An Intelligent GUI Enhancements a Robust Model for Detecting Fake News

Raji Krishna[1], D. Jayanthi[2], Naveen Kumar Maurya[3](✉), K. Kavitha[4], Sathini Sindhuja[2], Togari Sneha[2], and Mulinti Sharanya[2]

[1] School of Electrical Engineering, Vellore Institute of Technology, Chennai, India
[2] Department of Electronics and Communication Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India
[3] Department of Electronics and Communication Engineering, Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, India
naveenkumar.m@vishnu.edu.in
[4] Department of Electronics and Communication Engineering, V.S.B. Engineering College, Karur, Tamil Nadu, India

Abstract. The Fake news is becoming more and more common due to the significant changes made to the information distribution landscape by the rapid advancement of technology. Misinformation can spread quickly in this environment because digital platforms are widely used and because creating and sharing content is simple. Modern technological solutions are required to identify and address the increasing issue of misinformation in digital media. This study examines cutting-edge methods for detecting fake news's. This study also presents Machine Learning models that can reliably identify fake news in various challenging scenarios. The machine learning models that are employed are Logistic Regression (LR), Decision Tree Classifier (DTC), Gradient Boosting Classifier (GBC), and Random Forest Classifier (RFC). LR, DTC, GBC, and RFC models' accuracy were approximately 98.69%, 99.69%, 99.56%, and 99.03%, respectively. We also created a Graphical User Interface (GUI). That is built up with a Python library, namely tkinter. We get output on the dialogue box if we input the top box. Attributes are taken according to our choice. As our code is in Python, we created a GUI with Python library rather than preferring HTML, CSS, or other languages.

Keywords: Fake news · Logistic Regression · Decision Tree Classifier · Gradient Boosting Classifier · Random Forest Classifier Graphical User Interface

1 Introduction

The internet's extensive distribution capabilities have resulted in the pervasive propagation of misinformation through social media. As an illustration, Facebook referrals contribute 20% of the traffic to trustworthy news websites and 50% to fraudulent news websites. A few machine learning techniques that we can use to create a robust model that combines various aspects for identifying fake news are Extreme Gradient Boosting

P. Pareek et al. (Eds.): IC4S 2024, LNICST 599, pp. 221–234, 2025.
https://doi.org/10.1007/978-3-031-77081-4_18

(XGBoost), Neural Network, LR, Support Vector Machine (SVM), Random Forest (RF), classification and regression trees (CART) [1]. The impact of fake news has grown significantly from a minor annoyance to a severe threat to societal stability and order. Because of its multifaceted nature, which touches on social, religious, political, and economic dimensions, it incentivizes various groups to spread biased information, conspiracies, and fraudulent content. This manipulation of public sentiment has increased because of the ease of creating and disseminating fake news on social media and messaging platforms. Combating this 'infodemic' has relied heavily on artificial intelligence (AI) algorithms and social network analysis. Despite these advances, scaling up an effective solution remains a significant challenge. The complexities of detecting and mitigating fake news pose a formidable challenge, encompassing human, technical, and economic factors that preclude the development of a one-size-fits-all solution [2]. This strategy is intriguing because it aims to capture the relationships and contexts within the news content and the text itself.

Fake News Detection Multitask Learning (FDML) presents a task gate in the multitask learning aspect that integrates representations based on distinct tasks in a selective manner. Notably, this model performs better than current approaches in these crucial areas, indicating that it has the potential to make a substantial contribution to the realm of counterfeit news identification. The FDML model uses two key findings to increase detection accuracy: an author's propensity to publish fake news and a topic's tendency to produce fake news discussed in [3]. Strengthening the hybrid model is highlighted by the significant increase in accuracy when both linguistic and fact-verification features are used together instead of separately. This combination of approaches results in a synergistic effect, as evidenced by the significant increase in accuracy (89.4% with linguistic features alone versus 81.2% with fact-verification features alone). Further validating the system's effectiveness could involve examining its performance on various datasets or assessing its resilience in different situations. As demonstrated by the impressive evaluation results on the BuzzFeed political news dataset, the suggested system significantly outperformed the RFC, achieving an accuracy rate of 94.4%. These results are more accurate than the results of using fact-verification features alone and linguistic features alone. Furthermore, increasing the system's capacity to identify additional forms of misleading information, like biased or inaccurate news or confusing or unclear news, suggests a thorough method for combating false information in all its forms and categories [4].

