Fake News Detection using Machine Learning

Dhanyashree[1], Aparna N[2], Prathibha[3], Deeksha[4], Nisha[5]
[1,3,4,5]*Student, Department of Information Science and Engineering, Srinivas Institute of Technology, Mangalore, Karnataka, India*
[2]*Assistant Professor, Department of Information Science & Engineering, Srinivas Institute of Technology, Mangalore, Karnataka, India*
doi.org/10.64643/IJIRTV12I12-206751-459

***Abstract*—The world of digital media is increasing at a very rapid pace in Recent Years. Due to this, too much fake news is also being created. It is a serious problem that is becoming one of the greatest threats of the world. It alters the thinking of people and creates misunderstanding. Additionally, it causes numerous misunderstandings or befuddlement. In addition to this, it also has social and political effects. Though many people read the news properly, some don't. It is becoming hard to contain fake news because of this. For building a fake news detector, we are using the model building method of machine learning. We will be able to recognize valid news and fake news with the help of this. Initially, data collection was done by us. The cleaning process was performed afterward. In this step, we did the tokenization, removal of stop words and normalisation. At first, these are important but little confusing. Then, we used TF-IDF to change the text into numbers. The model can't understand the text, so we convert all the text to numbers. Subsequently, algorithms such as Logistic Regression, Passive-Aggressive Classifier, Naïve Bayes, and SVM were utilized. Some models yield better results than others do. We have built a web application using Flask. Users can input the news text and acquire the outcome in this. It is not complicated, easy to use. For performance, we use accuracy and Precision and recall and F1 score. This model is correct up to 99%. It could have a bit of error sometimes, but overall, it is pretty much correct. The method to detect the false information includes machine learning and NLP for feature extraction and classification of the text. Passive aggressive classifier, TF-IDF vectorization.**

***Index Terms*—False news detection, feature extraction, machine learning, natural language processing, passive aggressive classifier, text classification, TF-IDF vectorization.**

I. INTRODUCTION

The internet has been identified as a tool for obtaining information in the contemporary world due to the enormous number of websites and social networking sites that provide instant information. However, the internet may be considered advantageous in terms of obtaining information in a number of ways; however, there is the problem of authenticity of information owing to the large number of misinformation sources on the internet. Such cases were not many; however, due to the fast dissemination that takes place in the modern day, they have become very common, thereby questioning the accuracy of the information posted in the websites.

It is no longer possible to ascertain the accuracy of any piece of information in the modern day due to the presence of numerous information in the internet. In order to overcome such shortcomings, there is a need for automated approaches to effectively interpret and categorize the information at hand. With machine learning algorithms, it would be possible to analyze vast amounts of data and recognize any hidden patterns that would not otherwise be obvious using traditional approaches. This would allow news articles to be evaluated and classified according to their validity.The proposed project is related to developing an algorithm through which we will be able to classify the news article into either a fake or a genuine news piece. First, we will carry out the text pre-processing phase, followed by conversion using TFIDF method.

After completing the above two tasks, we can move on to classification through machine learning methods like logistic regression, passive-aggressive classifier, and naive Bayes classifier. The application is developed through Flask, and our project is turned into a web-based application.

II. LITERATURE REVIEW

For instance, Potthast et al. (2017) have researched how writing style and vocabulary could help recognize

false or biased information. The approaches they employed were Random Forest and SVM. The findings of their study revealed that hyper partisan news items employ emotionally and persuasively loaded language similarly regardless of which political camp they represent. However, it was also noted that analysis of writing style alone cannot be considered sufficient when it comes to recognizing false news content. As a solution to the issue regarding the availability of good-quality data, Wang (2017) suggested the LIAR data set, which comprises more than 12,000 statements relating to politics, which were collected from the site of PolitiFact. The use of logistic regression, SVM, and LSTM models on this data set has proven that better results are achieved by considering speaker information and text together.

