Int. j. inf. tecnol. (June 2024) 16(5):3233–3241
https://doi.org/10.1007/s41870-023-01564-5

ORIGINAL RESEARCH

Leveraging contextual features to enhanced machine learning models in detecting COVID-19 fake news

Amal Esmail Qasem[1] · **Mohammad Sajid**[1]

Received: 12 June 2023 / Accepted: 22 September 2023 / Published online: 25 October 2023

Abstract The proliferation of fake news on online social networks, particularly Twitter, has become a major issue in recent years. False and potentially harmful information can spread quickly and cause panic or confusion among the public. To mitigate this, accurate fake news detection is crucial. This work introduces a novel approach by leveraging domain knowledge to extract high-quality features from text data. These features, including word count, hashtag count, and sentiment, complement tweet embeddings derived from the Term Frequency-Inverse Document Frequency technique. The resulting combined representation enhances accuracy. Four machine learning models, i.e., Logistic Regression, Support Vector Machine (SVM), Decision Tree, and Gradient Boosting Decision Tree, are employed to classify text as real or fake, using the combined enriched features. The models are evaluated on a COVID-19 fake news benchmark dataset, measuring their performance across four key metrics: accuracy, precision, recall, and F1-score. The results reveal a 0.5–2% performance boost compared to baseline models. Notably, SVM achieved the highest accuracy at 93.74%. This highlights the efficacy of augmenting models with quality features for improved fake news detection.

Keywords Fake news · COVID-19 · Classification · Machine learning · Contextual feature · Support vector machine

✉ Mohammad Sajid
sajid.cst@gmail.com

[1] Department of Computer Science, Aligarh Muslim University, Aligarh, India

1 Introduction

Millions of people use the internet daily and publish news content on social media platforms like Twitter and Facebook. With so many online sources of information, it can be difficult to determine which content is based on facts and which is misleading. Using digital platforms to spread false information can have a powerful and far-reaching impact, influencing others to accept it as fact. Fake news can also be used to provoke and exacerbate social conflict, impacting all areas of society. Its impact is particularly significant when it relates to the health of individuals, such as during the COVID-19 pandemic this virus affected almost 10 million people in the world [1, 2]. Generally, using machine learning (ML) and deep learning (DL) methods can significantly aid in detecting fake news content on social media platforms. These methods have proven valuable in addressing various real-world challenges such as sentiment analysis [3, 4], sarcasm [5], etc. They are trained to verify and tag text into predefined labels, such as "positive" or "negative" in case of sentiment analysis. Natural Language Processing (NLP) is a subfield of artificial intelligence that involves using natural language to understand human interaction with machines. In order to interpret the meaning of a text, it is necessary to understand its context. Using domain knowledge to extract useful, meaningful, and high-quality features from the text can improve its representation and lead to more accurate models.

Researchers are trying to find the best ML classifier to determine fake news. The model's accuracy is essential and must be considered because it can harm different individuals if it fails to detect fake news [6]. These models' performance depends mainly on the data preprocessing [7] and the features' quality in the training phase [8]. It has been proved that leveraging features engineering into ML classifiers can

enhance the classifiers' performance and increase their accuracy [9]. Thus, this work focuses to increase the performance of the traditional ML models, i.e. Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), and Gradient Boosting Decision Tree (GBDT) in detecting fake news by enriching its input features extracted by Term Frequency –Inverse Document Frequency (TF-IDF) with more text-representative features. This approach is evaluated on the COVID-19 benchmark dataset and shows the impact of adding extra context features to enhance the models' overall performance. The main contributions of this paper are as follows:

- Extracting eleven context features from each tweet and investigating and analyzing its impact on the performance of the ML models.
- Experiments have carried out the evaluation of four ML classifiers with each of the eleven extra features to identify fake news on the publicly available COVID-19 dataset.
- Comparing the performance of this approach to the baseline models (without extra features).

The remaining content is organized as follows: the related works are explored in Sect. 2. The proposed work methodology is described in Sect. 3. The experimental setup is discussed in Sect. 4. Section 5 involves the results and discussion, whereas, the conclusion and future work in Sect. 6, followed by references.

