



PAPER • OPEN ACCESS

An Approach to detect fault text in articles

To cite this article: Pakala Prahasit Reddy *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1042** 012032

View the [article online](#) for updates and enhancements.

You may also like

- [Relation between Boltzmann and Gibbs entropy and example with multinomial distribution](#)
Paško Županovi and Domagoj Kui
- [Multifractal analysis of financial markets: a review](#)
Zhi-Qiang Jiang, Wen-Jie Xie, Wei-Xing Zhou et al.
- [A perturbative probabilistic approach to quantum many-body systems](#)
Andrea Di Stefano, Massimo Ostili and Carlo Presilla



The Electrochemical Society
Advancing solid state & electrochemical science & technology

247th ECS Meeting
Montréal, Canada
May 18-22, 2025
Palais des Congrès de Montréal

Showcase your science!

**Abstracts
due
December
6th**

An Approach to detect fault text in articles

Pakala Prahasit Reddy, Yempati Prasheela, Avula Uday Kumar Reddy,
Rajanikanth Aluvalu

Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, Telangana, India

rajanikanth.aluvalu@gmail.com

Abstract. The problem of Fault text (that contains fake news) has spread enormously in recent years. Social media is the major source for this type of news because it is easily and freely accessible to everyone and this type of news can be shared easily and rapidly, this enabled the widespread of news, so there is a need to detect this type of news. This news will mislead the users creating chaos and unrest. So the detection has become mandatory to avoid chaos and to maintain the pleasant news. We use Machine Learning Algorithms and basic Natural Language Process techniques to detect this type of news. In this work, Machine Learning Algorithms are used to parse the article and extract the summary from the article, this summary is used to predict the credibility. A passive-aggressive classifier and multinomial naive Bayes algorithms are used to train the model. A web app is created using the Flask API. One can put the URL in the web page to detect the news as "FAKE" or "REAL".

Keywords: TF-IDF vectorizer, Flask Framework, Passive Aggressive Classifier, Multinomial Naïve Bayes, Fake text, Real text.

1 INTRODUCTION

With the increase in technology, data became easily accessible to everyone free of cost. The Internet is the major source to provide freely accessible data, but the trustworthiness of information provided by them relies on a wide variety of factors. A large amount of data is produced through electronic media, online, and print media, but it is difficult to tell whether the data is genuine or not.

In the last few years, there has been an extensive increase in the use of social media and browsing the internet. Taking this as an advantage some people have started publishing incredible news to the users. In which there is a discrepancy between the title and the text of that article. In many news articles, fake news spread around us has resulted in chaos among people and are unable to predict the credibility of the news. Also, in this hectic life schedule, one does not have time to check the credible nature of the news. This type of news creates chaos and unrest in society [1]. For example, in the 2016 US elections, there are a lot of sources suggest that President Trump had won the election due the fake propaganda and also creating fake accounts on Facebook to spread propaganda to impact the opinion of voters [2].

A study portrayed that diffusion of fault text had a non-ignorable impact on USA twenty-sixteen elections [3]. A few facts on fault news in the United States:

- Around 20 crore US population gets the news updates from social media [4]
- Fault text has a lion's share on Facebook than other external sources. [5].

In recent years the study on this area has increased through Artificial Intelligence and Machine Learning. The real objective of this work is to detect whether the data contain any Faulty news (Fake news) or not. If the data contain fault text then it is termed as "FAKE" and if data doesn't contain fault text then it is termed as "REAL". This project uses efficient data mining algorithms [6][7][8] and basic natural language(text) processing techniques [9]. A subset of AI is called machine learning in which statistical methods are used to give systems the ability to learn the data without using explicit programming. Natural language processing is the study of interaction with computers and natural languages particularly dealing with the analysis of natural text [10].

In this paper data is transformed into a TF-IDF vectorizer, then Machine Learning algorithms like Passive-Aggressive classifier and Multinomial Naive Bayes classifier [7] are used to train the model.