Apart from CNN and Capsule Networks methods, another method of detecting fake news called CSI was developed by Rucha sky et al. (2017), which also made use of deep learning. Here, the algorithm known as LSTM was utilized in order to analyse user interaction with news. This means that the algorithm does not only consider the news article but the interactions between users and the particular news. In the case of the CSI algorithm, the authors Monti et al. (2019) improved it through GCNN methodology. Specifically, they made use of geometric deep learning techniques in order to describe social media interactions, suggesting that fake news can be identified irrespective of the content.

In another study, a method similar to that used here was employed in Katiyar et al. (2021), where the authors came up with the concept of Fake BERT. In this case, a combination of the BERT model with CNN architecture is used. The performance of this approach was higher compared to other approaches like LSTM, SVM, and basic BERT.

III. METHODOLOGY

The methodological framework for the implementation of the suggested system involves systematic approaches to data processing that begin with the extraction of information from datasets consisting of actual and false news stories. Next, the acquired data will undergo preprocessing that involves the removal of noise such as stop words, punctuation marks, and other unnecessary symbols.

This will be followed by tokenization and normalization. The TF-IDF vectorization technique will be applied to convert data to numeric form to understand the significance of a particular word in a document.

The next stage will involve the utilization of machine learning algorithms on the data set for learning and classification. Machine learning algorithms' performance will be assessed, and the best-performing one will be selected. Lastly, the selected algorithm will be deployed using flask web application. This will allow users to input a news article and obtain immediate results.

Hardware Requirements:
Processor, RAM, Storage specs

Software Requirements:
Python 3.8, Flask, Scikit-learn, MySQL, Jupiter, libraries list

Hardware Requirements:

- Hardware Requirements: Processor, RAM, Storage specs
- Software Requirements: Python 3.8, Flask, Scikit-learn, MySQL, Jupiter, libraries list

IV. SYSTEM ARCHITECTURE

It is important for users to deal with the data in a way such that they can enter the data into the system without changing anything about it. It was thus decided that a user-friendly interface should be developed using Flask, which functions through the Internet. User-friendly interfaces ensure efficient interaction between users and the system. Flask allows users to perform actions much better and faster. There are a number of things that will be required while making videos using the script. As is claimed by professionals, the use of these tools improves the quality of the scripts and improves the accuracy of timekeeping; thus, all the elements take their rightful place. As these tools influence the scripts, the result is improved in its quality. The process of data processing starts with the cleansing of the data. This is done through the removal of unnecessary words, punctuation marks, and other elements from the data. Through the use of this tool, the data is cleansed for its quality requirements. Individuals assume that it is impossible to get any information from the data

without coding it into a language understandable by the computer.

Actually, one of the main functions that are found in the system is Flask interface. In consequence of the use of the system, people will be able to perform their tasks much quicker and more efficiently. No matter whether it is a video production process or data preparation, all the operations should be simplified and become more accurate owing to this software.

People will be able to manipulate with data and get all the information that is necessary for them because of such a useful system. It is very effective and easy to use, which enables people to pay attention to other things. The visualize predictions module identifies crucial linguistic indicators while the visualize predictions module allows people to understand the logic of the machine. Classification Process: The data extracted from the text is fed into the machine learning model as the model needs to find out repetitive patterns. These mathematical models, according to experts, operate in such a way that they are used to make a decision regarding whether the information is fake or true. Since the systems vary in their structure, there is variation in the working of the classification process for each system. After the analysis is done, the machine provides a decision of either "Fake" or "Real". Moreover, it provides a confidence level as well. It tells us how confident the machine was regarding its prediction.

However, the fact that confidence levels are generated does not guarantee the accuracy of predictions. But on this issue, it is certain that the visualization process will never be completed. Visualization Prediction Module: The predictions data will be transferred to the visualization module for display purposes. It is beyond doubt that the visualization process of predictions and graph data is of benefit to the one the visualize predictions module identifies crucial linguistic indicators while the visualize predictions module allows people to understand the logic of the machine.