Lemmatization is used to reduce dictionary size and remove superfluous characters and numbers from the data as part of the framework's preprocessing step. Essential features are extracted using feature extraction techniques such as word embedding, document-to-vector algorithm. The analysis of variance and chi-square algorithms are employed to minimize features. Three online datasets are used for evaluation: ISOT, Media-Eval, and Fake-or-Real-News. Five metrics evaluate the system's performance: f1-score, area under the curve, recall, accuracy, and precision. Remarkably, this framework obtains high accuracy rates on all datasets: 100% for ISOT, 92.3% for Media-Eval, and 94.6% for Fake-or-Real. A comparative study with alternative classification algorithms demonstrates how well this suggested framework performs compared to previous efforts, particularly attaining a marginal but noteworthy increase in accuracy of 0.2% for the ISOT dataset. Furthermore, investigating the framework's effectiveness in various fake news scenarios or datasets may improve its legitimacy and usefulness. The intention is to annotate these datasets in order to make it possible to identify phoney images and videos, thereby greatly expanding the scope of identifying phoney news [5].

The optimized convolutional neural network (OPCNN-FAKE) was designed to identify false news. Utilizing four benchmark datasets, compare OPCNN-FAKE to other models such as long short-term memory (LSTM), recurrent neural network (RNN), and six classic machine learning techniques: K-Nearest Neighbour (KNN), DT, LR, SVM, RF, and Naive Bayes (NB). Benchmark datasets are used to test feature extraction techniques such as N-gram, glove word embedding for Deep Learning (DL) models, and term frequency-inverse document frequency (TF-IDF) for ML models. The results demonstrate that the OPCNN-FAKE model consistently performs better than alternative models in all datasets. Furthermore, it outperforms other models in the testing and cross-validation stages, indicating its superiority in identifying false news. This all-encompassing method, which employs a range of ML and DL techniques and different feature extraction techniques, shows a solid approach to tackling the problem of detecting fake news. OPCNN-FAKE has demonstrated its superiority over a variety of evaluation metrics and datasets, indicating its potential as a valuable tool in the fight against disinformation [6]. Another algorithm presents a novel training approach designed to efficiently utilize user graphs in the context of Korean fake news detection. Studies comparing and analyzing data indicate that this model outperforms baseline techniques in identifying false news. These results emphasize how meaningful user interactions are in identifying false information, even when hate speech exists.

Moreover, this study confirms the reliability of stance information by expanding its scope and tackling the problem of class imbalance, thereby emphasizing its potential significance in enhancing approaches for detecting fake news. This study introduced a novel approach to detecting fake news using a graph-based learning strategy. Unlike traditional single-article body-centred learning methods, the model effectively categorized fake news based on the reader's information and the article's content. In addition, ablation research and comparing experiments were conducted to investigate methods of incorporating user information into the challenge of detecting bogus news [7]. The proliferation of fake news, whether from human or machine sources, has a detrimental impact on society and individuals in both political and social domains. The rapid turnover of news in the social media era makes it difficult to assess its veracity quickly. Convolution neural network (CNN) and LSTM capabilities are combined in a hybrid neural network architecture, along with two different dimensionality reduction techniques: principal component Analysis (PCA) and chi-square are discussed.

The suggested model raises the accuracy and $F1-$ score by about 4% and 20%, respectively. According to the experimental results, PCA performs with 97.8% accuracy better than state-of-the-art techniques and chi-square. Unlike earlier research that focused only on individual sentences or phrases, a model for identifying fake news stances based on the headline and news body was presented in this study. PCA and Chi-Square extract high-quality features that are fed into the CNN and LSTM models in the proposed model, combining CNN and LSTM [8]. This study evaluates three DL and five ML models using holdout cross-validation on two datasets, one false and the other natural and of varying sizes. To obtain text representation, embedding techniques, term frequency, and term frequency-inverse document frequency were also used. It is impressive how this work explores different algorithms and datasets to handle the issue of false news identification. A systematic approach to performance improvement is indicated by the stacking method proposed and the use of McNemar's test for model comparison. This technique's applicability and dependability can be enhanced by collecting diverse datasets from different nations, expanding experiments to new languages, and investigating more models. Their suggested noval stacking model produced accuracy levels of 96.05% on the KDnugget dataset and 99.94% on the ISOT dataset [9].

The current hetero-generous graph based fake news detection model performs better efficiency and scalability than traditional content detection models because it focuses primarily on the semantic consistency analysis between external knowledge and news content. Nevertheless, the framework fails to consider that short text makes up most of the node content in heterogeneous graphs. Short text data presents a sparsity issue that frequently makes extracting useful features difficult for such methods. Furthermore, the various fake news writing styles. This approach proposes a fake news detection (FND) method centred around topic awareness and utilizes heterogeneous graphs. The model improves the discriminative power of fake news detection by examining the impact of news topics on detecting fake news. The model explores how news topics affect FND and improve its capacity for discrimination in [10]. This model fully utilizes three types of information by introducing semantically enhanced topic node information in the fake news detector: outside information news topics, news content, and Wikipedia. As a result, it can improve the performance of fake news detection. Despite much research, developing an accurate detection system remains difficult. To enhance the detection accuracy of fake news, this article suggests a novel model. The discussion utilizes two datasets to assess a deep neural network architecture's efficacy in detecting fake news. Some of the research's contributions include the use of TF-IDF and deep NN for feature extraction and model testing on various dataset sizes in the classification stage. Two distinct datasets have been used for the experiments. For most techniques, the proposed model aims to create a new fast classifier by extracting and combining the text's global, spatial, and temporal features.