A passive-aggressive classifier is an online learning algorithm that works as an interpreter step by step. This algorithm is best useful for analyzing social media data as it gets updated every day. One more algorithm that is in use is Multinomial Naïve Bayes which is good for word counts for classification of text (NLP problems). Naïve Bayes works well for detecting tag of text, in our case fake or real.

In this paper, the training process is created by using machine learning algorithms and this process is deployed using flask to further analyze the nature of news of other articles. Flask is an API that is used to create the web application, it was developed by Armin Ronacher and it is easy to learn and it has less base code to implement the simple web application [11]. Our project needs only a simple web application to detect the nature of news, so Flask API is suitable for our project.

Finally, if we enter the URL “localhost: 5000” a web page will be displayed where the URL of news is entered to detect the credibility of news.

2. RELATED STUDY

This section consists of previous studies that are already performed to which the present proposed study is related or have similar features. Numerous studies have been done that have a focus on detecting fake news. They have applied different data mining techniques for detection and achieved different probabilities for different methods.

Aswini Thota et al have proposed a process to detect the fake news using a transformed Tf-IDF combined with DNN(stands for Dense neural networks), we're capable to surpass previous process architectures by 2.25% and attained an accuracy of 93.21% on the validation set(test data) [12].

In a dissertation, the author built a model that used features related to frequency-based to train the processes including some classifiers like stochastic gradient classifier, Naïve Bayes classifier, support vector machine classifier, and an online supervised Passive Aggressive Classifier. SVM(support vector machine)and Stochastic gradient classifier algorithm combined with the Tf-Idf vector attains good accuracy. Performance metrics used for this work are classification accuracy, confusion matrix, the area under the ROC curve, and classification report. The train: the test is 68:32 is used for this work [13].

Victoria L. Rubin et al described that there exist many kinds of fault news, each with distinguished potential textual indicators. Thus, existing works plan hand-wrought features which are not only hard but strongly depends on the specific dataset and the accessibility of domain knowledge to plan the apt features [14].

3 PROPOSED MODEL

Proposed model combines Passive Aggressive Classifier (PAC) and Naïve Bayes processes to improve the accuracy of fault text detection. A feature extraction method called TF-IDF vectorizer is used in this training process. The ML, Passive Aggressive classifier is performed using the TF-IDF feature extraction approach [15]. This model is trained to classify fault news as either fake or real. A web page is created and this model is deployed to that page using Flask web framework. Our proposed model works as:

- Step 1: Get the data set
- Step 2: Apply data cleaning pre-processing
- Step 3: Extracting the training data.
- Step 4: Generate TF-IDF vectorizer
- Step 5: Split the data for test and train
- Step 6: Train model using PAC and Naïve Bayes

Step 7: Calculate the accuracy of the model

Step 8: Deploy the model using the Flask web framework

Trained process accuracy is calculated as

Trained process accuracy = (correctly classified samples) / (total samples)