V. MODEL TRAINING

Training Dataset: Two datasets named “Fake.csv” and “True.csv” were selected because the model requires specific information to learn. Such journalistic reports provide the reporting information because this reporting information helps the model understand the difference between inaccurate and accurate stories. Information from these files is employed for training the model to detect truth. Experts claim that the reporting information must be examined and refined properly before the actual training starts. Even though some unnecessary values might remain inside the reporting information, the dataset is suitable for training the model because the dataset remains high in quality. Algorithms: Various machine learning algorithms perform the main tasks within this project. Logistic Regression, Passive Aggressive Classifier, Naïve Bayes and Support Vector Machine (SVM) are It's used for this purpose.

We've noticed that each one algorithm follows a different logic to reach its goal. Certain models achieve a higher efficiency while other models show a lower quality. A Passive Aggressive Data sources (news, social media Data Integration Layer Data preprocessing, cleaning, tokenizing Layer feature engineering NLP + Metadata + Graph ML models (Transformers, GNN, etc) Prediction services UI/Dashboard/API layer Classifier is currently providing superior outcomes in this specific situation because the Passive Aggressive Classifier matches the data characteristics.

Vectorization Technique: Since machine learning models cannot understand text directly, the text is converted into numerical form. For this, TF IDF (Term Frequency - Inverse Document Frequency) is used. It converts text into vectors based on word importance. N - grams are also used to improve the feature extraction. This helps the model to understand the con text better. For this, TF IDF (Term Frequency - Inverse Document Frequency) is used. It converts text into vectors based on word importance. N - grams are also used to improve the feature extraction. The extracted features improve the representation of textual data for classification tasks. Following preprocessing and feature transformation, the dataset is provided to the selected algorithms for training. Here, the models will be able to recognize some features for distinguishing real news from fabricated ones. The amount of time taken depends on the amount of data involved and the capacity to process that data. In the end, once the model has been trained, the model is tested using another data set.

The attained accuracy is 99.34%, which shows that the model is doing an outstanding job. The other metrics, including Precision, Recall, and F1-score, are all above 0.99, meaning that the model performs

accurately more often than not. Accuracy of Testing Data Set: The accuracy of this model is 99.34% while its precision, recall, and F1-score are all around 0.99. This means that the accuracy level of classification is very high, which shows that this system is doing an excellent job since most of the predictions are accurate. Despite having some small errors here and there, the accuracy rate remains very high.

A. Python 3.8

Python is a high-level programming language and it is used in data science and machine learning a lot. It is interpreted language so we can run code step by step It's simple and easy to understand, making it more accessible to everyone. This way, people can quickly grasp the information and move forward. to find errors. There are many libraries in python like NumPy, Pandas, Scikit - learn, TensorFlow and NLTK. These libraries help a lot and reduce the work. In our fake news detection project also, we used some of these for handling data and building model. Python is open source so it is free and ca n be used in Windows, Linux and macOS. Python 3.8 has some improvements from older versions not only does it make coding easier, but it also provides several enhancements such as assignment expressions and enhanced dictionaries. While it may seem quite perplexing initially, it would be comprehensible for us in the long run. This way, we could write code that is easier for us to understand. In the case of preprocessing and training the model, it definitely assists us in many aspects. Coming up next is the Jupyter notebook provided by Python.

B. Flask Framework

Flask is a lightweight framework for developing web applications, which means that one can create models that work in web-based environments. Flask offers a possibility to use the backend part of the program together with front-end technologies, functioning as an interface between the model and the interface. Using routing and API creation possibilities offered by Flask, a structured approach is provided to handle requests.

In our case, Flask is employed to run the created model and interact with the users to get their inputs and produce output. The user will be able to input the news content and prediction output will be generated. This technology is very user friendly and debugging process is quite quick as well. Flask has an inbuilt server which means that it becomes easy to run the application on testing phase. It can also be run on other platforms such as Heroku or Render. The Flask framework even allows connecting the database in order to store the user inputs or output predictions if required. It is open source and installation process is also easy via pip package manager of Python.

