IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 9, September 2023

DOI: 10.17148/IJARCCE.2023.12907

FAKE NEWS DETECTION USING NLP

G. Agasthiya[1], Mrs. S. Jancy Sickory Daisy[2]

Student, CSE, Anand Institute of Higher Technology, Chennai, India[1]

Assistant Professor, CSE, Anand Institute of Higher Technology, Chennai, India[2]

Abstract: The field of Natural Language Processing (NLP) has gained significant attention in recent years, particularly in the context of fake news detection and categorization. NLP techniques offer powerful tools to analyse and understand textual data, allowing us to identify patterns, sentiments, and linguistic features that can help distinguish between true and false news. In this project, we aim to predict false news and determine their respective categories using NLP techniques. To achieve this, we will employ a combination of supervised machine learning algorithms and NLP methods. Firstly, we will gather a dataset consisting of news articles labelled as either true or false. The dataset will also include information regarding the category or topic of each news article. These categories may range from politics and sports to entertainment and science. Next, we will pre-process the textual data by performing tasks such as tokenization, stop-word removal, and stemming. These steps will help to clean and transform the raw text into a format suitable for analysis.

Keywords: False news prediction, news categorization, NLP techniques, supervised machine learning, textual data, tokenization, stop-word removal, stemming, TF-IDF, word embedding's, logistic regression, random forests, support vector machines.

I. INTRODUCTION

There was a time once if anyone required any news, he or she would sit up for the next day newspaper. With the expansion of on-line newspapers un agency update news nearly instantly, individuals have found a more robust and quicker thanks to learn of the matter of his/her interest. Today social-networking systems, on-line news portals, and alternative on-line media became the most sources of reports through that fascinating and breaking news are shared at a fast pace news are shared at a fast pace. Several news portals serve interest by feeding with distorted, part correct, and typically fanciful news that is probably to draw in the eye of a target cluster of individuals. Faux news has become a significant concern for being harmful typically spreading confusion and deliberate misinformation among the individuals. The term faux news has become a buzz word lately. A united definition of the term faux news remains to be found. It may be outlined as a sort of info that consists of deliberate information or hoaxes unfold via ancient print and broadcast print media or on-line social media. These are revealed sometimes with the intent to mislead to wreck a community or person, produce chaos, and gain financially or politically. Since individuals are usually unable to pay enough time to see reference and take care of the credibleness of reports, machine-driven detection of pretend news is indispensable. Therefore, it's receiving nice attention from the analysis community. The previous works on faux news have applied many ancient machine learning ways and neural networks to detect faux news. They need targeted on police investigation news of specific variety. Accordingly, they developed their models and designed options for specific datasets that match their topic of interest. It's probably that these approaches would suffer from dataset bias and are probably to perform poorly on news of another topic. Number of the present studies have additionally created comparisons among totally different ways of pretend news detection. Prevaricator and experimented some existing models on the dataset. The comparison result hints. Totally different models will perform on a structured dataset like prevaricator. The length of this dataset isn't ample for neural network analysis and a few models were found to suffer from overfitting. Several advanced machine learning models, e., neural network primarily based ones don't seem to be applied that are established best in several text classification issues

A. OBJECTIVE

The objective of fake news detection is to identify and flag news articles, stories, or content that contain false or misleading information, and to prevent the spread of such content. The spread of fake news can have serious consequences, including misinformation, public panic, and the manipulation of public opinion, which can have a significant impact on society, politics, and public policy. Fake news detection involves using a variety of techniques and tools, such as natural language processing, machine learning, and data analysis, to identify patterns and characteristics in news articles that indicate they may be fake or misleading. These techniques can include analyzing the source of the news, examining the language used in the article, and cross-referencing the information with other reliable sources to determine its veracity.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 9, September 2023
DOI: 10.17148/IJARCCE.2023.12907

B. SCOPE

There are numerous factors that contribute to the dissemination of fake news. The first is due to a lack of data among the public. The readers are uninformed of the sources' legitimacy, and hence the news' veracity. Being listened to this will have a huge detrimental influence on the public.

The lack of automatic fact-checking procedures is the second reason. Fake news detection is attempted by websites such as Politifact1, Full Fact2, and AltNews3, but the time-consuming human process is just too slow to prevent the initial dissemination of false information. Detecting false news automatically is a difficult task that defies present content-based analysis method.

II. RELATED WORK

A. Paper [1] "Fake news detection" is defined as the task of categorizing news along a continuum of veracity, with an associated measure of certainty. Veracity is compromised by the occurrence of intentional deceptions. The nature of online news publication has changed, such that traditional fact checking and vetting from potential deception is impossible against the flood arising from content generators, as well as various formats and genres. These methods have emerged from separate development streams, utilizing disparate techniques. In this survey, two major categories of methods emerge: 1. Linguistic Approaches in which the content of deceptive messages is extracted and analyzed to associate language patterns with deception; and 2. Network Approaches in which network information, such as message metadata or structured knowledge network queries can be harnessed to provide aggregate deception measures. Both forms typically incorporate machine learning techniques for training classifiers to suit the analysis.