The suggested model is divided into two stages: in the first, TF-IDF extracts global features, a CNN extracts spatial features, and a BiLSTM extracts temporal features simultaneously. Next, quickly using a learning network, the features are efficiently classified. Numerous tests were carried out with the ISOT and FA-KES fake news datasets, both accessible to the general public. Due to the differences in size between these two, a much better evaluation of the suggested architecture CNN and Bi-LSTM is possible. The result's shows that the suggested models are better than earlier efforts [11]. The public and society have suffered grave consequences due to the dissemination of fake news in several domains. Ultimately, a determination of whether the news is authentic or fraudulent is made by feeding the discriminator with the complete feature set. The SLFEND model's usefulness and effectiveness are demonstrated by experiments conducted on the weibo21 dataset. The final overall feature representation is obtained by extracting multi-domain features from news using soft labels. Lastly, a classification module with a multilayer perceptron structure is used to assess the news's veracity. Tests conducted on the weibo21 dataset demonstrate the improved performance of the model [12]. This unbridled freedom has also facilitated the proliferation of fake news spread worldwide during the COVID-19 outbreak, adversely affecting authorities' decisions and people's health. Consequently, governments, media outlets, and academic institutions have established fact-checking units and automated detection systems.

Building knowledge bases that can be used as a resource for fact-checking and researching writing styles and propagation patterns were the main focus areas for research methods used to confirm the accuracy of news reports. It presents a multilingual framework based on available evidence and a source's credibility score to assess the credibility of online news. It used the covid-19-related news as a case study to demonstrate the benefits of this method [13]. Social media sites like Facebook, WhatsApp, Twitter, and Telegram are important channels for disseminating information in the modern world, and individuals trust these sources even when they are not entirely credible. Due to social media's ease of use, spreading false information has captivated users' worldwide accessibility, affordability, and simplicity of information exchange. It is possible to create fake news to deceive the public for financial or personal benefit. It can also be used for other private gain, like discrediting well-known individuals or changing laws. Therefore, numerous research studies have been conducted to detect fake news accurately and avert its disastrous consequences. Next, it trains a selection of ML models discussed in [14].

On social networks, fake news can be considered an anomaly, and the fundamental technique for unsupervised learning is autoencoder. Thus, the Un-supervised fake news method based on autoencoder (UFNDA) is suggested in [15, 16]. The residual is then reconstructed to identify false news. The experimental findings were contrasted with two real-world datasets. The presence of four supplementary approaches illustrates that UFNDA yields advantageous outcomes [17–19]. This model addresses the issue of unsupervised fake news detection by presenting an autoencoder-based approach. Using fake news as anomalous data, first extract and combine Twitter's user, image, propagation, and text content features in social networks. Next, apply the suggested method UFNDA to analyze the extracted and fused data, and lastly. Even with its good performance, UFNDA could still use some work. Many features found in social networks, like comments, news distribution, and videos, are crucial for identifying fake news. Furthermore, more thorough categorization must be considered because current events are neither genuine nor fraudulent. The implementation of methodology covered in Sect. 2. Section 3 presents the experimental results and Lastly, Sect. 4 Lastly provides a summary of the conclusions.

2 Methodology

In section outlines our project's procedures and techniques to determine whether news is fake. Additionally, all the information about the datasets is discussed here. Explains data preprocessing, data exploration, and the use, operation, and application of the algorithms employed. Figure 1 depicts the project's overall layout. Both genuine and fake news are included in the datasets used in this investigation. We have two datasets: one contains fake news, and the other contains real news. Table 1 depicts the information regarding datasets which were taken from the Kaggle platform. There are over twenty-one thousand examples of real and fake news in each file. The datasets we gathered took into account the class, date, subject, title, and text. Depending on its intended use, thousands to millions of records could be in the dataset. We preprocessed, trained, and test-split the dataset after it had been analyzed. We then applied the four machine learning classification models to it and tested them using the test set. The labelled data is used to train machine algorithms that determine.