2 Related work

Starting with the baseline study of this work [10], which acquired COVID-19 tweets from different online sources and applied ML classifiers such as SVM, LR, DT, and GDBT. The results revealed that SVM achieved superior results in validation and test datasets. In [11] several ML and DL models were compared to identify disinformation about COVID-19 automatically. The experiments were conducted on two datasets, and the results were evaluated using various metrics. The results showed that traditional ML models performed better than DL models in predicting fake news. Specifically, both Random Forest (RF) and LR had superior results compared to other models. Additionally, Long Short-Term Memory (LSTM) performed better than Convolutional Neural Network (CNN). Similarly, [1] has experimented various ML and transformer models, including Naïve Bayes (NB) and SVM models, as well as Bidirectional Encoder Representations from Transformers (BERT), DistilBert, and Roberta, with TF-IDF and word2vec representation methods. They found that SVM performed the best among the other models when used with TF-IDF. However, using Word2vec decreased the performance of the models. Additionally, the transformer models showed the best accuracy and f1-score results. Variant classifiers ranging from traditional ML and DL, along with different extraction techniques like TF-DF with n-gram were evaluated on four COVID-19 fake news datasets by authors in [12]. The results demonstrated significant achievement by the baseline compared to the existing state of art. Other works emphasized the need for feature engineering to efficiently address fake news detection, such as [9], which used five DL models and features engineering, such as emotion, and features, including term frequency, stop word count ratio, and average sentence length. Also, [13] used ML classifiers with linguistic features, such as n-grams, readability, emotional tone, and punctuation, and found that linear SVM performed the best with an f1-score of 95.19% on the unseen set. In a different study [14] various experiments were conducted using ML and DL models to detect fake content. NLP features such as the number of mentions, hashtags, and tweet length were extracted from tweets and used as metadata in the models. The performance of the models is evaluated using each feature, and found that this approach slightly improved the English dataset with an f1-score of 0.93%.

3 Proposed work methodology

Figure 1 introduces steps followed in conducting fake news detection which explained in details in the sub-section below:

3.1 Proposed work components

3.1.1 Dataset description

COVID-19 dataset [10] consists of tweets regarding COVID-19 pandemic. Each tweet has a label indicating whether the tweet's is real or fake. It contains three CSV files: train, validation, and test, that includes 6420 samples, 2140 samples, and 2140 samples, respectively.

3.1.2 Preprocessing

Here, NLP techniques are used to minimize noise by removing irrelevant data for fake news classification such as, lower casing, removing URLs, replacing symbols and tags, and removing stop words [15]. This ensures that the data is properly prepared for feature extraction and further analysis.

3.1.3 Feature engineering and extraction

Feature engineering It is considered the most essential part of text classification. Different types of features can be

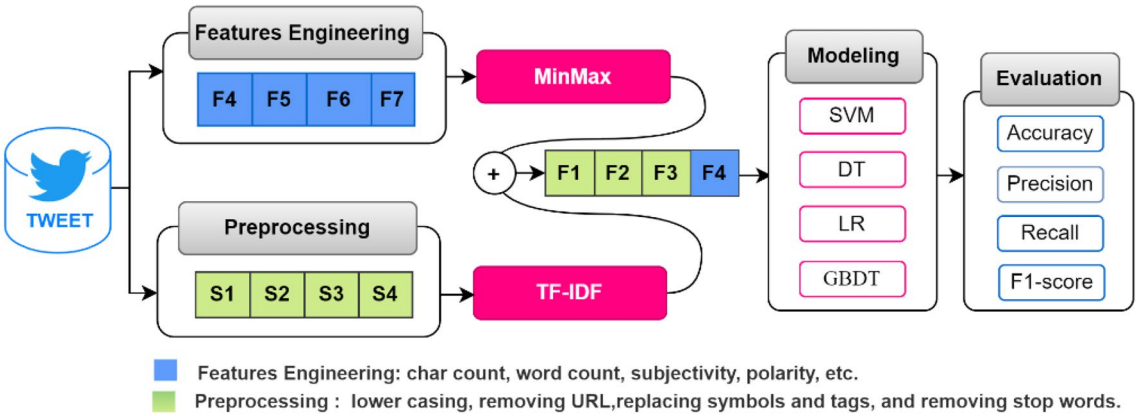

Fig. 1 Proposed work methodology steps

engineered from the given dataset, which can be used in the classification models are described in Table 1.