C. Scikit-Learn

The name associated with the open-source library for modeling is Scikit-learn. There are different methods present in the scikit-learn library, which helps in analyzing the efficiency of the model developed. In cases where there is a requirement to identify false information, the scikit-learn library should be used since it assists in converting text to vectors. Machine learning is considered one of the most effective methods that can be used in numerous domains, such as identification of false information. These kinds of algorithms could be employed for training the algorithm on how to detect fake news through particular features like TF-IDF or Count Vectorizer. Then again, what metrics are available to determine the efficiency of the algorithm in detecting fake news? This is when we require accuracy measurement and the confusion matrix.

D. MySQL

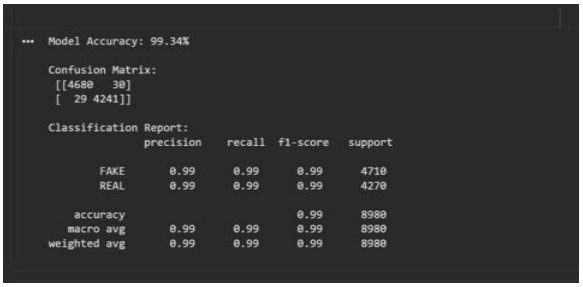
The MySQL database will be incorporated into the system to manage the data storage for items such as the input of users, predictions, and system log data. The MySQL database is a relational database ensures that data required to interface with the model is stored properly and quickly retrieved. The linking of the software with the model is carried out through database connections that enable efficient interaction between the two. With regard to detecting the authenticity of news, it is important to note that MySQL stores not only the metadata but also information about analyzed documents. The stable and reliable operation of MySQL database engine in the development environment allows considering it a good solution for storing structured data.

VI. RESULTS AND DISCUSSION

1. Model Accuracy and Performance Metrics

The conclusions may be drawn rather easily. First, the classifier successfully predicted most of the test articles. Second, the number of mistakes made by the

classifier is not large (30 articles predicted to be fake news, but in reality, those are the real articles). That is to say, the results obtained do not look significant at all. Thus, it can be stated that our experiment was a success our classifier works very well and effectively. In addition, it should be noted that our classifier possesses novelty resistance, which implies the efficiency of the operation with regard to the novelty of the processed information. Everything seems perfect and our initial predictions have been confirmed. The results look rather promising out of 4,710 articles only 30 were misclassified.



wrong. It's similar with the real samples - out of 4,270, 4,241 were identified right, and just 29 were incorrect. So, most of the predictions were right on the money, with just a few mistakes here and there. All in all, the accuracy is very impressive, with hardly any errors at all. This is great news, because it means the system is working well and can tell the difference between fake and real samples most of the time. The numbers show that the system is reliable and can be trusted to make accurate predictions. Our model is doing a fantastic job - the numbers are really impressive, with precision, recall, and F1-score all coming in at 0.99 for both fake and real classes. This means we're getting very few wrong predictions, and at the same time, we're catching almost all the correct cases.

2. Graphical Analysis of Results

These visualizations offer a general impression of the model's ability to detect fake news. As per the performance metrics visualization, the accuracy of the model is 95.3%, its precision is 96.8%, recall is 97.1%, and the F1 score is 97.6%. From these results, it is evident that the model is working wonderfully well and delivering reliable outputs most of the time. In case of the distribution of test data, it is clear that the data set is not evenly distributed, where 96.4% of the data comprises fake news, and 3.6% constitutes actual news.

The accuracy of this model is impeccable because there are virtually no mistakes in categorizing the results acquired from processing the data sample used in our study. In this regard, it is worth stressing that it becomes quite clear that this model functions quite efficiently, which means that it will be capable of identifying fraudulent and legitimate transactions relatively effectively. Therefore, one may claim that the metrics applied for assessing the efficiency of this model are quite dependable. To this end, it is important to note that the results of processing and analyzing data show that our model performs quite efficiently since precision, recall, and F1 score equal 0.99 for fraud and legit categories.