Table 1. Datasets

S. no	Dataset	Frequency
1	Fake News	23471
2	True News	21407

Prior to the stages of training, testing, and modelling, the data must be pre-processed. Fake and real news are combined before going on to these stages. We eliminated the columns from the datasets not required for processing during the dataset-cleaning procedure. Additionally, the stop words and punctuation were eliminated. Stop-words are words that crop up a lot, like "I," "are," "will," "shall," "is it," and so on. Letters in uppercase were changed to lowercase. The dataset appeared good and was prepared for the next exploration stage after cleaning it. Nevertheless, the dataset exploration was finished on the cleaned and uncleaned data to conduct more thorough research. The fictitious and real datasets were combined into a data set for exploration. We used four machine learning classifications: LR, DTC, GBC, and RFC.

After data pre-processing and exploration, we applied it to the training and testing section. This part of the data-set is used to instruct or train the machine learning models. It comprises input-output pairs, where the corresponding labels or target values are the outputs and the inputs are the data features. To generate predictions, the model analyses relationships and patterns found in the training set of data. This model modifies its internal parameters in the training phase in response to patterns found in the training data. The model should generalise well to new, untested data. Here, we used four machine learning models to achieve better results. The testing data is an independent subset of the dataset that the model did not encounter during training, also known as the validation or evaluation set. It evaluates the trained model's ability to generalise to fresh, unobserved examples. The testing data offers insights into the models capacity to generate precise predictions outside the training set and aids in estimating the models performance on real-world data.

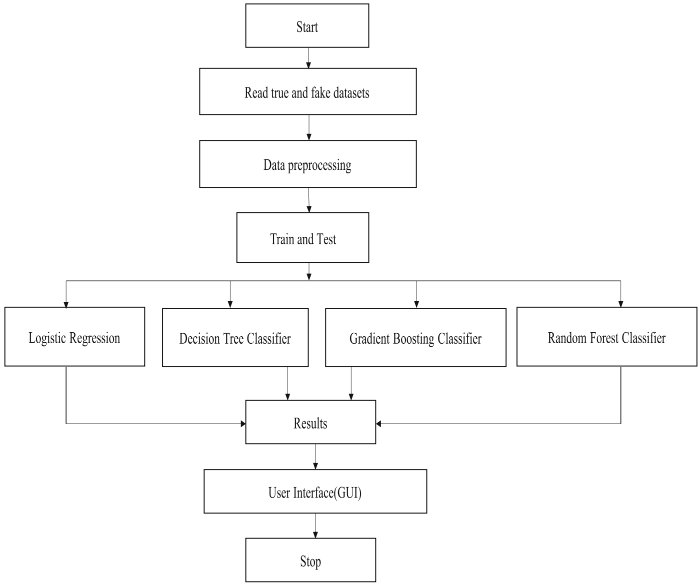

Fig. 1. Architecture of fake news detection model

The dataset splitting produces two subsets: the training and testing sets. Typically, the split is conducted randomly to guarantee that both subsets accurately represent the whole data distribution. The model is trained on the training set through iterative parameter adjustments, aiming to minimize the disparity between the predictions made by the machine learning model and the actual labels in the training data. Once the model has undergone training, the testing set is employed for evaluation. Various measures, including accuracy, precision, recall, and F1 score, are employed to evaluate the model's performance, depending on the type of task, be it classification or regression. The training and testing in our model are done through all four ML algorithms. One method is enough for training and testing, but we compared it with classifiers for better results and accuracy. When we trained and tested with LR, also called a sigmoid function, a supervised machine learning algorithm gave an accuracy of approximately 98%. It produces a probability value between 0 and 1 based on inputs that are independent variables. As an illustration, there are two classes: class 0 and class 1. The sigmoid function is $\partial(z) = 1/1 + e\hat{}(-z)$, where z is the linear combination of input features and weights. Results from LR are comprehensible. Each feature's coefficient can be understood as how that feature affects the log odds of the expected result. It also gives the classification report accuracy, recall, F1-score, and precision.

Decision trees, while in the supervised learning technique category, are predominantly employed to resolve classification difficulties. Moreover, they can also be employed to address regression difficulties. The structure of DTC is hierarchical, like a tree. The leaf nodes correspond to individual results, the branches reflect decision rules, and the core nodes indicate dataset attributes. A decision tree comprises two types of nodes: the decision node, which represents a decision or a question, and the leaf node, which represents an outcome or a result. Decision nodes are utilized for making determinations and possess numerous branches, while leaf nodes represent the result of decisions and do not encompass any additional branches. The attributes of the given dataset are utilized to guide the decisions or the examination. It is a graphical tool that shows all options for solving a problem or deciding given specific parameters. Due to its advanced features, it gives better results than the LR. The precision of DTC is almost 99%. Gradient descent is employed to train each new model to minimize the loss function, which may be the previous model's cross-entropy or mean squared error. Gradient boosting is a robust algorithm that enhances the performance of poor learners by transforming them into strong learners. During each iteration, this technique calculates the gradient of the loss function concerning the predictions given by the current ensemble.