Term frequency-inverse document frequency (TF-IDF) It encodes any type of text as a statistic number indicating the frequency of each word or phrase throughout the whole document [16]. It is considered a text vectorizer that converts the provided text into a numerical vector. Each value in this vector is calculated as the following formula, which multiplies two concepts, TF and IDF [17]

$$w_{i,j} = tf_{i,j} \times \log\left(N/df_i\right) \tag{1}$$

where TF represents how many times a given word appears in the text divided by the total no. of words in the exact text. In comparison, IDF is the log of the no. of documents divided by the no. of documents that contain the word. It specifies the weight of rare vocab among the dataset, as in the following formulas [18, 19].

$$tf_{i,j} = n_{i,j} / \sum_k ni,j \tag{2}$$

$$idf(w) = \log\left(N/df_t\right) \tag{3}$$

3.1.4 Model building

Logistic regression (LR) It is a probability-based predictive analytic algorithm that uses a statistical model based on the sigmoid or logistic function. When given a real-valued input, the output of an S-shaped curve is mapped between 0 and 1. Where 0 is the bias or intercepts term, and 1 is the coefficient for the independent variable [20].

Support vector machine (SVM) It is a supervised ML method for classification tasks [21] that creates a straight line separating samples of two classes with the highest margin. It works in an N-dimensional space, making the line as far away from the closest data points as possible. It is suitable for regression and classification tasks [22, 23].

Decision tree (DT) It is a robust and more popular supervised learning method due to its easy understanding [24] and implementation. Like SVM, DT can be used for regression and classification tasks and works well with numeric and categorical data. It works by separating the given dataset into small sets according to criteria, and the tree is built incrementally. The leaf nodes of a decision tree represent the classification results [25].

Gradient boosting decision tree (GBDT) It involves using an algorithm for gradient lifting and an algorithm for decision trees to correct the errors made by its predecessor. The primary function of gradient boosting is to reduce residuals or to generate a decision tree in the direction of a negative gradient to minimize final residuals. The fundamental principle of boosting theory is to continuously decrease the loss function as the model is established, meaning that the model is continually being optimized [26].

Table 1 Feature list description

Shortcut	Description	Shortcut	Description	Shortcut	Description
Cap_Ch_C	No. of capital chars	Stop_W_C	No. of stop words	Polarity	Polarity value
Stop_Vs_W	Stop words vs words	Unique_Vs_W	Unique words vs words	Word_C	No. of words
Unique_W	No. of unique words	Subjectivity	No. of subjectivity	Sent_C	No. of sentences
Char_C	No. of characters	Cap_W_C	No. of capital words		

3.1.5 Performance evaluation

The objective of the performance evaluation step is to evaluate the performance of the generated models on unseen data. For this purpose, we utilized accuracy, precision, recall, and f1-score performance evaluation metrics which are calculated using the functions available in the Python Scikit-learn Metrics module [3, 4, 27].