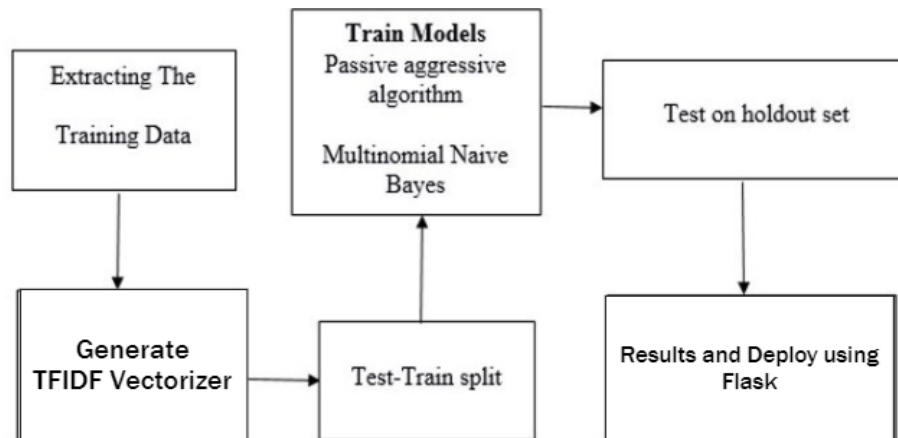


Fig 1: Architectural Model

4 EXPERIMENTAL SETUP

Experimental setup began with importing all the necessary libraries. Then the data is read into a data frame using `read_csv()` which is a pandas function to read the CSV files. If there is any redundant data in the dataset, the `drop_duplicates` function is used to remove redundant tuples. Then labels are created from the data frame. Secondly, partition the dataset into a training dataset and a test dataset in the ratio 80%,20% respectively. Thirdly, initialize the `Tfidfvectorizer` with a maximum document frequency of 0.7 using `max_df=0.7` where TF (Term Frequency) frequency of a word that appears in a document is coined as Term Frequency [16].

• $tf(t,d) = (\text{frequency of term } t \text{ in a document}) / (\text{total frequency of terms in a document})$.

IDF(Inverse Document Frequency) is the number of times a word appears on the total set of documents [17].

• $idf(t) = \log_e((\text{frequency of documents present}) / (\text{frequency of documents with term } t \text{ present in it}))$.
 $tfidf = tf(t,d) * idf(t)$

We initialize the Passive Aggressive Classifier

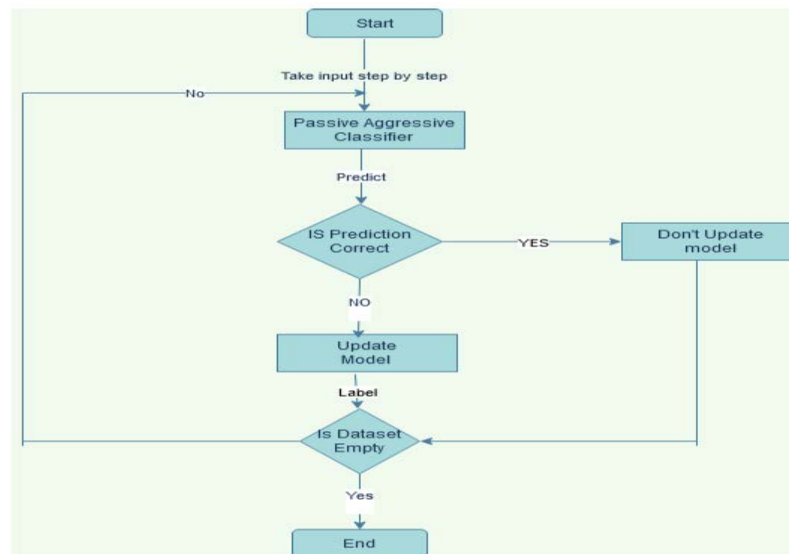


Fig 2: Flowchart of PAC

As shown in figure 2 in the Passive-Aggressive Algorithm, the input data comes in sequential order and the model is updated step-by-step. If the prediction is correct, we keep the model and do not update it. If the prediction is incorrect, we will make changes to the model and update it. We continue this process until the dataset becomes empty.

Then, predict the test set data and calculate the accuracy. Then find the confusion matrix(2*2) to find the percentages of tp,fp, fn, and then plot the confusion matrix[18].

Finally, initialize Multinomial Naïve Bayes, train the model, and dump it into the hard disk.

This Naïve Bayes classifier is derived from the Bayes theorem(A probability theorem).

$P(q|a)$ where q is the total outcomes possible and a is the object which has to be classified, with few features [19].

$$P(q|a) = P(a|q) * P(q) / P(a)$$

$P(q)$ = priori probability of class)

$P(a|q)$ = posteriori probability of q

$P(a)$ = priori probability of a (predictor)

Our model is trained in the filename model.py

Our flask API is used in the filename app.py

index.html is used to display web page.