Subsequently, a novel weak model is trained to minimize this gradient. The forecasts generated by the novel model are subsequently incorporated into the ensemble, and this iterative process continues until a predetermined termination criterion is satisfied, enhancing the precision. Finally, we used RFC, the well-known ML algorithm. It can be used to solve regression and classification-based machine-learning problems. It improves the functionality of the model. It creates many decision trees in the training phase. At the conclusion, it generates a class that is the average prediction (regression) or the class mode (classification) of each tree [20, 21]. The use of randomness in selecting data points and features for each tree's construction distinguishes RF. A random subset of the training data is sampled with replacement (bootstrapped) at each node during the construction of each tree, and a random subset of features is considered for splitting. The diversity improves the robustness and generalisation performance of the model as a whole that this randomness fosters among the trees. All four of the ML algorithms' performances were examined and contrasted [22, 23]. Following an analysis of the four machine learning algorithms' performances [24, 25], we also suggested the interface for user interaction. The point of interaction between a user and a computer or software program is called a User Interface (UI). It includes every aspect that allows users to interact with a system: visual, aural, and tactile. The term GUI refers primarily to the UI's visual elements, which include windows, menus, buttons, and icons, and with which users interact. A user-friendly experience is the goal of effective GUI design, which takes accessibility, usability, and aesthetics into account. For an easy-to-use and effective interaction, it should lead users through tasks, provide clear feedback, and remain consistent. To display input and output, we created a GUI interface. That is built up with a Python library, namely tkinter. If we give input to the top box, we get output from the dialogue box. Attributes are taken according to our choice. As our code is in Python, we created a GUI with Python library rather than preferring HTML, CSS, or other languages.

3 Methodology

The results were implemented on Jupyter Notebook, i.e. from Anaconda, using all four models, such as LR, DTC, GBC, and RFC. Python 3 was utilised for this. We used matplotlib, sklearn, seaborn, pandas, numpy, and Matlab libraries for training and testing. The evaluated results provide the classification report of accuracy, precision, recall, F1-score, and support. Initially, the test dataset was used to assess the LR. It achieved an accuracy of 98.69%. The accuracy of DTC is 99.69%, the accuracy of GBC is 99.56%, and the accuracy of RFC is 99.03%. Table 2 illustrates the accuracy, precision, recall and F1-score of all four models. To assess each proposed model, we compare its precision, accuracy, F1 score and recall. Several evaluation metrics were used to assess the effectiveness of the established models, including recall, F1-score, accuracy, and precision. A key performance indicator in machine learning, accuracy gauges how accurate a model is overall in all of its class predictions. It is frequently used to assess a model's performance on a particular dataset in classification tasks. The ratio of accurate predictions to total predictions is used to calculate accuracy. A model's accuracy gives an overall impression of how well it works across all classes. However, it might not be the most helpful metric when there is an imbalance in the classes. Precision is a crucial metric in machine learning, particularly in classification tasks. It is a metric for how well a model predicts the positive outcomes. The ratio of actual optimistic predictions to all of the model's positive predictions is known as precision. A high precision value means that the model does not misclassify negative cases as positive and is good at correctly identifying positive cases.

Table 2. Classification report of four machine learning algorithms

Model	Logistic Regression	Decision Tree Classifier	Gradient Boosting Classifier	Random Forest Classifier
Accuracy	0.9869	0.9969	0.9956	0.9903
precision	0.99	0.99	0.99	0.99
Recall	0.99	1.00	1.00	0.99
F1-score	0.99	1.00	1.00	0.99

Recall, also called sensitivity or actual positive rate, is a metric that assesses a model's capacity to detect and accurately identify every instance that pertains to a given class. It is especially crucial when there is a significant risk of missing positive cases (false negatives). The ratio of accurate optimistic predictions to the total number of positive instances is known as recall. In applications where missing positive instances can have detrimental effects, recall is an essential metric as it sheds light on the model's capacity to identify positive instances accurately. It is crucial to remember that recall and precision frequently trade-off. While precision highlights the accuracy of the optimistic predictions made by the model, recall concentrates on capturing all positive instances. F1 score strikes a balance between recall and precision by combining the two into a single number. It is beneficial when there is an imbalance between binary classification tasks' positive and negative classes. When precision and recall are crucial, and a balanced strategy is required, the F1 score helps assess a model's performance. In simple words, accuracy, precision, F1-Score, and recall are defined as follows: accuracy is the ratio of correctly classified samples to total samples in a test dataset (1). *Precision* is the ratio of true positive samples to the sum of true positive and false positive samples (2). The precision and recall weighted average are known as the F1-score. *Recall* is the ratio of true positive samples to the total of false negative and true positive samples (3). The efficiency of machine learning algorithms is assessed using the F1-score value (4). The true negative rate represents the specificity. Regarding the accuracy, precision, recall, and F1-score, the following explanations are applicable: TP, TN, FP, and FN represent the quantities of true negative, true positive, false positive, and false negative samples, respectively.