3.2 Proposed work algorithm

	Input: *COVID-19 tweet dataset (training and testing data)*
	Output: *Evaluation of the classifiers*
	Begin Algorithm
1.	***STEP 1: Load COVID-19 dataset***
2.	*- Train_data <- Load CSV train data file*
3.	*- Test_ data <- Load CSV test data file*
4.	***STEP 2: Pre-processing***
5.	*- For each tweet in the (Train_data, Test_data):*
6.	*- Convert the text to lowercase.*
7.	*- Remove URLs.*
8.	*- Replace symbols and tags with appropriate representations.*
9.	*- Remove stop words.*
10.	*- preprocessed_Train_data, preprocessed_Test_data <- Save (Train_data, Test_data)*
11.	*- End*
12.	***STEP 3: Feature Engineering and Extraction***
13.	***- STEP 3.1: Perform Feature Engineering***
14.	*- For each feature_function in the (Table1: Char_count, Word_count Subjectivity, etc.):*
15.	*- For each tweet in the (Train_data, Test_data):*
16.	*- Calculate feature_function*
17.	*- Add its output as additional feature to the dataset*
18.	*- End*
19.	*- End*
20.	*- Apply MinMax normalization function*
21.	***- STEP 3.2: Perform Feature Extraction***
22.	*- Initiate object for TF-IDF vectorizer*
23.	*- Use TF-IDF object to Fit and transform the preprocessed_Train_data*
24.	*- Use TF-IDF object to transform the preprocessed_Test_data*
25.	*- Combine tf-idf matrix with the additional features*
26.	***STEP 4: Model Building***
27.	*- For each model (LR, SVM, DT, GBDT):*
28.	*- Step 4.1: Train the model on train data*
29.	*- Step 4.2: Save the model*
30.	*- End*
31.	***STEP 5: Model Testing***
32.	*- For each trained model (LR, SVM, DT, GBDT):*
33.	*- For each tf-idf matrix data with one additional feature*
34.	*- Test the model on test data*
35.	*- End*
36.	*- End*
37.	***STEP 6: Performance Evaluation***
38.	*- For each trained model (LR, SVM, DT, GBDT):*
39.	*- Evaluate its performance using accuracy, precision, recall, and F1-score.*
40.	*- Plot relevant charts to visualize the model's performance.*
41.	*- End*
42.	*- Compare the performance of all models to determine the most effective one for the task.*
	End Algorithm

4 Implementation

Initially, we collected the dataset [10] related to COVID-19 fake news from Kaggle and performed various feature engineering techniques to construct additional features for the dataset to help ML models identifying different patterns. We applied the MinMax scaler, a standardization technique from the Scikit-learn library, on all of the extracted features to ensure that they were all in the same range of values for more ML performance efficiency. The text data was preprocessed to remove irrelevant words and characters followed steps in [10], and the TF-IDF technique was used for feature extraction. These preprocessing and feature extraction steps discussed earlier were also applied to the training and validation data. It is applied to each tweet to calculate that tweet vector. This vectorization results in a matrix representing each sentence as a vector. The vector has the same length as our vocabulary. We experimented with several ML models including those suggested on the model building section. We applied these models to the text data to form baseline results. Then, additional features were added, each impact is evaluated on the performance of the models. To evaluate and compare the performance of the models, a test is conducted on a separate validation set to estimate how well the model generalizes to unseen data on the suggested evaluation metrics.

5 Results analysis and discussion

5.1 Experiments results

Table 2 and Fig. 2 show the results of the SVM classifier on the validation dataset. It is shown that enriching the model with individual extra features can enhance the model slightly. Specifically, the "subjectivity" feature improves the baseline model on all performance evaluation metrics by ~ 0.3%. Table 3 and Fig. 3 similarly, show the results of the LR classifier, which demonstrated that utilizing "polarity" feature can enhance the model slightly in terms of accuracy. Some other features have the same accuracy as the LR baseline model but improve the precision performance a little bit by ~ 0.02.

Also, Table 4 and Fig. 4 shown that enriching the DT model with the "Char count" feature can enhance the model accuracy by a good margin, 1%. Finally, Table 5 and Fig. 5 demonstrated that enriching GBDT model with most of the

Table 2 The impact of features engineering on the performance of SVM classifier

Features	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Subjectivity	93.74	93.76	93.74	93.74
Polarity	93.69	93.73	93.69	93.69
Char_count	93.64	93.67	93.64	93.65
Unique_vs_words	93.64	93.68	93.64	93.65
Capital_char_count	93.60	93.63	93.60	93.60
Word_count	93.50	93.54	93.50	93.51
Sent_count	93.50	93.54	93.50	93.51
Stopwords_vs_words	93.50	93.54	93.50	93.51
Capital_word_count	93.46	93.50	93.46	93.46
Stopword_count	93.46	93.50	93.46	93.46
Unique_word_count	93.46	93.49	93.46	93.46
Baseline	93.46	93.48	93.46	93.46