We create a normal HTML page used to take a URL from the user by using Flask API. The process is trained by the naïve Bayes so it can give the custom output when custom input. We enter the localhost:5000 to see the webpage. Operate the terminal from your project directory yourself. Produce the training process using machine learning by running the model.py file. This would serialize the trained process as model.pkl or model.pickle Execute app.py using the below command to start Flask API python app.py. By default, the flask will run on port 5000.The user can enter the URL in the webpage, after entering the URL [20], we should press the predict button then we get the nature of the news as “the news is “REAL” or “the news is “FAKE”.”

5 RESULT ANALYSIS

In the Existed System, we check the credibility of the news based on the publisher of the news nonetheless of the content of the news. This is never a good approach because a news source could be categorized as mistrustful and still produce a true fact. In our proposed system, We have mainly three files for this work

HTML-index.html, CSS-style.css --frontend.

model.py-training the model.

app.py-for running and initializing the FLASK API.

Our frontend looks like:

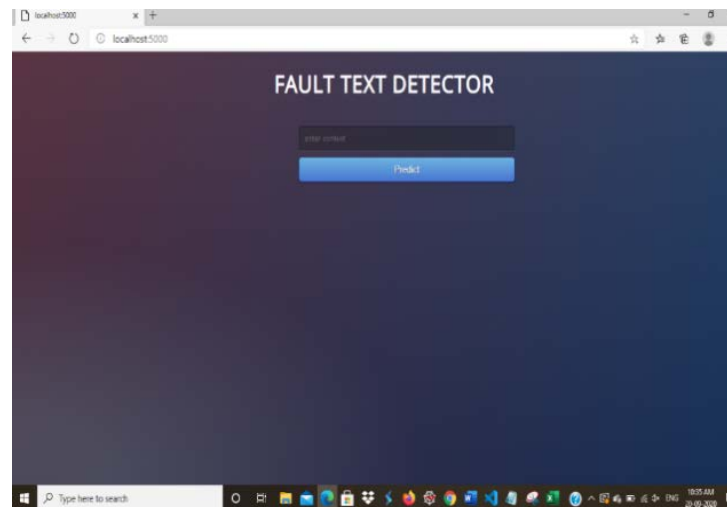


Fig 3: Webpage for user

The dataset news.csv was taken from Kaggle. Dataset was of size $7796 * 4$ records. A passive-aggressive classifier is used for calculating the accuracy of the model. We have taken the 80:20 tests: train ratio. Multinomial Naïve Bayes algorithm is used to train the process and then we dump the training process into the disk. The confusion matrix plotting for the model is as shown in figure 4.

After dumping the file into disk, load the model into app.py and run the flask API. After running app.py, the terminal states to visit the browser and open the localhost: 5000. The URL gives the webpage in which we can enter the URL to detect the nature of the news.

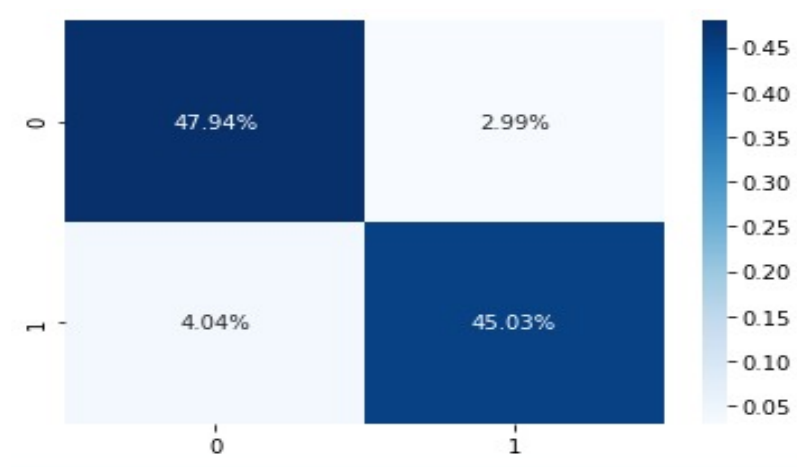


Fig 4: Confusion matrix plotting

6 CONCLUSION

Our model has analyzed the detection of fake news which is now prevalent in social media platforms and websites. We have used the TF-IDF feature extraction technique [21], Passive Aggressive Classifier, and Naïve Bayes Algorithm for training our model. We have achieved an accuracy of 93.94% by our model. Therefore by using Machine Learning techniques we can conclude that any news from a large or small dataset can be classified as fake or not fake with previous data set values in less time which helps the user to believe in particular news that appears on social media or other sources.