$$Accuracy = TP + TN/TP + TN + FP + FN \quad (1)$$

$$Precision = TP/TP + FP \quad (2)$$

$$Recall = TP/TP + FN \quad (3)$$

$$F1 - score = 2 \times Precision \times Recall\,/Precision + Recall \quad (4)$$

The accuracy, recall, precision, F1-score, and specificity of each suggested model are displayed in Table 2. It also shows the comparison of all four models. The precision of all four algorithms was the same, i.e. approximately 99%. The recall calculated of LR, RFC was approximately 99% and DTC, GBC was 100%. The F1-Score is the same as the recall. After testing the results using all four algorithms, we analysed and compared the performance by accuracy, precision, recall and F1-score. Figure 2 depicts the performance of each model. Here, we used two datasets: i.e. one for fake news and the other for real news. For better performance, we used a dataset comprising almost twenty-one thousand examples. All four ML algorithms that worked and predicted on this dataset are observed to be similar. Scale: 1 unit = 0.002% of accuracy.

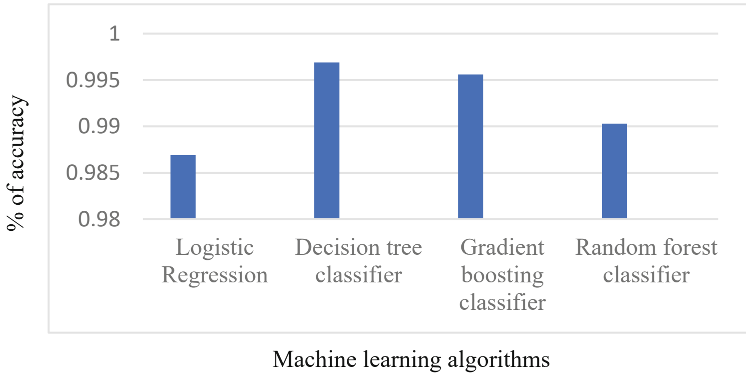

Fig. 2. Performance comparison of four machine learning algorithms

After manual testing, we created a user interface to make technology accessible and easy to use; fake news detection systems require a GUI. GUIs provide an intuitive interface for input, configuration, and result interpretation, allowing users with varying backgrounds and levels of technical expertise to interact with the system with ease.

Figure 3a and 3b depicts the output of the GUI interface. The fake news detection system becomes more accessible to a broader audience. GUIs enhance interpretability, transparency, and real-time interactions by including explanations and visualizations while allowing users to comprehend and rely on the model's predictions. Additionally, GUIs enable feedback, error correction, and customization, providing a seamless and customized user experience. Their ability to be integrated with other tools and systems and their function in outreach and education initiatives further emphasize the significance of encouraging the broader use and usefulness of fake news detection technologies through the use of GUIs. In our model to display input and output, we created an interface GUI. That is built up with a Python library, namely tkinter. If we give input to the top box, we get output from the dialogue box. Attributes are taken according to our choice. Instead of a GUI interface, we can create web pages or other applications. However, as our code is in Python, we created a GUI with a Python library rather than preferring HTML, CSS, or other languages. This model could be highly useful in context with the future smart city application [26, 27].