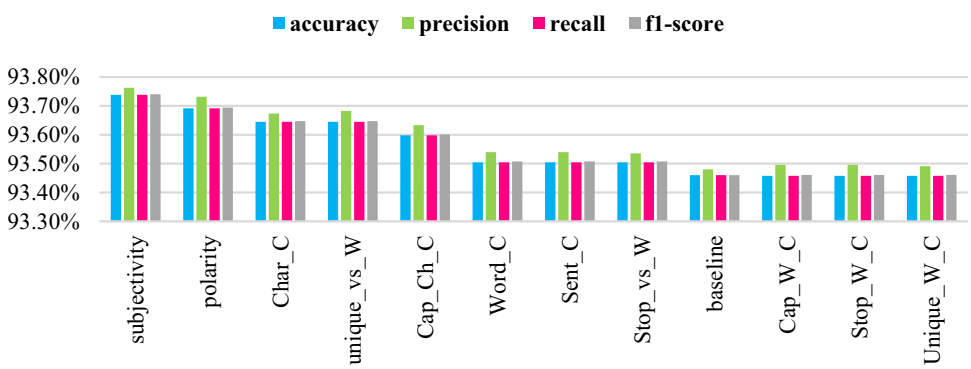

Fig. 2 The performance of SVM on the validation dataset based on the engineering features

Table 3 The impact of features engineering on the performance of LR classifier

Features	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Polarity	92.85	92.90	92.85	92.85
Char_count	92.76	92.81	92.76	92.76
Word_count	92.76	92.81	92.76	92.76
Sent_count	92.76	92.81	92.76	92.76
Stopword_count	92.76	92.81	92.76	92.76
Unique_word_count	92.76	92.81	92.76	92.76
Baseline	92.76	92.79	92.76	92.75
Capital_word_count	92.71	92.76	92.71	92.71
Unique_vs_words	92.71	92.78	92.71	92.71
Capital_char_count	92.62	92.67	92.62	92.62
Stopwords_vs_words	92.29	92.33	92.29	92.29
Subjectivity	92.15	92.19	92.15	92.15

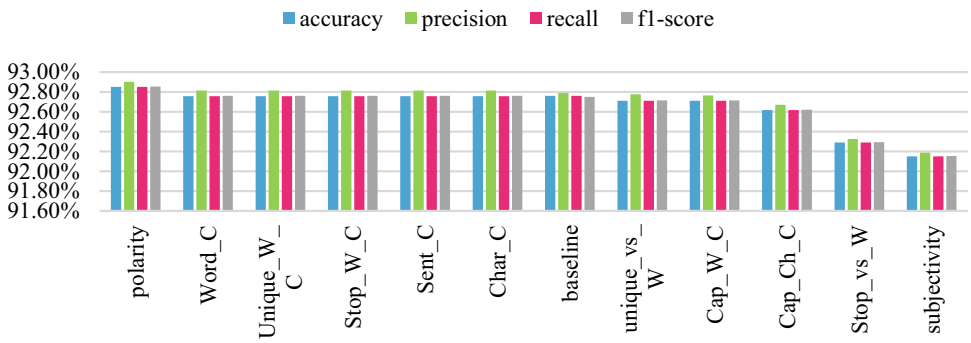

Fig. 3 The performance of LR on the validation dataset based on the engineering features

Table 4 The impact of features engineering on the performance of DT classifier

Features	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
char_count	86.21	86.21	86.21	86.21
capital_word_count	85.65	85.66	85.65	85.64
word_count	85.56	85.56	85.56	85.56
sent_count	85.51	85.52	85.51	85.52
capital_char_count	85.23	85.24	85.23	85.22
baseline	85.23	85.31	85.23	85.25
unique_word_count	85.19	85.18	85.19	85.18
stopword_count	84.95	84.98	84.95	84.93
polarity	84.95	84.95	84.95	84.94
unique_vs_words	84.72	84.73	84.72	84.70
stopwords_vs_words	84.67	84.67	84.67	84.67
subjectivity	84.53	84.54	84.53	84.52

features can improve the model by a good margin, ~ in the range of [1, 2%] in terms of accuracy except "unique word count" and "subjectivity," which got less accuracy than the baseline.