B. Paper [2] fake news is because identifying these entities require measuring the news propagation, which has shown to be complex and resource intensive [3]. Trend Micro, a cyber security company, analyzed hundreds of fake news services provider around the globe. They reported that it is effortless to purchase one of those services. In fact, according to the report, it is much cheaper for politicians and political parties to use those services to manipulate election outcomes and people opinions about certain topics [4, 5]. Detecting fake news is believed to be a complex task and much harder than detecting fake product reviews given that they spread easily using social media and word of mouth. We present in this paper an n-gram features based approach to detect fake news, which consists of using text analysis based on n-gram features and machine learning classification techniques.

C. Paper [3] Facebook post prediction through real or fake labeling can be done through naïve Bayes and it performs well [6]. A proposed method can separate fake contents in three categories: serious fabrication, large scale hoaxes and humorous fake [11]. It can also provide a way to filter, vet and verify the news. PHEME was a three-year research project funded by the European Commission from 2014-2017, studying NLP techniques for dealing rumor detection, stance detection [8] and [9], contradiction detection and analysis of social media rumors. Fake news stories can be easily shared on social media platforms but it is difficult to identify fake content automatically.

D. Paper [4] fake news is intentionally written to mislead readers, which makes it nontrivial to detect simply based on news content. The content of fake news is rather diverse in terms of topics, styles and media platforms, and fake news attempts to distort truth with diverse linguistic styles while simultaneously mocking true news. For example, fake news may cite true evidence within the incorrect context to support a non-factual claim [22]. Thus, existing hand-crafted and data-specific textual features are generally not sufficient for fake news detection. Other auxiliary information must also be applied to improve detection, such as knowledge base and user social engagements. Second, exploiting this auxiliary information actually leads to another critical challenge: the quality of the data itself. Fake news is usually related to newly emerging, time-critical events, which may not have been properly verified by existing knowledge bases due to the lack of corroborating evidence or claims.

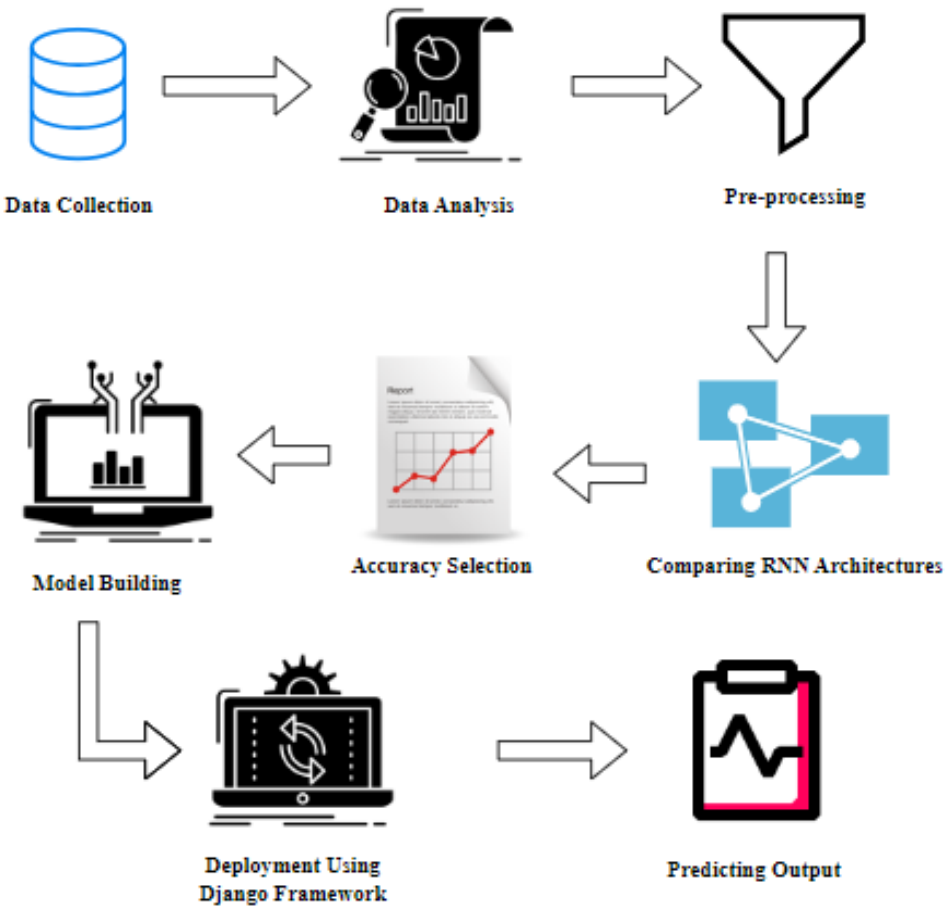
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 9, September 2023