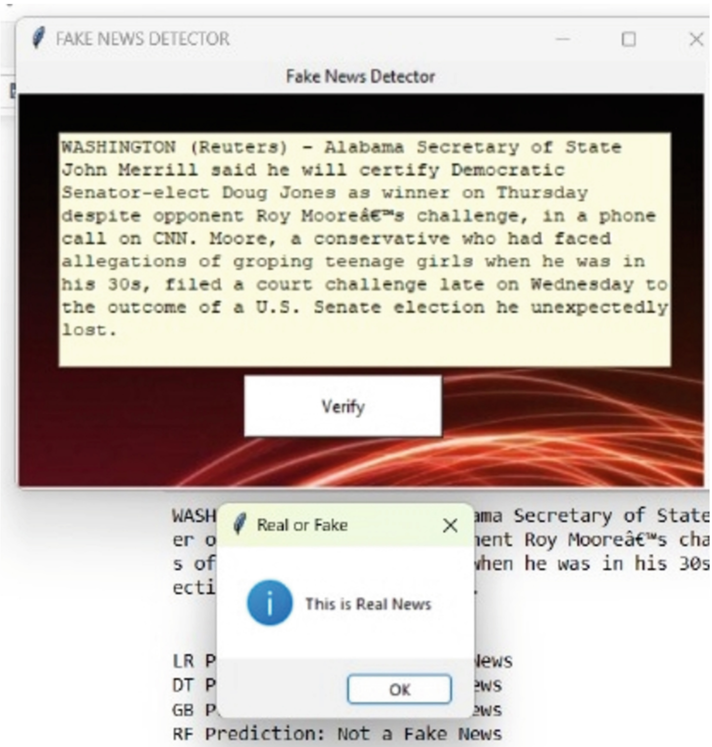

Fig. 3. a. Interface Output

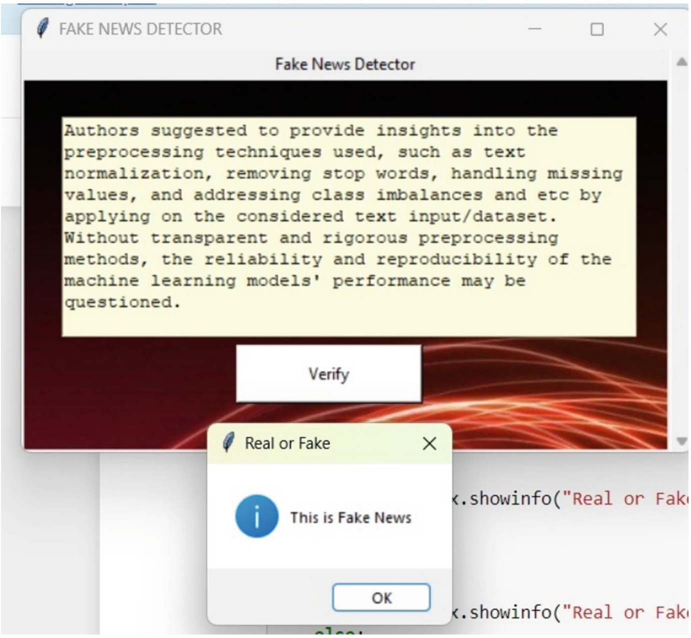

Fig. 3. b. Interface Output

4 Conclusion

Our social media produce every type of news, most fake. Typically, when two realities conflict over a similar issue, we question if either is true. We got ourselves into a bind trying to decide which source to trust. It is essential to clean the dataset. It is important because it alters the conclusions of the study. However, when the dataset still needs to be cleaned, terms like the appear most frequently. When used alone, these words have no identity and no meaning until combined with other terms. Therefore, the datasets should be cleaned for accurate results. In closing, the writer would like to point out that while disseminating false information occasionally makes people happy, it usually makes people sad. As soon as possible, fake news should be stopped from spreading. I use top-notch machine-learning algorithms like LR, DTC, GBC, and RFC. We were able to achieve some fantastic outcomes in our research. The algorithms displayed nearly flawless accuracy of over 98%–99%. LR, DTC, GBC and RFC models' accuracy were approximately 98.69%, 99.69%, 99.56% and 99.03%, respectively. As a result of this research, people who are somewhat addicted to the internet should no longer be afraid of fake news. In the end, the presented paper had some shortcomings and restrictions. These happen when a dataset is unbalanced or has not been cleaned; otherwise, it may be ineffective and produce inaccurate results. Regarding processing time, Spark machine learning, an extensive data framework, may produce better results. Furthermore, datasets containing fake news could be used with deep learning-enabled big data models.