5.2 Discussion and comparison

Prior research has explored various approaches for fake news detection as in related work section, some of which were applied to the same dataset used in this study [10, 11]. Specifically, the proposed approach distinguishes itself from existing methods [10, 11] at the component level. Unlike [11], which used distinct ML with tfidf features and additional deep learning models (CNN and LSTM) with Glove (the later exhibits low performance compared to traditional ML models), and [10] that utilized the similar models and tf-idf features, but this approach additionally introduced thirteen knowledge base features,

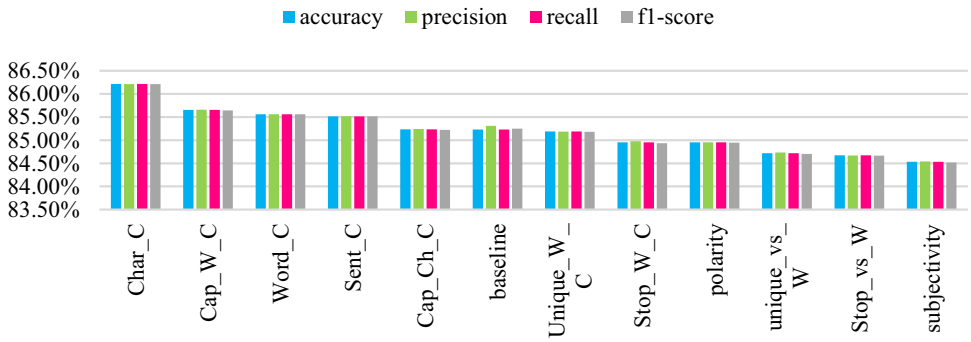

Fig. 4 The performance of DT on the validation dataset based on the engineering features

Table 5 The impact of features engineering on the performance of GBDT classifier

Features	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Capital_char_count	88.32	88.52	88.32	88.32
Stopword_count	87.90	88.00	87.90	87.90
Unique_vs_words	87.57	87.77	87.57	87.58
Char_count	87.48	87.47	87.48	87.47
Stopwords_vs_words	87.38	87.64	87.38	87.39
Capital_word_count	87.15	87.46	87.15	87.15
Sent_count	87.06	87.06	87.06	87.05
Word_count	86.96	86.98	86.96	86.97
Polarity	86.87	87.14	86.87	86.87
Baseline	86.82	87.08	86.82	86.82
Unique_word_count	86.68	86.68	86.68	86.68
Subjectivity	86.50	86.62	86.50	86.50

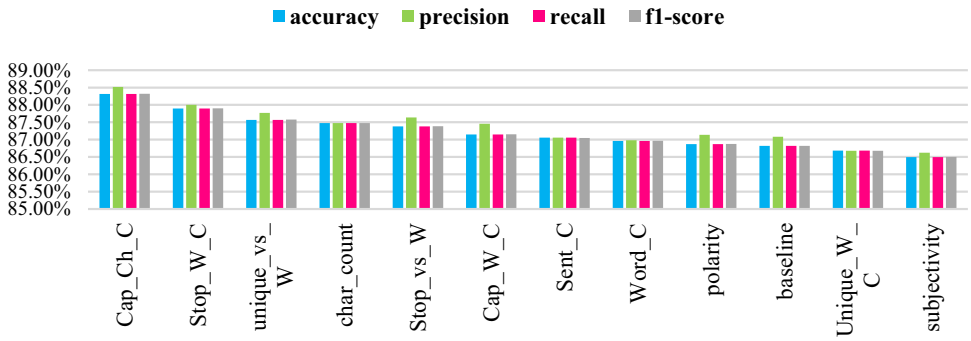

Fig. 5 The performance of GBDT on the validation dataset based on the engineering features

enhancing the model's ability to prioritize knowledge base-relevant characteristics extracted from the text. This innovation sets our approach apart from methods primarily relying on fixed feature extraction techniques and traditional deep learning models, resulting in improved performance and generalization in the detection of COVID-19 fake news.