DOI: 10.17148/IJARCCE.2023.12907

III. SYSTEM ARICHITECTURE DIAGRAM



IV. IMPLEMENTATION

Algorithm implementation:

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

Performance Metrics to calculate:

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

DOI: 10.17148/IJARCCE.2023.12907

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

True Positive Rate(TPR) = TP / (TP + FN)
False Positive rate(FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:
Accuracy = (TP + TN) / (TP + TN + FP + FN)
Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.
Precision = TP / (TP + FP)
Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labelled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:
F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

The below 3 different algorithms are compared:
- RNN Archietecture
- Long-Short term memory networks

RNN Archietecture:

A Recurrent Neural Network (RNN) is a type of artificial neural network designed for processing sequential data. Unlike traditional feedforward neural networks, RNNs have connections that loop back on themselves, allowing them to maintain a hidden state that captures information from previous time steps in the sequence. This looping mechanism makes RNNs well-suited for tasks involving sequences, such as natural language processing, speech recognition, and time series prediction.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 9, September 2023

DOI: 10.17148/IJARCCE.2023.12907

Here is a detailed explanation of the architecture and key components of an RNN:

Basic Structure:
An RNN consists of a series of interconnected layers. At each time step t, it takes an input vector (or sequence) and produces an output vector (or sequence).

The key feature of an RNN is its hidden state, denoted as "h." This hidden state is a representation of the network's memory, and it is updated at each time step.

Input and Output:
At each time step t, the RNN takes an input vector or element x(t). This input can be a single element of a sequence, a word in a sentence, a pixel in an image, etc.

The RNN produces an output vector or element y(t) at each time step. The output can be used for various tasks, such as predicting the next element in a sequence or classifying the sequence as a whole.

Hidden State:
The hidden state h(t) is a vector that captures information from previous time steps. It serves as the memory of the network.

The hidden state is computed at each time step using the current input x(t) and the previous hidden state h(t-1).

Output Computation:
The output at each time step can be computed based on the current hidden state or a combination of the hidden state and the input at that time step.

Backpropagation Through Time (BPTT):
Training an RNN involves using a variant of backpropagation called Backpropagation Through Time.

It is similar to standard backpropagation but accounts for the sequential nature of the data.

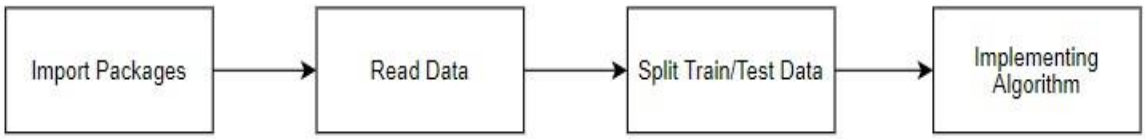
The gradients are computed at each time step and accumulated over the entire sequence to update the network's weights.
Issues with Standard RNNs:

Standard RNNs have limitations, including the vanishing gradient problem, which makes it challenging to capture long-range dependencies in sequences.

To address these issues, more advanced RNN architectures have been developed, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which are designed to better capture long-term dependencies.

In summary, an RNN is a neural network architecture that can process sequential data by maintaining a hidden state that captures information from previous time steps. It is a fundamental building block for various sequence-based tasks in machine learning and deep learning.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data
output : getting accuracy

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 9, September 2023

DOI: 10.17148/IJARCCE.2023.12907

Long-Short term memory networks:

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem and better capture long-range dependencies in sequential data. They were introduced to overcome the limitations of standard RNNs and have become a fundamental building block in many applications involving sequences, such as natural language processing, speech recognition, and time series analysis. Here's a comprehensive explanation of LSTM networks:

Basic Structure:
An LSTM network is composed of LSTM cells arranged in a sequence. Each LSTM cell has an internal structure that enables it to store and retrieve information over long sequences.

Like standard RNNs, LSTM networks take input vectors or elements sequentially and produce output vectors or elements at each time step.

The key innovation in LSTM cells is their ability to maintain a cell state, which can capture long-term dependencies in the data.

Components of an LSTM Cell:

An LSTM cell consists of three main gates and a cell state:
Forget Gate: Decides what information from the cell state should be thrown away or kept.
Input Gate: Determines what new information should be added to the cell state.
Output Gate: Controls what information from the cell state should be used to generate the output.
Cell State: The cell state runs throughout the entire sequence and can carry information over long distances.