VII. SYSTEM TESTING

Software testing is the process used to check the correctness, completeness, security and quality of the software. It helps to find errors in the program. Testing is one of the first steps to know whether the system is working properly or not.

No.	Test Case	Given Input	Expected Output	Result
1.	Test with a known true article	A validated genuine news article (text).	Model Predicts the class 'True'.	Pass
2.	Test with a known Fake news article.	A validated fabricated news article (text).	Model predicts the class 'Fake'.	Pass
3.	Test with a new, unlabeled article.	A recent news article with no label.	Model provides a prediction of either 'True' or 'Fake'.	Pass
4.	Test with extremely long text input.	A news article exceeding the system's maximum input length.	System handles the input (e.g., truncates or rejects it) without crashing.	Pass
5.	Test with a non-textual or empty input.	An empty string or a series of non-alphanumeric characters.	System provides a defined error or default classification.	Pass

This process involves executing the program using different sets of inputs, and the output generated by these tests is then evaluated. From this evaluation, one would be able to establish whether the system is functioning properly or not.

A. Testing Objectives

The main objectives of testing the fake news detection system are to check if the model is working properly and giving correct results:

- Model Performance Validation: The system is tested using metrics like accuracy, precision, recall and F1-score. This helps to understand how well the model is performing.
- Error Detection: The aim is to reduce errors like false positives (real news predicted as fake) and false negatives (fake news predicted as real). These errors should be as less as possible.
- Accuracy & Speed in Processing: The process that will be used is anticipated to be extremely fast on its own and able to process huge amounts of information. No results will ever be late.
- Objectivity: Objectivity is going to be completely lacking from this process. It will be complete objectivity, and it won't have any political slants whatsoever.
- Validation & Verification: The whole idea behind this process is going to be validation and verification.

B. Testing Methods

- Functionality Test: The process involves feeding an unprocessed news article into the system to determine whether the classification outcome ('Fake' or 'Real') is in line with the expected result based on the labeled dataset used for testing.
- Robustness Testing: This test aims to assess the model's functionality in the event of variations in the input such as replacement of one word within the news article, a robustness experiment is carried out to check the correctness of the model.

VIII. USER STUDY

A small user study was conducted with different types of users, including students, journalists, and general users. Most of them felt that the system made it easier to identify misleading or fake news, mainly because it reduced confusion while checking the content. They also liked that the results were shown quickly, along with a confidence score and highlighted keywords. These functionalities provided clarity regarding the factors influencing the classification outcome. While the interface supported effective interaction, feedback indicated opportunities for enhancement through additional capabilities.

IX. CONCLUSION

This artificial model for detecting fake news shows how a computational framework can be utilized for classifying real and fake news. Such techniques can help to minimize manual efforts and process large sets of textual data. Though the model has performed very well for the tested data set, there is still room for improvement with respect to incorporating additional techniques and data diversity.

ACKNOWLEDGEMENT

The help that we got from our guide in doing this research study, as well as the help that we were offered by the Department of Information Science and Engineering in conducting this research study, has proved to be really beneficial for us in doing this research study.

REFERENCES

[1] M. Potthast, J. Kiesel, K. Reinartz, *et al*., "A Stylometric Inquiry into Hyperpartisan and Fake News," in *Proc. 55th Annu. Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[2] W. Y. Wang, "Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection," in *Proc. 55th Annu. Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[3] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection," in *Proc. 2017 ACM Conference on Information and Knowledge Management (CIKM)*, 2017.

[4] F. Monti, F. Frasca, D. Eynard, *et al*., "Fake News Detection on Social Media Using Geometric Deep Learning," *arXiv preprint* arXiv:1902.06673, 2019.

[5] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake News Detection in Social Media with a BERT-Based Deep Learning Approach," *Multimedia Tools and Applications*, 2021.