In the comparative analysis Table 6, baseline_1 [10] consistently demonstrates strong performance across various models, achieving high levels of performance in all the measurement metrics used, while baseline_2 [11] exhibits comparable performance in terms of LR but showcases a distinct pattern with high recall and lower accuracy and precision for SVM, i.e., due to the "gamma" parameter that set as a kernel instead of "linear". Also, compared to the DT in baseline_2 [11], our approach achieves slightly lower accuracy at 86.21% versus 85.23%, but maintains consistent in the other matrices. Conversely, when compared to baseline_1 [10], our approach demonstrates ~ 1and ~ 2% high margin using DT and GBDT, respectively, in all the matrices. Interestingly, our variations consistently outperform their corresponding baselines, with the introduced features notably enhancing model effectiveness. The careful selection of features, including "subjectivity," "polarity," "char_count," and "capitalchar_count," significantly contributes to improving

Table 6 Performance Comparison with the baseline models

Model	Paper	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SVM	Baseline_1 [10]	93.46	93.48	93.46	93.46
	Baseline_2 [11]	65	60	99	75
	Ours + subjectivity	93.74	93.76	93.74	93.74
LR	Baseline_1 [10]	92.76	92.79	92.76	92.75
	Baseline_2 [11]	91	93	90	92
	Ours + polarity	92.85	92.90	92.85	92.85
DT	Baseline_1 [10]	85.23	85.31	85.23	85.25
	Baseline_2 [11]	87	91	84	87
	Ours + char_count	86.21	86.21	86.21	86.21
GBDT	Baseline_1 [10]	86.82	87.08	86.82	86.82
	Baseline_2 [11]	–	–	–	–
	Ours + capitalchar_count	88.32	88.52	88.32	88.32

baseline model performance. Ultimately, the proposed approach yields superior results compared to methods solely relying on the baseline models baseline_1 [10] and baseline_2 [11]. Performance comparisons among these algorithms are summarized in Table 6.

6 Conclusion

To summarize this work, it focused to increase the performance of the traditional ML models in detecting fake news related to COVID-19 pandemic by enriching its input features extracted by TF-IDF with more text-representative features. Firstly, the investigation of employing extra context feature has been carried out for all baseline models and found that: different features affect the performance of different classifiers, applying a scaler to the extracted features can enhance the model's performance. In addition, SVM and LR have been improved slightly, whereas DT and GBDT have been improved with a good margin. Moreover, "Char count", "Word count", and "unique words" are the most representative features among all others. Finally, this innovative approach has consistently outperformed alternative strategies reliant solely on baseline TF-IDF and Word2Vec techniques. This paper makes the research open to investigate multiple features and advance deep learning models in detecting fake news.

Data availability Data will be made available on request.

References

1. Raha T et al (2021) Identifying COVID-19 fake news in social media. http://arxiv.org/abs/2101.11954.
2. Khanday AMUD, Khan QR, Rabani ST (2021) Identifying propaganda from online social networks during COVID-19 using machine learning techniques. Int J Inf Technol 13(1):115–122. https://doi.org/10.1007/s41870-020-00550-5
3. Ali Salmony MY, Faridi AR (2021) An enhanced twitter sentiment analysis model using negation scope identification methods. In: Proceedings of 2021 8th international conference on computing for sustainable global development INDIACom 2021, pp 864–869. https://doi.org/10.1109/INDIACom51348.2021.00155
4. Salmony A, Rasool Faridi A (2021) Supervised sentiment analysis on amazon product reviews: a survey. In: Proceedings of 2021 2nd International conference on intelligent Engineering and management ICIEM 2021, pp 132–138. https://doi.org/10.1109/ICIEM51511.2021.9445303
5. Madani Y, Erritali M, Bouikhalene B (2021) Using artificial intelligence techniques for detecting covid-19 epidemic fake news in Moroccan tweets. Results Phys 25:104266. https://doi.org/10.1016/j.rinp.2021.104266
6. Ahmed AAA et al (2021) Detecting fake news using machine learning: a systematic literature review. Psychol Educ J 58(1):1932–1939. https://doi.org/10.17762/pae.v58i1.1046
7. Pandey S, Prabhakaran S, Reddy NVS, Acharya D (2022) Fake news detection from online media using machine learning classifiers. J Phys Conf Ser 2161(1):012027. https://doi.org/10.1088/1742-6596/2161/1/012027
8. Alam S, Yao N (2019) The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. Comput Math Organ Theory 25(3):319–335. https://doi.org/10.1007/s10588-018-9266-8
9. Melville W, Rd OT, Setauket E (2021) Efficiencies of feature engineering in the machine learning approach for fake news classification Katrin Donetski. https://doi.org/10.20944/preprints202111.0024.v1
10. Patwa P et al (2021) Fighting an infodemic: COVID-19 fake news dataset. Commun. Comput Inf Sci 1402:21–29. https://doi.org/10.1007/978-3-030-73696-5_3
11. Alhakami H, Alhakami W, Baz A, Faizan M, Khan MW, Agrawal A (2022) Evaluating intelligent methods for detecting covid-19 fake news on social media platforms. Electron 11(15):1–15. https://doi.org/10.3390/electronics11152417
12. Abdelminaam DS, Ismail FH, Taha M, Taha A, Houssein EH, Nabil A (2021) CoAID-DEEP: an optimized intelligent framework for automated detecting COVID-19 misleading information on twitter. IEEE Access 9(December 2019):27840–27867. https://doi.org/10.1109/ACCESS.2021.3058066
13. Felber T (2021) Constraint 2021: machine learning models for COVID-19 fake news detection shared task. pp 1–10. http://arxiv.org/abs/2101.03717