References

1. Minjung, P., Sangmi, C.: Constructing a user-centered fake detection model by using classification algorithms in machine learning techniques. IEEE Access, 71517–71527 (2023)
2. Gupta, A., Kumar, N., Prabhat, P., et al.: Combating fake news: stakeholder interventions and potential solutions. IEEE Access, 78268–78289 (2020)
3. Liao, Q., Chai, H.., Han, H., et al.: An integrated multi-task model for fake news detection. IEEE Trans. Knowl. Data Eng., 5154–5165 (2022)
4. Seddari, N., Derhab, A., Belaqued, M., et al.: A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media. IEEE Access, 62097–62109 (2022)
5. Elsaeed, E., Ouda, O., Elmogy, M.M., Atwan, A., El-Daydamony, E.: Detecting fake news in social media using voting classifier. IEEE Access, 161909–161925 (2021)
6. Saleh, H., Alharbi, A., Alsamhi, S.H.: OPCNN-FAKE: optimized convolutional neural network for fake news detection. IEEE Access, 129471–129489 (2021)
7. Kang, M., Seo, J., Park, C., Lim, H.: Utilization strategy of user engagements in Korean fake news detection. IEEE Access, 79516–79525 (2022)
8. Umer, M., Imtiaz, Z., Ullah, S., MehMood, A., Choi, G.S., On, B.-W.: Fake news stance detection using deep learning architecture (CNN-LSTM). IEEE Access, 156695–156706 (2020)
9. Jiang, T., Li, J.P., Ul Uaq, A., Saboor, A., Ali, A.: A novel stacking approach for accurate detection of fake news. IEEE Access, 22626–22639 (2021)
10. Sun, L., Wang, H.: Topic-aware fake news detection based on heterogeneous graph. IEEE Access, 103743–103752 (2023)
11. Almarashy, A.H.J., Feizi-Derakshi, M.R., Salehpour, P.: Enhancing fake news detection by multi-feature classification. IEEE Access, 139601–139613 (2023)
12. Wang, D.A., Zhang, W., Wu, W., Guo, X.: Soft-label for multi-domain fake news detection. IEEE Access, 98596–98606 (2023)
13. Hammouchi, H., Ghogho, M.: Evidence-aware multilingual fake news detection. IEEE Access, 116808–116818 (2022)
14. Rohera, D., Shethna, H., Patel, K., et al.: A taxonomy of fake news classification techniques: survey and implementation aspects. IEEE Access, 30367–30394 (2022)
15. Dun, L., Haimei, G., Zhenfei, W., Zhiyun, Z.: Unsupervised fake news detection based on autoencoder. IEEE Access, 29356–29365 (2021)
16. Alekya, L., Lakshmi, L., Susmitha, G., Hemanth, S.: A survey on fake news detection in social media using deep neural networks. Int. J. Sci. Technol. Res. **9** (2020)
17. Raji, K., Hemamalini, S.: Optimal energy management of virtual power plants with storage devices using teaching-and-learning-based optimization algorithm. Int. Trans. Electr. Energy Syst., 1727524–1727541 (2022)
18. Sangeeth, G., Jayanthi, D., Raji, K., Ilanchezhian, P., Sam, D.S.S.: Malware identification in advanced interconnects on SOC. In: 2022 International Virtual Conference on Power Engineering Computing and Control: Developments in Electric Vehicles and Energy Sector for Sustainable Future (PECCON), Chennai, India, pp. 1–7 (2022)
19. Raji, K., Hemamalini, S., Zhou, N.: Forecasting uncertainty parameters of virtual power plants using decision tree algorithm. Electr. Power Compon. Syst. (2023)
20. Satyanarayana, B.V.V., Kumar, G.P., Varma, A.K.C., Dileep, M., Srinivas, Y., Budumuru, P.R.: Alzheimer's disease detection using ensemble of classifiers. In: Gupta, N., Pareek, P., Reis, M. (eds.) IC4S 2022. LNICSSITE, vol. 472, pp. 55–65. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28975-0_5
21. Venkata Subbarao, M., Challa Ram, G., Ramesh Varma, D., Girish Kumar, D., Prema Kumar, M.: Automatic fault detection in industrial smart grids using KNN and ensemble classifiers. ECJSE **10**(2), 240–252 (2023)

22. Raji, K., Srujana, B., Jayanthi, D., Kavitha, K., Maurya, N.K.: Investigation of cylindrical gate all around field effect transistors exploring variation in low voltage using visual TCAD. In: 2023 International Conference on Next Generation Electronics (NEleX), Vellore, India, pp. 1–5 (2023)
23. Megha, P.S., Ramnath, V., Raji, K., Babitha, V., Chacko, B.: Production of zinc enriched designer eggs through dietary supplementation. J. Vet. Anim. Sci., 77–80 (2021)
24. Raji, K., Suganya, R., Yaganti, M.S., Buhari, S.M.: Shape organization for enhanced design optimization in generative systems using clustering technology. In: 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, pp. 569–574 (2023)
25. Raji, K., Sariki, T.P., Sigamsetty, S.M., Kumar, G.B.: Aviation safe landing prediction utilizing long-short term memory algorithm. In: 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, pp. 569–574 (2023)
26. Pareek, P., Maurya, N.K., Singh, L., Gupta, N., Reis, M.J.C.S.: Study of smart city compatible monolithic quantum well photodetector. In: Gupta, N., Pareek, P., Reis, M. (eds.) IC4S 2022. LNICSSITE, vol. 472, pp. 215–224. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28975-0_18
27. Maurya, N.K., Kumari, S., Pareek, P., Ghosh, J., Reis, M.J.C.S.: Tunable UWB metasurface absorber for smart city compatible IoT applications. In: Pareek, P., Gupta, N., Reis, M.J.C.S. (eds.) IC4S 2023. LNICSSITE, vol. 537, pp. 231–240. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-48891-7_20