide the testing techniques on the combined geographic area over time feature with content and context features.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study

References

- [1] N. Smitha and R. Bharath, "Performance comparison of machine learning classifiers for fake news detection," in *Second Int. Conf. on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, pp. 696–700, 2020.
- [2] A. Benamira, B. Devillers, E. Lesot, A. K. Ray, M. Saadi *et al.*, "Semi-supervised learning and graph neural networks for fake news detection," in *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM)*, Vancouver, BC, Canada, pp. 568–569, 2019.
- [3] M. D. P. P. Goonathilake and P. P. N. V. Kumara, "CNN, RNN-LSTM based hybrid approach to detect state-of-the-art stance-based fake news on social media," in *20th Int. Conf. on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka, pp. 23–28, 2020.
- [4] Ghinadya and S. Suyanto, "Synonyms-based augmentation to improve fake news detection using bidirectional lstm," in *8th Int. Conf. on Information and Communication Technology (ICOICT)*, Yogyakarta, Indonesia, pp. 1–5, 2020.
- [5] A. Kesarwani, S. S. Chauhan and A. R. Nair, "Fake news detection on social media using k-nearest neighbor classifier," in *Int. Conf. on Advances in Computing and Communication Engineering (ICACCE)*, Las Vegas, NV, USA, pp. 1–4, 2020.
- [6] D. Dementieva and A. Panchenko, "Fake news detection using multilingual evidence," in *IEEE 7th Int. Conf. on Data Science and Advanced Analytics (DSAA)*, Sydney, NSW, Australia, pp. 775–776, 2020.
- [7] M. M. M. Hlaing and N. S. M. Kham, "Defining news authenticity on social media using machine learning approach," in *IEEE Conf. on Computer Applications (ICCA)*, Yangon, Myanmar, pp. 1–6, 2020.
- [8] I. Ahmad, M. Yousaf, S. Yousaf and M. O. Ahmad, "Fake news detection using machine learning ensemble methods", *Hindawi, Complexity*, vol. 2020, pp. 1076–2787, 2020.
- [9] N. R. Oliveira, D. S. V. Medeiros and D. M. F. Mattos, "A sensitive stylistic approach to identify fake news on social networking," *IEEE Signal Processing Letters*, vol. 27, pp. 1250–1254, 2020.
- [10] A. B. Hani, O. Adedugbe, F. A. Obeidat, E. Benkhelifa and M. Majdalawieh, "Fane-KG: A semantic knowledge graph for context-based fake news detection on social media," in *Seventh Int. Conf. on Social Networks Analysis, Management and Security (SNAMS)*, Paris, France, pp. 1–6, 2020.
- [11] N. X. Nyow and H. N. Chua, "Detecting fake news with tweets' properties," in *IEEE Conf. on Application, Information and Network Security (AINS)*, Pulau Pinang, Malaysia, pp. 24–29, 2019.
- [12] H. Saqib, A. Mamoun, K. Suleman, G. Thippa, R. Praveen *et al.*, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2019.
- [13] B. Pranav and S. Zongru, "Fake news detection with semantic features and text," *International Journal on Natural Language Computing (IJNLC)*, vol. 8, no. 3, pp. 17–22, 2019.
- [14] M. S. A. Rakhami and A. M. A. Amri, "Lies kill, facts save: Detecting covid-19 misinformation in twitter," *IEEE Access*, vol. 8, pp. 155961–155970, 2020.