14. Gupta A, Sukumaran R, John K, Teki S (2021) Hostility detection and covid-19 fake news detection in social media. http://arxiv.org/abs/2101.05953
15. Khanday AMUD, Rabani ST, Khan QR, Rouf N, Mohi Ud Din M (2020) Machine learning based approaches for detecting COVID-19 using clinical text data. Int J Inf Technol 12(3):731–739. https://doi.org/10.1007/s41870-020-00495-9
16. Kotiyal B, Pathak H, Singh N (2023) Debunking multi-lingual social media posts using deep learning. Int J Inf Technol 15(5):2569–2581. https://doi.org/10.1007/s41870-023-01288-6
17. Ahuja R, Chug A, Kohli S, Gupta S, Ahuja P (2019) The impact of features extraction on the sentiment analysis. Procedia Comput Sci 152:341–348. https://doi.org/10.1016/j.procs.2019.05.008
18. de Beer D, M. Matthee S, (2021) Approaches to identify fake news: a systematic literature review, vol 136. Springer International Publishing, Cham (**no. Macaulay 2018**)
19. Das B, Chakraborty S (2018) An improved text sentiment classification model using TF-IDF and next word negation. http://arxiv.org/abs/1806.06407
20. Shaikh J, Patil R (2020) Fake news detection using machine learning. In: Proceedings of 2020 IEEE international symposium on sustainable energy, signal processing and cyber security iSSSC 2020, vol 2020. https://doi.org/10.1109/iSSSC50941.2020.9358890
21. Adjuik TA, Ananey-Obiri D (2022) Word2vec neural model-based techniqueto generate protein vectors for combating COVID-19: a machine learning approach. Int J Inf Technol 14(7):3291–3299. https://doi.org/10.1007/s41870-022-00949-2
22. Alenezi MN, Alqenaei ZM (2021) Machine learning in detecting covid-19 misinformation on twitter. Futur Internet 13(10):1–20. https://doi.org/10.3390/fi13100244
23. Qasem AE, Sajid M (2022) Exploring the effect of n-grams with BOW and TF-IDF representations on detecting fake news. In: International conference on data analytics for business and industry 2022
24. Gopi AP, Jyothi RNS, Narayana VL, Sandeep KS (2023) Classification of tweets data based on polarity using improved RBF kernel of SVM. Int J Inf Technol 15(2):965–980. https://doi.org/10.1007/s41870-019-00409-4
25. Garg H, Goyal A (2020) Techniques of fake news detection. Int J Civil Mech Energy Sci. 6(2):6–9. https://doi.org/10.22161/ijcmes.622
26. Huang Y, Wang X, Wang R, Min J (2021) Analysis and recognition of food safety problems in online ordering based on reviews text mining. Wirel Commun Mob Comput. https://doi.org/10.1155/2022/4209732
27. Alharbi NM, Alghamdi NS, Alkhammash EH, Al Amri JF (2021) Evaluation of sentiment analysis via word embedding and RNN variants for amazon online reviews. Math Probl Eng. https://doi.org/10.1155/2021/5536560