7. FUTURE SCOPE

The spread of misinformation has extremely harmful effects on users and the social environment. Fake news is designed to deceive the user which makes it difficult to detect it in the first place. There are many sources from which fake news is spread which causes chaos among the people and society. A future enhancement would be to identify the source of the fake news and to stop the increased spread of fake news on online platforms and in social media platforms. It would also have the capability to track and find the sources of this fake news so that we can stop the people who are trying to address the public with these malicious intentions. They would also identify the social account of the people spreading fake news and rumours so that they can stop them before it's too late. These things can engulf society with positivity and a healthy life. Finally, we want to expand this work by carrying out like analysis on a different dataset such as Twitter, Facebook, and Instagram. By categorizing fake news from social media platforms, we hope to get one step forward towards building a programmed fake news diagnosis platform.

REFERENCES

- [1] David O Klein and Joshua R Mueller. Fak news: A legal perspective. 2017
- [2] Alessandro Bessi and Emilio Ferrara. Social bots distort the 2016 us presidential election online discussion First Monday, 21(11), 2016.
- [3] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. In Journal of Economic Perspective, volume 31, 2017.

- [4] Jeffrey Gottfried and Elisa Shearer. News Use Across Social Media Platforms 2016. Pew Research Center, 2016
- [5] Craig Silverman and Lawrence Alexander. How teens in the Balkans are duping trump supporters with fake news. BuzzFeed News, 3, 2016.
- [6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.
- [7] Harry Zhang. The Optimality of Naive Bayes.
- [8] Mykhailo Granik, Volodymyr Mesyura, "Fake news detection using naïve Bayes classifier", Published in 2017 IEEE First Ukraine Conference (UKRON), Kyiv, Ukraine.
- [9] James Thorne, Mingjie Chen, Giorgos Myrithous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. Fake news stance detection using a stacked ensemble of classifiers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 80–83, 2017.
- [10] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.
- [11] Devendra Ghimire, "Comparative study on Python web frameworks: Flask and Django", A Thesis for Bachelor of Engineering, Metropolia University of Applied Sciences, 2020
- [12]. Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, Nibrat Lohia, "Fake News Detection: A Deep Learning Approach", A data science review, Southern Methodist University.
- [13]. Aayush Ranjan, Associate Prof. Manoj Kumar, "Fake News Detection Using Machine Learning", A Dissertation for Master of Technology in Software Engineering, Delhi Technological University.
- [14] Niall J Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015.
- [15] Vasu Agarwal, H. Parveen Sultana, Srijan Malhotra, Amitrajit Sarkar, "Analysis of Classifiers for Fake News Detection", Volume 165, 2019, Pages 377-383
- [16] Simon Lorent, Ashwin Itoo, "Fake news detection using machine learning", A thesis presented for the degree of Master in Data Science, University of Liège.
- [17] Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for IDF, 2004
- [18] Nicole O'Brien, "Machine Learning for Detection of Fake News", A thesis Master of Engineering in Electrical Engineering and Computer Science, Massachusetts Institute of Technology
- [19] Faraz Ahmad and Lokeshkumar R, "A Comparison of Machine Learning Algorithms in Fake News Detection", *International Journal on Emerging Technologies*, Published by Research Trend, Website: www.researchtrend.net
- [20] S. S. Uddin, M. Umar and K. Keerti Chennam, "Dynamic control structure in gameplay mechanism as a solution to gaming addiction : Tartarus – Conceptual approach using a simple FSM pattern," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-8, doi: 10.1109/ICCCI48352.2020.9104164.
- [21] M. T. Ahmed, M. M. Hussain and K. K. Chennam, "Designing a consensus ranking algorithm for same domain entities," 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2017, pp. 12-16, doi: 10.1109/CESYS.2017.8321253.