- [15] O. F. Altunbey and B. Alatas, "Fake news detection within online social media using supervised artificial intelligence algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 540, pp. 1–17, 2020.
- [16] K. M. Caramancion, "Understanding the impact of contextual clues in misinformation detection," in *IEEE Int. IOT, Electronics and Mechatronics Conf. (IEMTRONICS)*, Vancouver, BC, Canada, pp. 1–6, 2020.
- [17] W. Antoun, F. Baly, R. Achour, A. Hussein and H. Hajj, "State of the art models for fake news detection tasks," in *IEEE Int. Conf. on Informatics, IoT, and Enabling Technologies (ICIOT)*, Doha, Qatar, pp. 519–524, 2020.
- [18] A. B. Hani, O. Adedugbe, E. Benkhelifa, M. Majdalawieh and F. A. Obeidat, "A semantic model for context-based fake news detection on social media," in *IEEE/ACS 17th Int. Conf. on Computer Systems and Applications (AICCSA)*, Antalya, Turkey, pp. 1–7, 2020.
- [19] A. Monther and A. Alwahedi, "Detecting fake news in social media networks," *Procedia Computer Science*, vol. 141, pp. 215–222, 2018.
- [20] W. Liang, "Misinformation in social media: Definition, manipulation, and detection," *ACM SIGKDD Explorations Newsletter*, vol. 21, no. 2, pp. 80–90, 2019.
- [21] H. Shao, "Misinformation detection and adversarial attack cost analysis in directional social networks," in *29th Int. Conf. on Computer Communications and Networks (ICCCN)*, Honolulu, HI, USA, pp. 1–11, 2020.
- [22] L. Simone, G. Rizzo and M. Morisio, "Automated classification of fake news spreaders to break the misinformation chain," *Information*, vol. 12, no. 6, pp. 248, 2021.
- [23] S. De and D. Agarwal, "A novel model of supervised clustering using sentiment and contextual analysis for fake news detection," in *Third Int. Conf. on Multimedia Processing, Communication & Information Technology (MPCIT)*, Shivamogga, India, pp. 112–117, 2020.
- [24] N. Kousika, S. Deepa, C. Deephika, B. M. Dhatchaiyine and J. Amrutha, "A system for fake news detection by using supervised learning model for social media contents," in *5th Int. Conf. on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, pp. 1042–1047, 2021.
- [25] D. Xishuang, U. Victor and L. Qian, "Two-path deep semisupervised learning for timely fake news detection," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1386–1398, 2020.
- [26] A. Verma, V. Mittal and S. Dawn, "FIND: Fake information and news detections using deep learning," in *Twelfth Int. Conf. on Contemporary Computing (IC3)*, Noida, India, pp. 1–7, 2019.
- [27] K. R. Kumar, A. Goswami and P. Narang, "Echofaked: Improving fake news detection in social media with an efficient deep neural network," *Neural Computing and Applications*, vol. 33, no. 14, pp. 8597–8613, 2021.
- [28] R. Barua, R. Maity, D. Minj, T. Barua and A. K. Layek, "F-NAD: An application for fake news article detection using machine learning techniques," in *IEEE Bombay Section Signature Conf. (IBSSC)*, Mumbai, India, pp. 1–6, 2019.
- [29] H. Ahmed, I. Traore and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *Int. Conf. on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, Cham, Springer, pp. 127–138, 2017.
- [30] Abdullah, J. Awan, Mazhar, F. Shehzad, H. Muhammad *et al.*, "Fake news classification bimodal using convolutional neural network and long short-term memory," *International Journal Emerging Technology*, vol. 11, pp. 209–212, 2020.
- [31] Y. Yanagi, R. Orihara, Y. Sei, Y. Tahara and A. Ohsuga, "Fake news detection with generated comments for news articles," in *IEEE 24th Int. Conf. on Intelligent Engineering Systems (INES)*, Reykjavik, Iceland, pp. 85–90, 2020.
- [32] K. Shu, FakeNewsNet, *Dataset*, 2019. <https://doi.org/10.7910/DVN/UEMMHS>
- [33] S. S. Jadhav and S. D. Thepade, "Fake news identification and classification using DSSM and improved recurrent neural network classifier," *Applied Artificial Intelligence*, vol. 33, no. 12, pp. 1058–1068, 2019.
- [34] Y. Parfenenko, A. Verbytska, D. Bychko and V. Shendryk, "Application for medical misinformation detection in online forums," in *Int. Conf. on e-Health and Bioengineering (EHB)*, Iasi, Romania, pp. 1–4, 2020.

- [35] S. Sridhar and S. Sanagavarapu, "Fake news detection and analysis using multitask learning with bilstm capsnet model," in *11th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, pp. 905–911, 2021.
- [36] S. Schwarz, A. Theóphilo and A. Rocha, "EMET: Embeddings from multilingual-encoder transformer for fake news detection," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 2777–2781, 2020.
- [37] T. Jiang, J. P. Li, A. U. Haq, A. Saboor and A. Ali, "A novel stacking approach for accurate detection of fake news," *IEEE Access*, vol. 9, pp. 22626–22639, 2021.
- [38] K. Shu, L. Cui, S. Wang, D. Lee and H. Liu, "Defend: Explainable fake news detection," in *Proc. of the 25th ACM SIGKDD, Int. Conf. on Knowledge Discovery & Data Mining*, Anchorage AK USA, pp. 395–405, 2019.
- [39] M. Goldani, M. Hadi, S. Momtazi and R. Safabakhsh, "Detecting fake news with capsule neural networks," *Applied Soft Computing*, vol. 101, pp. 10–16, 2021.
- [40] K. Shu, S. Wang and H. Liu, "Exploiting tri-relationship for fake news detection," arXiv preprint arXiv:1712.077098, 2017.
- [41] M. Avinash and E. Sivasankar, "A study of feature extraction techniques for sentiment analysis," in *Emerging Technologies in Data Mining and Information Security*, Singapore: Springer, pp. 475–486, 2019.
- [42] D. Deepa, Raaji and A. Tamilarasi, "Sentiment analysis using feature extraction and dictionary-based approaches," in *Third Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, pp. 786–790, 2019.
- [43] L. Zhang, L. Jiang, C. Li and G. Kong, "Two feature weighting approaches for naive Bayes text classifiers," *Knowledge-Based Systems*, vol. 100, pp. 137–144, 2016.
- [44] A. A. Tanvir, E. M. Mahir, S. Akhter and M. R. Huq, "Detecting fake news using machine learning and deep learning algorithms," in *7th Int. Conf. on Smart Computing & Communications (ICSCC)*, Sarawak, Malaysia, pp. 1–5, 2019.
- [45] D. Duan, X. Gai, Z. Han and B. Liu, "Micro-blog misinformation detection based on gradient boost decision tree," *Journal of Computer Applications*, vol. 38, no. 2, pp. 4–10, 2018.