Information Flow:

The forget gate (f(t)) controls what information from the previous cell state (C(t-1)) should be retained.
The input gate (i(t)) determines what new information from the candidate cell state (ĉ(t)) should be added to the cell state.
The cell state (C(t)) is updated based on the forget gate, input gate, and candidate cell state.
The output gate (o(t)) controls what information from the cell state should be used to produce the hidden state (h(t)).

Backpropagation Through Time (BPTT):

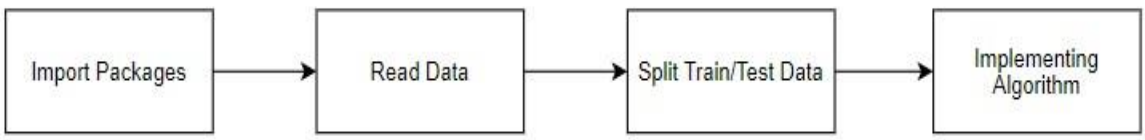
LSTM networks are trained using Backpropagation Through Time, similar to standard RNNs. BPTT computes gradients for the network's parameters to minimize a loss function.

Advantages of LSTMs:

LSTMs can capture long-range dependencies in sequences.
They mitigate the vanishing gradient problem, allowing for more effective training on long sequences.
They are suitable for a wide range of sequence-based tasks and have been extended into more advanced variants like Gated Recurrent Units (GRUs).

In summary, LSTM networks are a type of recurrent neural network that incorporates memory cells with gates to selectively store, update, and retrieve information over long sequences. This architecture has proven effective in capturing complex patterns in sequential data and has become a cornerstone of deep learning in fields that involve sequences.

MODULE DIAGRAM



DOI: 10.17148/IJARCCE.2023.12907

GIVEN INPUT EXPECTED OUTPUT
input : data
output : getting accuracy

Deployment:
Django (Web FrameWork) :

Django is a micro web framework written in Python.
It is classified as a micro-framework because it does not require particular tools or libraries.
It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Django supports extensions that can add application features as if they were implemented in Django itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Django was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. According to Ronacher, the idea was originally an April Fool's joke that was popular enough to make into a serious application. The name is a play on the earlier Bottle framework.

When Ronacher and Georg Brand created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed.

In April 2016, the Pocoo team was disbanded and development of Django and related libraries passed to the newly formed Pallets project.

Django has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

The micro-framework Django is part of the Pallets Projects, and based on several others of them.

Django **is** based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD licence. It was developed at pocoo by Armin Ronacher. Although Django is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let's take a closer look into Django, so-called "micro" framework for Python.

V. CONCLUSION AND FUTURE ENHANCEMENT

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set of higher accuracy score algorithm will be find out. The founded one is used in the application which can help to find the Real and fake news

Deploying the project in the cloud.
To optimize the work to implement in the IOT system.

REFERENCES

[1]. S. Maheshwari, How fake news goes viral: A case study, Nov. 2016. [Online]. Available: https://www.nytimes. com/2016/11/20/business/media/how-fake-news- spreads.html (visited on 11/08/2017).
[2]. A. Mosseri, News feed fyi: Addressing hoaxes and fake news, Dec. 2016. [Online]. Available: https://newsroom. fb.com/news/2016/12/news-feed - fyi - addressing - hoaxes-and-fake-news/ (visited on 11/08/2017).
[3]. S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Association for Computational Linguistics, 2012, pp. 171-175.
[4]. N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1-4, 2015.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 9, September 2023

DOI: 10.17148/IJARCCE.2023.12907

[5]. V. W. Feng and G. Hirst, "Detecting deceptive opinions with profile compatibility.," in IJCNLP, 2013, pp. 338- 346.
[6]. V. L. Rubin and T. Lukoianova, "Truth and deception at the rhetorical structure level," Journal of the Association for Information Science and Technology, vol. 66, no. 5, pp. 905-917, 2015.
[7]. G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks," PloS one, vol. 10, no. 6, c0128193,2015.
[8]. Opensources. [Online]. Available: http://www. opensources.co/ (visited on 11/08/2017).
[9]. D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?" In Pro- ceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016., 2016, pp. 42-47. [Online]. Available: http://ccur-ws.org/Vol- 1568/paper8.pdf.
[10]. Business financial news, u.s international breaking news. [Online]. Available:http://www.reuters.com/ (visited on 11/08/2017).
[11]. Explosion, Spacy, Sep. 2017. [Online]. Available: https: //github.com/explosion/spaCy (visited on 11/08/2017). [12] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. Selje- botn, and K. Smith, "Cython: The best of both worlds," Computing in Science Engineering, vol. 13, no. 2, pp. 31-39, 2011, ISSN: 1521-9615. DOI: 10.1109/ MCSE.2010.118.

